Zero-Shot Learning from Wikipedia Descriptions

Pedro Rodriguez
Department of Computer Science
University of Maryland
College Park, MD 20742
pedro@cs.umd.edu

Mozhi Zhang
Department of Computer Science
University of Maryland
College Park, MD 20742
mozhi@cs.umd.edu

Abstract

We study the task of zero-shot learning of fine-grained bird types from Wikipedia articles. We qualitatively investigate and improve GAZSL [29], a state-of-the-art system for this setting. In this we show qualitative and quantitative evidence that models use only a small fraction of words from each article — a complementary finding to prior textual noise suppression on this task —, but nevertheless learn correct visual semantics such as colors. However due to GAZSL's unigram-based text encoding it cannot model the semantic difference between a bird with a YELLOW BEAK AND WHITE BODY, and a bird with a YELLOW BODY AND WHITE BEAK. To address this limitation we modify the model to allow for higher orders of composition, and empirically validate improvements in test accuracy, marginally beating the state-of-the-art on the default split of Caltech-UCSD Bird-2011.

1 Introduction

Traditional supervised recognition techniques require a large number of training data for every category. Unfortunately, objects have a long-tailed distribution, and it can be hard to find examples for some "rare" categories. Zero-shot learning aims to solve this problem by transferring knowledge from seen classes to unseen ones.

The key challenge of zero-shot learning is to find an intermediate representation for each class. Previous work uses attributes [8, 13, 14], word embeddings [9, 16, 28], textual descriptions [6, 7, 15, 20, 21, 29], and knowledge graphs [26] to learn class semantic representation. In this work, we study the zero-shot learning setting where each category is associated with a Wikipedia article. We focus on a state-of-the-art system for this setup [29] and address two questions:

1. What textual semantics does the model learn?
2. Can we improve test accuracy with a more powerful text encoder?

For the first question, we find that the model predictions are mostly based on only a few words from the text descriptions. In particular, the category names are visually predictive by themselves. Our qualitative analysis demonstrate that the model can correctly align simple visual attributes such as colors to words.

For the second question, we experiment with adding bigram features to capture higher order word compositio, which slightly improves the performance. We also try initializing word embedding layer with pre-trained word vectors, but this surprisingly hurts test accuracy.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

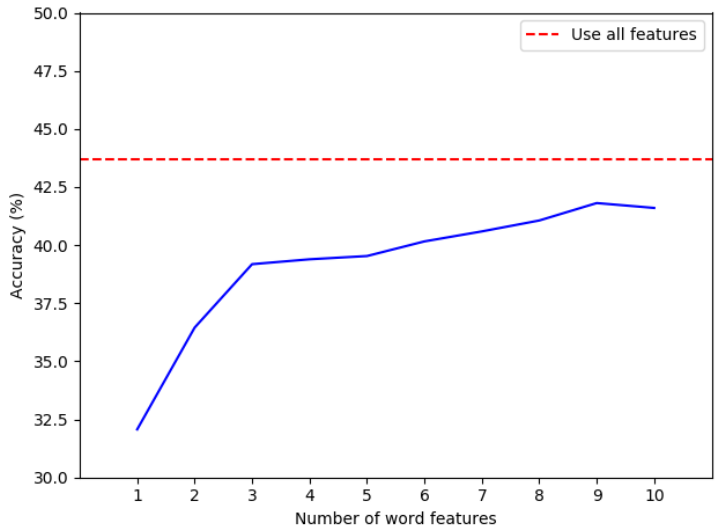

Figure 1: Test accuracy on CUB with different number of word features. For each test class, we remove all words except for the top-k word types (ranked by TF-IDF). Using only $k = 10$ word types achieves a similar test accuracy as using all word features (41.6% vs. 43.7%), indicating that most words are irrelevant for the model.

2 Background: GAZSL

This section introduces our starting point, GAZSL [29], a zero-shot learning approach based on generative adversarial networks (GAN). The high-level idea of GAZSL is to train a conditional GAN [22] to "imagine" image features from a text description. This reduces zero-shot learning to classic supervised learning, since we can now generate training examples for unseen classes from its description. We summarize the method below.

The first step of GAZSL is to extract visual features for each training image, which serve as real examples for the GAN. In particular, GAZSL uses a Visual Part Detector/Encoder network [27] as the image encoder. For each training class, the Wikipedia description is encoded by transforming the TF-IDF vector with a multi-layer perceptron. The text feature vector is concatenated with a Gaussian noise as input to the generator network.

The generator is trained to output realistic image features, and the discriminator is trained to differentiate real image features from fake ones. The authors use auxiliary classifier GAN [17] with Wasserstein distance objective [1] and gradient penalty [11]. To help the generator match the real data distribution, a visual pivot regularizer is applied to encourage the mean of the generated features for each class to match the mean of the real samples.

At test time, sixty "hallucinated" image features are generated for each test class, and class labels are predicted with nearest neighbor search. For all our experiments, we use Caltech-UCSD Birds 2011 (CUB) [25] dataset with default split. The original GAZSL has a state-of-the-art 43.7% test accuracy[1] on this dataset.

3 What does the Model Learn?

In our first set of experiments, we investigate what textual semantics GAZSL can capture by answering two questions: (1) how many word features are predictive for the model, and (2) how the model learns to map words into image feature space.

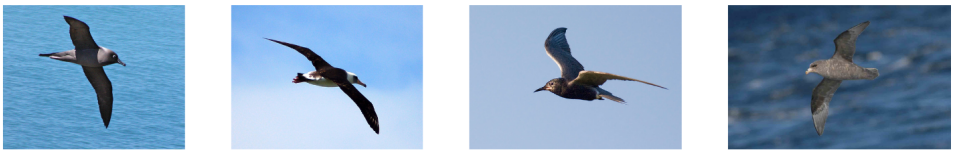

Figure 2: Top four nearest training classes for the word *albatross* in the image feature space. The first two classes are albatross, and the other two are visually similar to an albatross.

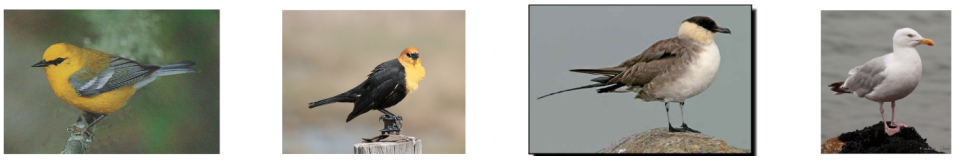

Figure 3: Top four nearest training classes for the word *yellow* in the image feature space. All four classes have yellow parts: head and body, head, neck, and bill (from left to right).

3.1 Varying Number of Word Features

Wikipedia articles are long and noisy. For example, only three out of fifteen paragraphs in the article for black-footed albatross contain information relevant to visual recognition. Therefore, we conjecture that only a few keywords from each article are visually predictive. To verify our hypothesis, we experiment with removing a large portion of text. For each test class, we rank the word types by TF-IDF scores and remove all words except the top-k word types.[2] Figure 1 shows test accuracy for different k. The test accuracy only drops for 2% (41.6% vs. 43.7%) when using 10 word features per class, which confirms that most words are not useful to the model.

We notice that category names are often predictive by themselves. The model has 37.3% test accuracy (6.4% drop) if we only use the category name as its text description. One reason is that many category names in CUB have super-category information. For example, one of the test classes is the black-footed albatross, and it is closely related to two training classes: sooty albatross and laysan albatross. As a result, the model can use the word *albatross* to transfer knowledge from the two training albatross class to the unseen black-footed albatross. When evaluating on a super-category-exclusive split of CUB proposed in [7], GAZSL only has 10.3% test accuracy, which proves that the model heavily relies on super-categories to transfer knowledge.

3.2 Text to Image Feature

The generator in GAZSL maps text to the image feature space. We qualitatively study this map with a nearest-neighbor analysis. Given some text w, we compute an average image feature as following: we feed w into the generator, sample sixty image features, and take their average as the image embedding for w. We use this process to generate embeddings for all training classes (from their Wikipedia article) and every word in the vocabulary.

Figure 2 shows the nearest training classes for the word *albatross*. As expected, the top-2 classess are both albatross.[3] The other neighboring classes seem to be visually similar to albatross. Interestingly, the model seems to know the concept of *yellow* — the four closest training classes all have some yellow part (Figure 3). We observe a similar pattern for other colors, which shows that the model is capable of matching some visual attributes to the correct word.

4 Improving Text Encoder

In this section, we report our experiments on improving the text encoder of GAZSL. As mentioned before, all our experiments are run on the CUB dataset with default split, and we compare models

[1] We only use top-1 accuracy in this paper.

[2] Stop words are already removed in the original GAZSL model.

[3] There are only two albatross training classes.

Table 1: Top-1 accuracy on the default split of CUB. The systems are sorted according to best accuracy. The highlighted row is the replicated state-of-the-art result of the original GAZSL [29]. Our best model marginally beats the baseline. We find TF-IDF weighting is crucial for good performance, and stemming slightly hurts test accuracy. Suprisingly, initializing with pre-trained word embeddings does not improve accuracy.

Feature	N-gram	Stemming	Embedding	Best Accuracy (Mean $\pm$ Std)
TF-IDF	1 + 2	False	Random	**44.562** (43.721 ± 0.748)
TF-IDF	1	False	Random	44.050 (43.800 ± 0.276)
TF-IDF	1	True	Random	43.778 (43.562 ± 0.290)
TF-IDF	1 + 2	True	Random	43.539 (43.414 ± 0.120)
TF-IDF	1	False	GloVe	41.493 (40.175 ± 1.182)
TF-IDF	1	True	GloVe	41.186 (41.096 ± 0.129)
TF-IDF	1	True	FastText	40.198 (39.914 ± 0.309)
Count	1	False	Random	39.277 (38.232 ± 1.291)
TF-IDF	1	False	FastText	39.107 (38.879 ± 0.336)
Count	1 + 2	False	Random	38.220 (37.879 ± 0.325)
Count	1	True	GloVe	30.003 (27.446 ± 2.238)
Count	1	True	FastText	27.924 (26.821 ± 1.314)

with top-1 accuracy. Each experiment is repeated three times, and we report the max, mean, and standard deviation of accuracies. We experiment with two extensions: adding bigram features and initialize with pre-trained word embeddings. Table 1 summarizes our experiments, which we will discuss in detail.[4]

4.1 Yellow Beak or Yellow Neck: Compositionality

Since the text model in GAZSL is based on stemmed unigrams, it cannot model compositionality beyond co-occurrence. This naturally leads to the inability to pair descriptors like color with corresponding body parts. For example, *yellow body and white beak* has the same bag-of-word feature vector as *yellow beak and white body*, and yet the two phrases describe very different birds. To partially alleviate the lack of composition in GAZSL we modify the original model to use a combination of unigrams and bigrams[5] with results shown in Table 1. We find that while the test accuracy improves, the gain is limited (0.8% absolute increase).

In implementing bigram features, we also experiment with turning off the Porter stemmer and use a count vector instead of TF-IDF. We confirm that TF-IDF weighting is crucial as it suppresses some noise in the text. However, we find that stemming actually hurts test accuracy.

4.2 Pre-trained Word Embedding

Although we do not experimentally validate, we suspect that the model fails to learn the semantics of many bigrams due to the small size of the text corpus (200 Wikipedia articles). A natural solution to data-scarcity challenges is to use transfer learning. In GAZSL the word representations are initialized with Xavier random initialization [10] which implies that the entirety of their text representations is based a small corpus of 200 articles. In contrast, it is standard and effective in natural language processing systems to initialize the word embedding layer with weights learned from large unlabeled corpora [4]. To test if GAZSL can be improved we compare initializing word embeddings with pre-trained GloVe [18] and FastText [3] word vectors, which are trained on billion-word corpora.

Surprisingly we find that pre-trained word embeddings degrades test accuracy in every configuration. We hypothesize that this may be related to learning dynamics affected by not using Xavier initialization so compared the accuracy curves versus optimization iteration in Figure 4. We observe that initial performance is significantly better using GloVe, but both GloVe and FastText initialized models converge to a worse solution than randomly initialized embeddings. We leave further investigation of

[4] In our experiments we tried significantly more configurations than shown, totally 175 experimental runs.

[5] To restrict the vocabulary size, we only use 20,000 bigrams selected by highest IDF scores.

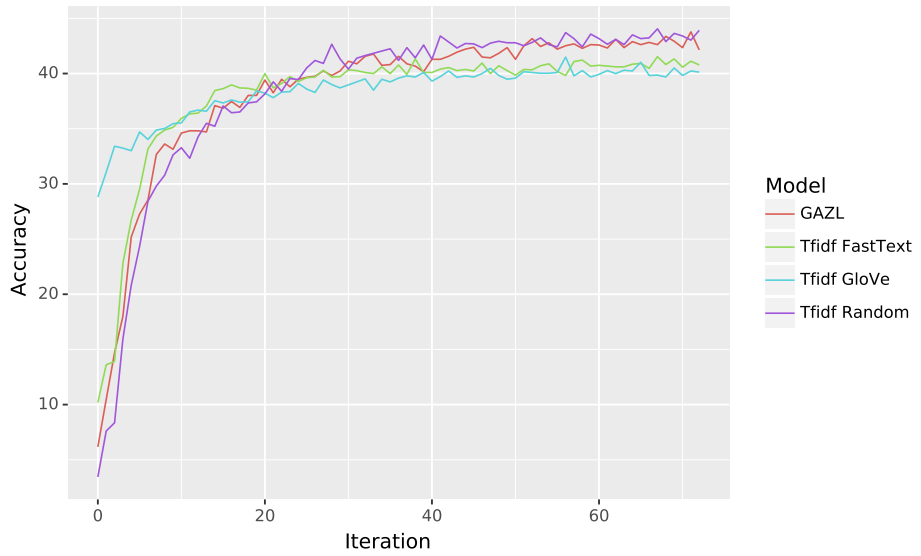

Figure 4: Learning curve for the best unigram based models from Table 1. When initializing with Glove or FastText embeddings, the test accuracy is better at the beginning, but converges to a worse solution.

this behavior to future work, and note that regularization techniques such as Batch Normalization [12] and Layer Normalization [2] may prove effective.

5 Conclusion and Future Work

In this work, we study the task of zero-shot recognition with the help of Wikipedia articles. We carefully study a state-of-the-art system, GAZSL. We find that the model uses only a fraction of text for prediction, and the model is capable of aligning some words to visual attributes such as color. We extend the text encoder by adding bigram features and initializing with pre-trained word embeddings. Our best system marginally improves the test accuracy on CUB by 0.8%.

Our initial attempts at improving over GAZSL are focused on using text encoders with a sophisticated composition function. For example, we attempt to use sequence models (RNN, CNN, and Transformer [24]) and pretrained contextualized word embeddings [5, 19]. The primary challenge in using these methods is that the input text are too long, which makes training probihitively expensive. One possible solution is to down-sample the text, which we leave to future work. Another challenge is that the training signal may be too noisy for a sequence model — as the authors identify and we confirm, only a small fraction of terms on a page matter. Therefore, we instead experiment with adding bigram features, which adds a little expressiveness to the model while keeping training tractable.

In the future, we plan to repeat our experiments on other datasets, such as the super-category-exclusive split for CUB [6] and the North America Bird [23] dataset. We also plan to look beyond Wikipedia articles — we hope to experiment with sequence models using a textual knowledge source that is less noisy than Wikipedia descriptions.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GANs. *arXiv preprint arXiv:1701.07875*, 2017.

[2] J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 2017.

[4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *Proceedings of Computer Vision and Pattern Recognition*, 2013.

[7] M. Elhoseiny, Y. Zhu, H. Zhang, and A. M. Elgammal. Link the head to the "beak": Zero shot learning from noisy text description at part precision. In *Proceedings of Computer Vision and Pattern Recognition*, 2017.

[8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proceedings of Computer Vision and Pattern Recognition*, 2009.

[9] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *Proceedings of Advances in Neural Information Processing Systems*, 2013.

[10] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. 2010.

[11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of Wasserstein GANs. In *Proceedings of Advances in Neural Information Processing Systems*, 2017.

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference of Machine Learning*, 2015.

[13] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In *Advances in neural information processing systems*, 2014.

[14] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[15] J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of Computer Vision and Pattern Recognition*, 2015.

[16] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *Proceedings of the International Conference on Learning Representations*, 2014.

[17] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the International Conference of Machine Learning*, 2017.

[18] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of Empirical Methods in Natural Language Processing*, 2014.

[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[20] 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 Computer Vision and Pattern Recognition*, 2016.

[21] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of Computer Vision and Pattern Recognition*, 2016.

[22] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *Proceedings of the International Conference of Machine Learning*, 2016.

[23] 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 Computer Vision and Pattern Recognition*, 2015.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, 2017.

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

[26] X. Wang, Y. Ye, and A. Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of Computer Vision and Pattern Recognition*, 2018.

[27] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas. SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition. In *Proceedings of Computer Vision and Pattern Recognition*, 2016.

[28] L. Zhang, T. Xiang, S. Gong, et al. Learning a deep embedding model for zero-shot learning. In *Proceedings of Computer Vision and Pattern Recognition*, 2017.

[29] 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 Computer Vision and Pattern Recognition*, 2018.