



Using Sentences as Semantic Representations in Large Scale Zero-Shot Learning

Yannick Le Cacheux, Hervé Le Borgne, Michel Crucianu

► To cite this version:

Yannick Le Cacheux, Hervé Le Borgne, Michel Crucianu. Using Sentences as Semantic Representations in Large Scale Zero-Shot Learning. ECCV 2020 workshop Transferring and adapting source knowledge in computer vision (TASK-CV), Aug 2020, Glasgow, United Kingdom. hal-03003689

HAL Id: hal-03003689

<https://hal.science/hal-03003689>

Submitted on 13 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using Sentences as Semantic Representations in Large Scale Zero-Shot Learning

Yannick Le Cacheux^{1,2}, Hervé Le Borgne¹, and Michel Crucianu²

¹ Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
 {yannick.lecacheux, herve.le-borgne}@cea.fr

² CEDRIC – CNAM, Paris, France
 michel.crucianu@cnam.fr

Abstract. Zero-shot learning aims to recognize instances of unseen classes, for which no visual instance is available during training, by learning multimodal relations between samples from seen classes and corresponding class semantic representations. These class representations usually consist of either attributes, which do not scale well to large datasets, or word embeddings, which lead to poorer performance. A good trade-off could be to employ short sentences in natural language as class descriptions. We explore different solutions to use such short descriptions in a ZSL setting and show that while simple methods cannot achieve very good results with sentences alone, a combination of usual word embeddings and sentences can significantly outperform current state-of-the-art ³.

1 Introduction and Related Work

Zero-shot learning (ZSL) is useful when no visual samples are available for certain classes, provided we have semantic information for these classes [11]. It is then possible to train a model to learn the relations between the visual and semantic features using *seen* classes, for which both modalities are available; these relations can later be employed to classify instances of *unseen* classes, for which no visual sample is available during training, based on their semantic *class prototypes* [18, 7, 24, 12, 13]. The semantic information can consist of vectors of attributes, e.g. binary codes for “has fur”, “has stripes”, etc. if classes are animal species. In a large-scale setting, it can be impractical to devise and provide attributes for hundreds or even thousands of classes. Word embeddings are then typically used to represent classes. However, a large performance gap still exists between these two types of class representations [14].

An ideal solution could be to use short natural sentences to describe each class, as this is less time-consuming than providing comprehensive attributes and can be more visually informative than word embeddings derived from generic text corpora. The use of sentences as class descriptions in ZSL is not well studied. Some works already employed sentences for ZSL, but not in a convenient setting. For instance, Akata *et al.* [1] rely on 10 sentences *per image* instead

³ A short version of this article was published in the Task-CV workshop @ECCV 2020

of 1 per class. Ba *et al.* [15] and Elhoseiny *et al.* [6] use tf-idf from large text collections such as Wikipedia articles and different neural architectures to obtain semantic representations. Zhu *et al.* [28] generate visual features from noisy text description (again from Wikipedia articles) using Generative Adversarial Network. These works as well as others such as [5,22] tend to focus on fine-grained recognition and explicitly or implicitly assume that classes are similar (consisting for example of birds species) and that class descriptions will contain specific information (for example regarding the beak, the wings, the plumage...). On the other hand, we focus on generic object recognition and make close to no assumption regarding the nature of these descriptions. The closest work to ours is probably Hascoet *et al.* [9], in which different methods to obtain prototypes from WordNet definitions are evaluated, but reported performance is significantly below that of usual word embeddings.

In this article, we explore several ideas to leverage sentences to build semantic embeddings for ZSL and show that this can significantly outperform previous best reported performance. These proposals are easy to implement and computationally light. We provide the code⁴ as well as the corresponding embeddings which can be used out-of-the-box as better quality semantic representations.

2 Proposed Method

To deal with a zero-shot learning (ZSL) task, one considers a set \mathcal{C}_s of *seen* classes used during training and a set \mathcal{C}_u of *unseen* classes that are available for the test only, with $\mathcal{C}_s \cap \mathcal{C}_u = \emptyset$. Each class is also associated to a semantic *class prototype* $\mathbf{s}_c \in \mathbb{R}^K$ that characterizes it. Let us consider a training set $\{(\mathbf{x}_i, y_i), i = 1 \dots N\}$ with labels $y_i \in \mathcal{C}_s$ and visual features $\mathbf{x}_i \in \mathbb{R}^D$. The goal of the ZSL task is to learn a compatibility function $f : \mathbb{R}^D \times \mathbb{R}^K \rightarrow \mathbb{R}$ assigning a similarity score to a visual sample \mathbf{x} and a class prototype \mathbf{s} . It is usually a parametrized function that is learned by minimizing a regularized loss \mathcal{L} :

$$\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^{|\mathcal{C}_s|} \mathcal{L}(f(\mathbf{x}_i, \mathbf{s}_c), y_i) + \lambda \Omega[f] \quad (1)$$

where Ω is a regularization term weighted by λ . Once f is learned, the testing phase consists in determining the label $\hat{y} \in \mathcal{C}_u$ corresponding to a visual sample \mathbf{x} such that $\hat{y} = \arg \max_{c \in \mathcal{C}_u} f(\mathbf{x}, \mathbf{s}_c)$.

A simple linear model is presented in Section 2.1. Our contribution presented in Section 2.2 consists in leveraging sentences to build semantic prototypes \mathbf{s}_c

2.1 Linear ZSL model.

We adopt a linear ridge regression model, in which semantic prototypes are projected into the visual features space so as to minimize a regularized least

⁴ <https://github.com/yannick-lc/zsl-sentences>

square loss. This choice is motivated by the good and consistent results of this model in [14], as well as its low computational cost and ease of reproducibility. The choice of the visual space as the projection space is based on [25], which argues that this choice enables to mitigate the hubness problem [23]. Results with other models from [8] are provided in Section 3.3.

The visual features $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ are extracted with a pre-trained network, so that $\mathbf{x}_n \in \mathbb{R}^D$, D the dimension of the visual space. We thus have a corresponding training matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$. The semantic representations $\{\mathbf{s}_1, \dots, \mathbf{s}_C\}$ of the C seen (training) classes lie in \mathbb{R}^K . We write $\mathbf{T} = (\mathbf{t}_1, \dots, \mathbf{t}_N)^\top \in \mathbb{R}^{N \times K}$ the semantic representations associated with each training sample \mathbf{x}_n with label y_n so that $\mathbf{t}_n = \mathbf{s}_{y_n}$.

The resulting loss for the ridge regression with parameters $\boldsymbol{\Theta} \in \mathbb{R}^{K \times D}$ is

$$\frac{1}{N} \|\mathbf{X} - \mathbf{T}\boldsymbol{\Theta}\|_F^2 + \lambda \|\boldsymbol{\Theta}\|_F^2 \quad (2)$$

where λ a hyperparameter weighing the ℓ_2 regularization penalty $\|\boldsymbol{\Theta}\|_F^2$, and $\|\cdot\|_F$ is the Frobenius norm. This minimization problem has a closed-form solution:

$$\boldsymbol{\Theta} = (\mathbf{T}^\top \mathbf{T} + \lambda N \mathbf{I}_K)^{-1} \mathbf{T}^\top \mathbf{X} \quad (3)$$

During the prediction phase, given a testing sample \mathbf{x} and the semantic representations $\{\mathbf{s}'_1, \dots, \mathbf{s}'_{C'}\}$ of the C' unseen classes, $\mathbf{s}'_{c'} \in \mathbb{R}^K$, we predict label \hat{y} corresponding to the class whose prototype's projection is closest to \mathbf{x} :

$$\hat{y} = \underset{c'}{\operatorname{argmin}} \|\mathbf{x} - \boldsymbol{\Theta}^\top \mathbf{s}_{c'}\|_2 \quad (4)$$

2.2 Sentence-Based Embeddings

A simple baseline consists in averaging the embeddings of the words in the definition, as is usually done for class names consisting of several words: if a sentence s describing a class has N words with respective embeddings $\{\mathbf{w}_1, \dots, \mathbf{w}_N\}$, then the corresponding semantic representation is $\mathbf{s} = \frac{1}{N} \sum_{n=1}^N \mathbf{w}_n$. We call this baseline the *Def_{average}* approach. However, as illustrated in Fig. 2, not all words are equally important in a short sentence description. We therefore explore the use of attention mechanisms: the sentence embedding is a weighted average of the embeddings of its words, so that more important words contribute more to the result. We consider two ways to achieve this.

Visualness Based Approach The first, called the *Def_{visualness}* approach, aims to estimate how “visual” a word is. For a given word w_n , we thus collect the 100 most relevant images from Flickr using the website’s search ranking. We obtain representations $\{\mathbf{r}_1^i, \dots, \mathbf{r}_M^i\}$ with 2048 dimensions ($\mathbf{r}_m \in \mathbb{R}^{2048}$) using a pre-trained ResNet (see Section 3.1). We then measure the average distance of

vectors \mathbf{r}_m^i to the mean vector $\bar{\mathbf{r}}^i = \frac{1}{M} \sum_{m=1}^M \mathbf{r}_m^i$ to obtain the additive inverse of the visualness v^i :

$$v^i = -\frac{1}{M} \sum_{m=1}^M \|\mathbf{r}_m^i - \bar{\mathbf{r}}^i\|_2 \quad (5)$$

We hypothesize that words with high visual content have features close to each other, so that the quantity v^i from Equation 5 can be used as a qualitative measure of the intuitive concept of how “visual” a given word is. Examples of words with high and low visualness shown in Section 3.4 tend to confirm that this hypothesis is reasonable.

Figure 1 shows the distribution of the inverse of the visualness for the 4059 words from class names and WordNet definitions of classes from [8]. Equation 5 is then used to obtain weights for each word based on its visualness in order to use a weighted average of the word embeddings from a definition as class prototype. As the initial scale of the average distances / negative visualnesses is arbitrary, a temperature τ is introduced in the softmax to weigh each word, so that the resulting sentence embedding is

$$\mathbf{s} = \sum_{n=1}^N \frac{\exp(v_n/\tau)}{\sum_{k=1}^N \exp(v_k/\tau)} \mathbf{w}_n \quad (6)$$

The value τ is cross-validated on the validation set – see Section 3.1. In practice, this often leads to selecting $\tau = 5$.

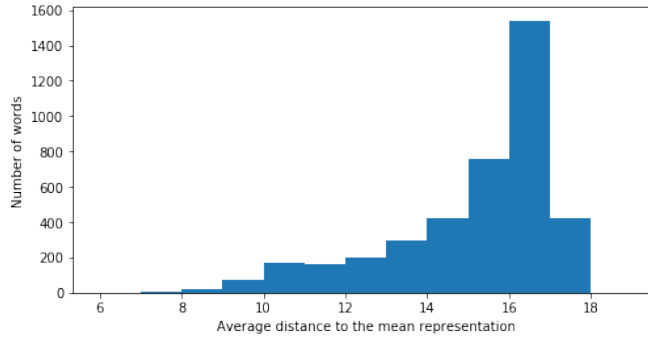


Fig. 1. Inverse of the visualness (low values correspond to high visualness) for the 4059 words from class names and WordNet definitions.

Learned Attention Based Approach A second approach called *Def_{attention}* aims to learn to predict the visualness v^i of word w^i from its embedding $\mathbf{w}^i \in \mathbb{R}^K$ such that $v^i = \boldsymbol{\theta}^\top \mathbf{w}^i$, where $\boldsymbol{\theta}$ are learned parameters. Equation 5 can then be used to create a prototype from a class’ definition. As the v^i are directly learned,

it is no longer necessary to account for their initial scale. We can thus discard the temperature in the softmax by setting $\tau = 1$ in Equation 5.

Different ways could be considered to learn θ . A straightforward approach could consist in randomly initializing θ along with Θ from Equation 2, computing visualnesses $v^i = \theta^\top \mathbf{w}^i$, computing class prototypes \mathbf{s}_c using the v^i and Equation 5, computing \mathbf{T} and finally computing the loss from Equation 2; we could then use back-propagation and gradient descent to update θ and Θ until convergence.

However, we instead take advantage of the closed-form solution from Equation 3 and proceed as follows: we randomly initialize θ and compute class prototypes \mathbf{s}_c and \mathbf{T} using Equation 5 as previously. We then directly estimate Θ using the closed-form solution from Equation 3, and use this value to compute the loss in Equation 2. We then back-propagate the gradient and perform gradient descent on θ only, the value of Θ being estimated with Equation 3 at each iteration. We repeat this process for 50 “epochs”. An illustration of the resulting weights from both approaches is shown in Fig. 2.

The resulting sentence embeddings can then be compared to the standard class prototypes obtained by embedding the class name (*Classname* approach). Since recent results show that hierarchical and graph relations between classes contain valuable information [8], in the *Classname+Parent* approach we combine the *Classname* prototype with the prototype of its parent class.

Finally, we experiment with different combinations of the base approaches:

- *Classname+Def_{average}*,
- *Classname+Def_{visualness}*
- *Classname+ Def_{visualness} +Parent*.

All these combinations simply consist in finding a value $\mu \in [0, 1]$ such that the combined prototype is $\mathbf{s} = \mu \mathbf{s}_1 + (1 - \mu) \mathbf{s}_2$, given \mathbf{s}_1 and \mathbf{s}_2 the prototypes of two approaches.

3 Experiments

3.1 Settings

Selecting hyperparameters. Hyperparameter values such as λ in Equation 2 or τ in Equation 5 are selected by cross-validation on *seen* classes, keeping 200 of the 1000 seen training classes as unseen validation classes. The model is then retrained on all 1000 training classes with the selected values.

When several prototypes \mathbf{s}_1 and \mathbf{s}_2 are combined with $\mathbf{s} = (1 - \mu) \mathbf{s}_1 + \mu \mathbf{s}_2$, $\mu \in [0, 1]$, for instance in the *Classname+Def_{average}* or *Classname+Def_{visualness}* approaches, μ is also considered to be a hyperparameter and its value is selected by cross-validation along with the other hyperparameters. In these cases, it leads to surprising results with a frequently selected value of $\mu = 0.7$, meaning the combined prototype is 70% from the definition prototype and 30% from the classname prototype, even though the definition alone performs far worse than the classname alone. Nonetheless, this still led to consistent results.

On the other hand, when combining a prototype with the prototype of its parent class, cross-validation tended to yield very inconsistent and unstable values. When combining a prototype with its parent prototype, we therefore fixed $\mu = 0.25$ (meaning 75% base class and 25% parent class) for all methods and all word embeddings. Note that in the ***Classname+Def_{visualness}*** and ***Classname+ Def_{visualness} +Parent*** approach, the prototype from the parent class is also built with *Classname+Def_{visualness}*, and the value of μ is the same as the one selected for the child class.

Pre-trained word embeddings. We used widely adopted pre-trained embeddings available on the Internet for Word2vec [16], FastText [3] and Glove [20]. We used a Word2vec embedding model pre-trained on Wikipedia as we found it gave better results than other version, such as a Word2vec embedding model trained on Google News. For the same reason, we used FastText and Glove models pre-trained on Common Crawl. We used a 300-dimension version for all three. The pre-trained embedding models were downloaded from the following links:

- Word2vec: <https://wikipedia2vec.github.io/wikipedia2vec/pretrained/> (English version trained on Wikipedia).
- Fasttext: <https://fasttext.cc/docs/en/english-vectors.html> (version trained on Common Crawl with 600B tokens, no subword information).
- Glove: <https://nlp.stanford.edu/projects/glove/> (version trained on Common Crawl with 840B tokens).

For Elmo [21], we similarly adopted a pretrained version obtained from <https://allennlp.org/elmo>. We used the original version with 93.6 million parameters, pre-trained on the 1 Billion Word Benchmark. Elmo embeddings have dimension 3×1024 (a 1024-dimensional embedding from each of the three layers). For the sake of simplicity we combine the three layers using the same weight of 0.33 to obtain a single 1024-dimensional representation for each word, as weights fine-tuned for our specific task gave similar results.

Visual and semantic features. Visual features are obtained from the last pooling layer of a ResNet [10] model pre-trained on ImageNet [4]. They therefore have 2048 dimensions. We use the ResNet-101 implementation of PyTorch [19]. Visual features are normalized to have unit ℓ_2 -norm.

Classname ZSL embeddings are obtained from synsets as follows: a synset in WordNet [17] can consist of several lemmas, each lemma consisting of several words. The prototype of a lemma is the mean of the embeddings of its words when such embeddings exist in the pre-trained model. The prototype of the synset and thus of the class is the mean of the prototypes of its lemmas. In the approximately 6 (depending on the pre-trained embedding) cases where no word from any lemma had an embedding in the pre-trained model, we used the embedding from the word “*thing*” as class prototype. All semantic prototypes are ℓ_2 -normalized, for the *Classname* approach as well as all the others.

Table 1. Comparison of approaches on ImageNet with a linear model. Best previous reported result is 14.1 in [8]; the result marked with * corresponds to the same setting (use of *Classname* with Glove embeddings).

	Word2vec	FastText	Glove	Elmo
<i>Classname</i>	12.4	14.8	14.5*	10.9
<i>Classname+Parent</i>	13.4	15.9	15.4	11.4
<i>Def_{average}</i>	9.7	10.6	10.0	8.7
<i>Def_{visualness}</i>	10.5	10.9	10.5	9.5
<i>Def_{attention}</i>	10.5	11.0	10.2	9.5
<i>Classname+Def_{average}</i>	14.6	17.2	16.9	12.2
<i>Classname+Def_{visualness}</i>	14.8	17.3	16.8	12.1
<i>Classname+Def_{visualness}+Parent</i>	15.4	17.8	17.3	12.5

Evaluation protocol We measure ZSL accuracy on the large-scale ImageNet dataset, with synsets as classnames and corresponding WordNet definitions as sentences. We also use the WordNet hierarchy to determine parent classes. We employ the experimental protocol of [8], with the same ResNet features and, especially, the same train/test splits since [8] evidenced significant structural bias with previous popular splits. Glove embeddings are employed in [8]; we also compare with Word2vec and FastText [2], as well as Elmo contextual embeddings using pre-trained embeddings available on the Internet (details are provided in the supplementary material). The ZSL model is a ridge regression from the semantic to the visual space as it gives good, consistent and easily reproducible results (see [14]).

3.2 Results with Linear Model

Results are shown in Table 1. The baseline *Def_{average}* approach performs poorly compared to the usual *Classname* approach. Attention provides a slight improvement (both *Def_{visualness}* and *Def_{attention}* give comparable results), but still does not match *Classname*, even though attention weights seem reasonable (some examples are shown in Fig. 2). The use of parent information improves results compared with *Classname* alone, which is consistent with [8] where the best methods make use of hierarchical relations between classes.

The combination of *Classname* and *Def* approaches brings significantly better scores than either separately. Surprisingly, while any *Def* alone has lower performance than *Classname* alone, the best trade-off obtained by cross-validation is 70% definition and 30% classname in every case, meaning that the definition has a much stronger presence than the class name in the resulting embedding.

The word embedding method has an impact on performance; FastText consistently ranks above the other methods. Elmo has surprisingly low performance including with fine-tuned attention, even though one could expect attention to be more effective here as Elmo considers the role of the word in the sentence.

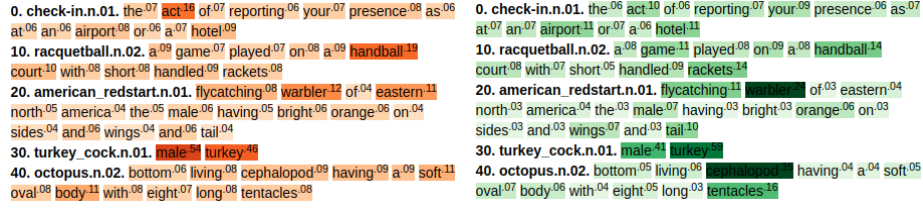


Fig. 2. Illustration of attention scores on some test classes. *Left*: weights from $Def_{visualness}$ after softmax (the temperature is $\tau = 5$ so differences are less pronounced than initially). *Right*: weights learned with $Def_{attention}$, with FastText embeddings.

3.3 Results with Other Models

We report results with the best performing prototypes for a few other ZSL models in addition to the linear model already described. We specifically report results for models used in [8] (except for graph-convolutional models which use a different setting). For ConSE [18], DeVISE [7] and ESZSL [24], we use the implementation from [8]. We report results averaged over 5 runs with different random initializations of parameters for ConSE and DeVISE. Since the other models use closed-form solutions, it is not necessary to report results averaged over several runs as there is no variance in the results. We also provide results with another simple linear model, consisting in a ridge regression from the visual to the semantic space. This model is called $Linear_V \rightarrow S$, as opposed to $Linear_S \rightarrow V$ which is the ridge regression model from the semantic to the visual space described in Section 2.1.

We report results with the best class prototypes as determined in Table 1 of the main paper, i.e. the ones obtained with the $Classname + Def_{visualness} + Parent$ approach with the FastText embeddings. These prototypes will be made available to the community along with the others. Top-1, top-5 and top-10 accuracies for the models mentioned above with these prototypes are provided in Table 2.

Table 2. Top-k ZSL accuracy for different models, using the $Classname + Def_{visualness} + Parent$ prototypes constructed from FastText embeddings.

	top-1	top-5	top-10
$Linear_S \rightarrow V$	17.8	43.6	56.7
$Linear_V \rightarrow S$	9.1	26.2	36.7
ESZSL	16.3	40.6	52.4
ConSE	12.7	31.8	42.4
DeVISE	14.0	38.3	52.1

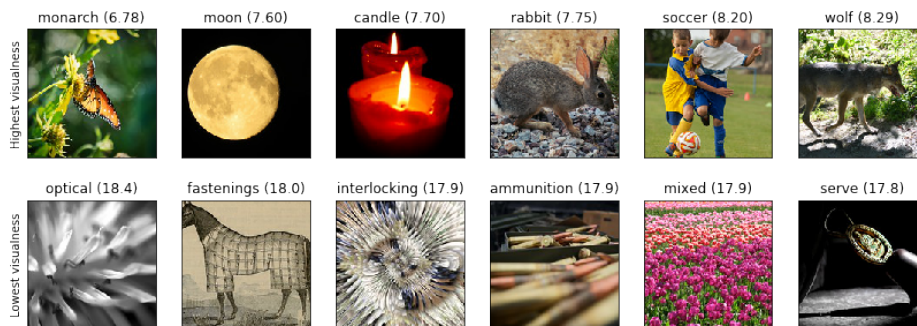


Fig. 3. *Top*: words with highest visualness. *Bottom*: words with lowest visualness. The visualness of a word is the inverse of the mean distance (shown in parenthesis) to the mean representation of visual features from the top 100 corresponding images from Flickr, see Section 3.4. Top 1 image with no copyright restriction is displayed.

3.4 Illustration of visualness

We provide a few examples of the words with the highest and lowest visualness in Figure 3, as well as the corresponding inverse visualness (the mean distance to the mean feature representations for images associated with this word) and the corresponding top image result with no copyright restriction from Flickr.

4 Conclusion.

To scale zero-shot learning to very large datasets it is important to solve the problem of providing class prototypes for many classes. The use of class name embeddings scales better than the provision manually-defined attributes but the resulting performance is nowhere near. We suggest that low-cost textual content, consisting in one sentence per class, can bring substantial performance improvements when combined with class name embeddings, when they are processed with the proposed approaches. The improved class prototypes for ImageNet are available at <https://github.com/yannick-lc/zsl-sentences>.

Beyond the case of zero-shot learning, other tasks are known to suffer from an incomplete semantic information, such as image retrieval, classification [31] or annotation [29]. The work we presented in this article can thus benefit to these tasks, in particular when one addresses large-scale datasets. Predicting automatically the important words into sentences with regard to their visualness leads to semantic representation that can be mixed to visual representation to build intrinsic multimedia representations [30] that can be used to address cross-modal retrieval [26] and classification [27].

References

1. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2927–2936 (2015)
2. Almeida, F., Xexéo, G.: Word embeddings: A survey. arXiv preprint arXiv:1901.09069 (2019)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
5. Elhoseiny, M., Saleh, B., Elgammal, A.: Write a classifier: Zero-shot learning using purely textual descriptions. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2584–2591 (2013)
6. Elhoseiny, M., Zhu, Y., Zhang, H., Elgammal, A.: Link the head to the” beak”: Zero shot learning from noisy text description at part precision. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6288–6297. IEEE (2017)
7. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al.: Devise: A deep visual-semantic embedding model. In: Advances in Neural Information Processing Systems (2013)
8. Hascoet, T., Ariki, Y., Takiguchi, T.: On zero-shot recognition of generic objects. In: Computer Vision and Pattern Recognition. pp. 9553–9561 (2019)
9. Hascoet, T., Ariki, Y., Takiguchi, T.: Semantic embeddings of generic objects for zero-shot learning. EURASIP Journal on Image and Video Processing **2019**(1), 13 (2019)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. Pattern Analysis and Machine Intelligence **36**(3), 453–465 (2014)
12. Le Cacheux, Y., Le Borgne, H., Crucianu, M.: From classical to generalized zero-shot learning: A simple adaptation process. In: International Conference on Multimedia Modeling. pp. 465–477. Springer (2019)
13. Le Cacheux, Y., Le Borgne, H., Crucianu, M.: Modeling inter and intra-class relations in the triplet loss for zero-shot learning. In: International Conference on Computer Vision. Seoul, Korea (2019)
14. Le Cacheux, Y., Popescu, A., Le Borgne, H.: Webly supervised embeddings for large-scale zero-shot learning. In: Asian Conference on Computer Vision (2020)
15. Lei Ba, J., Swersky, K., Fidler, S., et al.: Predicting deep zero-shot convolutional neural networks using textual descriptions. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4247–4255 (2015)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119 (2013)
17. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)

18. Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G.S., Dean, J.: Zero-shot learning by convex combination of semantic embeddings. In: International Conference on Learning Representations. pp. 488–501 (2014)
19. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems. pp. 8026–8037 (2019)
20. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics (2014)
21. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of NAACL-HLT. pp. 2227–2237 (2018)
22. Qiao, R., Liu, L., Shen, C., Van Den Hengel, A.: 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)
23. Radovanović, M., Nanopoulos, A., Ivanović, M.: Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research* **11**(Sep), 2487–2531 (2010)
24. Romera-Paredes, B., Torr, P.: An embarrassingly simple approach to zero-shot learning. In: International Conference on Machine Learning. pp. 2152–2161 (2015)
25. Shigeto, Y., Suzuki, I., Hara, K., Shimbo, M., Matsumoto, Y.: Ridge regression, hubness, and zero-shot learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 135–151. Springer (2015)
26. Tran, T.Q.N., Le Borgne, H., Crucianu, M.: Aggregating image and text quantized correlated components. In: Computer Vision and Pattern Recognition. Las Vegas, USA (june 2016)
27. Tran, T.Q.N., Le Borgne, H., Crucianu, M.: Cross-modal classification by completing unimodal representations. In: ACM Multimedia 2016 Workshop: Vision and Language Integration Meets Multimedia Fusion. Amsterdam, The Netherlands (october 2016)
28. Zhu, Y., Elhoseiny, M., Liu, B., Peng, X., Elgammal, A.: A generative adversarial approach for zero-shot learning from noisy texts. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1004–1013 (2018)
29. Znaidia, A., Le Borgne, H., Hudelot, C.: Tag completion based on belief theory and neighbor voting. In: Proceedings of the 3rd ACM conference on International Conference on Multimedia Retrieval (ICMR). pp. 49–56. ACM (2013)
30. Znaidia, A., Shabou, A., Le Borgne, H., Hudelot, C., Paragios, N.: Bag-of-multimedia-words for image classification. In: Pattern Recognition (ICPR), 2012 21st International Conference on. pp. 1509–1512. IEEE (2012)
31. Znaidia, A., Shabou, A., Popescu, A., Le Borgne, H., Hudelot, C.: Multimodal feature generation framework for semantic image classification. In: ACM International Conference on Multimedia Retrieval (ICMR 2012) (june 2012)