Learning Zero-Shot Multifaceted Visually Grounded Word Embeddings via Multi-Task Training

Hassan Shahmohammadi  Hendrik P. A. Lensch  R. Harald Baayen
University of Tübingen
hassan.shahmohammadi@uni-tuebingen.de
{hendrik.lensch, harald.baayen}@uni-tuebingen.de

Abstract

Language grounding aims at linking the symbolic representation of language (e.g., words) into the rich perceptual knowledge of the outside world. The general approach is to embed both textual and visual information into a common space—the grounded space—confined by an explicit relationship. We argue that since concrete and abstract words are processed differently in the brain, such approaches sacrifice the abstract knowledge obtained from textual statistics in the process of acquiring perceptual information. The focus of this paper is to solve this issue by implicitly grounding the word embeddings. Rather than learning two mappings into a joint space, our approach integrates modalities by implicit alignment. This is achieved by learning a reversible mapping between the textual and the grounded space by means of multi-task training. Intrinsic and extrinsic evaluations show that our way of visual grounding is highly beneficial for both abstract and concrete words. Our embeddings are correlated with human judgments and outperform previous works using pretrained word embeddings on a wide range of benchmarks. Our grounded embeddings are publicly available here.

1 Introduction

The distributional hypothesis asserts that words occurring in similar contexts are semantically related (Harris, 1954). Current state-of-the-art word embedding models (Pennington et al., 2014; Peters et al., 2018a), despite their successful application to various NLP tasks (Wang et al., 2018), suffer from the lack of grounding in general knowledge (Harnad, 1990; Burgess, 2000), such as captured by human perceptual and motor systems (Pulvermüller, 2005; Therriault et al., 2009). To overcome this limitation, research has been directed to linking word embeddings to perceptual knowledge in visual scenes. Most studies have attempted to bring visual and language representations into close vicinity in a common feature space (Silberer and Lapata, 2014; Kurach et al., 2017; Kiela et al., 2018).

However, studies of human cognition indicate that the brain processes abstract and concrete words differently (Paivio, 1990; Anderson et al., 2017) due to the difference in associated sensory perception. According to Montefinese (2019), similar activity for both categories are observed in the perirhinal cortex, a region related to memory and recognition, whereas in the parahippocampal cortex, associated with memory formation, higher activity only occurs for abstract words.

We argue that forcing the textual and visual modalities to be represented in a shared space causes grounded embeddings to suffer from the bias towards concrete words as reported by Park and Myaeng (2017); Kiela et al. (2018). Therefore, we propose a zero-shot approach that implicitly integrates perceptual knowledge into pre-trained textual embeddings (GloVe (Pennington et al., 2014) and fastText (Bojanowski et al., 2017)) via multi-task training. Our approach learns multifaceted grounded embeddings which capture multiple aspects of words’ meaning and are highly beneficial for both concrete and abstract words.

Figure 1 lays out the architecture of our model. It learns a reversible mapping from pre-trained text-based embeddings to grounded embeddings which maintains the linguistic co-occurrence statistics while integrating visual information. The architecture features a similar structure as an auto-encoder (Press and Wolf, 2017) translating from words to grounded space and back. The training is carried out as multi-task learning by combining image captioning in two directions and image-sentence pair discrimination. At the core is a mapping matrix that acts as an intermediate representation between the grounded and textual space, which learns to visually ground the textual word vectors. This mapping is trained on a subset of words and then is
applied to ground the full vocabulary of textual embeddings in a zero-shot manner.

We evaluate our grounded embeddings on both intrinsic and extrinsic tasks (Wang et al., 2019) and show that they outperform textual embeddings and previous related works in the majority of cases. Overall, our contributions are the following:

a) we design a language grounding framework that can effectively ground different pre-trained word embeddings in a zero-shot manner;
b) we create visually grounded versions of two popular word embeddings and make them publicly available;
c) unlike many previous works, our embeddings support both concrete and abstract words;
d) we show that visual grounding has the potential to refine the irregularities of a text-based vector space.

2 Related Works

The many attempts to combine images and text in order to obtain visually grounded word/sentence representations can be grouped into the following categories.

Feature Level Fusion: where the grounded embedding is the result of combining the visual and textual features. Combining strategies range from simple concatenation to adopting SVD and GRU gating mechanisms (Bruni et al., 2014; Kiela and Bottou, 2014; Kiros et al., 2018).

Mapping to Perceptual Space: this is usually a regression task predicting the image vector given its corresponding textual vector. The grounded embedding are extracted from an intermediate layer in auto-encoders (Silberer and Lapata, 2014; Hasegawa et al., 2017), the output of an MLP (Collell Talleda et al., 2017) or an RNN (Kiela et al., 2018). Another method is mapping both modalities into a common space in which their distance is minimized (Kurach et al., 2017; Park and Myaeng, 2017).

Equipping Distributional Semantic Models with Visual Context: here images are treated as a context in the process of computing the word vectors. Many of these approaches modify the Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) models by incorporating image features to the context for concrete words (Hill and Korhonen, 2014; Kottur et al., 2016; Zablocki et al., 2017; Ailem et al., 2018); minimizing the max-margin loss between the image-vector and its corresponding word vectors (Lazaridou et al., 2015); providing social cues based on child-directed speech along with visual scenes (Lazaridou et al., 2016; Zablocki et al., 2017; Ailem et al., 2018); or by extracting the relationship between words and images using multi-view spectral graphs (Fukui et al., 2017).

Hybrid: this category covers the combination of previous methods and other strategies. Here, the grounded word vectors are usually the results of updating the textual word vectors during training (Mao et al., 2016) or the output of sentence encoders such as LSTM (Hochreiter and Schmidhuber, 1997). Such methods include predicting the image vector along with training a language model (Chrupała et al., 2015) or generating an alternative caption at the same time (Kiela et al., 2018). Other approaches such as using the coefficients of
classifiers for grounded representation have also emerged (Moro et al., 2019). Our model falls in the hybrid category as we take a multitasking approach. However, unlike some previous works (Kiela et al., 2018; Collell Talleda et al., 2017; Bordes et al., 2019) we do not impose explicit constraints between the image features and their captions. Our model learns the relationship indirectly via multi-task training.

3 Multi-Task Visual Grounding

In this section, we present the details of the developed method. The training data set \( D \) consists of image–caption pairs, \((S_k, I_k) \in D\), with \( S_k = [w_1, w_2...w_n] \) being a sentence with \( n \) words describing the image \( I_k \). We use the Microsoft_COCO_2017 dataset (Lin et al., 2014) in our experiments. Let \( T_e(w) \in \mathbb{R}^d \) be a pre-trained textual embedding of the word \( w \), which has been trained on textual data only (e.g., GloVe). The objective is to train a mapping matrix \( M \) to ground the word vector \( T_e(w) \) visually, resulting in a grounded embedding \( G_e(w) = T_e(w) \cdot M \), where \( G_e(w) \in \mathbb{R}^c \). To do so, we train the matrix \( M \) to refine the visual vector space via two image-based language model tasks and a binary discrimination task on image-sentence pairs. For the language models, a GRU (Cho et al., 2014) is trained to predict the next word, given the previous words in the sentence provided as image caption, and its associated image vector. The transpose of the textual embedding \( T_e \) is used to compute the probability distribution over the vocabulary (see Figure 1). We employ an identical scenario to form a second language model task using another GRU, where the sentence is fed backward into the model.

The image-sentence discrimination is a binary classification task predicting whether the given sentence \( S_k \) represented in the grounded space matches the image \( I_k \). By training the model simultaneously on these three tasks confined by a linear transformation, we augment the visual information into the grounded embeddings (output of mapping matrix in Figure 1) while preserving the underlying structure of the textual embeddings.

3.1 Language Model

Given the input caption associated with image \( I_k \) as \( S_k = [w_1, w_2...w_n] \), we first encode the words using a pre-trained textual embedding \( T_e \) to obtain the embeddings as \( S_t = [t_1, t_2...t_n] \). We then linearly project these embeddings from the textual space into the visually grounded space via the trainable mapping matrix \( M \) as \( G_e(S_k) = S_t \cdot M \), to obtain a series of grounded vectors \( G_e(S_k) = [x_1, x_2...x_n] \) where \( x_j \in \mathbb{R}^c \). In the grounded space, the perceptual information of the image \( I_k \) corresponding to \( S_k \) is fused using a single-layer GRU \( (G_f (f\text{-forward})) \) in Figure 1) that predicts the next output \( h_{t+1} = GRU_f(x_i, h_t; \theta) \), where \( \theta \) denotes the trainable parameters, \( x_t \) the current input \( (G_e(w_t)) \), and \( h_t \in \mathbb{R}^c \) the current hidden state.

Image information is included by initializing the first hidden state \( h_0 \) with the image vector of \( I_k \). The GRU update gate propagates perceptual knowledge from images into the mapping matrix. This has been shown to be more effective than providing the image vector at each time step as input (Mao et al., 2016).

The transpose of the mapping matrix \( (M^\top) \) is used to map back from grounded space to the textual space. That is, the output of the GRU in each time-step is mapped back into the textual space as \( w_{\text{next}} = h_{t} \cdot M^\top \), where \( w_{\text{next}} \in \mathbb{R}^d \) is an approximation of the next word’s textual embedding. The mapping matrix \( M \) is used to both encode and decode into/from the grounded space. This improves generalization (Press and Wolf, 2017) and prevents the vanishing gradient problem compared to the case where the mapping matrix is only used at the beginning of the network (Mao et al., 2016). \( w_{\text{next}} \) is fed into the transpose of the textual embeddings in the same scenario: \( z = T_e^\top(w_{\text{next}}) \), where \( z \in \mathbb{R}^{|V|} \) and \( V \) indicates the vocabulary. The final probability distribution over \( V \) is computed by a softmax:

\[ P(y = j | z) = \frac{e^{z^\top W_j}}{\sum_{c=1}^{V} e^{z^\top W_c}} \tag{1} \]

Defining the input (previous words and the image vector) and the predicted output (next word prediction) as above, we minimize the categorical cross entropy which is computed for batch \( B \) as:

\[ L_{FW}(\theta) = -\frac{1}{|B|} \sum_{i \in B} \sum_{c \in V} y_{i,c} \log(\hat{y}_{i,c}), \tag{2} \]

Where \( \hat{y}_{i,c} \) and \( y_{i,c} \) are the predicted probability and ground truth for sample \( i \) with respect to the class \( c \).

Moreover, we define a second similar task: Given the input caption associated with image \( I_k \) as \( S_k = [w_1, w_2...w_n] \), we reverse the order of
the words: $S_k = [w_n, w_{n-1}... w_1]$ and use another 
GRU ($G_b$ (b-backward) in Figure 1) with identical 
structure trained on the loss $L_{BW} (\theta)$. The rest of 
the network is shared between these two tasks. Hav-
ing this backward language model is analogous to 
bi-directional GRUs (Schuster and Paliwal, 1997) 
which, however, can not be used directly since the 
ground truth would be exposed by operating in both 
directions.

3.2 Image-sentence discrimination

Even though context-driven word representations 
are a powerful way to obtain word embeddings 
(Pennington et al., 2014; Peters et al., 2018a), the 
performance of such models varies on language-
vision tasks (Burns et al., 2019). Therefore, we 
propose yet another task to align the textual word 
embeddings to their real-world relations in the images. 
The discrimination task predicts whether the given 
image and sentence describe the same content or 
not (shown by ‘caption-image match?’ in Figure 1). 
These types of tasks have been shown effective for 
learning cross-modality representations (Lu et al., 
2019; Tan and Bansal, 2019).

Given the input caption for image $I_k$ as $S_k = 
[w_1, w_2... w_n]$, after projecting the embeddings into 
the grounded space as before, we encode the whole 
sentence by employing a third single-layer GRU 
($G_m$ in Figure 1) with the same structure as before 
$h_n = GRU_m (G_e (S_k), h_0|\theta)$. Where the last out-
put $h_n$ encodes the whole sentence. $h_0$ is again 
initialized with the image vector of $I_k$. The final 
output is computed by a sigmoid function. This 
task shares the mapping matrix $M$ and the textual 
embeddings $T_c$. We minimize the binary cross en-
tropy, which could be computed for each batch as:

$$L_B(\theta) = -\frac{1}{|B|} \sum_{i\in B} y_i \log(\hat{y}_i) + (1-y_i) \log(1-\hat{y}_i).$$ (3)

For negative mining, half of the captions in each 
batch are replaced with captions of different, ran-
images.

3.3 Regularization and overall loss

All the three tasks explained above share the pre-
trained textual embeddings (see Figure 1) which 
gives rise to the question of whether the textual 
embeddings should be updated or kept fixed dur-
ing training. By updating, we might distort the 
pre-trained semantic relations, especially given our 
limited training data. Keeping them fixed, on the 
other hand, does not provide the flexibility to gen-
erate the desired grounding as these embeddings 
are noisy and not perfect (Yu et al., 2017). To pre-
vent distorting the semantic information of words 
while retaining sufficient flexibility, we propose the 
following regularization on the embedding matrix $T_c$: 

$$R(\alpha, \beta) = \frac{\alpha}{|V|} \sum_{w\in V} |\frac{w_c \cdot w_n}{\|w_c\|\|w_n\|} - \beta|,$$ (4)

where $\alpha$ controls the overall impact and $\beta$ controls 
how much the new word vectors $w_n$ are allowed to 
devote from the pre-trained embedding $w_c$. $\beta = 1$ 
indicates no deviation and $\beta = 0$ allows for up to 
90 degree deviation from $w_n$ when minimizing the 
equation. We join all the tasks into a single model 
and minimize the following loss:

$$L_{Alt}(\Theta) = L_{FW}(\theta) + L_{BW}(\theta) + L_B(\theta) + R(\alpha, \beta)$$ (5)

where $\Theta$ denotes all the trainable parameters.

4 Experimental setup

We use the Microsoft_COCO_2017 dataset (Lin 
et al., 2014) for training. Each sample contains 
an image with 5 captions. The dataset is split into 
118k train and 5k validation samples. Each batch 
includes 256 image vectors along with one of their 
captions. Hence, multiple image vectors might oc-
cur in each batch. Image vectors are obtained by 
transferring the penultimate layer of pre-trained 
Inception-V3 (Szegedy et al., 2016) trained on 
ImageNet (Deng et al., 2009). A NN with one 
hidden layer and tanh activation is employed to 
project the image vectors into the initial hidden 
state of the GRUs; $h_l \in \mathbb{R}^{1024}$. We lowercase all 
the words, delete the punctuation marks, and only 
keep the top 10k most frequent words. Two pop-
ular pre-trained textual word embeddings namely 
GloVe (crawl−300d−2.2M−cased) and fastText 
(crawl−300d−2M−SubW) are used for ini-
tialization of the embedding $T_c$. The mapping ma-
trix $M$ transforms the textual embeddings into 
the grounded space. We investigate the best dimension 
of this step and the improvement over pure textual 
embeddings in the next sections. Batch normali-
zation (Ioffe and Szegedy, 2015) is applied after each 
GRU. For the regularization, $R(\alpha = 0.001, \beta = 1)$ 
for GloVe and $R(\alpha = 0.01, \beta = 0)$ for fastText 
yielded the best relative results by meta parame-
ter search. This shows that FastText embeddings
require more deviation ($\beta = 0$ indicates 90 degree deviation) to adapt to the proposed tasks. We trained the model for 20 epochs with 5 epochs tolerance early stopping using NAdam (Dozat, 2016) with a learning rate of 0.001.

As we train a single mapping matrix $M$ for projecting from textual to grounded space, it can be used after the training to transfer out-of-vocabulary (OoV) word-vectors into the grounded space in a zero-shot manner. This way, visually grounded versions of both Glove and fastText are obtained despite being exposed to only 10k words.

5 Evaluations

While the question of what is a good word embedding model is still open (Wang et al., 2019), there are two main categories of evaluation methods: intrinsic and extrinsic. Intrinsic evaluators measure the quality of word embeddings independent of any downstream tasks. For instance, quality can be assessed by comparing similarities between embeddings with word similarities as perceived by human raters. Extrinsic evaluators on the other hand assess the performance based on sentence-level downstream tasks. There is not necessarily a positive correlation between intrinsic and extrinsic methods for a word embedding model (Wang et al., 2019). Nonetheless, we use both types of evaluators to compare our visually grounded embeddings with those presented in related works as well as to purely text-based embeddings.

**Baselines:** we considered two types of embeddings as baselines 1) the pre-trained textual embeddings $T_e$, 2) $T_e$ refined based only on the captions without injecting any image information using a similar language modeling task $L_{FW}$ with a one-layer GRU ($h_t \in \mathbb{R}^{1024}$) followed by a fully connected layer. We refer to the second baseline as C_GloVe and C_fastText for Glove and fastText trained only on captions.

**Intrinsic Evaluators:** We evaluate on some of the common lexical semantic similarity benchmarks: MEN (Bruni et al., 2014), SimLex999 (Hill et al., 2015), Rare-Words (Luong et al., 2013), MTurk771 (Halawi et al., 2012), WordSim353 (Finkelstein et al., 2001), and SimVerb3500 (Gertz et al., 2016). The evaluation metric is the Spearman correlation between the predicted cosine similarity vector and the ground truth.

**Extrinsic Evaluators:** We evaluate on the semantic textual similarity benchmarks (STS) from year 2012 to 2016 using SentEval (Conneau and Kiela, 2018). Here, the task is to measure the semantic equivalence of a pair of sentences solely based on their cosine coefficient. We are particularly interested in these benchmarks for two reasons. 1) They evaluate the generalization power of the given vector space without any fine-tuning. 2) Since they contain sentences from various sources such as news headlines and public forums, they reveal whether abstract knowledge is still preserved by our framework. We used BoW (averaging) to obtain sentence representations. While BoW is a simple sentence encoder, it is a great tool to evaluate the underlying structure of a vector space. For instance, the BoW representation of a pair of sentences such as ‘her dog is very smart’ and ‘his cat is too dumb’ are, unfortunately, very similar in a vector space that does not distinguish dissimilar from related words (e.g., smart and dumb). We will show that our model properly refines the textual vector space and alleviates these kinds of irregularities.

<table>
<thead>
<tr>
<th>Model</th>
<th>RW</th>
<th>MEN</th>
<th>WSim</th>
<th>MTurk</th>
<th>SimVerb</th>
<th>SimLex</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GloVe</td>
<td>45.5</td>
<td>80.5</td>
<td>73.8</td>
<td>71.5</td>
<td>28.3</td>
<td>40.8</td>
<td>56.7</td>
</tr>
<tr>
<td>C_GloVe</td>
<td>46</td>
<td>82.1</td>
<td>74.1</td>
<td>72.3</td>
<td>29.3</td>
<td>43.3</td>
<td>57.8</td>
</tr>
<tr>
<td>VGE_G</td>
<td>52.6</td>
<td>85.1</td>
<td>78.9</td>
<td>73.4</td>
<td>37.4</td>
<td>51.8</td>
<td>63.2</td>
</tr>
<tr>
<td>FastText</td>
<td>56.1</td>
<td>81.5</td>
<td>72.2</td>
<td>75.1</td>
<td>37.8</td>
<td>47.1</td>
<td>61.6</td>
</tr>
<tr>
<td>C_FastText</td>
<td>49.2</td>
<td>68.3</td>
<td>58.1</td>
<td>56.8</td>
<td>30.3</td>
<td>41.9</td>
<td>50.76</td>
</tr>
<tr>
<td>VGE_F</td>
<td>57.8</td>
<td>83.6</td>
<td>73.9</td>
<td>76.1</td>
<td>39.2</td>
<td>49.0</td>
<td>63.2</td>
</tr>
</tbody>
</table>

Table 1: Intrinsic evaluation. Visual grounding (denoted by ‘VGE’) improves the results compared to the baselines on all test sets.

<table>
<thead>
<tr>
<th>Model</th>
<th>RW</th>
<th>MEN</th>
<th>WSim</th>
<th>MTurk</th>
<th>SimVerb</th>
<th>SimLex</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGE_G</td>
<td>52.6</td>
<td>85.1</td>
<td>78.9</td>
<td>73.4</td>
<td>37.4</td>
<td>51.8</td>
<td></td>
</tr>
<tr>
<td>VGE_F</td>
<td>57.8</td>
<td>83.6</td>
<td>73.9</td>
<td>76.1</td>
<td>39.2</td>
<td>49.0</td>
<td></td>
</tr>
<tr>
<td>Cap2Both</td>
<td>48.7</td>
<td>81.9</td>
<td>71.2</td>
<td>–</td>
<td>–</td>
<td>46.7</td>
<td></td>
</tr>
<tr>
<td>Cap2Img</td>
<td>52.3</td>
<td>84.5</td>
<td>75.3</td>
<td>–</td>
<td>–</td>
<td>51.5</td>
<td></td>
</tr>
<tr>
<td>Park et al.</td>
<td>83.8</td>
<td>77.5</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>58.0</td>
<td></td>
</tr>
<tr>
<td>Park, VG.</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>15.7</td>
<td></td>
</tr>
<tr>
<td>Collell et al.</td>
<td>81.3</td>
<td>–</td>
<td>–</td>
<td>28.6</td>
<td>41.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Comparison of grounded embeddings to previous work on intrinsic tasks. Ours are denoted by VGE.

6 Results

**Intrinsic Evaluation – Baselines:** Table 1 shows the intrinsic evaluation results for the baselines and our visually grounded embeddings (VGE_F and VGE_G for visually grounded fastText and Glove respectively). In general, fastText performs better on word-level tasks compared to GloVe, prob-
ably because it provides more context for each word by leveraging from its sub-words. The results also validate the efficacy of our proposed model since updating the embeddings on captions alone (C_fastText and C_GloVe) brings subtle or no improvements. By the proposed visual grounding, significant improvements are achieved on all datasets for both fastText and GloVe. Analyzing why the improvement varies across different datasets is difficult. However, the table reveals interesting properties. For instance, the improvement on SimLex999, which focuses more on the similarity between words, is larger than that on WSim353, which does not distinguish between similarity and relatedness. Hence, visual grounding seems to prioritize similarity over relatedness. Considering the overall performance, it enhances both embeddings to the same level despite their fundamental differences.

**Intrinsic Evaluation – Grounded Embeddings:**

We compare our model to related grounded embeddings by (Collell Talleda et al., 2017; Park and Myaeng, 2017; Kiros et al., 2018; Kiela et al., 2018) (Table 2). We limit our comparison to those who adopted the pre-trained GloVe or fastText since these pre-trained models alone outperform many visually grounded embeddings such as (Hasegawa et al., 2017; Zablocki et al., 2017) on many of our evaluation datasets.

Conceptually, Kiela et al. (2018) also induces visual grounding on GloVe by using the MSCOCO data set. Even though they propose a number of tasks for training (Cap2Img: predicting the image vector from its caption, Cap2Cap: generate an alternative caption of the same image; Cap2Both: training by Cap2Cap and Cap2Img simultaneously) our model clearly outperforms them as ours integrate visual information without degraded performance on abstract words.

Park and Myaeng (2017) proposed a polymodal approach by creating and combining six different types of embeddings (linear and syntactic contexts, cognition, sentiment, emotion, and perception) for each word. Even though they used two pre-trained embeddings (GloVe and Word2vec) and other resources, our model still outperforms their approach on MEN and WSim353, but their approach is better on Simlex999. This performance can be attributed to the many-modality training as using only their visually grounded embeddings (Park_VG) performs much worse. This clearly shows that their visual embeddings do not benefit abstract words (cf. Park and Myaeng, 2017). In summary, our approach benefits from capturing different perspectives of the words’ meanings by learning the reversible mapping in the context of multi-task learning.

**Fine-Grained Intrinsic Evaluation:** we further evaluate our model on the different categories of SimLex999 divided into nine sections: all (the whole dataset), adjectives, nouns, verbs, concreteness quartiles (from 1 to 4 increasing the degree of concreteness), and hard pairs. The hard section indicates 333 pairs whose similarity is hard to discriminate from relatedness. The results for our best embeddings on SimLex999 (VGE_G) are shown in Table 3. We see a large improvement over GloVe in all categories. Some previous approaches such as (Park and Myaeng, 2017) concluded that perceptual information would be beneficial only to concrete words (e.g., apple, table) and would adversely affect abstract words (e.g., happy, freedom). However, our model succeeds in maintaining high-precision co-occurrence statistics from the textual model while augmenting these with perceptual information, in such a way that the representations for abstract words are actually enhanced. Therefore, it outperforms GloVe not only on concrete pairs (conc-q4) but also on highly abstract pairs (conc-q1).

We compared the results on SimLex999 with another recent visually grounded model called Picturebook (Kiros et al., 2018), which employs a multi-modal gating mechanism (similar to a LSTM and GRU update gate) to fuse the Glove and Picturebook embeddings (Table 3). It uses image feature vectors pre-trained on a fine-grained similarity task with 100+ million images (Wang et al., 2014). Picturebook’s performance is highly biased toward concrete words (conc-q3, conc-q4) and performs worse than GloVe by nearly 29% on highly abstract words (conc-q1). Picturebook + GloVe on the other hand shows better results but still performs worse on highly abstract words and adjectives. Our model (VGE_G) can generalize across different categories and outperforms Picturebook+Glove with a large margin on most of the categories while being quite comparable on the others.

**Refining the Textual Vector Space:** Our grounded embeddings, while improving relatedness scores, prioritize similarity over relatedness. This is further demonstrated through inspection of nearest neighbors (Table 5). Given the word ‘bird’,
Table 3: SimLex999 (Spearman’s ρ) results. Conc-q1 and Conc-q4 contain the most abstract and concrete words respectively. Our embeddings (VGE_G) generalize across different word types and strongly outperform all the others on most of the categories.

Table 4: Comparison (Pearson correlation ×100) of our embeddings (VGE_*) with baselines (first two sections) and other word embeddings (bottom) on STS.

Extrinsic Evaluation: Table 4 shows the results on semantic similarity benchmarks. Both grounded embeddings strongly outperform their textual version on all benchmarks. While fastText outperforms GloVe on intrinsic tasks, GloVe is superior here. The reason might be that unlike fastText GloVe treats each word as a single unit and takes into account the global co-occurrences of words. This probably helps to capture the high-level structure of words (e.g., in sentences). Considering the mean score, our model boosts both embeddings approximately by 10 percent.

Furthermore, while we are well aware that our simple averaging model cannot compete with the state-of-the-art sequence models (Gao et al., 2021) on the sentence level STS task, we compare it to other word embeddings to highlight the contribution of visual grounding. Table 4 (bottom) shows the results of our best model (VGE_G) with other textual word embeddings namely ELMo (Peters et al., 2018b), Word2Vec (Mikolov et al., 2013), and Power-Mean (Rücklé et al., 2018) reported by (Perone et al., 2018). While the textual GloVe is the second-worst model (by mean score: 52.84) in the table, its grounded version VGE_G is the best one. Overall, these results confirm that 1) our grounding framework effectively integrates perceptual knowledge that is missing in purely text-based embeddings and 2) visual grounding is highly beneficial for downstream language tasks. It would be interesting to see if our findings extend to grounded sentence embedding models (Sileo, 2021; Bordes et al., 2019; Tan and Bansal, 2020) for instance by training transformer-based models such as BERT (Devlin et al., 2018) on top of our embeddings. However, we postpone this to the future since our focus here is on grounding word embeddings.

7 Model Analysis

We further analyze the performance of our model from different perspectives as follows.

Dependency on the Encoding Dimension c: We train our model with different dimensions of the

GloVe returns ‘turtle’ and ‘nest’ while grounded GloVe returns ‘sparrow’ and ‘avian’, which both reference birds. Moreover, our embeddings retrieve more meaningful words regardless of the degree of abstractness. For the word ‘happy’ for example, GloVe suffers from a bias toward dissimilar words with high co-occurrence such as ‘everyone’, ‘always’, and ‘wish’. This issue is intrinsic to the fundamental assumption of the distributional hypothesis that words in the same context tend to be semantically related. Therefore, Glove embeddings, even though trained on 840 billion tokens, still reports antonyms such as ‘smart’ and ‘dumb’ as very similar. In addition, common misspellings of words (e.g., ‘togheter’) while serving the same role, occur with different frequencies in changing context. Hence, they are pulled apart in purely text-based vector spaces. However, our visual grounding model clearly puts them in the same cluster. Our model therefore seems to refine the text-based vector space by aligning it (via the mapping matrix) with real-world relations (in the images). This refinement generalizes to all the words by using our zero-shot mapping matrix which explains the improvement on highly abstract words. A sample of nearest neighbors for FastText and VGE_F is available in Appendix B. However, since FastText already performs quite well on intrinsic tasks, the difference with its grounded version is subtle which also confirms the results in Table 1.
Table 5: Results of 10 nearest neighbors for Glove (G) and VGE_G (V). Only the differing neighbors are reported. While GloVe retrieves more related words, ours (VGE_G) focuses on similar words. Overall, VGE_G is closer to human judgment and retrieves highly semantically similar words.

Table 6: Sensitivity analysis (Spearman’s $\rho$) on intrinsic datasets. $\alpha = 1$ indicates no use of GloVe and $\alpha = 0$ means no use of VGE_G. Pure grounded embeddings alone yield the best results on 3 of the datasets.

Table 7: Mean score (Spearman’s $\rho$) on intrinsic datasets with respect to each task. $L_{All}$ refers to all the three tasks and $R(\alpha, \beta)$ the regularization loss.

Table 8: Effect of grounded word-vectors magnitude on intrinsic tasks. ‘G’ and ‘V’ refers to Glove and VGE_G respectively. Significant improvement is achieved even with the same size as the textual GloVe.

because these datasets focus on the relatedness of words while SimLex999 for instance distinguishes between similarity and relatedness.

Ablation Study: We further analyze the contribution of each task by performing an ablation evaluation. Table 7 shows the mean score on all the intrinsic datasets (see Table 1) with respect to each loss for both embeddings. While both GloVe and FastText show the same behaviour for language model tasks, fastText embeddings require more deviation ($\beta = 0$ in $R(\alpha, \beta)$) to adapt to the binary discrimination task ($L_{BW}$). Textual embeddings $T_e$ were frozen for all the cases except for $L_{All}$. Even though the best performance, considering all the datasets, is achieved by using all the losses (including the regularization), each loss contributes differently to the overall performance. A more detailed ablation study based on the SimLex999 dataset is provided in Appendix A.

Connections to Human Cognition: Motivated by the different processing patterns of abstract and concrete words in the brain Montefinese (2019), we showed that it is possible to benefit from visual information without learning the two modalities in a joined space. Our experiments show that leveraging visual knowledge to inform the distributional models about the real world might be a better way of integrating language and vision. These modalities while separated could be informed and aligned with each other.
We investigated the effect of integrating perceptual knowledge from images into word embeddings via multi-task training. We constructed the visually grounded versions of GloVe and fastText by learning a zero-shot transformation from textual to grounded space trained on the MSCOCO dataset. Results on intrinsic and extrinsic evaluation support that visual grounding benefits current textual word embedding models. The major findings in our experiments are as follows:

a) Our improvement of visual grounding is not limited to words with concrete meanings and covers highly abstract words as well.

b) Discrimination between relatedness and similarity is more precise when using grounded embeddings.

c) Perceptual knowledge can profitably be transferred to purely textual downstream tasks.

Moreover, we showed that visual grounding has the potential to refine the irregularities in textual vector spaces by aligning words with their real-world relations. This paves the way for future research on how visual grounding could resolve the problem of dissimilar words that occur frequently in the same context (e.g., small and big). In the future, we will investigate whether transformer blocks could profitably replace the GRU cells since they lead the state-of-the-art in many downstream sentence tasks. Moreover, while thus far our focus has been on words, a similar approach could be extended to obtain grounded sentence representations.

Acknowledgements

This work has been supported by EXC number 2064/1 – Project number 390727645, as well as by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Hassan Shahmohammadi. The third author was supported by ERC-WIDE (European Research Council – Wide Incremental learning with Discrimination nEtworks), grant number 742545.

References


Karol Kurach, Sylvain Gelly, Michal Jastrzebski, Philip Haeusser, Olivier Teytaud, Damien Vincent,


9 Appendix

A Fine-Grained Ablation Study

In this section, we provide a more detailed ablation study based on the SimLex999 dataset for both FastText and GloVe. Shown in Table 10, the results reveal interesting findings. The binary discrimination task ($\mathcal{L}_B$) is the most beneficial one for adjectives in the case of both embeddings. This improvement arguably comes from the missing information in textual representations such as shapes, colors, and sizes of the objects which are fused by this cross-modality alignment. $\mathcal{L}_B$ also boosts the performance of the ‘Hard’ section in which similarity is hard to distinguish from relatedness. The reason probably lies in the shift of focus toward similarity (see Table 5) which makes it easier to distinguish between similarity and relatedness. The language model tasks ($\mathcal{L}_{FW}$ and $\mathcal{L}_{BW}$) seem to contribute the most to nouns and verbs describing the scenes in the images. Moreover, our best model ($\mathcal{L}_{All} + \mathcal{R}(\alpha, \beta)$), regarding all the datasets, does not achieve the best result here because each dataset focuses on a different aspect of the language (e.g., similarity or relatedness). However, our final embeddings incorporate the information from different perspectives and improve on all the datasets.

B Refining the Textual Vector Space

Similar to the visually grounded GloVe embeddings, the grounded FastText (VGE_F) also refine the irregularities of textual vector space (referring to Section 6). Examples of differing nearest neighbors are reported in Table 9. Since FastText performs quite well on word-level tasks, the difference is very subtle. The improvement seems to mainly fall into alleviating the antonym problem (e.g., for ‘democracy’ in the table) and clustering typos together (e.g., ‘medicine’ and ‘medecine’). We can also observe tokens such as ‘round.And’ that FastText’s tokenizer has failed to split but have been cluster together by our approach. Overall, the table confirms the results in Table 1.
Table 9: Results of 10 nearest neighbors for FastText(F) and VGE_F (V). Only the differing neighbors are reported.

<table>
<thead>
<tr>
<th>Model</th>
<th>SimLex999</th>
<th>AdjS</th>
<th>Nouns</th>
<th>Verbs</th>
<th>Conc-q1</th>
<th>Conc-q2</th>
<th>Conc-q3</th>
<th>Conc-q4</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>GloVe</td>
<td>40.8</td>
<td>62.2</td>
<td>42.8</td>
<td>19.6</td>
<td>43.3</td>
<td>41.6</td>
<td>42.3</td>
<td>40.2</td>
<td>27.2</td>
</tr>
<tr>
<td>( \mathcal{L}_B )</td>
<td>42.5</td>
<td>70.1</td>
<td>41.3</td>
<td>25.1</td>
<td>45.8</td>
<td>45.9</td>
<td>43.8</td>
<td>46.6</td>
<td>28.1</td>
</tr>
<tr>
<td>( \mathcal{L}_FW )</td>
<td><strong>52.6</strong></td>
<td>70.1</td>
<td>53.1</td>
<td>37.8</td>
<td>54.4</td>
<td>54</td>
<td><strong>49.3</strong></td>
<td>55.2</td>
<td>38</td>
</tr>
<tr>
<td>( \mathcal{L}<em>FW + \mathcal{L}</em>{BW} )</td>
<td>52.5</td>
<td>69.7</td>
<td>52.6</td>
<td>40.6</td>
<td><strong>55.5</strong></td>
<td>54.1</td>
<td>48.7</td>
<td>55.4</td>
<td>38.3</td>
</tr>
<tr>
<td>( \mathcal{L}<em>FW + \mathcal{L}</em>{BW} + \mathcal{L}_B )</td>
<td>52.5</td>
<td>69.8</td>
<td><strong>53.5</strong></td>
<td>37.7</td>
<td>53.3</td>
<td>53.8</td>
<td>48.7</td>
<td><strong>58</strong></td>
<td><strong>39.3</strong></td>
</tr>
<tr>
<td>( \mathcal{L}_{All} + \mathcal{R}(\alpha,\beta) )</td>
<td>51.8</td>
<td><strong>72.1</strong></td>
<td>52.0</td>
<td>35</td>
<td>53.1</td>
<td><strong>54.8</strong></td>
<td>47.4</td>
<td>56.8</td>
<td>38.3</td>
</tr>
<tr>
<td>Fasttext</td>
<td>47.1</td>
<td>59.8</td>
<td>50.5</td>
<td>31.5</td>
<td>46.4</td>
<td><strong>46.8</strong></td>
<td>48.5</td>
<td>52</td>
<td>29.6</td>
</tr>
<tr>
<td>( \mathcal{L}_B )</td>
<td>38.5</td>
<td>64.9</td>
<td>41.7</td>
<td>23.8</td>
<td>37.2</td>
<td>37</td>
<td>41.2</td>
<td>48</td>
<td>26.2</td>
</tr>
<tr>
<td>( \mathcal{L}_FW )</td>
<td>50.2</td>
<td>59</td>
<td>55.8</td>
<td><strong>37.1</strong></td>
<td>47.1</td>
<td>46.1</td>
<td>51.9</td>
<td>60.2</td>
<td>32.1</td>
</tr>
<tr>
<td>( \mathcal{L}<em>FW + \mathcal{L}</em>{BW} )</td>
<td>50.8</td>
<td>59.3</td>
<td><strong>57.3</strong></td>
<td>36</td>
<td>46.1</td>
<td>46.2</td>
<td><strong>53.1</strong></td>
<td><strong>62.3</strong></td>
<td>32.7</td>
</tr>
<tr>
<td>( \mathcal{L}<em>FW + \mathcal{L}</em>{BW} + \mathcal{L}_B )</td>
<td>50.8</td>
<td>60.6</td>
<td>57.1</td>
<td>35.8</td>
<td><strong>48.1</strong></td>
<td>46.4</td>
<td>52.7</td>
<td>61.3</td>
<td><strong>33.4</strong></td>
</tr>
<tr>
<td>( \mathcal{L}_{All} + \mathcal{R}(\alpha,\beta) )</td>
<td>49.0</td>
<td>58.6</td>
<td>54.1</td>
<td>32.9</td>
<td>45.3</td>
<td>46.7</td>
<td>51.3</td>
<td>57.7</td>
<td>31.3</td>
</tr>
</tbody>
</table>

Table 10: Fine-grained ablation study on SimLex999 (Spearman’s \( \rho \)). Conc-q1 and Conc-q4 contain the most abstract and concrete words respectively. The hard section includes a set of word-pairs in which similarity is hard to distinguish from relatedness.