Hascoet *et al. EURASIP Journal on Image and Video Processing* (2019) 2019:13
https://doi.org/10.1186/s13640-018-0371-x

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Semantic embeddings of generic objects for zero-shot learning

Tristan Hascoet[1*], Yasuo Ariki[1,2] and Tetsuya Takiguchi[1,2]

Abstract

Zero-shot learning (ZSL) models use semantic representations of visual classes to transfer the knowledge learned from a set of training classes to a set of unknown test classes. In the context of generic object recognition, previous research has mainly focused on developing custom architectures, loss functions, and regularization schemes for ZSL using word embeddings as semantic representation of visual classes. In this paper, we exclusively focus on the affect of different semantic representations on the accuracy of ZSL. We first conduct a large scale evaluation of semantic representations learned from either words, text documents, or knowledge graphs on the standard ImageNet ZSL benchmark. We show that, using appropriate semantic representations of visual classes, a basic linear regression model outperforms the vast majority of previously proposed approaches. We then analyze the classification errors of our model to provide insights into the relevance and limitations of the different semantic representations we investigate. Finally, our investigation helps us understand the reasons behind the success of recently proposed approaches based on graph convolution networks (GCN) which have shown dramatic improvements over previous state-of-the-art models.

Keywords: ZSL, Semantic embedding

1 Introduction

Recent successes in generic object recognition have largely been driven by the successful application of convolutional neural networks (CNN) trained in a supervised manner on large image datasets. One main drawback of these approaches is that they require a large amount of annotated data to successfully generalize to unseen image samples. The collection and annotation of such dataset for custom applications can be prohibitively complex and/or expensive which hinders their applications to many real-world practical scenarios. To reduce the number of training samples needed for efficient learning, *few-shot learning* techniques are being actively researched. The zero-shot learning (ZSL) paradigm represents the extreme case of few-shot learning in which recognition models are trained to recognize instances of a set of target classes without any training sample to learn from.

To recognize unseen classes, ZSL models use descriptions of the visual classes, i.e., representations of the visual classes in a non-visual modality. Research in ZSL has been driven by relatively small-scale benchmarks [1, 2] for which human-annotated visual attributes are available as visual class descriptions. In the case of generic object recognition, however, manually annotating each and every possible visual class of interest with a set of visual attributes is impractical. Hence, generalizing the zero-shot learning approaches developed on such benchmarks to the more practical case of generic object recognition comes with the additional challenge of collecting suitable descriptions of the visual classes.

Finding such description presents two challenges: first, the collection of these descriptions must be automated so as to not require an expensive human annotation process. Second, the collected descriptions must be visually discriminative enough to enable the zero-shot recognition of generic objects. Word embeddings are learned in an unsupervised manner from large text corpora so that they can be collected in a large scale without human supervision. Furthermore, their successful application to a number of natural language processing (NLP) tasks has shown that word embedding representations encode a number of desirable semantic features, which have been naturally assumed to generalize to vision tasks. For these desirable properties, word embeddings have become the standard

*Correspondence: tristan.hascoet@me.cs.scitec.kobe-u.ac.jp
[1]Kobe University, 1-1 Rokkodaicho, Nada Ward, Kobe 657-0013, Japan
Full list of author information is available at the end of the article

visual class descriptions used by recent zero-shot generic object recognition models ([3–7]).

In addition to word embeddings, we argue that generic objects can also be described by either text documents or knowledge graph data that satisfy our requirements: these descriptions both contain visually discriminative information and are automatically collectible from the web in a large scale, without requiring human intervention.

This paper aims to discuss the role and the affect of semantic representations on zero-shot learning of generic objects. To do so, we conduct an extensive empirical evaluation of different semantic representations on the standard generic object ZSL benchmark. We investigate the use of both different raw descriptions (i.e., different text documents and knowledge graphs collected from the web) and different embedding models in each semantic modality (word, graph, and document embedding models).

The main result of our study is to show that a basic linear regression model using graph embeddings outperforms previous the state-of-the art ZSL models based on word embeddings. This result highlights the first-class role of semantic representations in ZSL accuracy, a topic that has been relatively little discussed in the recent literature. We believe that our results emphasize the need for a better understanding of the nature of the information needed from semantic representations to recognize unseen classes. To shed some light on these results, we then discuss the efficiency, relevance, and limitations of different semantic representations.

Finally, our investigation allows us to better understand the outstanding results recently presented in [8, 9]. In particular, we show that much of the improvement shown by these models over the previous state of the art can be attributed to their use of explicit knowledge about the relationships between training and test classes as made available by knowledge graphs.

The remainder of this paper is organized as follows: Section 2 reviews the literature for related work, Section 3 presents the preliminaries to understand our results, and Section 4 presents the methodology used in our experiments. Section 5 is the main section of our paper in which we present and discuss the results of our investigation.

2 Related work

While the majority of the works on zero-shot generic object recognition have used word embeddings as semantic features, some works have explored the use of different semantic features, which we present in this section.

In [10], the authors use different linguistic resources to derive semantic similarity scores between classes and between classes and attributes and to automatically mine attribute-classes correspondence. Similar to this work, they automate the acquisition of semantic data from knowledge bases, but they focus on deriving semantic similarity scores and part attributes while we evaluate graph embedding models.

Mensink et al. [11] uses visual class co-occurrence statistics to perform ZSL. Given a training set of multi-labeled images and similarity scores between known and unknown labels, they use the co-occurrence distribution of known labels to predict the occurrence of unknown labels in test images. Their multi-label classification setting differs from the ZSL setting in which input images are classified into a unique class.

Mukherjee and Hospedales [12] question the limits of using a single data point (word embedding vectors) as semantic representations of visual classes because this setting does not allow the representation of intra-class variance of semantic concepts. They used Gaussian distributions to model both semantic and visual feature distributions of the visual classes.

More related to this work, [13] investigates different semantic representations for zero-shot action recognition. They compare different representations of documents and videos, while we investigate the application of word, document, and knowledge graph embeddings to zero-shot recognition of generic objects.

A series of works of [14–16] compares the zero-shot classification accuracy obtained with semantic representations derived from words, taxonomy, and manual attribute annotations on fine-grained or small-scale ZSL benchmarks. Our investigation differs in that we are concerned with the more practical task of generic object recognition, and we investigate a broader class of semantic features.

3 Preliminaries

3.1 Semantic data acquisition

To conduct our study, we are heavily dependent on the data available to us in the form of image/semantic description pairs (x, y). We use the ImageNet dataset as our starting point as it has become the standard evaluation benchmark for generic object ZSL. In ImageNet, visual classes are indexed by WordNet [17] concepts, which are defined by three components that correspond to the three semantic modalities we investigate: their lemmas (a set of synonym words that refer to the concept), a definition in natural language, and a node connected by a set of predicate edges to other concept nodes of the WordNet knowledge graph. Figure 1 illustrates the different semantic descriptions provided by WordNet.

In addition to the descriptions directly provided by WordNet, we investigate the use of semantic descriptions gathered from larger databases openly available on the web. WordNet has been integrated to the linked open data [18] cloud, which provides an interlinking between resources of different open knowledge bases.

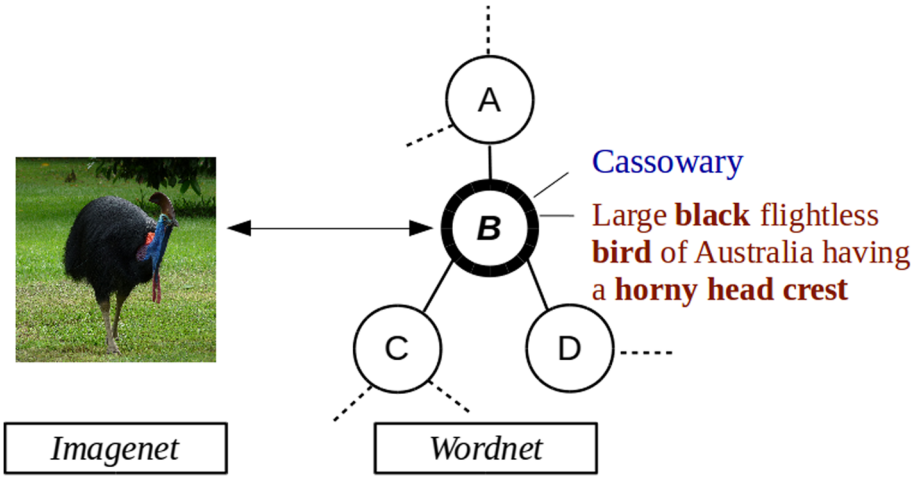

Fig. 1 Illustration of the different description levels of generic object classes. ImageNet classes are indexed by WordNet concepts which are defined by their lemmas (word level, in blue), definition (document level, in red), and structured data (graph level, in black)

These links define equivalences between WordNet concepts and resources of other knowledge bases. Following the links of the LOD cloud, as illustrated in Fig. 2, we are able to collect descriptions of ImageNet classes from different knowledge bases in a fully automated process. In our experiments, we use the BabelNet knowledge graph as augmented graph-level descriptions and Wikipedia articles as augmented document-level descriptions. Table 1 summarizes a few statistics about these datasets. Both the text and graph data collected from these datasets are considerably larger than the original WordNet descriptions.

3.2 Word embeddings

Distributional semantic models (DSM) and neural word embeddings are two related classes of models that learn continuous distributed representations of words. These models implement the distributional hypothesis that states that the meaning of words can be defined by

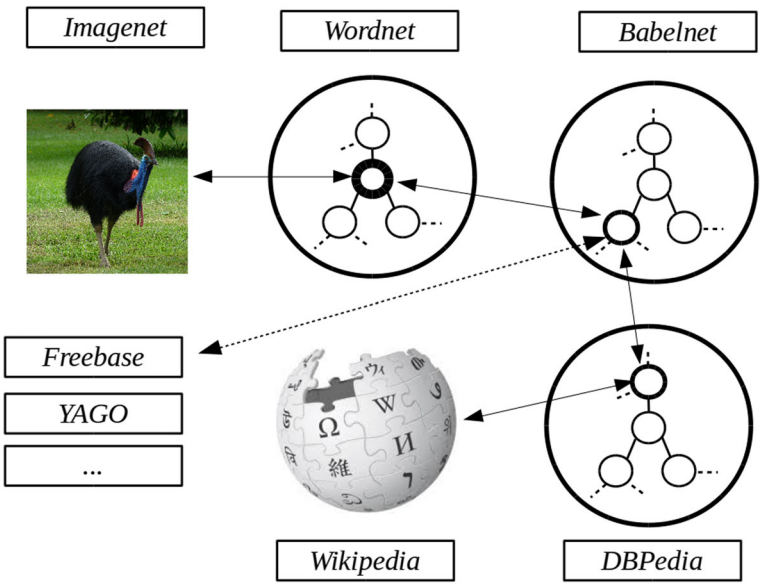

Fig. 2 Illustration of our linking process. ImageNet classes are indexed by WordNet concepts. WordNet concepts are linked to BabelNet concepts. BabelNet concepts are linked to DBPedia concepts. Each concept in DBPedia is linked to a Wikipedia article. Following links between linked open datasets allow us to collect rich descriptions of ImageNet visual classes. Our evaluation focused on BabelNet graph embeddings and Wikipedia articles embeddings but other knowledge bases such as Freebase or YAGO may be used interchangeably

Table 1 Comparison of knowledge base statistics

	Documents		Graphs		
	Doc	W/doc	Nodes	Edges	Triples
WordNet	117 k	10	117 k	20	372 k
BabelNet	–	–	15 M	2.3 k	1.3 G
Wikipedia	5.6 M	630	–	–	–

(Left) number of document and average size (word per documents) of WordNet definitions vs. Wikipedia articles. (right) number of nodes, edge types, and triples of WordNet vs. BabelNet knowledge graphs

the context in which they occur. DSMs explicitly factorize matrices of word co-occurrence statistics while neural word embedding models learn word representations by stochastic optimization methods. The latter typically samples individual words and their context from large text corpora, maximizing a similarity score between co-occurring words. These approaches have been extensively studied both theoretically [19] and practically [20]. In [19], the authors show that the skip-gram word2vec model with negative sampling implicitly factorizes a shifted PMI matrix, suggesting that both approaches are qualitatively similar. For the sake of our discussion in Section 5, we will consider that word embedding models do implicitly factorize matrices derived from word co-occurrence statistics following [19]. While qualitatively similar, the empirical study of [20] showed that neural embedding approaches tend to outperform DSM models on standard benchmarks. We evaluate three state-of-the-art embedding models on our ZSL benchmark: GloVe [21], FastText [22], and word2vec [23].

3.3 Graph embeddings

A knowledge graph can be formalized as a set of facts $\mathcal{G} = \{(s,p,o) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$. Each fact in the graph consists of a predicate (edge) $p \in \mathcal{R}$ and two entities (nodes) $s,o \in \mathcal{E} \times \mathcal{E}$, respectively referred to as the subject and object of the triple. Each triple denotes a relationship of type p between the subject s and the object o. Knowledge graph embedding models include tensor decomposition and neural embedding models. In [24], the authors show that simple neural embedding baselines such as DistMult [25] tend to outperform more sophisticated approaches on several benchmark knowledge base completion tasks, which leads us to focus on baseline neural embedding models. Neural embedding models learn d-dimensional vector representations of entities $\left\{e_i \in \mathbb{R}^d, \forall i \in \mathcal{E}\right\}$ and relations $\left\{r_i \in \mathbb{R}^d, \forall i \in \mathcal{R}\right\}$ by maximizing a scoring function $\psi\left(e_s, r_p, e_o\right)$ for triples $(s,p,o) \in G$. Learning is performed stochastically by minimizing a loss function $\mathcal{L}$ over the score of randomly sampled triples:

$$e^*, r^* = \operatorname{argmin}\left(\mathbb{E}_{(s,p,o)\in\mathcal{G}}\mathcal{L}\left(\psi\left(e_s, r_p, e_o\right)\right)\right) \quad (1)$$

Different embedding models differ in their choice of scoring function $\psi(e_s, r_p, e_o)$ and loss function $\mathcal{L}$ used for training. Table 2 summarizes the scoring function of popular models we evaluate in Section 5. While experimenting with these models, we found that a slight modification to the TransE objective function to yield better accuracy. We include these embeddings, denoted as TransE*, in our experiments.

Recently, a series of work have shown the merits of hyperbolic geometry to embed structured data in a continuous space. The Poincarre embeddings [28] have been successfully applied to embed the WordNet hierarchy, dramatically reducing the reconstruction error compared to previous embedding approaches based on Euclidean geometry. We include their model in our evaluation for the impressive results they report.

3.4 Textual embeddings

In this section, we present embedding models for text documents. As different models are concerned with documents of different scale, we separately present embedding models for short, sentence-like documents (i.e., WordNet definitions) and models concerned with full-text documents (i.e., Wikipedia articles).

Sentence embeddings—universal sentence embedding models learn continuous distributed representations of sentences. Different embedding models differ in their architecture and the objective function they use for training. Most current state-of-the-art universal sentence embedding models use either a bag of words (BoW) or a recurrent neural network (RNN) architecture. As training objectives, these models are either trained in an unsupervised or supervised manner on auxiliary task. Supervised objectives include natural language inference [29], image captioning [30], or the regression of word embeddings from dictionary definitions [30]. Unsupervised models can be classified into intra-sentence and inter-sentence training objectives. Intra-sentence objective models [31] learn sentence embedding based only on the words contained in the sentence. Inter-sentence objectives [32, 33] use the ordering of sentences in large text corpora as training signal, in a similar fashion to word embeddings. Table 3 summarizes the characteristics of the different embedding models evaluated in our study.

Table 2 Neural embedding scoring and loss functions

Model	$\psi(e_s, r_p, e_o)$	$\mathcal{L}$
TransE [26]	$\|e_s + r_p - e_o\|$	L_2
DistMult [25]	$\langle e_s, r_p, e_o \rangle$	Ranking
ConvE [27]	$f(vec(f([e_s; r_p] * w))W)e_o$	BCE
TransE*	$\langle e_s + r_p, e_o \rangle$	Triplet

Table 3 Sentence embedding models' architecture and objectives

Model	Architecture	Objective
Infersent [29]	RNN	SNLI
DictRep [30]	BoW	Word embedding regression
CapRep [30]	BoW	Image captioning
Sent2Vec [31]	BoW	Intra
SkipThought [32]	RNN	Inter
FastSent [33]	BoW	Inter

Document embeddings—two of the most popular methods for embedding documents are latent semantic indexing [38] and latent Dirichlet allocation [39]. Latent semantic indexing performs singular value decomposition on a matrix of term/document occurrence. The TF-IDF model was introduced as a weighting factor to reduce the impact of frequently occurring words and has been shown to improve search results on document retrieval tasks.

In Section 5, we evaluate Wikipedia article embeddings based on these models and apply sentence embedding models to WordNet definitions.

3.5 Graph convolution networks

Graph convolution networks (GCN) were first introduced in [34] for the task of semi-supervised classification on graph structured data. The work of [9] first adapted this model to zero-shot learning, and [8] proposed a number of improvements to their original method. These works cast the ZSL problem as a regression task in which the model learns to regress the weights of a softmax classifier on the set of unknown test classes. They use word embeddings as input features to the model and the WordNet hierarchy as graph structure. Through the local actions of the GCN layers, input features are propagated along the WordNet hierarchy until the top layer that regresses the weights of the test set classifier.

It should be noted that these works slightly differ from our study as they use visual supervision from the weights of a pretrained ResNet classifier to learn a similarity measure between semantic descriptions and images, while we restrict our study to semantic features learned independently from the visual and core ZSL modules to structure our results.

However, these works have shown impressive improvement over previous models: successfully combining graph and word embeddings, they have more than doubled the classification accuracy obtained by previous state of the art accuracy. In Section 5, we use the results of our investigation to explain these impressive results.

4 Method

The general architecture of ZSL models can be seen as the combination of three modules $\{V, S, E\}$, as illustrated in Fig. 3. The visual module V extracts high-level visual features $V(x)$ from raw input images x; the semantic module S extracts semantic features $S(y)$ from raw descriptions y of the visual classes, and the core ZSL module E computes

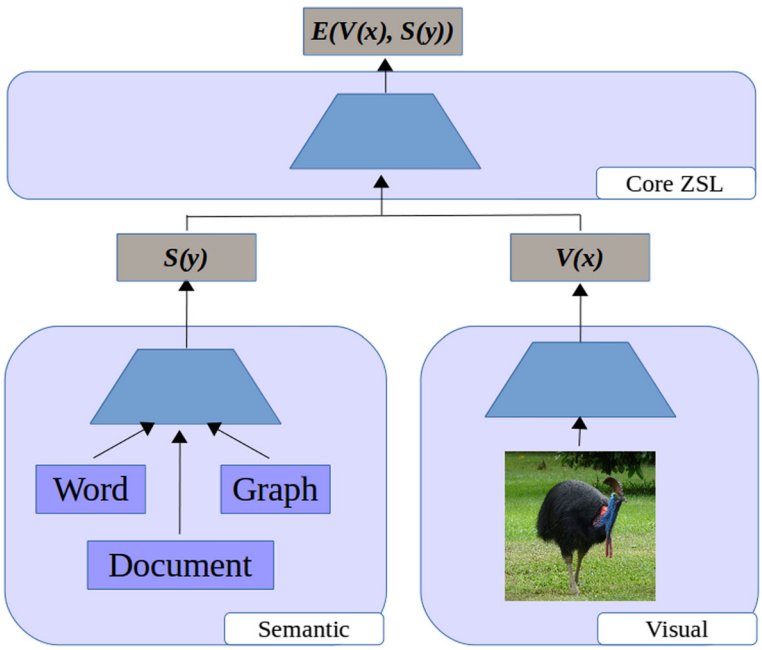

Fig. 3 Illustration of ZSL models architecture. The general architecture of ZSL can be seen as the combination of a visual module *V*, a semantic *S*, and a core ZSL module *E*

a similarity score $E(V(x), S(y))$ between semantic and visual features.

ZSL models aim to generalize the classification ability of traditional image classifiers to out-of-sample classes for which no image sample is available to learn from. To evaluate the out-of-sample recognition ability of models, ZSL benchmarks split the full set of classes C into disjoint training and test sets.

$$C_{\text{train}} \cup C_{\text{test}} \subset C \tag{2a}$$

$$C_{\text{train}} \cap C_{\text{test}} = \emptyset \tag{2b}$$

$$\text{Tr} = \{(x, y_c), c \in C_{\text{train}}\} \tag{2c}$$

$$\text{Te} = \{(x, y_c), c \in C_{\text{test}}\} \tag{2d}$$

Learning is performed by minimizing a loss function $\mathcal{L}$ over the regularized similarity score of the set of training samples with respect to the model parameters θ.

$$\theta^* = \text{argmin}_\theta \mathbb{E}_{(x,y)\in Tr} \mathcal{L}(E_\theta(V(x), S(y)) + \Omega(\theta) \tag{3}$$

At test time, an image x_{test} can be classified among the set of unseen test classes by retrieving the class description y of highest similarity score.

$$c = \text{argmax}_{c\in C_{\text{test}}} E(V(x_{\text{test}}), S(y_c)) \tag{4}$$

The visual and semantic modules can either be learned jointly with the core ZSL module in an end-to-end procedure by back-propagation of the error signal from the core ZSL module to the two lower modules, or they can be learned independently on unsupervised or auxiliary supervised tasks (e.g., pretraining the visual module on the ILSVRC classification task and pretraining the semantic module as an unsupervised word embedding model).

Our work focuses exclusively on the semantic module: we question what raw descriptions y and embedding module S provide semantic features $S(y)$ that are most visually discriminative so as to enable zero-shot recognition of generic objects. We restrict our study to embedding models S learned independently from the other modules, without visual supervision from the ZSL module.

We use the top layer activations of a pretrained ResNet50 as visual feature representations $V(x)$. We investigate different embedding models S and raw semantic descriptions y in the form of words, text documents, and knowledge graphs as semantic features $S(y)$. Our ZSL module consists of a ridge regression from the visual feature space to the semantic feature space.

Let us denote by (X, Y) the matrix representation of stacked visual and semantic features of the training set, we learn a projection matrix W from visual feature space to semantic feature space as:

$$W^* = \min_W \left(||XW - Y||^2 + \lambda||W||^2\right) \tag{5a}$$

$$W = \left(XX^T + \lambda I\right)^{-1} XY^T \tag{5b}$$

At test time, similarity scores are given by the Euclidean distance between the projection of test images x in semantic space and the test class semantic features

$$c^* = \text{argmin}_{c\in C_{\text{test}}} ||V(x)W - S(y_c)|| \tag{6}$$

We use the simplest ZSL module possible for interpretability, to emphasize the importance of the semantic features, although we found qualitatively similar results with more sophisticated models. Previous works [35] have shown that the Hubness problem negatively impact the ZSL accuracy of the ridge regression model. We found this to be a non-problem as normalizing semantic features to unit norm solves the distance concentration in semantic space.

Following previous works [3–7], we use the 1000 classes of the ILSVRC 2012 image classification dataset as training set and evaluate the accuracy of different semantic embeddings on the 2-hop, 3-hop, and all test splits proposed in [5].

When openly available, we always prefer the reference implementation of the semantic modules we evaluate. We re-implement the models for which no open implementation has been released. All implementations are made available on the GitHub page of the project[1]. Detailed hyperparameters and training settings are also accessible on the project page. Experiments with word embeddings were performed using the pretrained vectors as released by the original papers.

5 Results and discussion

In this section, we first present the evaluation of the different semantic embeddings on the standard ImageNet zero-shot learning benchmark in Section 5.1. In Section 5.2, we conduct an error analysis on the 2-hop test split to shed some light on the good performance of graph embeddings. Section 5.3 discusses the results of the recently proposed models of [8, 9] and relate their results to our study. Finally, Section 5.4 discusses the relevance of word embeddings and quantify some of their limitations.

5.1 Standard evaluation

Table 4 presents the results of our evaluation on standard test splits used in previous works [3–7].

State of the art—the bottom section of the table presents the state-of-the art results obtained using word2vec embeddings as reported in [4], as well as the recent results of GCN-based approaches [8, 9]. As mentioned in the previous sections, [8, 9] have brought dramatic improvements to previous states of the art.

Table 4 Results on the standard ImageNet ZSL test splits

			2-hop			3-hop			All		
ZSL module	Description	Semantic module	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10
Ridge reg.	WordNet lemmas	word2vec	7.66	21.00	29.90	3.08	9.35	13.78	0.89	2.70	4.23
		FastText	12.98	32.35	41.68	2.96	9.01	13.20	1.30	4.00	6.01
		Glove	**13.47**	**32.96**	**42.99**	**3.08**	**9.35**	**13.78**	**1.34**	**4.15**	**6.30**
Ridge reg.	WordNet graph	TransE	5.77	8.73	10.16	1.07	1.71	1.99	0.42	0.65	0.76
		DistMult	16.94	37.61	43.85	3.28	9.57	12.39	1.35	3.91	5.08
		TransE*	**20.13**	**48.32**	**58.06**	**3.65**	**11.84**	**17.05**	**1.51**	**4.90**	**7.21**
		ConvE	3.23	9.12	12.46	1.30	2.14	3.26	0.42	1.72	3.10
		Poincarre	11.81	28.86	37.53	2.02	5.93	8.79	0.79	2.32	3.46
Ridge reg.	BabelNet graph	TransE	2.82	5.11	7.16	1.03	1.41	1.75	0.37	0.88	1.01
		DistMult	8.42	20.31	27.33	1.82	5.14	7.64	0.78	2.23	3.40
		TransE*	17.76	42.46	53.47	3.62	10.82	15.65	1.53	4.69	6.97
Ridge reg.	WordNet definitions	InferSent	4.06	12.37	18.52	1.18	3.91	6.15	0.49	1.66	2.67
		DictRep	**6.06**	**18.74**	**27.28**	**1.52**	**5.58**	**9.05**	**0.63**	**2.32**	**3.84**
		CapRep	3.45	10.86	16.37	1.13	2.97	4.35	0.21	0.56	1.01
		Sent2vec	5.93	17.57	25.57	1.65	5.60	8.92	0.67	2.36	3.85
		FastSent	1.82	5.31	9.86	0.82	2.11	3.21	0.19	0.43	0.75
		SkipThought	0.50	1.38	2.11	0.17	0.46	0.67	0.06	0.17	0.26
Ridge reg.	Wikipedia articles	TFIDF	9.03	26.53	37.31	-	-	-	-	-	-
State of the art											
SYNC [6]	WordNet lemmas	word2vec	**9.26**	-	-	**2.29**	-	-	**0.96**	-	-
CONSE [7]			7.63			2.18			0.95		
ESZSL [36]			6.35			1.51			0.62		
ALE [16]			5.38			1.32			0.5		
LATEM [15]			5.45			1.32			0.5		
SJE [14]			5.31			1.33			0.52		
DEVISE [5]			5.25			1.29			0.49		
CMT [37]			2.88			0.67			0.29		
GCNZ [9]	Lemmas and graph	Glove&GCN	19.8	53.2	65.4	4.1	14.2	20.2	1.8	6.3	9.1
ADGPM [8]			**26.6**	**60.3**	**72.3**	**6.3**	**19.3**	**27.7**	**3.0**	**9.3**	**13.9**

The upper part of the table shows our results using a ridge regression model with different semantic representations. The bottom part of the table shows the state-of-the-art results as reported in [4], with the additional entries of [8, 9]. Bold entries represent the best results obtained in each category

Word embeddings—Table 4 highlights striking differences in the performance between the word2vec embeddings used in previous works [3–7] and both GloVe and FastText embeddings. GloVe embeddings almost double the word2vec the top-1 accuracy of the 2-hop split, effectively outperforming previous states of the art using word2vec, as presented in [4]. However, even the best-performing embeddings, GloVe, fall behind the graph embedding results. In Section 5.4, we further discuss the application of word embeddings to zero-shot learning and highlight some of their limitations that partially explain their relatively poor performance.

Sentence embeddings —we expected sentence embeddings to provide strong representations as sentence embedding has been a very active research topic for the past years. Surprisingly, sentence embedding models performed poorly, even slightly underperforming the word2vec embeddings. We observe that models trained with supervised training signals tend to perform relatively better than modes trained in an unsupervised manner.

Among unsupervised models, models trained with intra-sentence signals like Sent2vec seem to perform better than SkipThought or FastSent that use inter-sentence training signal.

Graph embeddings—graph embeddings performed remarkably well. In particular, the TransE∗ model outperformed the best performing word embedding model by an absolute 6.66% in top-1 accuracy on the 2-hop split (20.13% vs. 13.47%).

The Poincarre embeddings are learned from the WordNet hierarchy alone so that they do not embed explicit visual information such as object part attributes. It is remarkable that they outperform word2vec.

ConvE embeddings performed poorly in contrast. This result seems to make sense: the ConvE model uses a non-linear similarity function which is not designed to learn linearly separable embeddings whereas we use a linear model to measure the similarity between visual and semantic features.

In the next section, we further discuss the results of graph embeddings by analyzing the errors of the TransE* embeddings on the 2-hop dataset.

Data augmentation—Wikipedia document embeddings performed better than sentence embeddings. However, one major limitation of this approach is that many links between WordNet concepts and Wikipedia articles could not be recovered from LOD data. Only 60% of visual classes were successfully matched to a Wikipedia article. Hence, we manually recovered the missing links for the 2-hop set (623 missing classes), but we did not recover the missing links for classes of larger test splits as manual linking is very time consuming.

Surprisingly, augmenting the WordNet knowledge graph with BabelNet did not improve on the model's accuracy.

5.2 Error analysis on the 2-hop test split

To better understand the results presented in Table 4, we conduct an error analysis on the 2-hop test split. We are particularly interested in the 2-hop test split because its configuration allows for a trivial solution against which we can compare the output of our model. As illustrated in Fig. 4, the 2-hop split consists of the set of test classes that are directly connected to at least one of the training classes in the WordNet hierarchy. In other words, every test class of the 2-hop split is either a child class of one of the training classes or the parent class of at least one of the training classes.

In this section, we first introduce a simple procedure using the explicit information of the WordNet hierarchy to classify images into the 2-hop test set. We then show that the action of this procedure can be simulated by a trivial semantic embedding of the visual classes within our linear regression framework. Finally, we compare the classification output of this trivial solution to the output of different semantic embeddings. In Section 5.2.3, we elaborate on this comparison to get a better understanding of the reasons behind the high classification accuracy of graph embeddings.

5.2.1 Trivial algorithm

The configuration of the 2-hop test split, illustrated in Fig. 4, allows us to derive a simple procedure to classify test images into the 2-hop test set in two steps: first, classify a test image x into a class c' of the *training set* and then

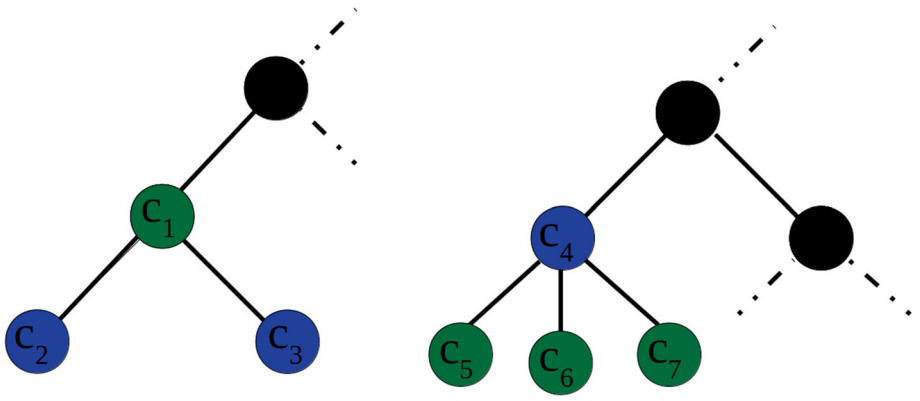

Fig. 4 Illustration of the 2-hop test set configuration. This figure illustrates the configuration of test classes ($\{c_1, c_5, c_6, c_7\} \in C_{\text{test}}$ in green) in relationship to training classes ($\{c_2, c_3, c_4\} \in C_{\text{train}}$ in blue) in the WordNet hierarchy. Test classes are directly connected to the training classes. This leads to two possible configuration. **a** On the right, test classes are direct parent of one or more training class: $\text{Tr}(c_1) = \{c_2, c_3\}$ and $\text{Te}(c_2) = \text{Te}(c_3) = \{c_1\}$. **b** On the left, test classes are direct children of one training class. There may be one or more children test classes for a given training class: $\text{Tr}(c_5) = \text{Tr}(c_6) = \text{Tr}(c_7) = \{c_4\}$ and $\text{Te}(c_4) = \{c_5, c_6, c_7\}$

perform the final classification by randomly assigning x to one of the child/parent *test class* c of c'. Denoting by $\mathrm{Te}(c')$ the set of child/parent test classes of a given training class c' (see Fig. 4b), we formalize this procedure in Algorithm 1.

Algorithm 1 Trivial solution algorithm

INIT Associate to each training class the subset of test classes to which they are directly connected within the Wordnet hierarchy:
$Te : C_{train} \rightarrow C \subset C_{test}$

For all test image x
Step 1: Classify x into a class $c' \in C_{train}$ using the pretrainedResNet classifier.
Step 2: Randomly assign x to a test class $c \in Te(c')$ associated to c'.
End For

It should be noted that this algorithm simply exploits a trivial solution by taking advantage of the configuration of the test split. As such, it does not represent an interesting end solution to the problem of zero-shot generic object recognition; rather, we introduce this algorithm as a mean to understand the good results obtained by graph embeddings.

5.2.2 Trivial embeddings

Interestingly, Algorithm 1 can be emulated within our framework using the following trivial semantic embedding scheme.

First, we assign randomly generated d-dimensional vectors as semantic embeddings to the training classes. Second, we assign to test classes the same semantic embedding as their child/parent training class. In the case where a test class c is the parent class of several training classes, we assign to c the mean of its children training class embeddings.

Denoting by $Tr(c)$ the set of child/parent training classes of a given test class c (see Fig. 4a), we formalize this embedding scheme as follow:

$$\mathrm{Tr} : C_{\mathrm{test}} \rightarrow C \subset C_{\mathrm{train}} \tag{7a}$$

$$\mathrm{Tr} : c \rightarrow \{c'_1, ..., c'_N\} \tag{7b}$$

$$y_{c'} \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \in \mathbb{R}^d, \qquad \forall c' \in C_{\mathrm{train}} \tag{7c}$$

$$y_c = \frac{1}{|Tr(c)|} \sum_{c' \in Tr(c)} y_{c'} + e_c \qquad \forall c \in C_{\mathrm{test}} \tag{7d}$$

$$e_c \sim \mathcal{N}(\mathbf{0}, \mathbf{1} \times 10^{-8}) \in \mathbb{R}^d, \qquad \forall c \in C_{\mathrm{test}} \tag{7e}$$

where $y_{c'}$ denotes the randomly generated semantic vectors of training classes, and y_c denotes the semantic vectors of test classes.

By assigning to test classes the same semantic embedding as their related training class, we effectively equate them to the same class. Hence, test images that are correctly classified into their child or parent training class are correctly classified into their actual test class.

e_c represents a small Gaussian noise we add to the test class embeddings to differentiate between children test classes of the same training class (Fig. 4b). This is necessary to get deterministic classification results at test time. Otherwise, classification (Eq. 6) would result in undefined behavior as several test classes share the same semantic embedding.

5.2.3 Result analysis

In this section, we compare the classification outputs of our model, using different semantic embeddings, to those of the trivial solution. To compare both outputs, we look at the Pearson correlation between their class-wise accuracy results. For readability, we only report the results of the best performing embedding of each semantic modality: Glove, Sent2vec, and TransE* in Table 5.

First, we observe that the trivial solution performs remarkably well. Compared to to the results presented in Table 4, the trivial solution outperforms all previous work on ZSL with the notable exception of [8, 9]. Second, and most interestingly, we observe that the classification outputs of graph embeddings are strongly correlated to the trivial solution.

To understand why this is, we need to consider the nature of the information contained in the knowledge graphs. Child/parent relationships between the training and test classes are explicitly stored in the knowledge graphs. Consequently, graph embeddings of the test classes are explicitly optimized to be close to their related training class in the semantic space.

The trivial solution precisely consists of the ideal case where test class embeddings are equal to their child/parent training class. If the child/parent relationships between test and training classes were the only information contained in the knowledge graph, graph embeddings would essentially converge to the solution

Table 5 Correlation between the trivial and semantic embeddings class accuracy on the 2-hop split obtained

Semantics	Top-1	Top-5	Top-10	ρ_{trivial}
Trivial	20.30	51.94	62.23	1.00
TransE∗	20.13	48.32	58.06	0.84
Sent2vec	5.48	16.46	23.94	0.19
Glove	11.47	29.22	38.72	0.44

where the test class embeddings are equal to their child/parent training class embeddings, similar to the trivial embeddings.

However, knowledge graphs also contain information about other properties of the visual classes, such as their part attributes. This additional information further constrains the graph embedding representations, but does it provide useful visual clues to increase zero-shot recognition accuracy? This is difficult to assess, but two different observations suggest otherwise: First, no graph embedding outperform the trivial solution; second, BabelNet is a much larger knowledge graph than WordNet, i.e., it contains many more such properties about the visual classes, in addition to the hierarchical information. Despite this rich information, BabelNet embeddings perform poorly, in comparison to the WordNet embeddings (see Table 4).

These observations suggest that the main reason behind the graph embedding efficiency is that they explicitly model the hierarchical relationships between training and test classes. Word embeddings and sentence embeddings, on the other hand, do not explicitly model these relationships, which explains their lower correlation to the trivial solution.

5.3 Graph convolutional networks

In this section, we relate the results of our study to the recently proposed methods based on GCN [8, 9] (cf. Section 3.5). To do so, we first conduct an ablation study on a vanilla 2-layer GCN model. We first replace the GloVe embeddings by randomly generated vectors as input features to the GCN model. This allows us to evaluate the accuracy of the GCN model due only to the explicit information of the WordNet hierarchy. We then evaluate the full GCN model with GloVe input features, as proposed in the original work. Table 6 shows the classification accuracy of both models, as well as their Pearson correlation score to the trivial solution, similar to Section 5.2. In addition, we show the training time of these models to highlight the relative simplicity of our model.

5.3.1 Random input features

The results of our ablation study are appealing. Even with random input features, the GCN model outperforms previous approaches based on word embeddings. Interestingly, the accuracy of the GCN model with random input features plateaus around the accuracy of the trivial solution, with strong correlation. This suggests that the GCN model with random input features learns a similar solution to the trivial solution.

The graph convolution layers of a GCN model have been shown to perform a Laplacian smoothing operation [38]. In other words, graph convolution operations effectively draw the internal representations of neighboring nodes closer together in feature space. Hence, the semantic representation of test classes are drawn closer to the semantic representations of their child/parent training class. The trivial solution corresponds to the ideal case in which test class embeddings are equal to their child/parent training class, which explains why, with random input features, the GCN seems to fit the trivial solution.

Table 6 Comparison of GCN

Semantics	Top-1	Top-5	Top-10	p_{trivial}	Training time
Trivial	20.30	51.94	62.23	1	$\approx$ 1 s
TransE*	20.13	48.32	58.06	0.84	$\approx$ 1 s
GCN — random	20.87	51.93	63.26	0.79	$\approx$ 30 min
GCN — GloVe	23.47	56.10	67.61	0.82	$\approx$ 30 min

5.3.2 Glove features

As expected, using GloVe embeddings instead of random input features improves the accuracy of the GCN model. However, it is surprising that the correlation coefficient with the trivial solution does further increases with GloVe input features. This observation suggests that the impact of GloVe embeddings are twofolds.

First, the GloVe embeddings of a parent and its child classes are statistically closer in the semantic space than randomly generated vectors, because these concepts are, by definition, semantically close. Hence, using GloVe embeddings instead of random input features facilitates the Laplacian smoothing between training classes and their parent/child classes by assigning input features that are already close to each other in the semantic space. As input features of neighboring nodes are already close to each other, the hidden features of test classes are easier to fit their parent/child training class representations. As a consequence, GloVe embeddings help the GCN model fit the trivial solution, as illustrated by the increase in their correlation score.

Second, the implicit information provided by word embeddings seems to provide a strong heuristic to break ties between test classes that share a common training class parent. The trivial solution randomly breaks ties between children test classes of the same training class (step 2 of Algorithm 1), which is sub-optimal. Instead, GloVe vectors provide better visual clues to differentiate between different neighboring test subclasses.

In conclusion, we showed that most of the improvement that GCN have brought over previous state of the art can be attributed to the use of the explicit WordNet hierarchy information. We showed that, without word embeddings, the GCN model converges toward a solution similar to the trivial and graph embedding solutions. However, the GCN model also efficiently combine the implicit information of word embeddings to the explicit relationships

of the WordNet hierarchy. In the next section, we further discuss the visual clues embedded in word embeddings.

5.4 Word embeddings

It is surprising that complex ZSL models using word embeddings as semantic features are largely outperformed by a simple regression model using graph embeddings as semantic features. To conclude this paper, we discuss the relevance of word embeddings in the context of ZSL. We then highlight and quantify some of their limitations that partially explains their poor performance.

5.4.1 Relevance

In NLP, word embeddings are implementations of the distributional hypothesis, which suggests that the meaning of words can be inferred from the context in which they occur. In contrast, ZSL models using word embeddings as semantic features make the assumption that the *appearance of generic objects* can be characterized by the context in which their lemmas occur. Table 7 shows the co-occurrence frequency of a few common visual class lemmas with words that explicitly characterize visual attributes. This table shows that visual class lemmas tend to share high co-occurrence frequency with either their part attributes (i.e., both car and truck co-occur more frequently with wheel than bird and cassowary do) or with other lexical forms that implicitly impacts the shape of these objects (i.e., both cars and trucks are "drivable" vehicles). This suggests that the co-occurrence patterns of words in large text corpora indeed contain implicit information regarding distinctive visual features shared among classes. A particular exception that stands out from Table 1 is the word cassowary, a species of bird, which we discuss below.

5.4.2 Limitations

We conjecture that while coarse-grained visual classes are well defined by common words such as "car," "truck," and "bird," several difficulties arise when considering more fine-grained visual classes such as "cassowary" or "moving van."

N-grams—different from coarse-grained concepts, fine-grained concepts are often not best described by single words but by composition of words (e.g., *n*-grams such as "polar bear" or "blue jeans" vs. their unigram parent class "bear" and "trousers"). Figure 5 shows that 54.2% of ImageNet class lemmas are actually not single words but *n*-grams.

In a first experiment, we evaluate the impact of *n*-grams on the classification accuracy of our model. To do so, we split the ImageNet dataset class-wise into two subsets: one consisting of visual classes whose lemmas are single words and one with *n*-grams classes only. We train our ZSL model on the ILSVRC training set, as done in the previous experiments, and evaluate the accuracy of the model separately on the *n*-gram and unigram test class splits. For each split, we randomly sample 200 classes and evaluate the accuracy of the model on a 200 class classification problem. As different randomly sampled test sets yield different results, we repeat this operation 20 times per split with different randomly sampled test class sets, and we present the mean, first quartiles, and extrema of each split's top-1 accuracy in Fig. 5. *N*-gram embeddings were computed as the mean of their individual word embeddings.

Against our expectations, ZSL accuracy does not seem to suffer from the "*n*-gram-ness" of visual class lemmas as *n*-gram lemmas even outperform single word lemmas by an average of 2%.

Rare words—we found that fine-grained visual classes that are correctly defined by a single word tend to be defined by rare words (e.g., the rare lemma "cassowary" vs. the common lemma "bird" of its parent class). Word embeddings are learned from their co-occurrence statistics in large text corpora. While visual clues are embedded in word co-occurrence patterns, a considerable amount of noise (i.e., non-visually discriminative information) stems from random word co-occurrences. The example of cassowary is illustrated in Table 7. The word "cassowary" only occurs 792 times in the English Wikipedia corpus in which it does not co-occur once with the visual bird-like attributes "wings" or "beak." Instead, "cassowary" randomly co-occurs once with the word "drive." We conjecture that frequently occurring words provide more visually discriminative representations than rare words because of the higher "visual signal-to-noise ratio" of their co-occurrence statistics. Figure 6a shows the occurrence count distribution of ImageNet lemmas in the English Wikipedia corpus. This figure shows a large number of visual classes are defined by rare words. For example, we found that 9.7% of ImageNet lemmas appear less than 10 times in Wikipedia.

Table 7 Lemmas co-occurrence with visually discriminative words

	Car	Truck	Bird	Cassowary
Wheel	1.3×10^{-4}	1.6×10^{-4}	2.0×10^{-6}	0.0
Drive	1.0×10^{-3}	1.0×10^{-3}	2.2×10^{-5}	2.4×10^{-3}
Wings	4.0×10^{-6}	0.0	1.6×10^{-4}	0.0
Beak	0.0	0.0	5.2×10^{-5}	0.0
Occ.	4.7×10^{5}	7.8×10^{4}	1.4×10^{5}	7.9×10^{2}

Statistics presented in this table were gathered from the English Wikipedia corpus with a context window size of five words. The columns correspond to visual class lemmas, and rows correspond to visually discriminative words. The last row shows the number of occurrence of the visual class lemmas in the corpus. Upper rows show the frequency of occurrence of visually discriminative words within the context of visual class lemmas. For example, the upper left value denotes $p(\text{wheel}|\text{car})$

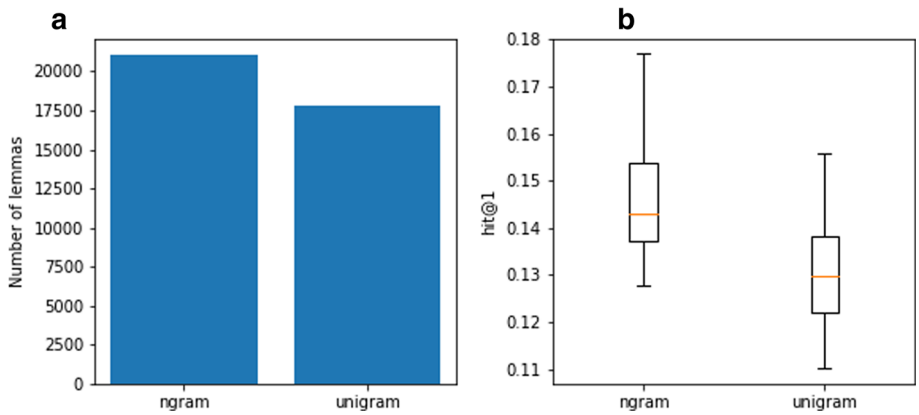

Fig. 5 Evaluation of ZSL accuracy of *n*-gram classes vs. unigram classes. Left—*n*-gram vs. unigram class lemmas distribution. Right—classification accuracy per lemmas type. **a** Distribution of *n*-gram vs. unigram lemmas. **b** ZSL accuracy results

To quantify the impact of lemma scarcity, we split the ImageNet dataset class-wise into 5 artificial subsets according to Fig. 6a. Each split k contains visual classes whose lemmas appear between 10^k and 10^{k+1} times in the English Wikipedia corpus. We evaluate each split individually on 20 randomly sampled subsets of 200 test classes, following the same protocol as the *n*-gram experiment.

Our results highlight a strong correlation between lemmas' frequency and classification accuracy (Fig. 6b). The first two splits (rare words) strikingly underperform mid-frequency terms with 6% and 7% mean accuracy compared to the 14% accuracy of the best performing splits. On the other hand, we found that very frequent words also show lower classification accuracy, which was unexpected.

Homonyms—finally, natural languages contain many homonyms which makes it difficult to uniquely identify visual classes with a single word. For example, a "(river) bank" and a "(financial) bank" share similar representations in a word embedding space while being two different visual concepts. The consequences of homonymy are twofolds: first, the semantic representation of homonym classes is learned from the co-occurrence statistics of the different meanings of the lemma which results in noisy embeddings; second, a mechanism to break ties between homonym visual classes must be given. We found that 13% of the ImageNet lemmas are shared with at least one other class and 38% of ImageNet classes share a lexical form with at least one other class. A rigorous evaluation of the impact of homonymy on ZSL

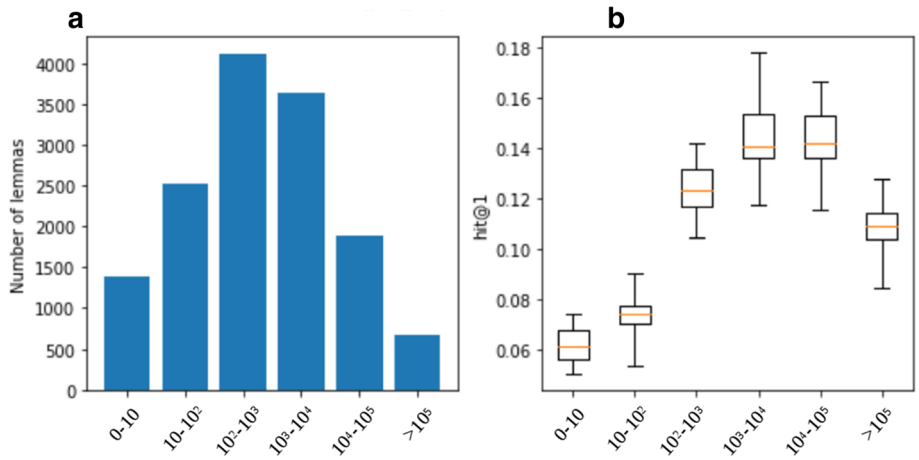

Fig. 6 Evaluation of ZSL accuracy by test class lemmas occurrence frequency. Left: visual class lemmas occurrence count distribution in the English Wikipedia corpus. Right: classification accuracy per occurrence count. **a** Lemmas occurence count distribution. **b** ZSL accuracy results

accuracy involves evaluating different heuristics to break ties between homonym classes which is beyond the scope of this work so we only mention it for completeness.

6 Conclusion

Zero-shot learning has the potential to be of great practical impact and to facilitate the wide-spread use of object recognition technologies. Despite almost a decade of active research [39], the accuracy of ZSL models on standard generic object recognition benchmarks remains too low to be considered for practical applications. In this paper, we presented a large-scale investigation of semantic representations applied to zero-shot recognition of generic objects. Our main result was to show that, given appropriate semantic embeddings, a basic linear regression model can outperform previous state-of-the-art models. In particular, we showed that explicit information about class relationships, as made available by knowledge graphs, provides strong empirical performance. Through these results, our investigation also indirectly lead us to a better understanding of the impressive gains in performance reported by recent works using GCN. We believe that these results call for a deeper discussion on the role of semantic representations and the nature of the information needed from these representations to achieve practical solutions to the problem of zero-shot recognition of generic objects.

Endnote

[1] https://github.com/TristHas/ZSL-semantics.

Abbreviations

CNN: Convolutional neural network; GCN: Graph convolution network; NLP: Natural language processing; ZSL: Zero-shot learning

Acknowledgements

The authors would like to thank the reviewers for their constructive comments and feedback.

Funding

This work was supported by a scholarship MEXT from the Japanese Ministry of Education, Culture, Sports, Science, and Technology. A part of this study is subsidized by JSPS Grant-in-Aid for Scientific Research and Research granted JP 17K00236. This work was supported in part by PRESTO, JST (Grant No. JPMJPR15D2).

Availability of data and materials

All the code and data used in this work are publicly available at https://github.com/TristHas/ZSL-semantics.

Authors' contributions

TH contributed the implementation and investigation presented in this work. TT and YA jointly supervised this project. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1] Kobe University, 1-1 Rokkodaicho, Nada Ward, Kobe 657-0013, Japan. [2] Association for Advanced Science and Technology, 1-1 Rokkodaicho, Nada Ward, Kobe 657-0013, Japan.

Received: 20 May 2018 Accepted: 1 November 2018
Published online: 15 January 2019

References

1. C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset (2011). California Institute of Technology
2. C. H. Lampert, H. Nickisch, S. Harmeling, in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference On*. Learning to detect unseen object classes by between-class attribute transfer (IEEE, 2009), pp. 951–958
3. E. Kodirov, T. Xiang, S. Gong, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Semantic autoencoder for zero-shot learning, (2017), pp. 3174–3183
4. Y. Xian, B. Schiele, Z. Akata, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Zero-shot learning-the good, the bad and the ugly, (2017), pp. 4582–4591
5. A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, *et al*, in *Advances in Neural Information Processing Systems*. Devise: a deep visual-semantic embedding model, (2013), pp. 2121–2129
6. S. Changpinyo, W. L. Chao, B. Gong, F. Sha, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Synthesized classifiers for zero-shot learning, (2016), pp. 5327–5336
7. M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, J. Dean, Zero-shot learning by convex combination of semantic embeddings (2013). arXiv preprint arXiv:1312.5650
8. M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, E. P. Xing, Rethinking knowledge graph propagation for zero-shot learning. arXiv preprint arXiv:1805.11724 (2018)
9. X. Wang, Y. Ye, A. Gupta, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Zero-shot recognition via semantic embeddings and knowledge graphs, (2018), pp. 6857–6866
10. M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, B. Schiele, in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference On*. What helps where–and why? semantic relatedness for knowledge transfer (IEEE, 2010), pp. 910–917
11. T. Mensink, E. Gavves, C. G. Snoek, in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference On*. Costa: co-occurrence statistics for zero-shot classification (IEEE, 2014), pp. 2441–2448
12. T. Mukherjee, T. Hospedales, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Gaussian visual-linguistic embedding for zero-shot recognition, (2016), pp. 912–918
13. Q. Wang, K. Chen, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Alternative semantic representations for zero-shot human action recognition (Springer, 2017), pp. 87–102
14. Z. Akata, S. Reed, D. Walter, H. Lee, B. Schiele, in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference On*. Evaluation of output embeddings for fine-grained image classification (IEEE, 2015), pp. 2927–2936
15. Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, B. Schiele, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Latent embeddings for zero-shot classification, (2016), pp. 69–77
16. Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for image classification. IEEE Trans. Pattern Anal. Mach. Intell. **38**(7), 1425–1438 (2016)
17. G. A. Miller, WordNet: a lexical database for English. Commun. ACM. **38**(11), 39–41 (1995)
18. L. Yu, Linked open data. Developer's Guide Semant. Web., 409–466 (2011)
19. O. Levy, Y. Goldberg, in *Advances in Neural Information Processing Systems*. Neural word embedding as implicit matrix factorization, (2014), pp. 2177–2185
20. M. Baroni, G. Dinu, G. Kruszewski, in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors, vol. 1, (2014), pp. 238–247
21. J. Pennington, R. Socher, C. Manning, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Glove: global vectors for word representation, (2014), pp. 1532–1543

22. A. Joulin, E. Grave, P. B. T. Mikolov, in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Bag of tricks for efficient text classification, vol. 2, (2017), pp. 427–431
23. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, in *Advances in Neural Information Processing Systems*. Distributed representations of words and phrases and their compositionality, (2013), pp. 3111–3119
24. R. Kadlec, O. Bajgar, J. Kleindienst, in *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Knowledge base completion: baselines strike back, (2017), pp. 69–74
25. B. Yang, S. W.-T. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases (2014). arXiv preprint arXiv:1412.6575
26. A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, in *Advances in Neural Information Processing Systems*. Translating embeddings for modeling multi-relational data, (2013), pp. 2787–2795
27. T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2D knowledge graph embeddings (2017). arXiv preprint arXiv:1707.01476
28. M. Nickel, D. Kiela, in *Advances in Neural Information Processing Systems*. Poincaré embeddings for learning hierarchical representations, (2017), pp. 6341–6350
29. A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Supervised learning of universal sentence representations from natural language inference data, (2017), pp. 670–680
30. F. Hill, K. Cho, A. Korhonen, Y. Bengio, Learning to understand phrases by embedding the dictionary. Trans. Assoc. Comput. Linguist. **4**, 17–30 (2016)
31. M. Pagliardini, P. Gupta, M. Jaggi, Unsupervised learning of sentence embeddings using compositional n-gram features. Technical report. (2017)
32. R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, S. Fidler, in *Advances in Neural Information Processing Systems*. Skip-thought vectors, (2015), pp. 3294–3302
33. F. Hill, K. Cho, A. Korhonen, in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Learning distributed representations of sentences from unlabelled data, (2016), pp. 1367–1377
34. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
35. Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, Y. Matsumoto, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Ridge regression, hubness, and zero-shot learning (Springer, 2015), pp. 135–151
36. B. Romera-Paredes, P. Torr, in *International Conference on Machine Learning*. An embarrassingly simple approach to zero-shot learning, (2015), pp. 2152–2161
37. R. Socher, M. Ganjoo, C. D. Manning, A. Ng, in *Advances in Neural Information Processing Systems*. Zero-shot learning through cross-modal transfer, (2013), pp. 935–943
38. Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning. arXiv preprint arXiv:1801.07606 (2018)
39. H. Larochelle, D. Erhan, Y. Bengio, in *AAAI*. Zero-data learning of new tasks. vol. 1. (2008), p. 3