Supplementary Material: Webly Supervised Semantic Embeddings for Large Scale Zero-Shot Learning

Yannick Le Cacheux, Adrian Popescu, and Hervé Le Borgne

Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

A Implementation details

A.1 Word embeddings

We provide additional details about the versions of the embedding implementations used, namely word2vec, GloVe and FastText, as well as their parameters.

We used the original implementation of each method available at:

– word2vec - https://code.google.com/archive/p/word2vec/
– GloVe - https://nlp.stanford.edu/software/GloVe-1.2.zip
– FastText - https://github.com/facebookresearch/fastText

The main parameters used for to create semantic embeddings are given in Table 1. These values were selected by following the guidelines from the original papers. We ran initial tests with a larger number of epochs and this did not improve results compared to the numbers presented in Table 1.

Table 1: Training parameters for the different semantic embedding models.

Parameter	word2vec	GloVe	FastText
Epochs	25	100	25
Learning rate	0.1	0.05	0.1
Window	10	10	10
Embedding dimension	300	300	300

The set of parameters used each time in order to facilitate reproducibility is reported in Table 2. We exclude the input, output and intermediary, as well as the number of threads because they do not influence directly the learning process.

We tried to add phrase representations [16], but it did not provide any improvement of results in ZSL experiments, thus it was not used in the final models.

Table 2: Command line used to train embeddings.

Model	Command
word2vec	-size 300 -window 1 -sample 1e-4 -negative 5 -hs 0 -binary 0 -cbow 0 -iter 25 -min-count 5
GloVe	-x-max 100 -iter 100 -eta 0.05 -vector-size 300 -alpha 0.75
FastText	skipgram -dim 300 -epoch 25 -minn 4 -maxn 6 -lr 0.1 -ws 10 -minCount 5

A.2 Visual features and ZSL models

For the ImageNet dataset, we use visual features provided by Hascoet *et al.* [22], which consist in the weights of the last pooling layer of a pre-trained ResNet. We also use a pre-trained ResNet to extract visual features for the CUB and AwA2 datasets, and we further apply 10-crop to the images.

On ImageNet and CUB, hyper-parameters of ZSL methods are selected using respectively 200 and 50 random classes for validation. For AwA2, we use the 8 classes which are not in the ILSVRC out of the 40 training classes.

A.3 Datasets

Some statistics regarding the word word frequencies in each dataset are available in Table 3.

Table 3: Mean word frequency and standard deviation (in thousands of occurrences) in a corpus for words present in a given dataset.

	$wiki$	$clue$	fl_{wiki}	fl_{cust}
ImageNet	51 ± 192	183 ± 886	49 ± 150.2	56 ± 149.7
CUB	104 ± 275	416 ± 1596	117.9 ± 260.6	146 ± 303.8
AwA2	40 ± 94	320 ± 714	73.1 ± 160.3	116.8 ± 207.9

B Additional results

We provide results for Linear$_{V\to S}$, Linear$_{S\to V}$ with no $\ell 2$ normalization applied to attributes, as well as for ESZSL with normalization (ESZSL^{norm}) as we found that normalizing attributes could have a significant impact on these models. Results are provided for ImageNet (Table 4) as well as CUB and AwA2 (Table 5) similarly to tables 1 and 2 of the main paper.

For comparison with other papers, we also provide top-5 and top-10 accuracy for the Linear$^{norm}_{S\to V}$ model trained on FastText fl_{cust} in Table 6.

Table 4: Results with and without ℓ2 normalization of attributes on the ImageNet dataset; this table is similar to Table 1 of the main paper. Normalized attributes are indicated with the *norm* exponent; results without the exponent correspond to unnormalized attributes.

Model	word2vec					GloVe					FastText				
Source	*pt*	wiki	clue	$\mathbf{fl_{wiki}}$	$\mathbf{fl_{cust}}$	*pt*	wiki	clue	$\mathbf{fl_{wiki}}$	$\mathbf{fl_{cust}}$	*pt*	wiki	clue	$\mathbf{fl_{wiki}}$	$\mathbf{fl_{cust}}$
$\text{Linear}_{V\to S}$	*2.0*	4.3	4.1	3.7	4.6	*5.4*	3.3	1.3	4.2	5.3	*1.8*	4.6	1.1	4.0	**4.9**
$\text{Linear}_{S\to V}$	*10.7*	12.1	12.5	12.4	17.0	*14.3*	7.7	8.7	8.2	10.6	*14.6*	12.5	2.5	12.8	**17.3**
ESZSL^{norm}	*13.4*	12.8	13.6	13.8	**18.0**	*16.1*	10.7	11.9	13.7	14.4	*16.0*	13.0	8.6	14.7	17.7

Table 5: Results with and without ℓ2 normalization of attributes on the CUB and AwA2 datasets; this table is similar to Table 2 of the main paper. Normalized attributes are indicated with the *norm* exponent; results without the exponent correspond to unnormalized attributes.

Model	word2vec					GloVe					FastText				
Source	*pt*	wiki	clue	$\mathbf{fl_{wiki}}$	$\mathbf{fl_{cust}}$	*pt*	wiki	clue	$\mathbf{fl_{wiki}}$	$\mathbf{fl_{cust}}$	*pt*	wiki	clue	$\mathbf{fl_{wiki}}$	$\mathbf{fl_{cust}}$
CUB dataset															
$\text{Linear}_{V\to S}$	*5.6*	12.1	10.7	11.7	**15.7**	*3.9*	13.4	5.5	11.5	12.1	*3.2*	12.0	7.9	11.7	15.2
$\text{Linear}_{S\to V}$	*14.3*	19.0	17.7	20.1	21.3	*20.6*	14.3	12.9	14.9	17.3	*18.0*	17.4	2.0	19.2	**22.4**
ESZSL^{norm}	*16.9*	20.6	16.7	20.9	23.6	*19.1*	18.3	18.8	21.2	22.0	*20.7*	17.4	19.9	21.5	**24.0**
Awa2 dataset															
$\text{Linear}_{V\to S}$	*27.3*	15.6	33.6	15.5	25.9	*30.6*	17.2	34.9	26.3	**42.3**	*7.8*	11.8	9.7	3.8	15.2
$\text{Linear}_{S\to V}$	*24.8*	45.0	53.2	56.1	56.6	*55.7*	48.4	50.5	41.7	**60.6**	*58.1*	47.6	2.2	47.9	55.2
ESZSL^{norm}	*41.6*	38.7	46.7	49.5	45.6	*55.3*	31.6	47.0	48.5	46.4	***55.9***	38.2	18.6	45.3	43.9

C Effect of User Filtering on Flickr Embeddings

In Section 3 of the main article, we reported the introduction of user filtering instead of raw co-occurrence frequency in Flickr in order the quality of embeddings. When user voting is exploited, each user gets to vote only once for a pair of words and the effect of bulk tagging is thus reduced. We compare the fl_{cust} results presented in Table 1 of the main paper, obtained with user filtering and those of fl_{cust}^{raw}, obtained with a simple count of word co-occurrences. We use FastText and all the tested ZSL methods of the main paper. The results, presented in Table 7, confirm that user filtering has a positive effect for all collection sizes and ZSL methods tested. This confirms the importance of an appropriate preprocessing of text collections.

D Effect of combining fl_{cust} and fl_{wiki}

In Subsection 4.2 of the main paper, we noted that fl_{cust}, the Flickr collection which includes metadata from the three test datasets, gave the best results

Table 6: Top-k accuracy on ImageNet, with FastText and fl_{cust}.

	top-1	top-5	top-10
$\text{Linear}_{S\to V}$	17.3	39.6	51.9
$\text{Linear}_{S\to V}^{norm}$	17.2	39.2	51.4
ESZSL	15.8	37.5	49.3
ESZSL^{norm}	17.7	40.0	51.4
ConSE^{norm}	14.5	32.4	42.0
Devise^{norm}	13.8	32.1	43.7

Table 7: ZSL accuracy on the ImageNet dataset for two versions of the fl_{cust} collection which exploit user voting (fl_{cust}) and raw counts (fl_{cust}^{raw}) to compute word co-occurrences.

Model	FastText	
Source	fl_{cust}	fl_{cust}^{raw}
$\text{Linear}_{S\to V}$	17.3	13.9
$\text{Linear}_{S\to V}^{norm}$	17.2	13.8
ESZSL	15.8	12.5
ESZSL^{norm}	17.7	15.5
ConSE^{norm}	14.5	12.6
Devise^{norm}	13.8	11.2

among the text collections tested. Since fl_{wiki} is collected from the same source but with a different set of concepts, we merged the two collections to observe the effect of results. The results are reported in Table 8 and they confirm that most of the performance gain is due to the use of fl_{cust}.

Table 8: ZSL accuracy for the ImageNet dataset.

Model	word2vec			GloVe			FastText		
Source	fl_{wiki}	fl_{cust}	fl_{merged}	fl_{wiki}	fl_{cust}	fl_{merged}	fl_{wiki}	fl_{cust}	fl_{merged}
$\text{Linear}_{S\to V}$	12.4	17.0	17.2	8.2	10.6	11.1	12.8	17.3	17.2
$\text{Linear}_{S\to V}^{norm}$	12.8	17.1	16.9	9.2	11.4	11.9	13.3	17.2	17.1
ESZSL	9.5	15.3	15.3	11.1	12.0	14.4	11.9	15.8	15.2
ESZSL^{norm}	13.8	18.0	17.9	13.7	14.4	17.1	14.7	17.7	17.9
ConSE^{norm}	11.9	13.5	14.1	11.3	11.9	12.7	12.6	14.5	14.2
Devise^{norm}	9.6	13.3	13.9	3.8	3.4	9.0	10.3	13.8	13.6

E Comparison with manual attributes

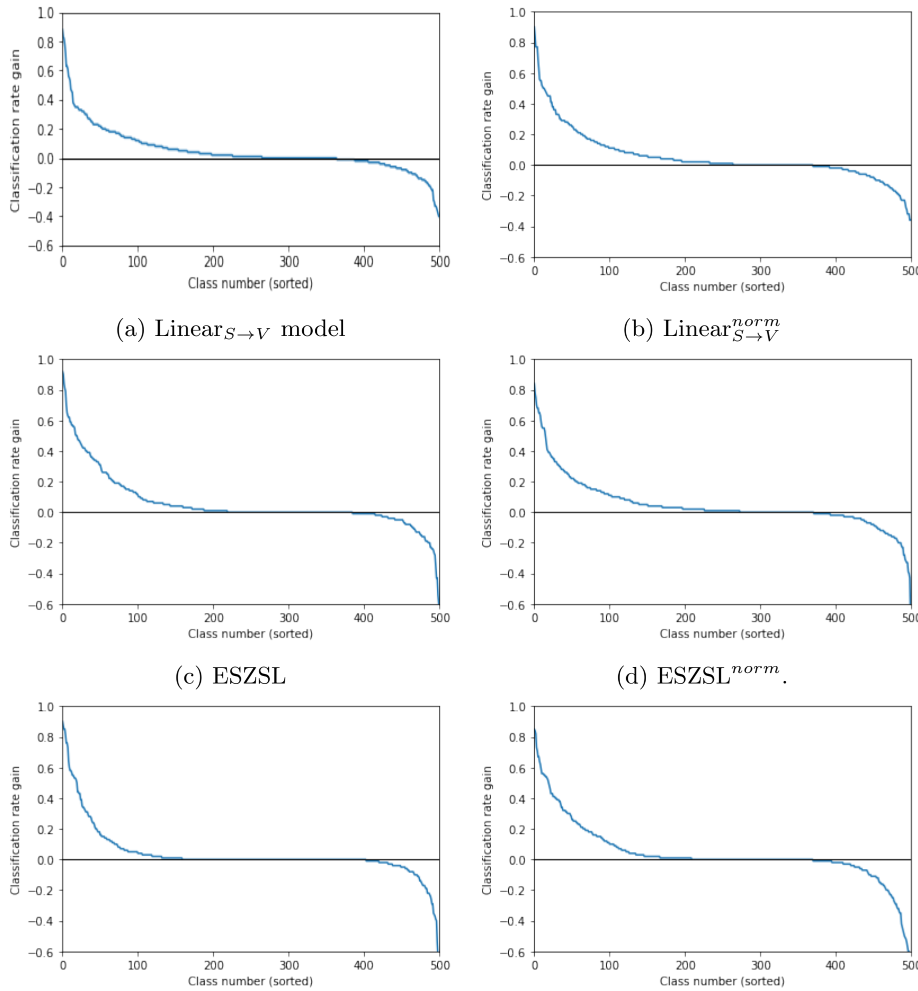
Table 9 contains the data used to create Figure 1 of the main paper. Note that when all attributes are selected, there is no randomness involved since $\text{Linear}_{S\to V}$ is deterministic, hence a standard deviation of 0.

F Performance gain of fl_{cust} over *wiki*

We present a comparison of FastText accuracy obtained for *wiki* and fl_{cust} for the ImageNet dataset with different models.Figure 1 provides a view of accuracy

Table 9: Performance with linear model on CUB and AwA2 with attributes randomly removed. Averaged on 10 runs.

CUB											
Number of attributes	312	250	200	150	100	50	20	15	10	5	2
Mean ZSL score	55.3	54.8	54.2	51.7	46.6	34.7	21.2	15.7	10.4	5.9	2.2
Standard deviation	0.0	0.5	0.9	1.9	3.9	3.3	3.8	3.5	2.9	1.8	0.9
AwA2											
Number of attributes		85	70	50	40	30	20	15	10	5	2
Mean ZSL score		66.0	65.8	61.3	59.7	57.4	46.2	42.2	42.2	25.7	8.8
Standard deviation		0.0	2.8	5.7	7.9	5.6	9.3	7.4	7.9	10.1	4.6



(a) $\text{Linear}_{S\rightarrow V}$ model (b) $\text{Linear}^{norm}_{S\rightarrow V}$

(c) ESZSL (d) ESZSL^{norm}.

(e) ConSE^{norm} (f) Devise^{norm}.

Fig. 1: Performance gain on each test class (by decreasing value) for the fl_{cust} collection w.r.t *wiki* collection, with several ZSL methods.

differences between fl_{cust} and *wiki* for ImageNet test classes. These differences are plotted in decreasing order from left to right. For the $\text{Linear}_{S\rightarrow V}$ model, fl_{cust} is better for 265 of ImageNet test classes, no change is observed for another 99 classes and *wiki* provides better results for the remaining 136 classes. For classes that perform better with fl_{cust}, the average gain is 0.13 and the maximal gain is 0.88. For those performing worse, the average loss is -0.08 and the maximal loss is -0.4. Trends are similar for other methods, indicating that performance gains are robust with respect to the ZSL methods used.

G ImageNet ZSL Full Graph

We provide a visualization of the full WordNet hierarchy for all 1000 (resp. 500) training (resp. testing) classes, as well as some intermediate nodes in Fig. 2. We only keep one parent per node. Fig. 3 of the main paper contains subsets of this visualization. For nodes which originally have several hypernyms, we keep the nodes corresponding to the longest path to the root node *"entity"*; we found that this leads to more meaningful paths, with fewer classes at each level. For example, we keep the path *"greyhound"* → *"hound"* → *"hunting_dog"* → *"dog"* → ... → *"animal"* (visible in Fig. 2) instead of *"greyhound"* → *"racer"* → *"animal"*. We remove intermediate nodes which are not direct hypernyms of either a training or a testing class, as well as some other hand-picked nodes to improve readability.

In addition to the remarks from the main paper, it is interesting to observe that ZSL training and testing classes are not homogeneous in the hierarchy: some tree branches contain very few unseen classes, *e.g. "carnivore"*, while other contain many unseen classes and not a single seen class, *e.g. "woody_plant"*. These latter classes appear very challenging to correctly predict.

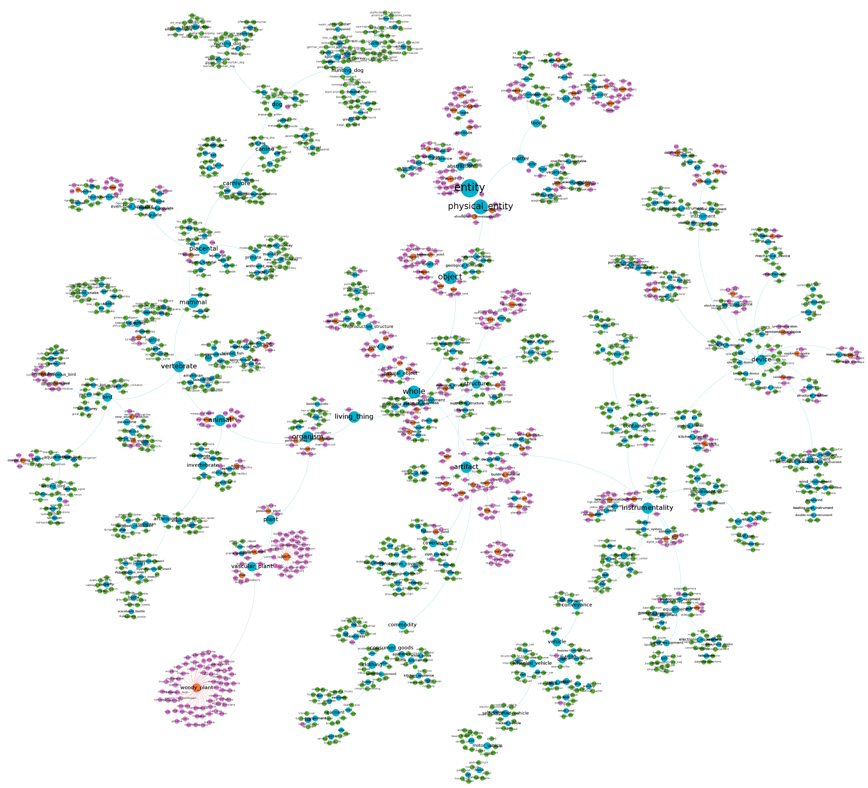


Fig. 2: Overview of the full class hierarchy. Pink nodes refer to test classes, green nodes refer to train classes, orange nodes have only test classes below them and blue nodes are other intermediate nodes. Best viewed in color with at least 600% zoom.