Semantic Web and Zero-Shot Learning of Large Scale Visual Classes

Tristan Hascoet, Yasuo Ariki, Tetsuya Takiguchi

Abstract

Zero-shot learning (ZSL) refers to the task of learning a model capable of classifying images into classes for which no sample is available as training data. This can be achieved by leveraging semantic features of the visual classes as an intermediate level of representation shared by both training classes (for which labeled images are provided as training data) and test classes (for which no image is available for training). Following the success of deep learning models in the traditional task of image classification, ZSL has recently attracted a lot of attention from the computer vision community as it holds the promise of scaling up the classification capacity of traditional image classifiers while easing the data collection process. While several models have recently been introduced for ZSL, arguably little attention has been given to the design of the visual class semantic features. In this paper, we propose to leverage the interlinking of knowledge bases published as Linked Open Data to provide different semantic feature representations of visual classes in a large-scale setting. Using a simple ZSL architecture, we compare the efficiency of the semantic features we extracted and find that some of them outperform the standard word embedding representations by a significant margin.

1. Introduction

The task of image classification naturally precedes ZSL. The introduction of the ImageNet dataset (Deng et al. 2009) and the related ImageNet Large Scale Visual Recognition Challenge have played an important role in the recent success of image classifiers as it provided computer vision practitioners with training data of a new scale. With thousands of image samples available for thousands of classes, ImageNet has set the stage for the success of Convolutional Neural Networks (CNNs). ImageNet's wide coverage of naturally occurring objects allows CNNs to learn efficient mid-level visual features that generalize well to unseen visual distributions so that CNNs pretrained on ImageNet can be successfully finetuned and applied to a variety of tasks (Oquab et al. 2014).

However, image classifiers require a balanced training set with a considerable amount of image samples per visual classes. As the granularity of the classes increases, two problems arise. First the image collection and annotation process becomes very expensive. Second, the computation itself becomes intractable. As state of the art n-way classifiers use one-hot encoding of classes, they cannot computationally handle an exponentially increasing number of classes. ZSL models offer the perspective of scaling up the discriminative capability of visual classifiers without requiring any expensive data collection. To achieve this, ZSL models embed visual classes in a semantic vector space. Then, using the correlations between training and test classes at the semantic level, the visual knowledge learned from a set of training classes can be transferred to a set of unseen test classes.

In ImageNet, visual classes are indexed by WordNet (Miller 1995) synonym set (synset) IDs. Synonym sets group together *word senses* conveying a similar meaning. They are linked to each other by semantic relationships into the WordNet hierarchy. The main advantage of using WordNet synsets as class identifiers is that it allows for word-sense disambiguation of the visual concepts. In the context of large-scale visual categories, this is important to avoid biases introduced by homonymy in natural languages. For example, learning distinct visual classes for *musical jams*, *traffic jams*, and *marmalade jam* seems more meaningful than learning one *jam* visual concept embedding these three different meanings of the same word. While in small-scale image classification settings, homonymy biases are unlikely to occur, large-scale settings (i.e, dealing with thousands of visual classes) inevitably suffer from homonymy biases if the visual classes are defined at the lexical level; i.e, by words in natural languages. For example, among the 21,845 WordNet synsets for which ImageNet provides images, 1,444 classes share their first lexical form with at least one other synset, with up to 6 different synsets sharing the same first lexical form *queen*. Despite this high degree of homonymy, all recent ZSL models applied in large-scale settings use word embeddings as semantic representations. Hence, they lose the benefit of ImageNet's use of the WordNet word-sense disambiguation feature.

Another interesting feature of WordNet that has been ignored by the computer vision community is that WordNet has been integrated to the Linked Open Data (LOD) cloud as part of its Linguistic LOD sub-cloud. Linked Data (Bizer, Heath, and Berners-Lee 2009) refers to a set of best practices to be adopted by web data publishers in order to integrate their data into a web of data; i.e, the *semantic web*. The LOD cloud references openly published datasets that follow the Linked Data best practices. As one important aspect of these best practices, LOD datasets contain links from their

resources to the resources of other LOD datasets.

In this paper, we propose to use the interlinking of WordNet to other knowledge bases integrated to the LOD cloud, namely the semantic network Babelnet (Navigli and Ponzetto 2012) and the knowledge graph DBPedia (Auer et al. 2007). We extract semantic feature representations of ImageNet visual classes from these two knowledge bases. In addition, we investigate two other semantic modalities. The first one is provided by (Iacobacci, Pilehvar, and Navigli 2015). It uses the neural word embedding model word2vec (Mikolov et al. 2013) trained on a word-sense disambiguated corpus of Wikipedia. The second modality uses topic-modeling techniques to extract semantic features from the full text of Wikipedia articles. Using LOD interlinkings, we are able to map WordNet synsets to Wikipedia articles in a fully automated process.

The main contributions of this paper are as follow:

- We show how LOD can be used to automate the generation of visual class semantic features. We extract rich semantic features, either directly from knowledge bases of the LOD cloud, or by using LOD knowledge bases in combination with text data from Wikipedia articles.
- We run a set of experiments to compare the efficiency of these semantic features in the context of large-scale ZSL, and find that they generally outperform the traditionally used word embeddings by a large margin.

The rest of this paper is organized as follows: We first present related work on ZSL in section 2. In section 3, we detail the automatic semantic feature extraction process, and we further present the extracted semantic features. In section 4, we present the ZSL model we used for our experiments and in Section 5 we describe our experiment settings and results. Section 6 concludes with comments on our present results and introduces future research.

2 Related Work

2.1 ZSL Models

Most early works on Zero-shot learning used human-annotated visual attributes as semantic descriptions of visual classes. In their pioneer work on attribute learning, (Lampert, Nickisch, and Harmeling 2009) proposed two frameworks for ZSL: Direct Attribute Prediction (DAP) models first map visual inputs to the attribute space, then classify among unseen test classes based on the attribute prediction scores and the attribute signature of the test classes. Indirect Attribute Prediction (IAP) first learns a n-way classifier on the training classes. The classification score of training classes is used to predict the attribute scores, then classification among unseen test classes is similarly performed based on the attribute prediction scores. (Frome et al. 2013) first explored the use of word embeddings for zero-shot learning in a large-scale image classification setting. A few works ((Norouzi et al. 2013),(Zhang, Xiang, and Gong 2016) and others) followed their setting and improved on their results. Interestingly (Norouzi et al. 2013) can be seen as a simple case of IAP whereas (Frome et al. 2013) implements a DAP model.

(Shigeto et al. 2015) identified the hubness problem as a main drawback to regression-based zero-shot classification. They advocate projecting semantic features into the visual feature space for nearest-neighbor search rather than conducting nearest neighbor search in the semantic space. Recently, transductive zero-shot learning (Fu et al. 2014) has been introduced. In a transductive setting, test images are made available during training as unlabeled data so that the global distribution of the test samples can be leveraged to alleviate some limitations of inductive ZSL models s.a. the hubness problem in (Dinu, Lazaridou, and Baroni 2014). In this paper however, we focus on the more general case of ZSL in an inductive setting. We found (Socher et al. 2013) give an instructive early overview of related fields and (Zhang, Xiang, and Gong 2016) provide a more complete and complementary review of recent works.

2.2 Semantic Data for ZSL

While the majority of works on ZSL use word embeddings as semantic features in large-scale settings and visual attributes in small-scale settings, some works have explored the use of different semantic features. In the early work of (Larochelle, Erhan, and Bengio 2008), the authors applied ZSL (which they refer to as zero-data learning) to perform license plate digit recognition. They trained a predictive model on decimal numbers, and used pictograms of both digits and Roman characters to transfer the visual knowledge and classify the Roman characters on the license plates. This contrasts with the rest of the literature we review as their semantic representations are simple image-like representations whereas both attribute-based and word embeddings are based on high level concepts formulated in natural language.

In (Rohrbach et al. 2010), the authors used different linguistic resources to derive semantic similarity scores between classes, between classes and attributes, and to automatically mine attribute-classes correspondence. Similar to our work, they automate the acquisition of semantic data from knowledge bases, but they focus on deriving semantic similarity scores and part attributes while we extract more elaborate semantic features. (Mensink, Gavves, and Snoek 2014) used visual classes 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. Although a multi-labeled image setting differs from our image classification setting in which both training and test images are given unique identifiers, their work represents an interesting line of research complementary to ours. (Mukherjee and Hospedales 2016) questioned 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 the intra-class variance of semantic concepts. They used Gaussian distributions to model both semantic and visual feature distributions of the visual classes.

As semantic representations drawn from different distributions are often complementary, combining them can improve recognition performance. (Akata et al. 2015) successfully used a combination of word embeddings, visual at-

tributes, hierarchical structure and Wikipedia article text as semantic features. In a small-scale setting, they were able to manually collect the Wikipedia articles corresponding to their visual classes. (Zhang, Xiang, and Gong 2016) combined visual attributes with sentence descriptions to substantially improve on the state of the art. The success of their model motivates our belief that rich and diverse semantic features are essential to performing ZSL recognition. This work aims to provide such semantic data.

3 Semantic Representations

Existing ZSL works in a large-scale setting use word embeddings as semantic representations, with the notable exception of the early work by (Rohrbach, Stark, and Schiele 2011). In this paper, we question the common practice of using lexical forms (words) to represent visual classes. Our motivation is two-folds: First, ImageNet classes are actually defined at the semantic level (as WordNet synsets), and not at the lexical level (as words). As similar words can carry very different semantic meanings (e.g, *musical/traffic/marmalade jam*), they similarly exhibit very different visual appearances when they refer to different meanings. Second, considering visual classes as semantic meanings gives us access to a wealth of knowledge resources from which we can extract rich semantic features. In this section we first present the automated process we used to match ImageNet classes to resources of various knowledge bases. We then present how we generate semantic feature vectors from these knowledge bases, and further describe the different semantic representations we used in our experiments.

3.1 Linking Process

In its simplest form, the World Wide Web consists of a set of HTML documents linked together by embedding the URI of other HTML documents within their content. Hence, it is a web of documents since HTML documents are the interlinked resources. The semantic web was born from the idea of creating a web of data; i.e, linking web resources at the data level instead of the document level. Linked Data defines best practices to interlink data in a standardized way within the semantic web. WordNet synsets, Babel synsets and DBPedia resources are all *resources* in the context of Linked Data. To map WordNet synsets to resources of rich knowledge bases, we crawl the RDF data of the LOD cloud for equivalence links between these resources. BabelNet uses the SKOS vocabulary to provide matching links between WordNet and Babel synsets. We use these links to associate ImageNet classes to a unique Babel synset ID. BabelNet also provides similar links between Babel synsets and DBPedia resources. We use the transitive nature of these links to associate a unique DBPedia entity to ImageNet classes. The mappings between DBPedia resources and Wikipedia articles are provided by DBPedia.

It should be noted that the matching of entities across knowledge bases do not always come as one-to-one associations. In the case of one-to-several mapping, we apply very simple heuristics to reduce them to one-to-one mappings. To evaluate the accuracy of our end-to-end mappings between Wordnet synsets and Wikipedia articles, we manually checked a set of one hundred (WordNet synset, Wikipedia page) associated pairs. Both the simple heuristics and the results of our manual evaluation are provided as supplementary material[1].

Furthermore, our extraction process does not match all WordNet entities across all datasets. The main bottleneck of this mapping process is the missing links from Babel synsets to DBPedia resources. In total, we are only able to generate matches across all knowledge bases for 11.069 out of the 21.845 WordNet synsets for which ImageNet provides images. In the rest of this paper, we focus on the subset of 11.069 ImageNet classes. More statistics on the generated mappings can be found in the supplementary material[1]. We refer to the i^{th} visual class as s^i so that $s^i \in S$, where S is the set of available classes, so we have $card(S) = 11,069$. Similarly, we refer by $s^i_{kb} \in S_{kb}$ to the resource of the i^{th} visual class in a knowledge base kb. For example s^7_{bn} stands for the Babel synset associated to the 7^{th} visual class.

3.2 Graph Propositionalization

In the previous section, we described how we mapped each visual class $s^i \in S$ to resources $s^i_{kb} \in S_{kb}$ in three knowledge bases: BabelNet, DBPedia and Wikipedia. Both DBPedia and BabelNet are graph structures $G_{kb} = \{E_{kb}, V_{kb}\}$ defined by a set of nodes V_{kb} and edges E_{kb}. So, in both cases, visual classes are mapped to a node in a graph $s^i_{kb} \in S_{kb} \subset V_{kb}$. However, ZSL models require semantic features in propositional form, i.e, vectors of numerical, nominal or binary values. The process of generating a vector representation of a graph node has been referred to as *propositionalization* (Ristoski and Paulheim 2014). Here we explore two propositionalization approaches:

First, we explore a Bag of Words approach. Let's consider a node $v^i_{kb} \in V_{kb}$ in a knowledge graph $G_{kb} = \{E_{kb}, V_{kb}\}$. It is connected to k other nodes in the graph by typed-edge relationships $v^i_{kb} \rightarrow e_{i,j} \rightarrow v_{i,j}$, with $(e_{i,j}, v_{i,j}) \in E_{kb} \times V_{kb}$. In this approach, we represent visual classes by the set of outgoing links $O_{kb}(s^i_{kb}) = \{(e_{i,j}, v_{i,j})\} \subset (E_{kb} \times V_{kb})^{k_i}$ connecting them to other nodes in the knowledge graph. The vocabulary of the resulting BoW model is defined by:

$$O_{kb} = \bigcup_{s^i_{kb} \in S_{kb}} O_{kb}(s^i_{kb}) \subset (E_{kb} \times V_{kb})^d$$

where $d = card(O_{kb})$ is the dimension of the resulting BoW encoding vectors. This approach is similar to the *Features Derived from Specific Relations* in (Ristoski and Paulheim 2014) and it is illustrated in Appendix C of the supplementary material. We apply the TF-IDF transform to the BoW vectors and reduce their dimension to 400 using truncated Singular Value Decomposition (SVD).

Second, we consider an adaptation of neural language models to graph structures introduced in (Ristoski and Paulheim 2016). Neural language models learn vector representations of words in a vocabulary from a coprus of words *sequences*. In order to apply these models to graph structure,

[1]The supplementary material can be found at https://github.com/TristHas/SNL-supplementary-material

a corpus of sequence must be extracted from the graph. In their original work, the authors first generate a corpus made of all depth-n paths of their graph. Each path consists of a sequence $(v_i \rightarrow e_{i,1} \rightarrow v_{i,1} \rightarrow ... \rightarrow e_{i,n} \rightarrow v_{i,n})$. Then, considering both nodes and vertices as words of a same vocabulary, they train a word2vec model on the generated corpus. We replicate their method and generate for each class node $s^i_{kb} \in S_{kb}$ a set of 200 depth-8 random walks initialized in s^i_{kb}. The corpus we gather contains 11.609×200 unique length-9 sequences. We train a word2vec model on this corpus with the following configurations: window size=5, dimension=400, n-gram model with negative sampling of 25 samples. The model is optimized through 5 iterations. We refer to features generated by this approach as rdf2vec after the name of the original work.

3.3. BabelNet

BabelNet (Navigli and Ponzetto 2012) is a multilingual semantic network originally created by the fusion of Wikipedia and WordNet. Since its original publication, it has evolved to integrate several other resources s.a. Wikidata and FrameNet. It consists of over 13M concept nodes $v \in V$ called *Babel synsets*, connected by a set of directed edges $e_j \in E$. Edges e_j are annotated with a language among BabelNet's 271 languages, a weight value, and a type which encodes one of 2.554 different semantic relationships. Semantic relationship types range from the formal 'Hypernym/Hyponym' relationships to the very loosely defined 'semantically related' relationship. In our experiments, we did not consider the language nor weight annotations of the edges, resulting in a graph $G_{bn} = \{E_{bn}, V_{bn}\}$, where $card(E_{bn}) = 2.554$ and $card(V_{bn}) = 13M$. We generate both rdf2vec and BoW semantic features from this graph.

3.4. DBPedia

DBpedia (Auer et al. 2007) is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web as LOD. Using a shallow ontology and automated extractors, it has become one of the most widely used Knowledge Graphs. DBpedia also plays a central role in the LOD cloud as many datasets link their resources to DBPedia resources. The full DBPedia knowledge graph is provided as several distinct datasets. In our experiments, we used the DBPedia subgraph made of the *page links*, *mapping based object properties*, *resource type*, and *resource categories* datasets extracted from English Wikipedia pages. More details on the DBPedia datasets we used in our experiments can be found in Appendix B of the supplementary material[1]. Similar to BabelNet, we conducted experiments with both rdf2vec and BoW semantic vectors.

3.5. Wikipedia

Using LOD, we associated each visual class to a matching Wikipedia article as described in section 3.1. As semantic feature vectors, we compute a Bag-of-Word representation of the text of these articles. Our BoW model uses an unrestricted vocabulary made of the full set of words contained in the set of 11,609 articles. We conduct experiments on the resulting vectors after two distinct transformations. First we directly reduce the dimension of the BoW feature vectors using truncated SVD (also known as Latent Semantic Analysis). Second, we first compute the TFI-IDF transformation of the BoW features, and then reduce the dimension of the transformed vectors using truncated SVD.

3.6. Sense Embeddings

The last semantic representation we investigate is provided by (Iacobacci, Pilehvar, and Navigli 2015). In this work, the authors perform word-sense disambiguation on the English Wikipedia corpus. Babelfly, a word-sense disambiguation system based on BabelNet, is used so that the words of the corpus are converted to BabelNet *word-senses*. Then, a word2vec model is trained on the disambiguated corpus to compute 400-dimension *word-senses* embedding vectors. This way, the model learns embeddings of *word senses* instead of *words* in their lexical form and we are able to directly associate a unique *word-senses* embedding to each of the ImageNet classes.

3.7. Word Embeddings

For comparison with existing works, we also conduct experiments using word embeddings as semantic features. To allow for fair comparison with the sense embeddings, we train a word2vec model with configurations similar to (Iacobacci, Pilehvar, and Navigli 2015): CBOW architecture, hierarchical softmax objective, window size 5 and 400-dimension embedding vectors. As WordNet synsets correspond to several words and most words appear in several synsets, ImageNet classes cannot be directly associated a unique word embedding vector. We replicate the procedure of the original work (Frome et al. 2013) as described in their supplementary material to deal with ambiguous situations.

4. ZSL Model

To conduct our experiments, we used a simple linear mapping between visual and semantic features similar to (Frome et al. 2013). As in the original work, we train our model with a hinge loss, using stochastic gradient descent.

During training, we consider a set of training classes $s_i \in S_{train}$ and a knowledge base kb. For each class, we have a set of n sample visual features $\{x_{i,k} | k \in [1, n]\}$ and a semantic feature vector $s^{kb}_i \in S^{kb}_{train}$. Training is performed by minimizing over W the following loss:

$$L(x_{i,k}) = \sum_{j \neq i} max(0, margin - s^{kb}_i W x_{i,k} + s^{kb}_j W x_{i,k}),$$

with $k \in [1, n]$ and $(s^{kb}_i, s^{kb}_j) \in (S^{kb}_{train})^2$.

At test time, we consider a set of test classes $s_i \in S_{test}$ so that $S_{test} \cap S_{train} = \emptyset$. Given a knowledge base kb and a test sample x_{test}, classification is performed by selecting the test class yielding the highest similarity score:

$$i = argmax_{s^{kb}_i \in S^{kb}_{test}} (s^{kb}_i W x_{test})$$

As visual features, we use the activation values of the top hidden layer of a ResNet-50 (He et al. 2016) pretrained on

the ILSVRC2012 image classification task. The supplementary material includes a visual illustration of this model. It differs from the original work in two ways:

We use a ResNet-50 to extract visual features, whereas the original work uses the AlexNet model.

For implementation convenience, we use the activation values of the top layer of our model as fixed visual features and we do not back-propagate the error gradient through the network for an end-to-end training.

5. Experiments

5.1. Experiment Settings

Few ZSL works report results on a large-scale setting. A comprehensive summary of existing work can be found in the results section of (Zhang, Xiang, and Gong 2016). Existing works follow either of the two experimental settings:

Experimental setting A: 800 classes are randomly sampled from the ILSVRC classification dataset and used as the training set and the remaining 200 classes are used as the test set.

Experimental setting B: 1000 classes of the ILSVRC image classification are used as the training set and test splits of increasing size and difficulty are selected from ImageNet.

The set of visual classes for which we have generated semantic features does not cover the full set of the ILSVRC classification dataset. Hence, we cannot replicate the exact same experiment setting for a fair comparison. Instead, we randomly sample a set of 1000 ImageNet classes for which we have generated semantic feature representations. Using this modified 1000-classes dataset, we replicate both settings in 5.2 and 5.3 respectively. In both cases, we compare the results we obtain for each semantic feature representation.

5.2. Experimental Setting A

Table 1: Top-k accuracy in Setting A (%)

Accuracy measure	Top-1	Top-5	Top-10
Word Embedding	9.64	29.53	43.44
Sense Embedding	12.67	36.78	51.09
Wikipedia$_{BoW}$	13.61	39.97	56.61
Wikipedia$_{tfidf}$	**14.0**	**42.98**	**57.57**
BabelNet$_{BoW}$	12.5	39.86	54.12
BabelNet$_{rdf2vec}$	3.64	12.74	21.47
DBpedia$_{BoW}$	11.45	37.55	51.83
DBpedia$_{rdf2vec}$	4.2	16.29	27.07

Table 1 shows the flat $top-k$ accuracy for different k and semantic features. We can see that **nearly all our semantic features outperform word embeddings**. Sense embeddings provide a **24.5%** relative improvement over word embeddings on the standard top-5 accuracy metric. This improvement is surprisingly high considering that they have been learned with a similar model. This illustrates the advantage of defining visual classes at the semantic level as *word senses* instead of the lexical level as *words*.

Both DBPedia and BabelNet BoW semantic features provide a relative improvement larger than 26.% over the word embeddings on the top-5 accuracy metric. We find DBPedia and BabelNet to show relatively similar performance and that simple BoW propositionalization approaches gave the best results. The rdf2vec approach yields disappointing results, even significantly inferior to word embeddings. We believe this might be due to the small size of the random walk corpus we generated. These corpora are made of roughly 2M length-9 sequences, which is orders of magnitude smaller than the full text corpus of the English Wikipedia on which both words and sense embeddings have been trained. Wikipedia articles yield the best results on all three metrics, with up to **45.2%** relative improvement of the top-1 accuracy.

5.3. Experimental Setting B

Table 2: Top-5 accuracy in setting B (%)

Test classes	100	200	500	1000	2000
Word Embedding	39.52	29.0	18.52	12.59	10.35
Sense Embedding	49.24	37.01	22.65	14.55	11.14
Wikipedia$_{BoW}$	49.76	41.27	24.48	14.73	11.77
Wikipedia$_{tfidf}$	**53.06**	**44.98**	**27.72**	**17.82**	**15.12**
BabelNet$_{BoW}$	49.78	39.86	23.18	14.51	11.91
BabelNet$_{rdf2vec}$	23.82	13.77	7.26	3.94	3.04
DBpedia$_{BoW}$	48.08	39.12	24.08	15.57	13.45
DBpedia$_{rdf2vec}$	28.02	18.36	9.10	5.362	4.83

Table 2 shows the flat $top-5$ accuracy results for different sizes n of the test set. These results show a similar trend to the one observed in Experimental setting A: BoW features extracted form either Wikipedia articles or knowledge graphs provide significant improvement over word embeddings. This experience also highlights another trend: the Wikipedia representations seem to scale better with larger test sets: On a relatively small 100-class test set, Wikipedia representations provide a 34.26% relative improvement, while for larger test sets, the relative improvement is between 45% and 50 %. The feature representation we used (textual BoW, TFIDF transform with Latent Semantic Analysis) has been developed to index large sets of documents, which might explain its better scaling to a large nearest neighbor search.

6. Conclusion

ZSL is a complex task that requires both powerful models and efficient semantic representations of the visual classes. In this paper we showed how Linked Open Data can be leveraged to generate rich semantic features of ImageNet's visual classes. Existing ZSL approaches label visual classes with words and leverage word embeddings as semantic features. Instead of that approach, we consider ImageNet classes for what they are: resources defined at the semantic level and integrated to the LOD cloud. This enables us to extract semantic representations either directly from LOD knowledge bases, or by jointly leveraging these knowledge bases with text data from Wikipedia. Using a simple ZSL

baseline model, we found that the semantic features we generated generally outperform word embeddings by a significant margin.

In future work, we plan on using these semantic features with more complex ZSL architectures. Recent works like (Zhang, Xiang, and Gong 2016) or (Akata et al. 2015) have been able to successfully combine different semantic views to improve on ZSL recognition rates. As the semantic features we generated are drawn from different data sources with different distributions, we believe they might prove complementary and further improve on the state of the art.

References

Akata, Z.; Reed, S.; Walter, D.; Lee, H.; and Schiele, B. 2015. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2927–2936.

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer. 722–735.

Bizer, C.; Heath, T.; and Berners-Lee, T. 2009. Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts* 205–227.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 248–255. IEEE.

Dinu, G.; Lazaridou, A.; and Baroni, M. 2014. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*.

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

Fu, Y.; Hospedales, T. M.; Xiang, T.; Fu, Z.; and Gong, S. 2014. Transductive multi-view embedding for zero-shot recognition and annotation. In *European Conference on Computer Vision*, 584–599. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Iacobacci, I.; Pilehvar, M. T.; and Navigli, R. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *ACL (1)*, 95–105.

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

Larochelle, H.; Erhan, D.; and Bengio, Y. 2008. Zero-data learning of new tasks. In *AAAI*, volume 1, 3.

Mensink, T.; Gavves, E.; and Snoek, C. G. 2014. Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2441–2448.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Mukherjee, T., and Hospedales, T. 2016. Gaussian visual-linguistic embedding for zero-shot recognition. In *arxiv*. EMNLP.

Navigli, R., and Ponzetto, S. P. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* 193:217–250.

Norouzi, M.; Mikolov, T.; Bengio, S.; Singer, Y.; Shlens, J.; Frome, A.; Corrado, G. S.; and Dean, J. 2013. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*.

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ristoski, P., and Paulheim, H. 2014. A comparison of propositionalization strategies for creating features from linked open data. *Linked Data for Knowledge Discovery* 6.

Ristoski, P., and Paulheim, H. 2016. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*, 498–514. Springer.

Rohrbach, M.; Stark, M.; Szarvas, G.; Gurevych, I.; and Schiele, B. 2010. What helps where–and why? semantic relatedness for knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 910–917. IEEE.

Rohrbach, M.; Stark, M.; and Schiele, B. 2011. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 1641–1648. IEEE.

Shigeto, Y.; Suzuki, I.; Hara, K.; Shimbo, M.; and Matsumoto, Y. 2015. Ridge regression, hubness, and zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 135–151. Springer.

Socher, R.; Ganjoo, M.; Manning, C. D.; and Ng, A. 2013. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, 935–943.

Zhang, L.; Xiang, T.; and Gong, S. 2016. Learning a deep embedding model for zero-shot learning. *arXiv preprint arXiv:1611.05088*.