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 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).

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 exhaus-
tive 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 Visual 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
describes how co-occurrence count matrices are used to learn word embeddings. (i) Log-bilinear Model (GloVe) learns representations from a single co-occurrence type. (ii) Multi-task Log-bilinear learns a shared, more compact embedding across different co-occurrence types. (iii) Multi-task with Select Transform learns separate embeddings for each co-occurrence type followed by concatenation.

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 $\phi_t$. Log-bilinear losses are computed in the projected embedding spaces. (iii) shows an approach where the different colored regions of the visual features are used as a surrogate label in a CBOW framework. Ab-...
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 $t \in T$ refers to a type from the set of types $T$. Also let $N_t$ and $Z_t$ 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

$$
\sum_{t \in T} \sum_{(i,j) \in N_t} (\phi_t(w_i)^T \phi_t(w_j) + b_i^t + b_j^t - \log X^t_{ij})^2 + \sum_{t \in T} \sum_{(i',j') \in Z_t} \max(0, \phi_t(w_{i'})^T \phi_t(w_{j'}) + b_i^{t'} + b_j^{t'}). \quad (2)
$$

Here $\phi_t : \mathbb{R}^d \rightarrow \mathbb{R}^{d_t}$ 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 $\phi_t$, 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 $\phi_t$. To understand the role of the transformations $\phi_t$ in learning from multiple co-occurrence matrices, consider the naïve approach of concatenating $|T| d_t$-dimensional word embeddings learned separately for each type $t$ using Eq. (1). Such an approach would yield an embedding with $d \geq |T| \min_{t} d_t$ 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?

<table>
<thead>
<tr>
<th>Transforms</th>
<th>$d$</th>
<th>$d_t$</th>
<th>$\phi_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>select (200)</td>
<td>200</td>
<td>50 $\forall t$</td>
<td>$\phi_t(w) = [w_{[i_0]}, \ldots, w_{[i_{10}]}]$ where ${i_0, \ldots, i_{10}}$ are indices pre-allocated for $t \in {0, \ldots, 200}$</td>
</tr>
<tr>
<td>linear (50)</td>
<td>50</td>
<td>50 $\forall t$</td>
<td>$\phi_t(w) = A_t w$ where $A_t \in \mathbb{R}^{50 \times 50}$</td>
</tr>
<tr>
<td>linear (100)</td>
<td>100</td>
<td>50 $\forall t$</td>
<td>$\phi_t(w) = A_t w$ where $A_t \in \mathbb{R}^{100 \times 100}$</td>
</tr>
<tr>
<td>linear (200)</td>
<td>200</td>
<td>50 $\forall t$</td>
<td>$\phi_t(w) = A_t w$ where $A_t \in \mathbb{R}^{200 \times 200}$</td>
</tr>
</tbody>
</table>

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

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 $\phi_t$), 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, $\phi_t$ 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 $\phi_t$ simultaneously for all $t$ in an end-to-end manner.

With this multi-task formulation the dimensions of $w$ can be chosen independently of $|T|$ 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 $\phi_t$ 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 $Z_t$ (zero co-occurrences) close together (high dot product). To suppress such spurious similarities, we include the max term which encourages all word pairs $(i', j') \in Z_t$ to have a small predicted log co-occurrence

$$
\log \tilde{X}_{i' j'}^{t} = \phi_t(w_{i'})^T \phi_t(w_{j'}) + b_{i'} + b_{j'}. \quad (3)
$$
In particular, the second term in the objective linearly penalizes positive predicted log co-ocurrences 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 $X_t$ and $Z_t$ 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 $T = \{oa, aa, c, oh\}$ denote the set of four co-occurrence types and $X^t_{ij}$ denote the number of co-occurrences of type $t \in T$ between words $i$ and $j$. We denote a synset and its associated set of words as $S$. All co-occurrences are initialized to 0. We now describe how each co-occurrence matrix $X^t$ is computed.

- Let $O$ and $A$ be the sets of object and attribute synsets annotated for an image region. For each region in VisualGenome, we increment $X^\text{oa}_{ij}$ by 1, for each word pair $(i, j) \in S_o \times S_a$, and for all synset pairs $(S_o, S_a) \in O \times A$. $X^\text{aa}_{ij}$ is also incremented unless $i = j$.
- For each region in VisualGenome, we increment $X^\text{oh}_{ij}$ by 1, for each word pair $(i, j) \in S_o \times S_a$, and for all synset pairs $(S_o, S_a) \in O \times A$.
- Let $C$ be the union of all object synsets annotated in an image. For each image in VisualGenome, $X^\text{oh}_{ij}$ is incremented by 1, for each word pair $(i, j) \in S_o \times S_a$, and for all synset pairs $(S_o, S_a) \in C \times C$.
- Let $H$ be a set of object synsets annotated for an image in ImageNet and its ancestors in WordNet. For each image in ImageNet, $X^\text{oh}_{ij}$ is incremented by 1, for each word pair $(i, j) \in S_o \times S_a$, and for all synset pairs $(S_o, S_a) \in H \times H$.

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).

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 homogeneous clusters of the 13 coarse categories than GloVe.

---

1We 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.
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 existing 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 representation 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 \( c \in C \). Let \( g: \mathbb{R}^m \rightarrow \mathbb{R}^d \) 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 \( \text{cosine}(-) \) is the cosine similari-
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.

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

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.

<table>
<thead>
<tr>
<th>Embeddings</th>
<th>Dim.</th>
<th>Fine</th>
<th>Coarse</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViCo(linear,50)</td>
<td>50</td>
<td>0.57</td>
<td>0.56</td>
</tr>
<tr>
<td>ViCo(linear,100)</td>
<td>100</td>
<td>0.60</td>
<td>0.59</td>
</tr>
<tr>
<td>ViCo(linear,200)</td>
<td>200</td>
<td>0.59</td>
<td>0.60</td>
</tr>
<tr>
<td>ViCo(select,200)</td>
<td>200</td>
<td>0.59</td>
<td>0.60</td>
</tr>
<tr>
<td>GloVe</td>
<td>300</td>
<td>0.50</td>
<td>0.52</td>
</tr>
<tr>
<td>GloVe+ViCo(linear,50)</td>
<td>300+50</td>
<td>0.60</td>
<td>0.66</td>
</tr>
<tr>
<td>GloVe+ViCo(linear,100)</td>
<td>300+100</td>
<td>0.61</td>
<td>0.65</td>
</tr>
<tr>
<td>GloVe+ViCo(linear,200)</td>
<td>300+200</td>
<td>0.60</td>
<td>0.65</td>
</tr>
<tr>
<td>GloVe+ViCo(select,200)</td>
<td>300+200</td>
<td>0.57</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Learning. The model (parameters of $f$, $g$, and $c$) is trained on images from the set of seen classes $S \subset C$. 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 $S$) on the test set. The selected model is evaluated on unseen category ($U = C \setminus S$) prediction accuracy computed on the test set.

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

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 outperforms 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-
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.

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.

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


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


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