ViCo: Word Embeddings from Visual Co-occurrences

Tanmay Gupta Alexander Schwing Derek Hoiem
University of Illinois at Urbana Champaign
{tgupta6, aschwing, dhoiem}@illinois.edu http://tanmaygupta.info/vico/

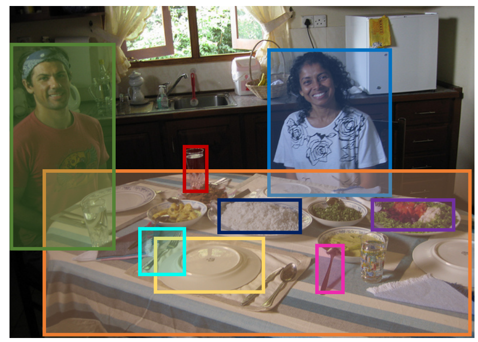
Abstract

We propose to learn word embeddings from visual co-occurrences. Two words co-occur visually if both words apply to the same image or image region. Specifically, we extract four types of visual co-occurrences between object and attribute words from large-scale, textually-annotated visual databases like VisualGenome and ImageNet. We then train a multi-task log-bilinear model that compactly encodes word "meanings" represented by each co-occurrence type into a single visual word-vector. Through unsupervised clustering, supervised partitioning, and a zero-shot-like generalization analysis we show that our word embeddings complement text-only embeddings like GloVe by better representing similarities and differences between visual concepts that are difficult to obtain from text corpora alone. We further evaluate our embeddings on five downstream applications, four of which are vision-language tasks. Augmenting GloVe with our embeddings yields gains on all tasks. We also find that random embeddings perform comparably to learned embeddings on all supervised vision-language tasks, contrary to conventional wisdom.

1. Introduction

Word embeddings, *i.e.*, compact vector representations of words, are an integral component in many language [46, 14, 23, 38, 36, 48, 43] and vision-language models [28, 52, 53, 2, 41, 40, 49, 12, 47, 6, 55, 16, 27]. These word embeddings, *e.g.*, GloVe and word2vec, are typically learned from large-scale text corpora by modeling textual co-occurrences. However, text often consists of interpretations of concepts or events rather than a description of visual appearance. This limits the ability of text-only word embeddings to represent visual concepts.

To address this shortcoming, we propose to gather co-occurrence statistics of words based on images and learn word embeddings from these visual co-occurrences. Concretely, two words co-occur visually if both words are applicable to the same image or image region. We use four types of co-occurrences as shown in Fig. 1: (1) *Object-*



Region	Object Words	Attribute Words
	man, person, adult, mammal	muscular, smiling
	woman, person, adult, mammal	lean, smiling
	table, tablecloth, furniture	striped, oval
	rice, carbohydrates, food	white, grainy, cooked
	salad, roughage, food	leafy, chopped, healthy, red, green
	glass, glassware, utensil	clear, transparent, reflective, tall
	plate, crockery, utensil	ceramic, white, round, circular
	fork, cutlery, utensil	metallic, shiny, reflective
	spoon, cutlery, utensil	serving, metallic, shiny, reflective

Type	Visual Co-occurrences
Object-Attribute	salad-chopped | table-oval | rice-white | salad-healthy | glass-clear | plate-ceramic | fork-metallic ...
Attribute-Attribute	grainy-cooked | green-leafy | leafy-healthy | clear-transparent | metallic-shiny | shiny-reflective ...
Context	man-woman | person-table | fork-spoon | plate-glass | table-tablecloth | rice-salad | plate-food ...
Object-Hypernym	man-mammal | woman-adult | table-furniture | rice-food | glass-utensil | fork-utensil | fork-cutlery ...

Figure 1. **Visual co-occurrences are a rich source of information for learning word meanings.** The figure shows regions annotated with words and attributes in an image, and the four types of visual co-occurrences used for learning ViCo embeddings.

Attribute co-occurrence between an object in an image region and the region's attributes; (2) *Attribute-Attribute* co-occurrence of a region; (3) *Context* co-occurrence which captures joint object appearance in the same image; and (4) *Object-Hypernym* co-occurrence between a visual category and its hypernym (super-class).

Ideally, for reliable visual co-occurrence modeling of a sufficiently large vocabulary (a vocabulary size of 400K is typical for text-only embeddings), a dataset with all applicable vocabulary words annotated for each region in an image is required. While no visual dataset exists with such exhaustive annotations (many non-annotated words may still be applicable to an image region), large scale datasets like VisualGenome [17] and ImageNet [8] along with their WordNet [32] *synset* annotations provide a good starting point. We use ImageNet annotations augmented with WordNet hypernyms to compute Object-Hypernym co-occurrences while the remaining types of co-occurrence are computed from VisualGenome's object and attribute annotations.

To learn ViCo, *i.e*., word embeddings from **Vi**sual **Co**-occurrences, we could concatenate GloVe-like embeddings trained separately for each co-occurrence type via a log-bilinear model. However, in this naïve approach, the dimensionality of the learned embeddings scales linearly with the number of co-occurrence types. To avoid this linear scaling, we extend the log-bilinear model by formulating a *multi-task* problem, where learning embeddings from each co-occurrence type constitutes a different task with compact trainable embeddings shared among all tasks. In this formulation the embedding dimension can be chosen independently of the number of co-occurrence types.

To test ViCo's ability to capture similarities and differences between visual concepts, we analyze performance in an *unsupervised clustering*, *supervised partitioning* (see supplementary material), and a *zero-shot-like* visual generalization setting. The clustering analysis is performed on a set of most frequent words in VisualGenome which we manually label with *coarse* and *fine-grained* visual categories. For the *zero-shot-like* setting, we use CIFAR-100 with different splits of the 100 categories into seen and unseen sets. In both cases, ViCo augmented GloVe outperforms GloVe, random vectors, *vis-w2v*, or their combinations. Through a qualitative analogy question answering evaluation, we also find ViCo embedding space to better capture relations between visual concepts than GloVe.

We also evaluate ViCo on five downstream tasks – a discriminative attributes task, and four vision-language tasks. The latter includes Caption-Image Retrieval, VQA, Referring Expression Comprehension, and Image Captioning. Systems using ViCo outperform those using GloVe for almost all tasks and metrics. While learned embeddings are typically believed to be important for vision-language tasks, somewhat surprisingly, we find random embeddings compete tightly with learned embeddings on all vision-language tasks. This suggests that either by nature of the tasks, model design, or simply training on large datasets, the current state-of-the-art vision-language models do not benefit much from learned embeddings. Random embeddings perform significantly worse than learned embeddings in our clustering, partitioning, and zero-shot analysis, as well as the discriminative attributes task, which does not involve images.

To summarize our contributions: (1) We develop a multi-task method to learn a word embedding from multiple types of co-occurrences; (2) We show that the embeddings learned from multiple visual co-occurrences, when combined with GloVe, outperform GloVe alone in unsupervised clustering, supervised partitioning, and zero-shot-like analysis, as well as on multiple vision-language tasks; (3) We find that performance of supervised vision-language models is relatively insensitive to word embeddings, with even random embeddings leading to nearly the same performance as learned embeddings. To the best of our knowledge, our study provides the first empirical evidence of this unintuitive behavior for multiple vision-language tasks.

2. Related Work

Here we describe non-associative, associative, and the most recent contextual models of word representation.

Non-Associative Models. Semantic Differential (SD) [34] is among the earliest attempts to obtain vector representations of words. SD relies on human ratings of words on 50 scales between bipolar adjectives, such as 'happy-sad' or 'slow-fast.' Osgood *et al*. [34] further reduced the 50 scales to 3 orthogonal factors. However, the scales were often vague (*e.g*., is the word 'coffee' 'slow' or 'fast') and provided a limited representation of the word meaning. Another approach involved acquiring word similarity annotations followed by applying Multidimensional Scaling (MDS) [21] to obtain low dimensional (typically 2-4) embeddings and then identifying meaningful clusters or interpretable dimensions [45]. Like SD, the MDS approach lacked representation power, and embeddings and their interpretations varied based on words (*e.g*., food names [45], animals [44], *etc*.) to which MDS was applied.

Associative Models. The hypothesis underlying associative models is that word-meaning may be derived by modeling a word's association with all other words. Early attempts involved factorization of word-document [7] or word-word [26] co-occurrence matrices. Since raw co-occurrence counts can span several orders of magnitude, transformations of the co-occurrence matrix based on Positive Pointwise Mutual Information (PPMI) [4] and Hellinger distance [22] have been proposed. Recent neural approaches like the Continuous Bag-of-Words (CBOW) and the Skip-Gram models [29, 31, 30] learn from co-occurrences in local context windows as opposed to global co-occurrence statistics. Unlike global matrix factorization, local context window based approaches use co-occurrence statistics rather inefficiently because of the requirement of scanning context windows in a corpus during training but performed better on word-analogy tasks. Levy *et al*. [24] later showed that Skip-Gram with negative-sampling performs implicit matrix factorization of a PMI word-context matrix.

Our work is most closely related to GloVe [37] which combines the efficiency of global matrix factorization approaches with the performance obtained from modelling local context. We extend GloVe's log-bilinear model to simultaneously learn from multiple types of co-occurrences. We

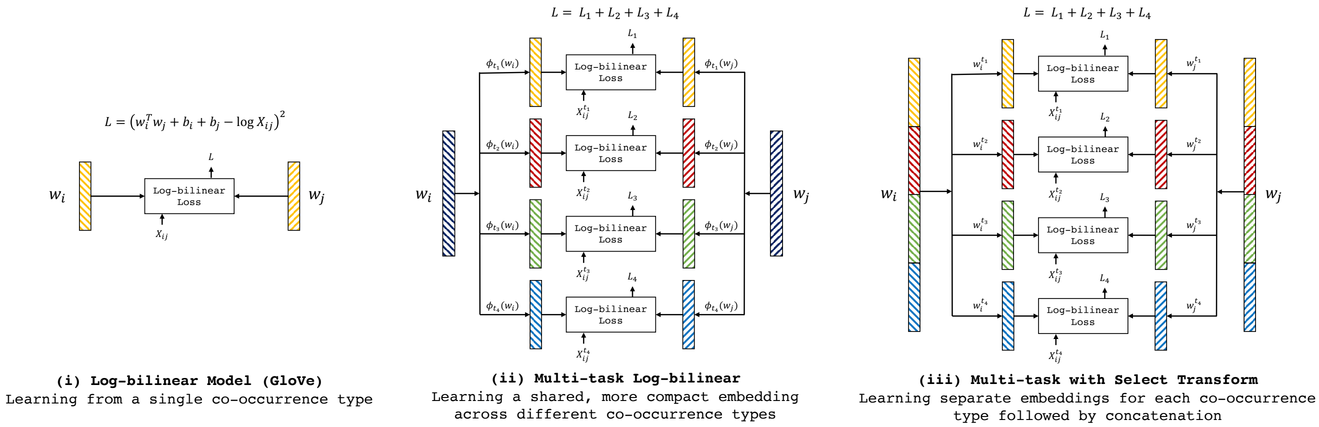


Figure 2. **Log-bilinear models and our multi-task extension.** We show loss computation of different approaches for learning word embeddings w_i and w_j for words i and j. The embeddings are denoted by colored vertical bars. (i) shows GloVe's log-bilinear model. (ii) is our multi-task extension to learn from multiple co-occurrence matrices. Word embeddings w_i and w_j are projected into a dedicated space for each co-occurrence type t through transformation ϕ_t. Log-bilinear losses are computed in the projected embedding spaces. (iii) shows an approach where the different colored regions of w_i (or w_j) are allocated to learn from different co-occurrence types. This approach, equivalent to training separate embeddings followed by concatenation, can be implemented in our multi-task formulation using a *select* transform (Tab. 1). Tab. 4 shows that an appropriate choice of ϕ (*e.g.*, *linear*) in the multi-task framework leads to more compact embeddings than (iii) without sacrificing performance since the correlation between different co-occurrence types is utilized.

also demonstrate that visual datasets annotated with words are a rich source of co-occurrence information that complements the representations learned from text corpora alone.

Visual Word Embeddings. There is some work on incorporating image representations into word embeddings. *vis-w2v* [18] uses abstract (synthetic) scenes to learn visual relatedness. The scenes are clustered and cluster membership is used as a surrogate label in a CBOW framework. Abstract scenes have the advantage of providing good semantic features for free but are limited in their ability to match the richness and diversity of natural scenes. However, natural scenes present the challenge of extracting good semantic features. Our approach uses natural scenes but bypasses image feature extraction by only using co-occurrences of annotated words. ViEW [13] is another approach to visually enhance existing word embeddings. An autoencoder is trained on pre-trained word embeddings while matching intermediate representations to visual features extracted from a convolutional network trained on ImageNet. ViEW is also limited by the requirement of good image features.

Contextual Models. Embeddings discussed so far represent individual words. However, many language understanding applications demand representations of words in context (*e.g.*, in a phrase or sentence) which in turn requires to learn how to combine word or character level representations of neighboring words or characters. The past year has seen several advances in contextualized word representations through pre-training on language models such as ELMo [39], OpenAI GPT [42], and BERT [9]. However, building mechanisms for representing context is orthogonal to our goal of improving representations of individual words (which may be used as input to these models).

3. Learning ViCo

We describe the GloVe formulation for learning embeddings from a single co-occurrence matrix in Sec. 3.1 and introduce our multi-task extension to learn embeddings jointly from multiple co-occurrence matrices in Sec. 3.2. Sec. 3.3 describes how co-occurrence count matrices are computed for each of the four co-occurrence types.

3.1. GloVe: Log-bilinear Model

Let X_{ij} denote the co-occurrence count between words i and j in a text corpus. Also let $\mathcal{N}$ be the list of word pairs with non-zero co-occurrences. GloVe learns d-dimensional embeddings $w_i \in \mathbb{R}^d$ for all words i by optimizing

$$\min_{w,b} \sum_{(i,j)\in\mathcal{N}} f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2, \quad (1)$$

where $f : \mathbb{R} \to \mathbb{R}$ is a weighting function that assigns lower weight to less frequent, noisy co-occurrences and b_i is a learnable bias term for word i.

Intuitively, the program in Eq. (1) learns word embeddings such that for any word pair with non-zero co-occurrence, the dot product $w_i^T w_j$ approximates the log co-occurrence count up to an additive constant. The word meaning is derived by simultaneously modeling the degrees of association of a single word with a large number of other words [33]. We also refer the reader to [37] for more details.

Note the slight difference between the objective in Eq. (1) and the original GloVe objective: GloVe replaces w_j and b_j with $\tilde{w}_j$ (context vector) and $\tilde{b}_j$ which are also trainable. The GloVe vectors are obtained by averaging w_i and $\tilde{w}_i$. However, as also noted in [37], given the symmetry in

Transforms	d	d_t	ϕ_t
select (200)	200	$50\ \forall\ t$	$\phi_t(w) = [w[i_0^t], \cdots, w[i_{49}^t]]$ where $\{i_0^t, \cdots, i_{49}^t\}$ are indices pre-allocated for t in $\{0, \cdots, 200\}$
linear (50)	50	$50\ \forall\ t$	$\phi_t(w) = A_t w$ where $A_t \in \mathbb{R}^{50\times 50}$
linear (100)	100	$50\ \forall\ t$	$\phi_t(w) = A_t w$ where $A_t \in \mathbb{R}^{50\times 100}$
linear (200)	200	$50\ \forall\ t$	$\phi_t(w) = A_t w$ where $A_t \in \mathbb{R}^{50\times 200}$

Table 1. **Description and parametrization of transforms.** $\phi_t : \mathbb{R}^d \to \mathbb{R}^{d_t}$ is a transform for co-occurrence type $t \in \mathcal{T}$. *select* corresponds to approach (iii) in Fig. 2 that concatenates separately trained d_t dimensional embeddings.

the objective, both vectors should ideally be identical. We did not observe a significant change in performance when using separate word and context vectors.

3.2. Multi-task Log-bilinear Model

We now extend the log-bilinear model described above to jointly learn embeddings from multiple co-occurrence count matrices X^t, where $t \in \mathcal{T}$ refers to a type from the set of types $\mathcal{T}$. Also let $\mathcal{N}_t$ and $\mathcal{Z}_t$ be the list of word pairs with non-zero and zero co-occurrences of type t respectively. We learn ViCo embeddings $w_i \in \mathbb{R}^d$ for all words i by minimizing the following loss function

$$\sum_{t\in\mathcal{T}} \sum_{(i,j)\in\mathcal{N}_t} (\phi_t(w_i)^T \phi_t(w_j) + b_i^t + b_j^t - \log X_{ij}^t)^2 + \sum_{t\in\mathcal{T}} \sum_{(i',j')\in\mathcal{Z}_t} \max(0, \phi_t(w_{i'})^T \phi_t(w_{j'}) + b_{i'}^t + b_{j'}^t). \quad (2)$$

Here $\phi_t : \mathbb{R}^d \to \mathbb{R}^{d_t}$ is a co-occurrence type-specific transformation function that maps ViCo embeddings to a type-specialized embedding space. b_i^t is a learned bias term for word i and type t. We set function $f(X)$ in Eq. (1) to the constant 1 for all X. Next, we discuss the transformations ϕ_t, benefits of capturing different types of co-occurrences, use of the second term in Eq. (2), and training details. Fig. 2 illustrates (i) GloVe and versions of our model (ii,iii).

Transformations ϕ_t. To understand the role of the transformations ϕ_t in learning from multiple co-occurrence matrices, consider the naïve approach of concatenating $|\mathcal{T}|$ d_t-dimensional word embeddings learned separately for each type t using Eq. (1). Such an approach would yield an embedding with $d \geq |\mathcal{T}| \min_t d_t$ dimensions. For instance, 4 co-occurrence types, each producing embeddings of size $d_t = 50$, leads to $d = 200$ dimensional final embeddings. Thus, a natural question arises – *Is it possible to learn a more compact representation by utilizing the correlations between different co-occurrence types?*

Word Pair	ViCo	Obj-Attr	Attr-Attr	Obj-Hyp	Context	GloVe
crouch / squat	0.61	0.74	0.72	0.18	0.25	0.05
sweet / dessert	0.66	0.78	0.76	0.56	0.79	0.43
man / male	0.71	0.98	0.8	0.38	1	0.34
purple / violet	0.75	0.93	1	0.24	0.03	0.52
hosiery / sock	0.52	0.27	0.18	0.87	0.07	0.23
aeroplane / aircraft	0.73	0.43	0.07	0.87	0.75	0.43
bench / pew	0.63	0.67	0.09	0.79	-0.14	0.1
keyboard / mouse	0.19	0.63	0.19	0.09	0.95	0.52
laptop / desk	0.39	0.23	0.24	0.1	0.94	0.28
window / door	0.59	0.46	0.35	0.53	0.93	0.67
hair / blonde	0.16	0.56	0.32	-0.15	0.17	0.51
thigh / ankle	0.09	0.19	0.03	0.01	0.39	0.74
garlic / onion	0.36	-0.03	0.3	0.37	0.56	0.77
driver / car	0.27	0.16	0.26	0.12	0.53	0.71
girl / boy	0.41	0.38	0.22	0.44	0.74	0.83

Figure 3. **Rich sense of relatedness through multiple co-occurrences.** Different notions of word relatedness exist but current word embeddings do not provide a way to disentangle those. Since ViCo is learned from multiple types of co-occurrences with dedicated embedding spaces for each (obtained through transformations ϕ_t), it can provide a richer sense of relatedness. The figure shows cosine similarities computed in GloVe, ViCo(linear) and embedding spaces dedicated to different co-occurrence types (components of ViCo(select)). For example, 'hosiery' and 'sock' are related through an object-hypernym relation but not related through object-attribute or a contextual relation. 'laptop' and 'desk' on the other hand are related through context.

Eq. (2) is a multi-task learning formulation where learning from each type of co-occurrence constitutes a different task. Hence, ϕ_t is equivalent to a task-specific head that projects the shared word embedding $w \in \mathbb{R}^d$ to a type-specialized embedding space $\phi_t(w) \in \mathbb{R}^{d_t}$. A log-bilinear model equivalent to Eq. (1) is then applied for each co-occurrence type in the corresponding specialized embedding space. We learn the embeddings w and parameters of ϕ_t simultaneously for all t in an end-to-end manner.

With this multi-task formulation the dimensions of w can be chosen independently of $|\mathcal{T}|$ or d_t. Also note that the new formulation encompasses the naïve approach which is implemented in this framework by setting $d = \sum_t d_t$, and ϕ_t as a slicing operation that 'selects' d_t non-overlapping indices allocated for type t. In our experiments, we evaluate this naïve approach and refer to it as the *select* transformation. We also assess *linear* transformations of different dimensions as described in Tab. 1. We find that 100 dimensional ViCo embeddings learned with *linear* transform achieve the best performance *vs.* compactness trade-off.

Role of max term. Optimizing only the first term given in Eq. (2) can lead to accidentally embedding a word pair from $\mathcal{Z}_t$ (zero co-occurrences) close together (high dot product). To suppress such spurious similarities, we include the max term which encourages all word pairs $(i', j') \in \mathcal{Z}_t$ to have a small predicted log co-occurrence

$$\log \tilde{X}_{i'j'}^t = \phi_t(w_{i'})^T \phi_t(w_{j'}) + b_{i'}^t + b_{j'}^t. \quad (3)$$

	Obj-Attr	Attr-Attr	Obj-Hyp	Context	Overall
Unique Words	$15,548$	$11,893$	$11,981$	$25,451$	$35,476$
Non-zero entries (in millions)	1.37	1.37	0.61	8.12	11.48

Table 2. **Co-occurrence statistics** showing the number of words and millions of non-zero entries in each co-occurrence matrix. For reference, GloVe uses a vocabulary of $400,000$ words with 8-40 billion non-zero entries.

In particular, the second term in the objective linearly penalizes positive predicted log co-occurences of word-pairs that do not co-occur.

Training details. Pennington *et al*. [37] report Adagrad to work best for GloVe. We found that Adam leads to faster initial convergence. However, fine-tuning with Adagrad further decreases the loss. For both optimizers, we use a learning rate of 0.01, a batch size of 1000 word pairs sampled from $\mathcal{N}_t$ and $\mathcal{Z}_t$ each for all t, and no weight decay.

Multiple notions of relatedness. Learning from multiple co-occurrence types leads to a richer sense of relatedness between words. Fig. 3 shows that the relationship between two words may be better understood through similarities in multiple embedding spaces than just one. For example, 'window' and 'door' are related because they occur in context in scenes, 'hair' and 'blonde' are related through an object-attribute relation, 'crouch' and 'squat' are related because both attributes apply to similar objects, *etc*.

3.3. Computing Visual Co-occurrence Counts

To learn meaningful word embeddings from visual co-occurrences, reliable co-occurrence count estimates are crucial. We use Visual Genome and ImageNet for estimating visual co-occurrence counts. Specifically, we use object and attribute *synset* (set of words with the same meaning) annotations in VisualGenome to get *Object-Attribute* (oa), *Attribute-Attribute* (aa), and *Context* (c) co-occurrence counts. ImageNet *synsets* and their ancestors in WordNet are used to compute *Object-Hypernym* (oh) counts. Tab. 2 shows the number of unique words and non-zero entries in each co-occurrence matrix.

Let $\mathcal{T} = \{oa, aa, c, oh\}$ denote the set of four co-occurrence types and X^t_{ij} denote the number of co-occurrences of type $t \in \mathcal{T}$ between words i and j. We denote a *synset* and its associated set of words as $\mathcal{S}$. All co-occurrences are initialized to 0. We now describe how each co-occurrence matrix X^t is computed.

- Let $\mathcal{O}$ and $\mathcal{A}$ be the sets of object and attribute synsets annotated for an image region. For each region in VisualGenome, we increment X^{oa}_{ij} by 1, for each word pair $(i, j) \in \mathcal{S}_o \times \mathcal{S}_a$, and for all *synset* pairs $(\mathcal{S}_o, \mathcal{S}_a) \in \mathcal{O} \times \mathcal{A}$. X^{oa}_{ji} is also incremented unless $i = j$.
- For each region in VisualGenome, we increment X^{aa}_{ij} by 1, for each word pair $(i, j) \in \mathcal{S}_{a_1} \times \mathcal{S}_{a_2}$, and for all *synset* pairs $(\mathcal{S}_{a_1}, \mathcal{S}_{a_2}) \in \mathcal{A} \times \mathcal{A}$.
- Let $\mathcal{C}$ be the union of all object *synsets* annotated in an image. For each image in VisualGenome, X^c_{ij} is incremented by 1, for each word pair $(i, j) \in \mathcal{S}_{c_1} \times \mathcal{S}_{c_2}$, and for all *synset* pairs $(\mathcal{S}_{c_1}, \mathcal{S}_{c_2}) \in \mathcal{C} \times \mathcal{C}$.
- Let $\mathcal{H}$ be a set of object synsets annotated for an image in ImageNet and its ancestors in WordNet. For each each image in ImageNet, X^{oh}_{ij} is incremented by 1, for each word pair $(i, j) \in \mathcal{S}_{h_1} \times \mathcal{S}_{h_2}$, and for all *synset* pairs $(\mathcal{S}_{h_1}, \mathcal{S}_{h_2}) \in \mathcal{H} \times \mathcal{H}$.

4. Experiments

We analyze ViCo embeddings with respect to the following properties: (1) Does unsupervised clustering result in a natural grouping of words by visual concepts? (Sec. 4.1); (2) Do the word embeddings enable transfer of visual learning (*e.g*., visual recognition) to classes not seen during training? (Sec. 4.2); (3) How well do the embeddings perform on downstream applications? (Sec. 4.3); (4) Does the embedding space show word arithmetic properties ($land - car + aeroplane = sky$)? (Sec. 4.4).[1]

Data for clustering analysis. To answer (1) we manually annotate 495 frequent words in VisualGenome with 13 coarse (see legend in the t-SNE plots in Fig. 4) and 65 fine categories (see appendix for the list of categories).

Data for zero-shot-like analysis. To answer (2), we use CIFAR-100 [20]. We generate 4 splits of the 100 categories into disjoint Seen (categories used for training visual classifiers) and Unseen (categories used for evaluation) sets. We use the following scheme for splitting: The list of 5 sub-categories in each of the 20 coarse categories (provided by CIFAR) is sorted alphabetically and the first k categories are added to Seen and the remaining to Unseen for $k \in \{1, 2, 3, 4\}$.

4.1. Unsupervised Clustering Analysis

The main benefit of word vectors over one-hot or random vectors is the meaningful structure captured in the embedding space: words that are closer in the embedding space are semantically similar. We hypothesize that ViCo represents similarities and differences between visual categories that are missing from GloVe.

Qualitative evidence to support this hypothesis can be found in t-SNE plots shown in Fig. 4, where concatenation of GloVe and ViCo embeddings leads to tighter, more homogenous clusters of the 13 coarse categories than GloVe.

[1] We also perform a *supervised partitioning* analysis which is included in the supplementary material. The results show that a supervised classification algorithm partitions words into visual categories more easily in the ViCo embedding space than in the GloVe or random vector space.

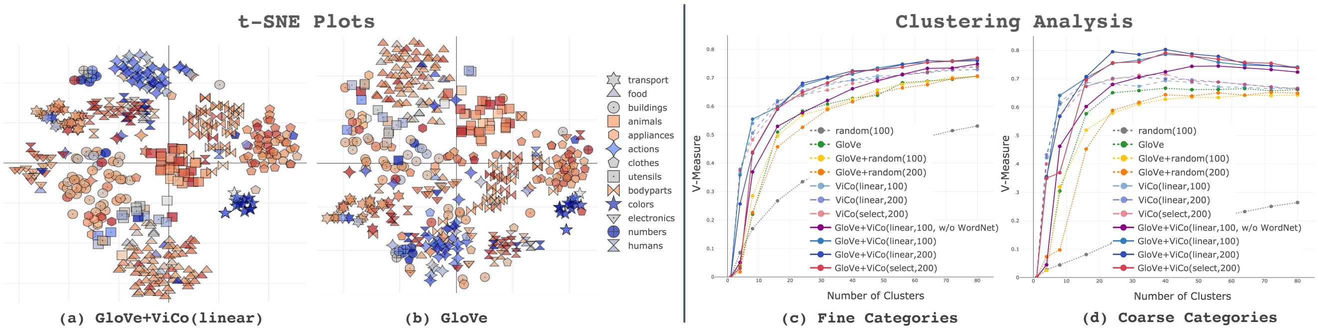


Figure 4. **Unsupervised Clustering Analysis.** (a,b) **Qualitative evaluation with t-SNE:** Plots show that ViCo augmented GloVe results in tighter, more homogenous clusters than GloVe. Marker shape encodes the annotated coarse category and color denotes if the word is used more frequently as an object or an attribute; (c,d) **Quantitative evaluation:** Plots show clustering performance of different embeddings measured through V-Measure at different number of clusters. All ViCo based embeddings outperform GloVe for both fine and coarse annotations (Sec. 4.1). See Tab. 3 and Tab. 4 for average performance across cluster numbers. Best viewed in color on a screen.

To test the hypothesis quantitatively, we cluster word embeddings with agglomerative clustering (cosine affinity and average linkage) and compare to the coarse and fine ground truth annotations using *V-Measure* which is the harmonic mean of *Homogeneity* and *Completeness* scores. *Homogeneity* is a measure of cluster purity, assessing whether all points in the same cluster have the same ground truth label. *Completeness* measures whether all points with the same label belong to the same cluster[2].

Plots (c,d) in Fig. 4 compare random vectors, GloVe, variants of ViCo and their combinations (concatenation) for different number of clusters using V-Measure. Average performance across different cluster numbers is shown in Tab. 3 and Tab. 4. The main conclusions are as follows:

ViCo clusters better than other embeddings. Tab. 3 shows that *ViCo* alone outperforms *GloVe*, *random*, and *vis-w2v* based embeddings. *GloVe+ViCo* improves performance further, especially for coarse categories.

WordNet is not the sole contributor to strong performance of ViCo. To verify that ViCo's gains are not simply due to the hierarchical nature of WordNet, we evaluate a version of ViCo trained on co-occurrences computed without using WordNet, *i.e.*, using raw *word* annotations in VisualGenome instead of *synset* annotations and without Object-Hypernym co-occurrences. Tab. 3 shows that *GloVe+ViCo(linear,100,w/o WordNet)* outperforms *GloVe* for both coarse and fine categories on both metrics.

ViCo outperforms existing visual word embeddings. Tab. 3 evaluates performance of existing visual word embeddings which are learned from abstract scenes [18]. *wiki* and *coco* are different versions of *vis-w2v* depending on the dataset (Wikipedia or MS-COCO [25, 5]) used for training word2vec for initialization. After initialization, both models are trained on an abstract scenes (clipart images) dataset [56]. *ViCo(linear,100)* outperforms both of these embeddings. *GloVe+vis-w2v-wiki* performs similarly to *GloVe* and *GloVe+vis-w2v-wiki-coco* performs only slightly better than *GloVe*, showing that the majority of the information captured by *vis-w2v* may already be present in *GloVe*.

Learned embeddings significantly outperform random vectors. Tab. 3 shows that random vectors perform poorly in comparison to learned embeddings. *GloVe+random* performs similarly to *GloVe* or worse. This implies that gains of *GloVe+ViCo* over *GloVe* are not just an artifact of increased dimensionality.

***Linear* achieves similar performance as *Select* with fewer dimensions.** Tab. 4 illustrates the ability of the multi-task formulation to learn a more compact representatio than *select* (concatenating embeddings learned from each co-occurrence type separately) without sacrificing performance. 50, 100, and 200 dimensional ViCo embeddings learned with linear transformations, all achieve performance similar to *select*.

4.2. Zero-Shot-like Analysis

The ability of word embeddings to capture relations between visual categories enables to generalize visual models trained on limited visual categories to larger sets unseen during training. To assess this ability, we evaluate embeddings on their zero-shot-like object classification performance using the CIFAR-100 dataset. Note that our *zero-shot-like* setup is slightly different from a typical zero-shot setup because even though the visual classifier is not trained on unseen class images in CIFAR, annotations associated with images of unseen categories in VisualGenome or ImageNet may be used to compute word co-occurrences while learning word embeddings.

Model. Let $f(I) \in \mathbb{R}^n$ be the features extracted from image I using a CNN and let $w_c \in \mathbb{R}^m$ denote the word embedding for class $c \in \mathcal{C}$. Let $g : \mathbb{R}^m \rightarrow \mathbb{R}^n$ denote a function that projects word embeddings into the space of image features. We define the score $s_c(I)$ for class c as $\text{cosine}(f(I), g(w_c))$, where $\text{cosine}(\cdot)$ is the cosine similar-

[2]Analysis with other metrics and methods yields similar conclusions and is included in the supplementary material.

Embeddings	Dim.	Fine	Coarse
random(100)	100	0.34	0.15
GloVe	300	0.50	0.52
GloVe+random(100)	300+100	0.50	0.49
vis-w2v-wiki [18]	200	0.41	0.43
vis-w2v-coco [18]	200	0.45	0.4
GloVe+vis-w2v-wiki	300+200	0.5	0.52
GloVe+vis-w2v-coco	300+200	0.52	0.55
ViCo(linear,100)	100	**0.60**	**0.59**
GloVe+ViCo(linear,100)	300+100	**0.61**	**0.65**
GloVe+ViCo(linear,100, w/o WN)	300+100	0.54	0.58

Table 3. **Comparing ViCo to other embeddings.** All ViCo based embeddings outperform GloVe and random vectors. *ViCo(linear,100)* also outperforms *vis-w2v*. *GloVe+vis-w2v* performs similarly to *GloVe* while *GloVe+ViCo* outperforms both *GloVe* and ViCo. Using WordNet yields healthy performance gains but is not the only contributor to performance since *GloVe+ViCo(linear,100, w/o WN)* also outperforms *GloVe*. **Best** and **second best** numbers are highlighted in each column.

Embeddings	Dim.	Fine	Coarse
ViCo(linear,50)	50	0.57	0.56
ViCo(linear,100)	100	**0.60**	0.59
ViCo(linear,200)	200	0.59	0.60
ViCo(select,200)	200	0.59	0.60
GloVe	300	0.50	0.52
GloVe+ViCo(linear,50)	300+50	0.60	**0.66**
GloVe+ViCo(linear,100)	300+100	**0.61**	**0.65**
GloVe+ViCo(linear,200)	300+200	**0.60**	**0.65**
GloVe+ViCo(select,200)	300+200	0.57	0.63

Table 4. **Effect of transformations on clustering performance.** The table compares average performance across number of clusters. The *linear* variants achieve performance similar to *select* with fewer dimensions. In fact, when used in combination with GloVe, *linear* variants outperform *select*. **Best** and **second best** numbers are highlighted in each column.

ity. The class probabilities are defined as

$$p_c(I) = \frac{\exp(s_c(I)/\epsilon)}{\sum_{c' \in \mathcal{C}} \exp(s_{c'}(I)/\epsilon)}, \tag{4}$$

where ϵ is a learnable temperature parameter. In our experiments, $f(I)$ is a 64-dimensional feature vector produced by the last linear layer of a 34-layer ResNet (modified to accept 32×32 CIFAR images) and g is a linear transformation.

Learning. The model (parameters of f, g, and ϵ) is trained on images from the set of seen classes $\mathcal{S} \subset \mathcal{C}$. We use the Adam [17] optimizer with a learning rate of 0.01. The model is trained with a batch size of 0.01 for 50 epochs.

Model Selection and Evaluation. The best model (among iteration checkpoints) is selected based on seen class accuracy (classifying only among classes in $\mathcal{S}$) on the test set. The selected model is evaluated on unseen category ($\mathcal{U} = \mathcal{C} \setminus \mathcal{S}$) prediction accuracy computed on the test set.

Fig. 5 compares chance performance ($1/|\mathcal{U}|$), random vectors, *GloVe*, and *GloVe+ViCo* on four seen/unseen splits.

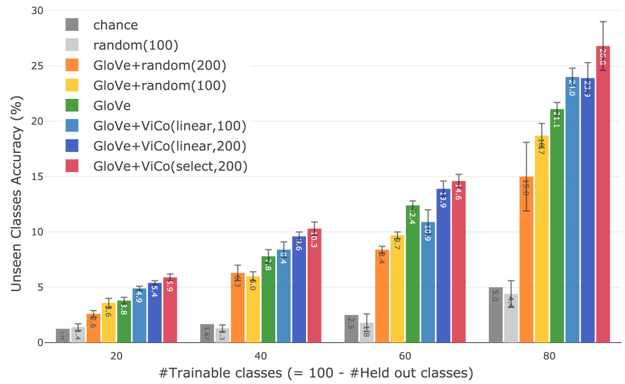


Figure 5. **Zero-Shot Analysis.** The histogram compares the transfer learning ability of a simple word embedding based object classification model. The x-axis denotes the number of CIFAR-100 classes (m) used during training. During test, we evaluate the classifier on its ability to correctly classify among the remaining $(100 - m)$ unseen classes. Results show that *GloVe+ViCo* leads to better transfer to unseen classes than GloVe alone (Sec. 4.2).

We show mean and standard deviation computed across four runs ($7 \times 4 \times 4 = 112$ models trained in all). The key conclusions are as follows:

ViCo generalizes to unseen classes better than GloVe. ViCo based embeddings, especially 200-dim. select and linear variants show healthy gains over *GloVe*. Note that this is not just due to higher dimensions of the embeddings since *GloVe+random(200)* performs worse than *GloVe*.

Learned embeddings significantly outperform random vectors. Random vectors alone achieve close to chance performance, while concatenating random vectors to *GloVe* degrades performance.

Select performs better than Linear. Compression to 100-dimensional embeddings using linear transformation shows a more noticeable drop in performance as compared to the *select* setting. However, *GloVe+ViCo(linear,100)* still outperforms *GloVe* in 3 out of 4 splits.

4.3. Downstream Task Evaluation

We now evaluate ViCo embeddings on a range of downstream tasks. Generally, we expect tasks requiring better word representations of objects and attributes to benefit from our embeddings. When using existing models, we initialize and freeze word embeddings so that performance changes are not due to fine-tuning embeddings of different dimensions. The rest of the model is left untouched except for the dimensions of the input layer where the size of the input features needs to match the embedding dimension.

Tab. 5 compares performance of embeddings on a word-only discriminative attributes task and 4 vision-language tasks. On all tasks *GloVe+ViCo* outpeforms *GloVe* and *GloVe+random*. Unlike the word-only task which depends solely on word representations, vision-language tasks are less sensitive to word embeddings, with performance of ran-

		Discr. Attr.	Im-Cap Retrieval		VQA				Ref. Exp.			Image Captioning			
		Avg. F1	Recall@1		Accuracy				Loc. Accuracy			Captioning Metrics			
Embeddings	**Dim.**	$m \pm \sigma$	Im2Cap	Cap2Im	Overall	Y/N	Num.	Other	Val	TestA	TestB	B1	B4	C	S
random	300	50.03 ± 2.26	43.1	30.6	66.1	82.0	44.8	57.5	71.3	73.5	66.3	**0.714**	**0.296**	**0.910**	**0.170**
GloVe	300	63.85 ± 0.04	**44.8**	33.5	**67.5**	83.8	**46.5**	**58.3**	72.2	**75.3**	66.8	0.708	0.290	0.891	0.167
GloVe + random	300+100	**63.88 ± 0.03**	44.3	**34.4**	**67.5**	**84.1**	45.9	58.2	**72.5**	75.1	**67.5**	0.707	0.288	0.881	0.166
GloVe + ViCo (linear)	300+100	**64.46 ± 0.17**	**46.3**	**34.2**	**67.7**	**84.4**	**46.6**	**58.4**	**72.7**	**75.5**	**67.5**	**0.711**	**0.291**	**0.894**	**0.168**

Table 5. **Comparing ViCo to GloVe and random vectors.** *GloVe+ViCo(linear)* outperforms *GloVe* and *GloVe+random* for all tasks and outperforms *random* for all tasks except Image Captioning. While random vectors perform close to chance on the **word-only** task, they compete tightly with learned embeddings on **vision-language** tasks. This suggests that vision-language models are relatively insensitive to the choice of word embeddings. **Best** and **second best** numbers in each column are highlighted.

dom embeddings approaching learned embeddings [3].

Discriminative Attributes [19] is one of the SemEval 2018 challenges. The task requires to identify whether an attribute word discriminates between two concept words. For example, the word "red" is a discriminative attribute for word pair ("apple", "banana") but not for ("apple", "cherry"). Samples are presented as tuples of attribute and concept words and the model makes a binary prediction. Performance is evaluated using class averaged F1 scores.

Let w_1, w_2, and a be the word embeddings (GloVe or ViCo) for the two concept words and the attribute word. We compute the scores s_g and s_v for GloVe and ViCo using function $s(a, w_1, w_2) = \text{cosine}(a, w_1) - \text{cosine}(a, w_2)$, where cosine$(\cdot)$ is the cosine similarity. We then learn a linear SVM over s_g for the *GloVe* only model and over s_g and s_v for the *GloVe+ViCo* model.

Caption-Image Retrieval is a classic vision-language task requiring a model to retrieve images given a caption or vice versa. We use the open source VSE++ [10] implementation which learns a joint embedding of images and captions using a *Max of Hinges* loss that encourages attending to hard negatives and is geared towards improving top-1 Recall. We evaluate the model using Recall@1 on MS-COCO.

Visual Question Answering [3, 11] systems are required to answer questions about an image. We compare the performance of embeddings using Pythia [55, 15] which uses bottom-up top-down attention for computing a question-relevant image representation. Image features are then fused with a question representation using a GRU operating on word embeddings and fed into an answer classifier. Performance is evaluated using overall and by-question-type accuracy on the test-dev split of the VQA v2.0 dataset.

Referring Expression Comprehension consists of localizing an image region based on a natural language description. We use the open source implementation of MAttNet [54] to compare localization accuracy with different embeddings on the RefCOCO+ dataset using the UNC split. MAttNet uses an attention mechanism to parse the referring expression into phrases that inform the subject's appearance, location, and relationship to other objects. These phrases are processed by corresponding specialized localization modules. The final region scores are a linear combination of module scores using predicted weights.

Image Captioning involves generating a caption given an image. We use the Show and Tell model of Vinyals *et al.* [51] which feeds CNN extracted image features into an LSTM followed by beam search to sample captions. We report BLEU1 (B1), BLEU4 (B4), CIDEr (C), and SPICE (S) metrics [35, 50, 1] on the MS-COCO test set.

4.4. Exploring Embedding Space Structure

Previous work [31] has demonstrated linguistic regularities in word embedding spaces through analogy tasks solved using simple vector arithmetics. Fig. 6 shows qualitatively that ViCo embeddings possess similar properties, capturing relations between visual concepts well.

Analogy	**Answer Candidates**	**GloVe**	**ViCo**
car:land::aeroplane:?	ocean, sky, road, railway	ocean	**sky**
clock:circle::tv:?	triangle, square, octagon, round	triangle	**square**
park:bench::church:?	door, sofa, cabinet, pew	door	**pew**
sheep:fur::person:?	hair, horn, coat, tail	coat	**hair**
monkey:zoo::cat:?	park, house, church, forest	park	**house**
leg:trouser::wrist:?	watch, shoe, tie, bandana	bandana	**watch**
yellow:banana::red:?	strawberry, lemon, mango, orange	mango	**strawberry**
rice:white::spinach:?	blue, green, red, yellow	blue	**green**
train:railway::car:?	land, desert, ocean, sky	land	**land**
can:metallic::bottle:?	wood, glass, cloth, paper	glass	**glass**
man:king::woman:?	queen, girl, female, adult	**queen**	girl
can:metallic::bottle:?	wood, plastic, cloth, paper	**plastic**	wood
train:railway::car:?	road, desert, ocean, sky	**road**	ocean

Table 6. **Answering Analogy Questions.** Out of 30 analogy pairings tested, we found both GloVe and ViCo to be correct 19 times, only ViCo was correct 8 times, and only Glove was correct 3 times. Correct answers are **highlighted**.

5. Conclusion

This work shows that in addition to textual co-occurrences, visual co-occurrences are a surprisingly effective source of information for learning word representations. The resulting embeddings outperform text-only embeddings on unsupervised clustering, supervised partitioning, zero-shot generalization, and various supervised downstream tasks. We also develop a multi-task extension of *GloVe*'s log-bilinear model to learn a compact shared embedding from multiple types of co-occurrences. Type-specific embedding spaces learned as part of the model help provide a richer sense of relatedness between words.

Acknowledgments: Supported in part by NSF 1718221, ONR MURI N00014-16-1-2007, Samsung, and 3M.

[3]See supplementary material for our hypothesis and test for why random vectors work well for vision-language tasks.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 8

[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 1

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 8

[4] John A. Bullinaria and J. P. Levy. Extracting semantic representations from word co-occurrence statistics: a computational study. *Behavior research methods*, 2007. 2

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 6

[6] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *CVPR*, 2017. 1

[7] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JASIS*, 1990. 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 3

[10] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improved visual-semantic embeddings. *BMVC*, 2018. 8

[11] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 8

[12] Tanmay Gupta, Kevin Shih, Saurabh Singh, and Derek Hoiem. Aligned image-word representations improve inductive transfer across vision-language tasks. In *ICCV*, 2017. 1

[13] Mika Hasegawa, Tetsunori Kobayashi, and Yoshihiko Hayashi. Incorporating visual features into word embeddings: A bimodal autoencoder-based approach. In *IWCS*, 2017. 3

[14] Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. Deep semantic role labeling: What works and whats next. In *ACL*, 2017. 1

[15] Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia. https://github.com/facebookresearch/pythia, 2018. 8

[16] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 1

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 2, 7

[18] Satwik Kottur, Ramakrishna Vedantam, José M. F. Moura, and Devi Parikh. Visualword2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 6, 7

[19] Alicia Krebs, Alessandro Lenci, and Denis Paperno. Semeval-2018 task 10: Capturing discriminative attributes. In *International Workshop on Semantic Evaluation*, 2018. 8

[20] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[21] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 1964. 2

[22] Rémi Lebret and Ronan Collobert. Word embeddings through hellinger pca. In *EACL*, 2014. 2

[23] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. *EMNLP*, 2017. 1

[24] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *NIPS*, 2014. 2

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[26] K. Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-ocurrence. 1996. 2

[27] Ruotian Luo and Gregory Shakhnarovich. Comprehension-guided referring expressions. In *CVPR*, 2017. 1

[28] Daniela Massiceti, N Siddharth, Puneet K Dokania, and Philip HS Torr. Flipdial: A generative model for two-way visual dialogue. In *CVPR*, 2018. 1

[29] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. 2

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 2

[31] Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, 2013. 2, 8

[32] George A Miller. Wordnet: a lexical database for english. *ACM*, 1995. 2

[33] Gregory Murphy. *The big book of concepts*. MIT press, 2004. 3

[34] Charles Egerton Osgood, George J Suci, and Percy H Tannenbaum. *The measurement of meaning*. University of Illinois press, 1957. 2

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 8

[36] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *EMNLP*, 2016. 1

[37] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 2, 3, 5

[38] Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. *ACL*, 2017. 1

[39] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. Deep contextualized word representations. In *NAACL-HLT*, 2018. 3

[40] Bryan A. Plummer, Paige Kordas, M. Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *ECCV*, 2018. 1

[41] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *ICCV*, 2017. 1

[42] Alec Radford. Improving language understanding by generative pre-training. 2018. 3

[43] Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. Event2mind: Commonsense inference on events, intents, and reactions. In *ACL*, 2018. 1

[44] Lance J Rips, Edward J Shoben, and Edward E Smith. Semantic distance and the verification of semantic relations. *Journal of verbal learning and verbal behavior*, 1973. 2

[45] Brian H. Ross and Gregory L. Murphy. Food for thought: Cross-classification and category organization in a complex real-world domain. *Cognitive Psychology*, 1999. 2

[46] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *ICLR*, 2017. 1

[47] Kevin J Shih, Saurabh Singh, and Derek Hoiem. Where to look: Focus regions for visual question answering. In *CVPR*, 2016. 1

[48] Gabriel Stanovsky, Julian Michael, Luke S. Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *NAACL-HLT*, 2018. 1

[49] Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *ECCV*, 2018. 1

[50] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 8

[51] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 8

[52] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *TPAMI*, 2019. 1

[53] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, 2018. 1

[54] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 8

[55] Yu Jiang*, Vivek Natarajan*, Xinlei Chen*, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0.1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018. 1, 8

[56] C Lawrence Zitnick and Devi Parikh. Bringing semantics into focus using visual abstraction. In *CVPR*, 2013. 6