Guessing Objects in Context

Karan Sharma*, Arun CS Kumar, Suchendra Bhandarkar
The University of Georgia

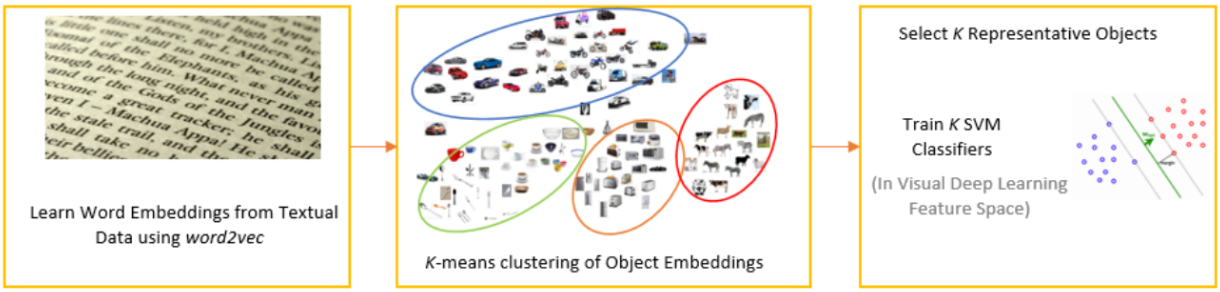

Figure 1: *1. From textual data, learn word embeddings using word2vec. 2. Cluster object embeddings that lets co-occurring objects fall in the same cluster. 3. Select representative object from each cluster that has maximal visual training data. Train visual classifiers for these representative objects. Given a test image, we guess objects (using word embeddings) in the image based on representative objects' detections*

Abstract

Large scale object classification has seen commendable progress owing, in large part, to recent advances in deep learning. However, generating annotated training datasets is still a significant challenge, especially when training classifiers for large number of object categories. In these situations, generating training datasets is expensive coupled with the fact that training data may not be available for all categories and situations. Such situations are generally resolved using zero-shot learning. However, training zero-shot classifiers entails serious programming effort and is not scalable to very large number of object categories. We propose a novel simple framework that can guess objects in an image. The proposed framework has the advantages of scalability and ease of use with minimal loss in accuracy. The proposed framework answers the following question: *How does one guess objects in an image from very few object detections?*

Keywords: Object recognition, word2vec

Concepts: •**Computing methodologies** → **Computer Vision;** *Object recognition;*

1 Introduction

Although large-scale image classification has seen tremendous performance gains in recent times, obtaining pre-annotated training data for a wide variety of object categories is expensive. This problem is further exacerbated, for many categories, reliable training data may not be available. In order to circumvent this situation, several zero-shot frameworks [Socher et al. 2013] have been proposed in the literature. However, training zero-shot frameworks for very large number of categories could be time expensive, complicated, and may not be scalable for thousands of categories. We propose a novel simple framework that can guess objects in an image, has an advantage of ease of use and is scalable while sacrificing little accuracy.

Research literature in cognitive science has shown that humans exploit context when recognizing objects in an image. Moreover, certain objects also tend to co-occur reliably with one another in the real world. Previous approaches to contextual object recognition have relied on recognizing all objects in an image followed by either smoothing the recognition results using contextual information [Rabinovich et al. 2007] or using zero-shot classifiers [Socher et al. 2013]. In the proposed framework we circumvent the need for running all possible object detectors on the image and training the zero-shot classifiers. We rely on the intuition is that if we are able to correctly identify even a single object in an image, we can easily guess the other plausible objects in the image. For example, just knowing that there is a *car* in an image, we can guess the presence of other plausible objects such as *road, tree, person* and *building* and we will be correct a statistically significant fraction of the time.

In order to guess the presence of plausible objects, we exploit word embeddings learned from textual data. The *word2vec* approach [Mikolov et al. 2013a, 2013b] learns word embeddings from textual data and tends to cluster semantically related objects in the embedding space, as is evident from the 2-D *t-sne* projection [Van Der Mateen and Hinton 2008] in Figure 2. Hence, if we can successfully detect even a single object in an image, we can regard the proximal objects in the *word2vec* space as reliable guesses for presence of other plausible objects in the image.

How do we decide the representative objects for training object classifiers? Note that the representative objects are the ones that are actually detected in the image and used to guess the presence of other plausible objects in the image. First, representative objects should be the ones that have maximal annotated training data so that the corresponding object classifiers can be learned with high precision and recall. Second, two distinct representative objects should have a low probability of co-occurrence in the real world. For example, *car* and *knife* are two good representative objects since they tend to NOT co-occur in the real world. In contrast, car and truck are not good representative objects since they tend to co-occur with high probability. In order to determine good candidates for representative objects, we cluster the word embeddings corresponding to all the objects in the domain of discourse. The objects that tend to co-occur will end up being placed in the same cluster. From each cluster, we select an object that has maximal annotated training data

*e-mail:karan@uga.edu

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA
ISBN: 978-1-4503-4371-8/16/07
DOI: http://dx.doi.org/10.1145/2945078.2945161

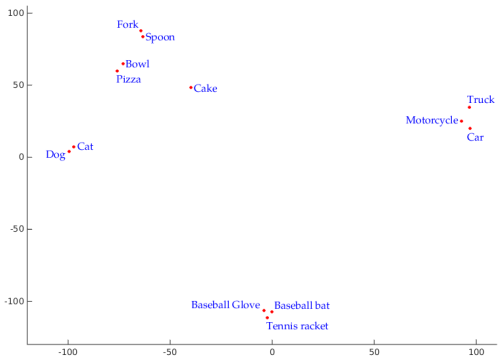

Figure 2: *2-d t-sne [Van Der Mateen and Hinton 2008] projection of word embeddings of objects.*

to be the representative object of that cluster. We train visual classifiers for all the representative objects and during testing, we run all the corresponding detectors for the representative objects on the test image. We select the top-n most likely objects in an image and guess the other objects in the image by exploiting the word embeddings.

In our current implementation, we use the top-3 object detections for guessing other objects in the image. Given a set of nouns N and top object classifications n_1, n_2 and n_3, the closest guesses from set N are given by:

$$\arg\max_i \ \{SIM(n_i, n_1) + SIM(n_i, n_2) + SIM(n_i, n_3)\} \quad (1)$$

where $SIM(n_i, n_j)$ denotes the cosine similarity of nouns n_i and n_j.

2 Experiments

Training: We learn word embeddings for all the words using a skip-gram model implementation of word2vec. We select the word embeddings corresponding to 80 annotated objects in the MicroSoft (MS) COCO dataset [Lin et al. 2014] as inputs to the k-means clustering algorithm. After clustering, semantically related or similar objects tend to be placed in the same cluster, as is evident from the *t-sne* diagram in Fig. 2. Currently, we set value of k to be 14. From each cluster, we select a representative object that has the most annotated instances in the training images. In most real world applications, we would want to select a representative object that has adequate training data available to predict other objects in an image. However, on the COCO dataset we demonstrate our approach only as proof of concept. For each representative object, we train the corresponding SVM classifiers using visual deep learning features. Currently, we train 14 classifiers corresponding to the 14 representative objects derived from their respective clusters.

Testing: For the purpose of testing, we use the 40,000 validation images from the MS COCO dataset as our test set. Given a test image, we run an SVM-based classifier for each representative object on the image. We select the top-3 object classifications, and predict (guess) 5 objects for each image using equation (1). The results are evaluated using the following metrics:
Top-5$_1$ score: where one of the predicted objects matches one of the objects in the ground truth annotations of image.
Top-5$_2$ score: where 2 of the predicted objects match two objects in the ground truth annotations of image.
Top-5$_3$ score: where 3 of the predicted objects match three objects in the ground truth annotations of image.
We also compare the following approaches:
Random baseline: Here we randomly guess 5 objects in an image.

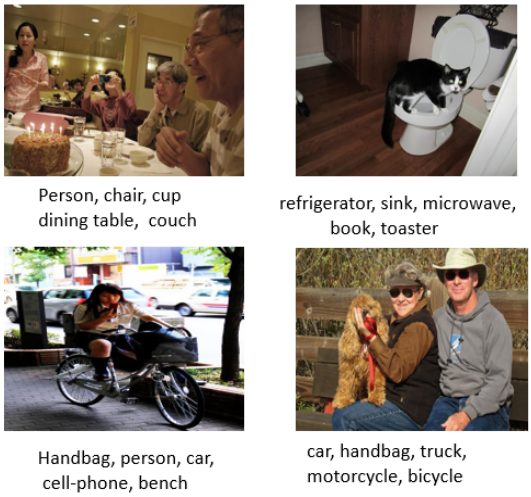

Figure 3: *Qualitative Results: Column 1: Good, Column 2: Bad*

Table 1: *Comparison of the proposed framework with a random baseline, most-frequent baseline, all visual classifiers baseline, and Guess$_1$ baseline.*

	Rand	Most-freq	CV-All	Guess$_1$	Guess$_3$
Top-5$_1$	17.3%	61.3%	95.7%	59.2%	**62.8%**
Top-5$_2$	14.4%	17.2%	57.7%	13.3%	**22.6%**
Top-5$_3$	$< 1\%$	4%	19.4%	3.2%	**6.3%**

Most-frequent baseline: Here we guess top-5 most frequently occurring categories in training set for all test images. This is a hard baseline because class imbalances may be hard to beat for frequently occurring categories.
CV-All: here we run visual classifiers for all 80 objects.
Guess-objs-3: here we guess 5 objects in an image using top-3 objects.
Guess-objs-1: here we guess 5 objects in an image using the top-1 (i.e., the top) object.

3 Results

The results of our experiments are shown in table 1. From the results, it is evident that *Guess-objs-3* beats *random, most frequent,* and *Guess-objs-1* baselines. Hence, using the top-3 object detections to guess other objects in an image is clearly a viable approach. As expected, the approach is inferior to the scenario where we run object classifiers for all 80 categories. But, our results clearly indicate that we can reliably guess objects in an image just by running very few object detectors while completely circumventing the need to train zero-shot classifiers.

References

LIN, T. Y. ET AL. 2014. Microsoft COCO: Common objects in context. *In Proc. ECCV*, (pp. 740-755).

MIKOLOV, T. ET AL. 2013. Distributed representations of words and phrases and their compositionality. *In Proc. NIPS*, (pp. 3111-3119).

MIKOLOV, T. ET AL. 2013. Efficient estimation of word representations in vector space. *In Proc. ICLR*.

RABINOVICH, A., ET AL. 2007. Objects in context. *In Proc. IEEE ICCV,* (pp. 1-8).

SOCHER, R., ET AL. 2013. Zero-shot learning through cross-modal transfer. *In Proc. NIPS*, (pp. 935-943).

VAN DER MAATEN, L., AND HINTON, G. 2008. Visualizing data using *t-SNE*. *In JMLR*, 9 (2579-2605), 85.