Practical Cross-modal Manifold Alignment for Grounded Language

Andre T. Nguyen,1,2 Luke E. Richards,1,2 Gaoussou Youssouf Kebe,1 Edward Raff,1,2 Kasra Darvish,1 Frank Ferraro,1 Cynthia Matuszek1
1 University of Maryland, Baltimore County, Baltimore, MD
2 Booz Allen Hamilton, Annapolis Junction, MD
Nguyen_Andre@bah.com

Abstract
We propose a cross-modality manifold alignment procedure that leverages triplet loss to jointly learn consistent, multi-modal embeddings of language-based concepts of real-world items. Our approach learns these embeddings by sampling triples of anchor, positive, and negative data points from RGB-depth images and their natural language descriptions. We show that our approach can benefit from, but does not require, post-processing steps such as Procrustes analysis, in contrast to some of our baselines which require it for reasonable performance. We demonstrate the effectiveness of our approach on two datasets commonly used to develop robotic-based grounded language learning systems, where our approach outperforms four baselines, including a state-of-the-art approach, across five evaluation metrics.

1 Introduction: Grounded Language Acquisition Through the Lens of Manifold Alignment

As robots become advanced and affordable enough to have in our daily lives, work needs to be done to make these machines as intuitive as possible. Language offers an approachable interface. However, understanding how natural language can best be grounded to the physical world is still very much an open problem. Combining language and robotics creates unique challenges that much of the current work on grounded language learning has not yet addressed.

Acquiring grounded language—learning associations between symbols in language and their referents in the physical world—takes many forms (Hamad 1990). With some exceptions (Thomason et al. 2016), the majority (Krishna et al. 2016, Salvador et al. 2017) of current work focuses on grounding language to RGB images. Due to the availability of large datasets consisting of up to millions of parallel RGB images and language (Marin et al. 2019, Krishna et al. 2016, Plummer et al. 2015), these tasks typically operate with a large pool of data. Large annotated datasets are rare in the field of grounded language for robotics, especially datasets containing depth information in the form of RGB-D.

This is a complex problem space, and has been demonstrated successfully in domains as varied as soliciting human assistance with tasks (Knepper et al. 2015), interactive learning (She and Chai 2017), and understanding complex spatial expressions (Paul et al. 2018). Previous work (Pillai and Matuszek 2018, Richards and Matuszek 2019) has made many simplifying assumptions such as using a bag-of-words language model and focusing on using domain-specific visual features for training classifier models. Our approach relaxes these assumptions: we do not assume any particular form of language model nor any specific visual features.

In particular, we demonstrate how to recast existing but disparate language and vision domain representations into a joint space. We do so by learning a transform of both language and Red Green Blue Depth (RGB-D) sensor data embeddings into a joint space using manifold alignment. This enables the learning of grounded language in a cross-domain manner and provides a bridge between the noisy, multi-domain perceived world of the robotic agent and unconstrained natural language. In particular, we use triplet loss in combination with Procrustes analysis to achieve the alignment of language and vision. Our approach to alignment attains state-of-the-art performance on the language enhanced University of Washington RGB-D Object Dataset (Richards and Matuszek 2019, Lai et al. 2011) as well as on the dataset of Pillai and Matuszek (2018). Importantly, our approach should be able to integrate with existing robot sensors and models with little additional overhead. The primary contribution of this work is the introduction of an easy to implement manifold alignment approach to the grounded language problem for systems where sensor data representations do not live in the same space. We additionally demonstrate generalizability to the unsupervised setting and examine the contribution of Procrustes analysis post-processing.

2 Related Work

We treat the language grounding problem as one of manifold alignment—finding a mapping from heterogeneous representations (commonly the case with language and sensor datasets) to a shared structure in latent space (Wang and Ma-hadevan 2013). This makes the assumption that there is an underlying manifold that datasets share, obtained by leveraging correspondences between paired data elements.

Jointly learning embeddings for different data domains to a shared latent space can yield a consistent representation of concepts across domains. Figure 1 illustrates the goal of aligning language and vision.

Given n different domains, the manifold alignment task
is to find $n$ functions, $f_1, ..., f_n$, such that each function maps each $m_i$-dimensional space to a shared latent $M$-dimensional space, $f_i : \mathbb{R}^{m_i} \to \mathbb{R}^M, i = 1, ..., n$. In our case, $n = 2$ where the domains correspond to RGB-D and natural language.

Applying manifold alignment to learning groundings between language and physical context is a relatively novel approach.

Most prior work in this area focuses on the cooking domain using the much larger Recipe1M dataset containing around one million cooking recipes and eight hundred thousand food images (Salvador et al. 2017; Carvalho et al. 2018; Fain et al. 2019). Our work differs from these previous works as we demonstrate the effectiveness of a manifold alignment approach on much smaller data (our datasets have less than one percent of the number of data points in the Recipe1M dataset). Lazaridou, Bruni, and Baroni (2014) learn a projection of image-extracted features to an existing and fixed language domain, the loss for that triplet is:

$$L = \max (d(f(x_a), f(x_p)) - d(f(x_a), f(x_n)) + \alpha, 0)$$

(1)

In the above example, $x_a$ could be a cat RGB-D image, $x_p$ a textual description of a cat, and $x_n$ a toaster image. Our primary method, called “Triplet Method” throughout this paper, uses cosine distance as the distance metric $d$.

Once the embedding alignment transformation $f_v$ and $f_l$ are learned, an optional fine-tuning step can be included in the form of a Procrustes analysis (Gower 1975) which finds the optimal translation, scaling, and rotation of two shapes to minimize the Procrustes distance between the shapes. The Procrustes distance is the Euclidean distance between the shapes after the learned optimal translation, scaling, and rotation of shapes. An optimal rotation matrix $R$ is found such that the Euclidean distance between the shapes after translation and scaling is minimized

$$R^* = \arg \min \left\| \frac{f_v(X_v) - m_v}{\|f_v(X_v) - m_v\|_F} - \frac{f_l(X_l) - m_l}{\|f_l(X_l) - m_l\|_F} R^T \right\|_2$$

(3)
and m where X three classes can be described using the word “fruit.” During Food Bag classes. The three examples shown in Figure 2 Figure 2 shows example data from the Tomato, Pear, and Food Bag classes. Each row corresponds to a class, and columns correspond to RGB, depth, and text descriptions.

evaluation, we desire that a good approach map the RGB-D and language representations of each object class near each other but far from the representations of objects from other classes.

4.2 Models

For the language feature extraction model b_l, we use a 12-layer BERT model pre-trained on lower case English text [Devlin et al. 2018]. The resulting BERT embedding is 3,072 dimensional.

For the vision feature extraction b_v, we use a ResNet152 pre-trained on ImageNet [He et al. 2016] with its last fully connected layer removed. The depth component is dealt with via colorization (which we shall call D2RGB) in a similar manner to the procedure from Eitel et al. (2015) which encodes a depth image as an RGB image where the information contained in the depth data is spread across all three RGB channels. This allows us to use the same pre-trained ResNet to process both the RGB image and the transformed depth information. The vectors resulting from the RGB images and the D2RGB depth-to-RGB colorization are concatenated to create a final 4,096 dimensional RGB-D vision embedding. This gives us b_v(x_RGB−D) = [ResNet(x_RGB); ResNet(D2RGB(x_D))].

Lu et al. (2019) introduce ViLBERT, joint representations of images and natural language. We note that while a pre-trained ViLBERT embedding could be used for the vision and language feature extraction, we do not use ViLBERT as our feature extractor in our experiments.

This is because ViLBERT learns vision and language embeddings jointly, and so the representations are already designed to work together. Our interest is in adapting embedding that have no prior relation, since it is increasingly common to use pre-trained models for different sub tasks.

In our experiments, the network architectures for our alignment models consist of an input layer, two hidden layers of size equal to the input layer size, and an output layer that has the size of the desired embedding dimensionality, set to 1,024 in our experiments. Rectified linear units were used as hidden layer activation functions, and Adam was

<table>
<thead>
<tr>
<th>Algorithm 1: Training Procedure for Triplet Method</th>
</tr>
</thead>
</table>
| **Input:** Dataset X of paired RGB-D and language feature vectors (x_v, x_l).  
**Output:** Embedding alignment functions f_v and f_l that map RGB-D and language data to a shared embedding space and a trained Procrustes transform.  
1. f_v, f_l ← randomly initialized neural networks with parameters θ_v and θ_l respectively  
2. while not converged do  
   3. x_v ← randomly selected vision or language feature vector from X  
   4. x_l ← randomly selected vision or language feature vector from X belonging to the same class as x_v  
   5. x_s ← randomly select any other vision or language feature vector from X belonging to a different class than x_v and x_l  
   6. Incur loss L using Equation 2 and backpropogate to update parameters θ_v and θ_l  
7. end  
8. m_v ← \[ l \frac{1}{|X|} \sum_{(x_v, x_l) \in X} f_v(x_v) \]  
9. m_l ← \[ l \frac{1}{|X|} \sum_{(x_v, x_l) \in X} f_l(x_l) \]  
10. s_v ← \|f_v(x_v) - m_v\|_F  
11. s_l ← \|f_l(x_l) - m_l\|_F  
12. R ← solution to Equation 3  
13. return f_v, f_l, m_v, m_l, s_v, s_l, R  

where X_v and X_l are the vision and language data respectively (where rows from each domain form pairs), where m_v and m_l are the means of f_v(X_v) and f_l(X_l), and \| \cdot \|_F is the Frobenius matrix norm. All the Procrustes analysis parameters are fit using the training set. As we will show, our method can benefit from, but does not require, Procrustes analysis, in contrast to some of our baselines which require it for reasonable performance. The full training procedure for the triplet method is described at a high level in Algorithm 1.
used as the optimizer \cite{Kingma2015}. The triplet loss uses cosine distance as the distance metric with a margin of \( \alpha = 0.4 \), where we did not tune the margin. The embedding space is chosen to be 1,024-dimensional and we fix the pre-trained feature extraction models \( b_l \) and \( b_v \) during training, only optimizing the alignment models.

The fixing of the feature extraction models directly connects to the robotics use-case where feature extraction model outputs may be used for multiple tasks and where there may be memory and latency constraints. By not having to store and process data through multiple feature extraction models, our approach is advantageous in how it can fit on top of existing state-of-the-art algorithms used by the robot for separate tasks. To illustrate, the feature extraction models together have 167,626,048 parameters, and the alignment models together have 59,785,216 parameters. In the case of an existing system with language and vision models currently being used for other tasks, the integration of manifold alignment would result in a 36\% increase in the number of parameters if the feature extraction models are reused whereas an 136\% increase in the number of parameters would occur if the feature extraction models are retrained.

4.3 Baselines

We compare our manifold alignment method with the following baselines. We also augment each of these baselines with a Procrustes analysis for additional, stronger baselines.

Canonical Correlation Analysis  Canonical Correlation Analysis (CCA) finds the linear combinations of variables within each of two datasets that maximizes the linear correlation between the combinations from each of the datasets \cite{Hotelling1992}.

Deep CCA  Deep Canonical Correlation Analysis (Deep CCA) is an extension of CCA where a nonlinear transformation of two datasets is learned to maximize the post-transformation linear correlation \cite{Andrew2013}. Deep CCA suffers from known numerical stability issues due to the need to backpropagate through eigen-decompositions. Additionally, mini-batched stochastic gradient descent cannot be directly used for optimization as correlation is a function of the training data in its entirety and does not decompose into a sum over data points. As a result, care needs to be taken and additional tricks potentially used when training Deep CCA \cite{Wang2019}. In particular, we found it necessary to reduce the dimensionality of the embedding space from 1,024 to 64 for Deep CCA in order to avoid numerical instability during the training process. Testing with larger dimensions resulted in a failure to converge.

4.4 Manifold Metrics

To evaluate the quality of the manifolds learned, we will use the three metrics specified below: Mean Reciprocal Rank (a measure of global order preservation), K-Nearest Neighbors (a measure of local order preservation), and Distance Correlation (a measure of global absolute distance preservation). A successfully manifold alignment approach should perform well in all three of these tasks. We do not argue that these are sufficient for determining all aspects about a manifold’s quality, but posit that they are useful to the tasks we are concerned with.

Mean Reciprocal Rank  Given an image and text pair, we can compute the distance in the joint embedding space between the text element and all data points in the vision domain. These distances can then be ranked with 1 being the closest, 2 being the second closest, and so forth. Common in information retrieval, Mean Reciprocal Rank (MRR) is the average across the data of the multiplicative inverse of the rank in embedding space of the nearest item from the same class that comes from the other domain \cite{Craswell2009}.

Distance Correlation  Intuitively, if two embedding manifolds are aligned, distances in one embedding should be correlated to distances in the other embedding. Specifically, if we select two image and text pairs, the distance between them in the vision embedding should be correlated with the distance between them in the language embedding space. To capture this property, we randomly select 10,000 pairs of image and text pairs and compute the distance between them. The Pearson correlation is then computed between the vision space distances and the language space distances, resulting in a metric between −1 and 1 where closer to 1 means better alignment. We call this metric Distance Correlation (DC) in this paper. The sampling is done due to the prohibitive cost to compute the pairwise correlation for the entire dataset.

K-Nearest Neighbors  As a final metric, we use K-Nearest Neighbors (KNN) classification accuracy with \( K = 5 \) in our experiments. This metric captures what performance would look like in an applied setting where a robot may need to associate natural language with a visual concept.

5 Supervised Alignment Evaluation

5.1 Grounded Language Learning

Our ultimate goal for manifold alignment is to enable the grounding of language to referents in the physical world. To directly assess the effectiveness of cross-modal manifold alignment for grounded language, we evaluate the aligned embeddings on the task of determining which objects in RGB-D space correspond to a given language description. In particular, every text description datum can be considered a separate classification task where the goal is the binary classification of all RGB-D images as relevant or not relevant given the text description.

For each of the classification tasks, an Area Under the Receiver Operating Characteristic Curve (AUC) score is obtained. Figure 4 shows cumulative counts over AUCs. We note that for any particular AUC score, our triplet method has more better scoring tasks than Deep CCA. In other words, Deep CCA has more and worse failure cases. We also compare our triplet method which uses cosine distance with a version of our triplet method that uses Euclidean distance instead. This ablation finds our cosine method best.

Table 1 summarizes the mean micro and macro averaged F1 scores across methods. The triplet method outperforms all of the other methods on the grounded language task. For the computation of F1 scores, the distance between the
Figure 3: Grounded language task cumulative counts over AUCs. Lower is better and perfect classification performance lies in bottom-right corner. Every text description datum can be considered a separate classification task (with its own AUC) where the goal is the binary classification of all RGB-D images as relevant or not relevant given the text description. The x-axis represents AUC score values, and the y-axis represents the number of classification tasks with an AUC less than a particular value. For any particular AUC score, our triplet method has more better scoring tasks than Deep CCA.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MRR</th>
<th>KNN</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triplet Method</td>
<td>0.802</td>
<td>0.787</td>
<td>0.686</td>
</tr>
<tr>
<td>Triplet. (w/out Procrustes)</td>
<td>0.758</td>
<td>0.742</td>
<td>0.692</td>
</tr>
<tr>
<td>Triplet. (Euclidean)</td>
<td>0.724</td>
<td>0.702</td>
<td>0.693</td>
</tr>
<tr>
<td>Triplet. (Eucl. w/out Proc.)</td>
<td>0.673</td>
<td>0.648</td>
<td>0.685</td>
</tr>
<tr>
<td>Triplet. (unsupervised)</td>
<td>0.754</td>
<td>0.736</td>
<td>0.773</td>
</tr>
<tr>
<td>Triplet. (unsup. w/out Proc.)</td>
<td>0.688</td>
<td>0.665</td>
<td>0.725</td>
</tr>
<tr>
<td>Cosine Baseline (w/out Proc.)</td>
<td>0.208</td>
<td>0.181</td>
<td>0.0306</td>
</tr>
<tr>
<td>Cosine Baseline (w/ Proc.)</td>
<td>0.113</td>
<td>0.0965</td>
<td>0.00108</td>
</tr>
<tr>
<td>CCA</td>
<td>0.0353</td>
<td>0.0267</td>
<td>0.0400</td>
</tr>
<tr>
<td>CCA (w/ Procrustes)</td>
<td>0.144</td>
<td>0.122</td>
<td>0.0665</td>
</tr>
<tr>
<td>Deep CCA</td>
<td>0.0232</td>
<td>0.0116</td>
<td>0.377</td>
</tr>
<tr>
<td>Deep CCA (w/ Procrustes)</td>
<td>0.870</td>
<td>0.860</td>
<td>0.359</td>
</tr>
</tbody>
</table>

Table 2: Evaluation of manifolds using Mean Reciprocal Rank (MRR), K-Nearest Neighbors (KNN), and Distance Correlation (DC) as metrics. Higher is better for all metrics.

Table 1: Metrics for grounded language task evaluated on held out test set. Best results are bolded.

5.2 Effective Deep Metric Learning using Triplet Loss

Table 2 shows the MRR, KNN accuracy, and DC for the triplet method as well as for our baselines. We find that while the triplet method has the highest DC and strong MRR and KNN accuracy, providing consistent performance across all manifold metrics. Deep CCA with the addition of Procrustes analysis has the highest MRR and KNN, at the cost of a $1.9 \times$ lower DC compared to our new approach. This disparity in performance means that Deep CCA with Procrustes is not learning a holistically useful manifold. As we saw in subsection 5.1, this translates to worse performance for grounded language learning. Deep CCA without Procrustes has a significantly reduced, and in fact the worst, MRR and KNN accuracy. CCA with and without Procrustes analysis both have poor performance. These results demonstrate the value of using Procrustes to improve the quality of a manifold alignment at little effort. We also note that while Procrustes is crucial for CCA and Deep CCA, our triplet method remains strong with only a slight decrease in MRR and KNN accuracy when Procrustes analysis is ablated.

To help confirm that our approach learns good manifolds, we would expect a visualization of the vision and language domains to have similar structure. We do this using UMAP (McInnes, Healy, and Melville 2018), which preserves global structure. Figure 4 shows the UMAP for the triplet method. Ten randomly selected classes are plotted for legibility purposes. We observe that classes are generally well clustered (items are close to other items from the same class and classes are separated) and are projected to similar locations across both the language and vision domains. Note that using our new approach, classes with wide dispersion (e.g., the water bottle) or compactness (e.g., cell phone) share this structure across domains. Figure 5 shows the UMAP for Deep CCA with Procrustes. In contrast with the triplet method, we observe that while data is well clustered in the language domain, data is less well clustered in the vision domain, in particular when it comes to class separation. Class alignment across domains is also less evident. While classes such as “cell phone” and “food bag” are well aligned, other classes such as “kleenex” and “calculator” are not. In these cases the structure is not successfully shared between the domains, indicating a lesser quality as a manifold.
Figure 4: Test set UMAP of the Triplet Method. 10 randomly selected classes are plotted.

Figure 5: Test set UMAP of Deep CCA with Procrustes. 10 randomly selected classes are plotted.

Figure 6: Distance Correlation visualization for the Triplet Method and for Deep CCA with Procrustes. Pairs of image and text pairs are randomly selected and the distance between them is plotted, with the x-axis representing the distance in the language domain and the y-axis representing the distance in the vision domain. The dashed line represents where points should lie under perfect manifold alignment.

Additionally, we can gain more insight into the DC results by plotting the vision space distances and the language space distances to compare their relationships. Subplots (a) and (b) in Figure 6 respectively show the distance relationships for the triplet method and Deep CCA with Procrustes. While the triplet method has the desired linear relationship between distances, Deep CCA with Procrustes lacks the desired relationship that would indicate well aligned manifolds.

5.3 Understanding the Contribution of Procrustes Analysis and Triplets

To better understand the role played by Procrustes analysis, we run ablation experiments, separately removing each of the Procrustes analysis components (translation, scaling, and rotation) one by one.

**Table 3**: Ablation metrics where various components of Procrustes analysis are disabled for the Triplet Method.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MRR</th>
<th>KNN</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triplet Method</td>
<td>0.802</td>
<td>0.787</td>
<td>0.686</td>
</tr>
<tr>
<td>No Translation</td>
<td>0.806</td>
<td>0.790</td>
<td>0.679</td>
</tr>
<tr>
<td>No Scaling</td>
<td>0.801</td>
<td>0.786</td>
<td>0.696</td>
</tr>
<tr>
<td>No Rotation</td>
<td>0.750</td>
<td>0.733</td>
<td>0.693</td>
</tr>
</tbody>
</table>

**Table 4**: Ablation metrics where various components of Procrustes analysis are disabled for the Triplet Method with Euclidean distance.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MRR</th>
<th>KNN</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triplet Method</td>
<td>0.724</td>
<td>0.702</td>
<td>0.693</td>
</tr>
<tr>
<td>No Translation</td>
<td>0.729</td>
<td>0.707</td>
<td>0.688</td>
</tr>
<tr>
<td>No Scaling</td>
<td>0.045</td>
<td>0.039</td>
<td>0.680</td>
</tr>
<tr>
<td>No Rotation</td>
<td>0.680</td>
<td>0.658</td>
<td>0.684</td>
</tr>
</tbody>
</table>

To better understand the role played by Procrustes analysis, we run ablation experiments, separately removing each of the Procrustes analysis components (translation, scaling, and rotation) one by one.

**Table 3** shows metrics for Procrustes analysis ablations on the triplet method. Metrics stay relatively similar when translation or scaling are removed. When rotation is removed, a decrease in MRR and KNN accuracy is observed without a decrease in DC.

**Table 4** shows metrics for Procrustes analysis ablations on a variant of the triplet method that uses Euclidean distance instead of cosine distance. Metrics stay similar when translation or rotation are removed. When scaling is removed, a significant decrease in MRR and KNN accuracy is observed. This suggests that the Euclidean distance without Procrustes maps data in each domain to similarly shaped manifolds of different scales. This result is consistent with the formulation of the Euclidean triplet loss, as differently scaled but otherwise similar manifolds can satisfy the relative distance constraints encouraged by the Euclidean triplet loss. This result demonstrates an advantage of the use of cosine distance in this context. A comparison of the performance of the triplet method with its Euclidean variant in **Table 1** and **Figure 3** confirms this advantage.

Similar ablation experiments can be run for Deep CCA with Procrustes analysis. **Table 3** suggests that both rotation and scaling are needed for Deep CCA to achieve high MRR and KNN accuracy.

We also explore the contribution of using triplets by adding a baseline which seeks to simply minimize the cosine distance between the positive and anchor points in the shared space. **Table 1** and **Table 2** show the performance of the cosine distance baseline, with and without Procrustes analysis. Overall, the triplet method performs significantly better than...
Table 5: Ablation metrics where various components of Procrustes analysis are disabled for Deep CCA.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MRR</th>
<th>KNN</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep CCA w/ Procrustes</td>
<td>0.870</td>
<td>0.860</td>
<td>0.359</td>
</tr>
<tr>
<td>No Translation</td>
<td>0.871</td>
<td>0.862</td>
<td>0.363</td>
</tr>
<tr>
<td>No Scaling</td>
<td>0.820</td>
<td>0.811</td>
<td>0.378</td>
</tr>
<tr>
<td>No Rotation</td>
<td>0.834</td>
<td>0.821</td>
<td>0.352</td>
</tr>
</tbody>
</table>

Table 6: Metrics for grounded language task comparing BERT, SBERT, and a fine-tuned SBERT.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg Micro F1</th>
<th>Avg Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triplet Met. (BERT)</td>
<td>0.984</td>
<td>0.735</td>
</tr>
<tr>
<td>Triplet Met. (SBERT)</td>
<td>0.982</td>
<td>0.748</td>
</tr>
<tr>
<td>Triplet Met. (SBERT fine-tuned)</td>
<td>0.984</td>
<td>0.734</td>
</tr>
</tbody>
</table>

Table 7: Manifold comparison for BERT, SBERT, and a fine-tuned SBERT.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MRR</th>
<th>KNN</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triplet Met. (BERT)</td>
<td>0.816</td>
<td>0.804</td>
<td>0.686</td>
</tr>
<tr>
<td>Triplet Met. (SBERT)</td>
<td>0.745</td>
<td>0.731</td>
<td>0.678</td>
</tr>
<tr>
<td>Triplet Met. (SBERT fine-tuned)</td>
<td>0.834</td>
<td>0.823</td>
<td>0.731</td>
</tr>
</tbody>
</table>

6.4 Comparison of Language Embeddings

Next, we investigate the effect of better feature extraction. Sentence-BERT (SBERT) is a sentence embedding oriented modification of BERT that achieves better performance on Semantic Textual Similarity (STS) tasks (Reimers and Gurevych 2019). We compare a BERT-based version of our triplet method to an off-the-shelf SBERT version and a fine-tuned SBERT version. We fine-tune SBERT using pairs of object descriptions from the same extended University of Washington dataset. Pairs describing the same instance of an object are given a score of 5 while pairs describing different instances of an object are given a score of 2.5, and pairs describing different objects are given a score of 0.

6.1 Sampling Negative Examples in an Unsupervised Setting

So far, the training of the triplet method has assumed the availability of class labels for triplet selection. However, the triplet method can still be trained when class ground truth is not available using unsupervised negative example selection. In this setting, the triplets are fixed to have a vision anchor and language negatives and positives. The positive is selected to be the anchor’s paired text, and the negative example is chosen through a semantic distance based technique similar to that used in (Pillai and Matuszek 2018). In particular, the cosine distances between all natural language descriptions can be computed, and the negative is sampled from the 25% of descriptions furthest away from the positive description. This can be interpreted as aligning vision to the manifold induced by the language embedding. Table 1 and Table 2 summarize the performance of the triplet method in this unsupervised setting. While there is a decrease in MRR and KNN accuracy, DC remains strong and even increases. On the grounded language task, performance also remains strong with only a 2% decrease in average micro F1 and a 4% decrease in average macro F1.

6.2 Effectiveness on a Smaller Dataset

We also test our triplet method on a dataset (Pillai and Matuszek 2018) containing fewer classes and fewer instances per class, with a lower computational cost vision extraction method, depth kernel descriptors (Bo et al. 2011) and average values for RGB channel values. Prior work (Pillai and Matuszek 2018; Pillai, Matuszek, and Ferraro 2019) used these same visual feature extraction methods with a word-as-classifier model. Pillai, Matuszek, and Ferraro (2019) combined depth kernel descriptors and averaged RGB channel values. We concatenate the kernel descriptors and the average RGB channel values into a single vision embedding vector. Each vision vector is paired with a natural language description of the object. On this dataset, the triplet method with Procrustes achieves a mean macro F1 score of 0.722, and the triplet method without Procrustes achieves a mean macro F1 score of 0.729, both of which are better than but still comparable to the reported 0.714 for the non-category based model from previous works.

7 Conclusions

We explored the use of the triplet loss enhanced with Procrustes analysis for manifold alignment in the context of grounded language. Our approach to alignment achieves state-of-the-art performance on two datasets, and integration with existing robot sensors and models would likely have minimal additional overhead. Next steps include the alignment of more than two modalities, integration with a robot system, and evaluation on a wider variety of tasks.

Ethics Statement

We have seen the integration of voice-assistant speakers in homes drastically increase in the recent years, and language may grow to become a preferred method for interacting with...
AI-enabled assistants. As the application of AI grows beyond location-fixed machines to robots that physically interact with our environment, effectively grounding natural language to referents in the physical world is critical. Advances in this space will have broad societal impacts by improving the quality of robotic assistance for elderly and handicapped users and more generally by improving the productivity and quality of life of all users (Kcoeska et al. 2019). We hope that this work will help to improve the state of grounded language for robotics.

More broadly, the manifold alignment approach has the benefit of interpretability, where the robot’s knowledge representation across modalities can be investigated and the manifold studied for AI quality assurance. Also, the ability to recast existing but disparate domain data representations into a joint space is useful in applications outside of robotics. For example, in the space of cyber security, an anti-virus solution may use different models for static and dynamic analysis, and fusing information into a joint space would benefit tasks such as the detection of new virus families (Rafi and Nicholas 2020).

We note that we have not studied the risks posed by the threat of adversarial attacks which could take the form of data poisoning during the learning of the manifold alignment or prediction time evasion attacks (Biggio and Roli 2018). Successful adversarial attacks on robot systems that can physically interact with our environment have the potential to cause significant damage and danger. While we hope that the explicit manifold representation of robot knowledge will help with the development of defenses against adversarial attacks, much work needs to be done in this area.

References


Fain, M.; Ponikar, A.; Fox, R.; and Bollegala, D. 2019. Dividing and Conquering Cross-Modal Recipe Retrieval: from Nearest Neighbours Baselines to SoTA.


Thomason, J.; Sinapov, J.; Svetlik, M.; Stone, P.; and Mooney, R. J. 2016. Learning Multi-Modal Grounded Linguistic Semantics by Playing” I Spy”. In *IJCAI*, 3477–3483.

Wang, C.; and Mahadevan, S. 2013. Manifold alignment preserving global geometry. In *23rd International Joint Conference on Artificial Intelligence (IJCAI)*.
