Grounding and Distinguishing Conceptual Vocabulary Through Similarity Learning in Embodied Simulations

Sadaf Ghaffari and **Nikhil Krishnaswamy**
Situated Grounding and Natural Language (SIGNAL) Lab
Department of Computer Science, Colorado State University
Fort Collins, CO, USA
{sadafgh,nkrishna}@colostate.edu

Abstract

We present a novel method for using agent experiences gathered through an embodied simulation to ground contextualized word vectors to object representations. We use similarity learning to make comparisons between different object types based on their properties when interacted with, and to extract common features pertaining to the objects' behavior. We then use an affine transformation to calculate a projection matrix that transforms contextualized word vectors from different transformer-based language models into this learned space, and evaluate whether new test instances of transformed token vectors identify the correct concept in the object embedding space. Our results expose properties of the embedding spaces of four different transformer models and show that grounding object token vectors is usually more helpful to grounding verb and attribute token vectors than the reverse, which reflects earlier conclusions in the analogical reasoning and psycholinguistic literature.

1 Introduction

A common critique of modern large language models (LLMs) is that they lack *understanding* in the sense of being able to link an utterance to a specific communicative intent (Bender and Koller, 2020). This shortcoming is often characterized as being due to a lack of ability to *ground* or link lexical items to real-world entities such as classes of objects, or associated properties or actions. For instance, a modern generative LLM like ChatGPT[1] may be able to generate coherent text describing an object (e.g., "a *coconut* has a hard, often hairy outer shell"), without any inherent underlying conceptualization of what the item actually *is*.

Crucially, these underlying conceptualizations necessarily invoke other modalities. Existing approaches to grounding in NLP typically treat the problem as one of making the correct kind of link between text and another modality, usually images (Socher et al., 2014; Yatskar et al., 2016; Zhu et al., 2020, 2021). However, still images do not capture the wealth of information humans receive when interacting with objects or experiencing events, and video data requires orders of magnitude more data and computational power to effectively process. Additionally, humans do not use vision alone as their only non-linguistic modality.

As humans develop object concept representations and map them to associated nouns, they are also learning to individuate objects from the perceptual flow not just based on visual features but also based on experience that includes interacting with them in real time (Spelke, 1985; Spelke et al., 1989; Spelke, 1990; Baillargeon, 1987). Gentner (2006) argues that Talmy (1975)'s findings on variability in verbal semantics helped to explain why nouns are typically learned before words for verbs or other properties. Concrete nouns are more easily "groundable" not just because of their visual manifestations but also because of their physical presences that leave traces in the world, and these physical properties provide a scaffold on which to build representations of related concepts that are supervenient upon understanding of objects.

In this paper we take an *embodied simulation* approach to grounding, using a virtual environment to create experiences for an agent interacting with objects. We show that similarity learning over data gathered during the agent's experience in the virtual world can not only make comparisons between objects, but also appears to learn information pertaining to more abstract properties of the objects. Fig. 1 shows a schematic view of our overall approach. We map token vectors from different transformer-based LLMs into the resulting representation space, and show that with just a few samples, grounding noun representations alone is

[1] https://chat.openai.com

Proceedings of the 15th International Conference on Computational Semantics pages 305–317
June 21–23, 2023. ©2023 Association for Computational Linguistics

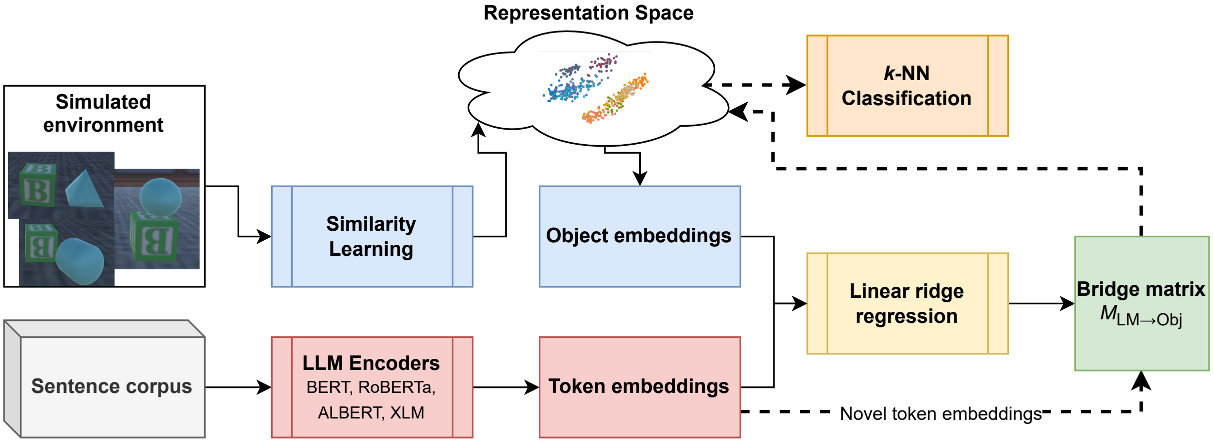


Figure 1: Overview of grounding architecture. In this figure, M denotes the computed affine transformation matrix between language model (LM) and object classifier (Obj) space. Similarity learning in this figure is performed only over a subset of the available classes (see Sec. 3.2). The solid lines depict the flow of information used to "train" or compute the affine transformation "bridge" matrix, and the dashed lines depict the flow of information of novel "test" samples, including transformation by the precomputed bridge matrix.

helpful for subsequent grounding of verbal tokens, abstract properties, and attributive terms, but that grounding verbal or attributive token representations is less helpful for subsequent grounding of object concepts.

2 Related Work

Multiple works in cognitive science have identified contrastive mechanisms, and the ability to analogize by applying previous experiences to novel scenarios, as a cornerstone of problem-solving (Gentner, 1983; Forbus et al., 1995; McLure et al., 2010; Hofstadter and Sander, 2013; Smith and Gentner, 2014; Lovett and Forbus, 2017).

In visual analogy, Hill et al. (2019) created analogies by contrasting relational structure. For solving Raven's Progressive Matrices (RPMs) (Raven, 1936), Małkiński and Mańdziuk (2022) applied a generalization of the Noise Contrastive Estimation (NCE) algorithm (Gutmann and Hyvärinen, 2010). Wu et al. (2018) performed feature learning using visual similarity via unsupervised learning at the instance-level with NCE. Oh Song et al. (2016) used deep feature embedding based on lifted structure loss, and evaluated their method via clustering and retrieval tasks on images from unseen classes. Bell and Bala (2015) trained a Siamese CNN with contrastive loss (Hadsell et al., 2006) to learn an embedding space of interior design images and applied the embeddings to image search applications like finding visually similar products across categories.

Since evaluating AI agents in physical environment can be expensive, many works have used both embodied and non-embodied simulations to explore language learning. Hermann et al. (2017) combined reinforcement and unsupervised learning to teach agents to correlate linguistic symbols with physical percepts and action sequences. However, this still-computationally-expensive method required millions of training episodes. The SNARE benchmark (Thomason et al., 2022) was evaluated on grounding to objects but not in context or under interaction. Tucker et al. (2021) demonstrated an emergent clustering of semantic tokens from a (non-embodied) continuous representation space and Tucker et al. (2022) extended that method with an application of an information bottleneck. Our work integrates concepts from the above areas: embodied simulation environments, language grounding using both situated and linguistic context, and emergent semantic categorization.

Merullo et al. (2022) examined 2D and 3D visual and interactive data for learning object affordances and found that 3D and interactive data performed better. Ebert et al. (2022) extracted verbal semantics from object trajectories in 3D space, but focused only on verbs whereas we examine nouns, verbs, and attributes. We show that objects and properties can also be encoded by object trajectories or behavior in 3D space, using a stacking task that exposes richer correlations between object properties and behaviors. Like us, Patel and Pavlick (2022) investigated word grounding but they evaluated on within-domain concepts (e.g., learning "left" to help ground "right" where we investigate how, say, learning "sphere" can help ground "round") in a grid world (our world representation is continuous), and where their transformation passed input through a whole LLM, our transformation is a simple affine map between embedding spaces. Lazaridou et al. (2015) mapped vision embeddings to language via ridge regression, but their Multimodal Skip-Gram used static word vectors, not contextualized vectors from transformers.

Pezzelle et al. (2021) evaluated the representations of transformer models and found that multimodal representations better align with human judgments in the domain of concrete nouns, but not abstract terms. Our work arrives at a related conclusion using cross-model transfer.

3 Methodology

Our methodology comprises two primary components: similarity learning to create a representation space of objects by making comparisons between geometric properties, and linear projection to ground language representations to this space.

3.1 Data

We use the dataset from Ghaffari and Krishnaswamy (2022), in which an agent in a simulated environment built on the VoxWorld platform (Krishnaswamy et al., 2022), stacks 9 different types of *theme* objects[2] on top of a cube. Each object's behavior when stacked is different, based on its geometric structure and therefore *affordances* (Gibson, 1977). For instance, a cube, if placed correctly on another cube, will remain stacked, while a sphere placed in the same position will roll off and keep moving. An egg will likely do the same, but the direction of motion may be subtly different based on the symmetry of the object. The dataset contains 10,000 total samples, each with 43 numerical values describing the behavior of the objects in the course of this stacking task: theme object type; object orientation before the agent acts upon it; numerical action describing the placement of the theme relative to the destination object; resulting spatial relations between the two objects; object orientation after the action; and position of the theme relative to the destination object before action, immediately after action, and after the world physics are applied to the scene. See Ghaffari and Krishnaswamy (2022) for further details. This information about object behaviors and trajectories in space, unlike still images, *situates* the objects in an embodied environment and encodes richer information than visuals alone do. The dataset does also contain images but these are not used here.

Two of the object types in the data, *cylinder* and *cone*, have both flat sides and round edges, and as this distinction strongly affects the behavior of these objects when stacked (i.e., given proper placement, a cylinder or cone will stack on top of a cube but only if also placed in the correct orientation), the dataset preserves these distinctions nicely, and we split the cone and cylinder samples into "flat-side-down" and "round-edge-down" for similarity learning of properties (Sec. 3.2).

3.2 Similarity Learning of Object Properties

Since comparing pairs of examples plays a role in analogy-making, we apply deep pair-based learning to compare structural object properties. The main goal in deep pair-based learning techniques is to learn an embedding space where embeddings of similar samples are closer together and dissimilar samples are pushed apart, after the projection of input space to the embedding metric space. In our case, the trained model should be able to infer contrasts and comparisons between different structural properties of objects (in this case *flatness* and *roundness*), and apply it to novel objects based on commonalities in behavior and relational structure.

In training, we consider only samples of *cube*, *rectangular prism*, *pyramid*, and *small cube* that stacked successfully, and samples of *capsule*, *sphere*, and *egg* that did not. For testing data, we take a test split of the same object classes, and also samples of *cone* and *cylinder*. These samples behave differently according to, among other things, their orientation when placed. We split cone and cylinder instances into "flat-side-down" (stacked successfully) and "round-edge-down" (did not stack successfully). Therefore we train on 7 classes and evaluate on 11 classes, including 4 never seen in training.

To train, we take 500 samples of each training class, zero-center the data and make it unit variance. Our model architecture consists of 4 1D convolutional layers (32, 32, 64, and 64 units, respectively, with filter size 3, stride length 1). The network applies ReLU activation to the output feature maps, with a max-pooling layer after the first two convolutional layers. The final convolutional layer output

[2] *cube*, *sphere*, *cylinder*, *capsule*, *small cube*, *egg*, *rectangular prism*, *pyramid*, and *cone*.

is flattened, followed by an L_2 normalized dense layer.

We use multi-similarity loss (Wang et al., 2019) which uses two iterative steps: pair-mining and weighting. This approach considers both self-similarity and relative similarity to collect more informative pairs, and takes a weighted combination of selected positive and negative pairs. Like other pairwise-based losses, this loss function maximizes the distance between dissimilar examples and minimizes it between similar examples.

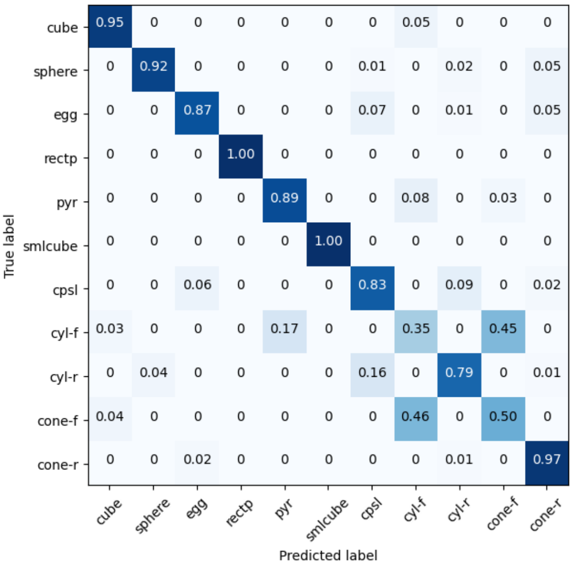
Equation 1 provides the formulation of the multi-similarity loss function:

$$\frac{1}{m}\sum_{i=1}^{m}\{\frac{1}{\alpha}\log[1+\sum_{k\in P_i}1+e^{-\alpha(S_{ik}-\lambda)}] \\ +\frac{1}{\beta}\log[1+\sum_{k\in N_i}1+e^{\beta(S_{ik}-\lambda)}]\}, \quad (1)$$

where N_i represents negative pairs (samples from different classes) in the batch while P_i denotes positive pairs (samples from the same class). S_{ik} represents element (i,k) of the similarity matrix, indicating the similarity of two samples $\{x_i, x_k\}$, $S_{ik} := f(x_i;\theta)\cdot f(x_k;\theta)$ where f is the neural network with parameters θ. The cosine embedding size is 64.

We use Adam optimization (Kingma and Ba, 2015) with a learning rate of 5×10^{-6}, with batch size 70, and train for 20 epochs. Training was performed on a Mac M1 Max with Metal acceleration. In every mini-batch, 10 inputs ($m = 10$) from each of the 7 training classes are randomly sampled. In Equation 1, $\alpha = 2$ (weight for positive pairs), $\beta = 40$ (weight for negative pairs), $\lambda = 0.5$ (used to weight the distance). Margin $\epsilon = 0.1$ is used to remove easy positive and negative pairs such that negative pairs are sampled if they are greater than $(\min_{y_i=y_k}(S_{ik})-\epsilon)$ where $\min_{y_i=y_k} S_{ik}$ represents the positive pair with the lowest similarity, and positive pairs are sampled if they are less than $(\max_{y_i\neq y_k}(S_{ik})+\epsilon)$ where $\max_{y_i\neq y_k} S_{ik}$ represents the negative pair with the highest similarity. y denotes the one-hot label vectors.

Since during training only *7* types of flat-sided and round objects are used, the model learns to output an embedding that represents pure round and flat objects samples in the cosine space. The extracted embeddings are indexed. Given that the index of the embedding space represents only purely round or flat objects, we consider 100 test samples each from all *11* classes (seen and unseen) and find the closest matches to the test samples using a nearest neighbour search ($K = 10$).



	cube	sphere	egg	rectp	pyr	smlcube	cpsl	cyl-f	cyl-r	cone-f	cone-r
cube	0.95	0	0	0	0	0	0	0.05	0	0	0
sphere	0	0.92	0	0	0	0	0.01	0	0.02	0	0.05
egg	0	0	0.87	0	0	0	0.07	0	0.01	0	0.05
rectp	0	0	0	1.00	0	0	0	0	0	0	0
pyr	0	0	0	0	0.89	0	0	0.08	0	0.03	0
smlcube	0	0	0	0	0	1.00	0	0	0	0	0
cpsl	0	0	0.06	0	0	0	0.83	0	0.09	0	0.02
cyl-f	0.03	0	0	0	0.17	0	0	0.35	0	0.45	0
cyl-r	0	0.04	0	0	0	0	0.16	0	0.79	0	0.01
cone-f	0.04	0	0	0	0	0	0	0.46	0	0.50	0
cone-r	0	0	0.02	0	0	0	0	0	0.01	0	0.97

Figure 2: Confusion matrix on the test split of 11 objects. Only 7 pure flat and round objects are used during training. `cyl-f` = cylinder, flat side down; `cyl-r` = cylinder, round edge down; likewise for `cone-f/r`. The values shown in the matrix are normalized between 0 and 1.

Similarity Learning Results Fig. 2 shows the confusion matrix for nearest neighbor search on the test split of objects, using 100 test samples per class. Interestingly, even though the model was not trained on any *cone* and *cylinder* instances, it is still able to not only match them to the correct object type, but also to the correct orientation. Where confusions arise, it is between different flat-sided objects and different round objects, but never across these categories. In other words, this model can capture and distinguish the main distinguishing concepts—roundness and flatness—in the different object classes, and draw comparisons between them across classes. It also applies what is already learned to novel objects to find similar examples with respect to these concepts. Overall classification accuracy is 82%.

3.3 Grounding Conceptual Vocabulary

First, we extracted the embeddings of 800 object test samples from the learned object space. These were 64D embeddings that defined the object representation space, and objects clustered into two broad regions defining the "flat-sided" and "round-sided" objects (see Fig. 3).

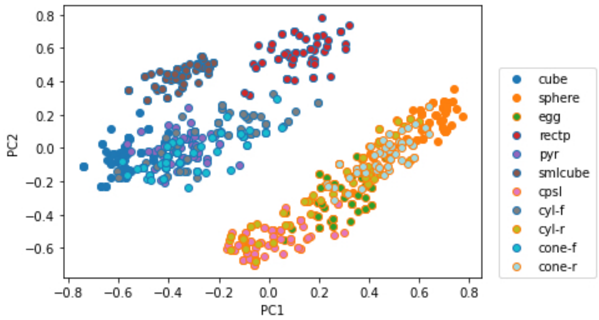


Figure 3: PCA of test object embeddings. Points outlined in blue represent "flat" object embeddings. Points outlined in orange represent "round" object embeddings.

Individual embedding vectors of different instances of the same object type form a region defining the object representation where some subset of these vectors form the region's spanning set; Ethayarajh (2019) observed similar phenomena in the representations of contextualized token vectors from LLMs, suggesting there exists a structure-preserving mapping between equivalent regions in different embedding spaces.

To assess this, we needed to generate appropriately contextualized vector representations of terms to ground to the object representation space. For this we turned to OpenAI's ChatGPT model to rapidly generate a sentence corpus. ChatGPT was given prompts to generate short, unique sentences that would explicitly mention the objects by name and describe their behavior in a stacking task (e.g., "*Write 40 short sentences about how cubes can be stacked*"). In the process, ChatGPT also generated mentions of properties of the objects (*flat/round*), associated behaviors (*stack/roll*), properties of the resulting structure (*stable/unstable*), and resulting state of the structure (*stand/fall*). We generated 40 sentences describing each object type, plus 20 sentences each for *block* and *ball*, synonyms for *cube* and *sphere*. In total, a 440-sentence corpus was generated.

We then took the most frequently-occurring domain-relevant terms (these were the object names and aforementioned related conceptual terms) and extracted the word-level embeddings for each occurrence. We extracted word embeddings using the BERT, RoBERTa, ALBERT (all 768D), and XLM (2,048D) base models. Embeddings were creating by summing over the encoder hidden states of the last four encoder layers. Where tokenization split the target word into multiple tokens, the individual contextualized token embeddings were averaged to create a single embedding.

To actually ground the word embeddings into the object space, we used a simple affine transformation. We took 5 contextualized embeddings of each target word, paired each with an embedding for the object whose name occurs in the sentence the target word came from, and use them to compute an affine transformation from LLM space to object embedding space, using a ridge regressor that minimizes the mean squared-error distance between the paired embeddings. The resulting transformation matrix serves as a "bridge" between the two representation spaces. This affine transformation technique has previously been used to compare image embeddings from different vision models (McNeely-White et al., 2022) and to map information from monolingual LLMs into multilingual LLMs (Nath et al., 2022). Here we apply this technique in a cross-modal setting.

We perform iterative experiments, starting by using only a subset of the different words and objects to compute the mapping, and incrementally add conceptual vocabulary to improve the quality of the calculated transformation. We evaluate the transformation by transforming word vectors for concepts not used in computing the transformation matrix and seeing if those embeddings cluster with the correct set of objects that bear those properties, have those affordances, etc. The order in which object concepts are introduced follows the order we used previously in Ghaffari and Krishnaswamy (2022), with the exception of moving *cylinder* and *cone* to the end, due to their exclusion from initial training of the similarity learning model, and pairing one flat-sided with one round object (e.g., *pyramid* + *capsule*) at each step.

As a final step, a "hint" is provided by adding 5 embedding pairs that explicitly include the concept to be grounded to the computation of the transformation. We evaluate this by transforming new instances of that concept into the object embedding space and seeing where they cluster. We quantify the clusters of different concepts when transformed into the object space using separation of cluster centers and a K-nearest neighbor (KNN) classifier with $K = 5$.

4 Results of Conceptual Grounding

For illustrative purposes, let us first examine the concepts of "flat" and "round" using word embeddings drawn from XLM, the best-performing model

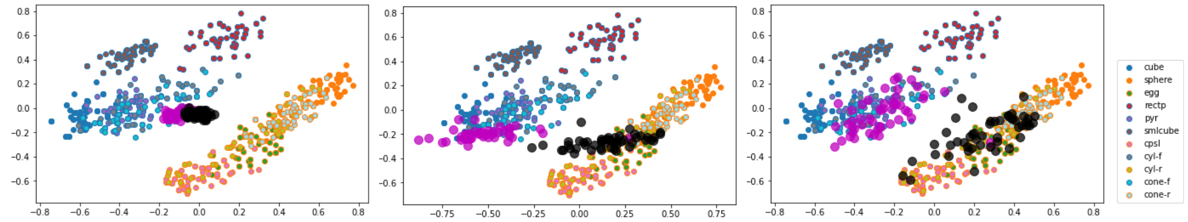


Figure 4: PCA of "flat" (pink) and "round" (black) test embeddings from XLM mapped into object embedding space. L: with mapping computed using only information about *cubes*, *spheres*, and *eggs*. C: using information about all objects. R: using all objects and a 5-sample "hint" about "flat" vs. "round."

in this domain when "hinting" is used. Further results from other models are given in the Appendix.

In Fig. 4, we see word embeddings for "flat" and "round" transformed into the object embedding space. At first (the left figure), when only information about cubes, spheres, and eggs are used to compute the mapping, there is only a slight separation between the two transformed embedding clusters and neither term clusters cleanly with either flat or round objects. When information about all objects is used to compute the transformation (center), the two word embedding clusters distinctly separate, with most "flat" embeddings clearly overlapping with the flat-sided objects, *mutatis mutandis* "round" embeddings and the round objects. Finally (right), the "hint" is provided, by explicitly pairing a small set of 5 "flat" or "round" word embeddings to object embeddings whose type appears in a generated sentence collocated with the target word (e.g., "*The cubes were flat on all sides, making it easy to stack them neatly.*"). With this hint we see that the "flat" and "round" word embeddings more completely overlap with the objects that have the respective attributes.

When little information about related object concepts is provided when computing the mapping from LLM space to object embedding space, the transformed clusters of contrasting terms share a high level of similarity in object space, but as more information about related object concepts is introduced into the transformation, the separation of the transformed novel concept clusters start to cleanly separate and become distinguished from each other. Fig. 5 shows the mean similarity of the transformed clusters of attributive, verbal, and object synonym terms as different object terms are mapped into the object space, using embeddings drawn from the four different LLMs. Fig. 6 shows the same change in the similarity between cluster centers, but this time evaluated over the transformed *object* terms when the transformation is computed using attributive and verbal terms. In both plots, dashed lines show where an explicit "hint" is given about specific concepts. We see that just by using a few samples of each concept and projecting them into object space using an affine transformation, grounding object terms is helpful in distinguishing the meaning of terms denoting related properties, attributes, and verbs, but grounding the more abstract concept vocabulary first does not usually cause the transformed clusters of object terms to separate before explicit hinting is provided, reflecting the psycholinguistic hypothesis of Gentner (1983).

Table 1 shows the results of the KNN classifier over the transformed attributive and verbal word embeddings, both when the transformation was computed using only object information (top section) and with "hints" about the attributive concepts (bottom). Table 2 shows KNN classifier results over transformed object embeddings without hints about the objects, and with. We report macroaveraged F1 scores, so that successful performance on high support classes does not obscure poor performance on low support classes. Numbers in parentheses show how much "hinting" helped improve performance of the particular model on the concept in question. *Block* and *ball* are included in both the "object" test set and the "predicate" test set (even though they are not predicative terms in this sense), because these terms were not used in computing the affine mapping in either case. They are included as synonyms for *cube* and *sphere*. Further discussion is provided in Sec. 5.

5 Discussion

Separation of conceptual clusters In Fig. 5, we can see that for object concept vectors from certain models, as information about certain other concepts is included in the transformation from LLM space to object space, the centers of the conceptual clusters in question start to organically separate. This is particularly true for ALBERT object word vectors and to some extent XLM and BERT vectors. In

	flat/round	**stack/roll**	**stable/unstable**	**stand/fall**	**block/ball**
Models	$N = 103$	$N = 56$	$N = 22$	$N = 10$	$N = 30$
BERT	0.89	0.16	**0.58**	0.60	0.33
RoBERTa	0.34	0.16	0.29	0.37	0.67
ALBERT	**0.92**	**0.65**	**0.58**	**0.89**	0.60
XLM	0.73	0.53	0.37	0.29	**0.79**
BERT+hint	0.96 (+0.07)	0.78 (+0.62)	0.91 (+0.63)	**1.00** (+0.40)	0.93 (+0.60)
RoBERTa+hint	0.90 (+0.56)	0.89 (+0.73)	**1.00** (+0.71)	**1.00** (+0.63)	0.90 (+0.23)
ALBERT+hint	0.89 (-0.03)	0.85 (+0.20)	0.86 (+0.28)	**1.00** (+0.11)	0.66 (+0.06)
XLM+hint	**0.98** (+0.25)	**1.00** (+0.47)	0.73 (+0.36)	**1.00** (+0.71)	**0.97** (+0.18)

Table 1: Macroaveraged KNN F1 over transformed attribute/verb/synonym word embedding test sets (mapping computed using object embeddings). Numbers in parentheses show performance increase with "hinting."

Models	**cube/sphere**	**pyr/cpsl**	**cyl-f/r**	**cone-f/r**	**block/ball**
BERT	0.77	0.46	0.34	0.40	**0.83**
RoBERTa	0.81	0.44	0.40	0.49	0.55
ALBERT	**0.88**	**0.88**	**0.81**	**0.78**	0.46
XLM	0.40	0.46	0.49	0.36	0.55
BERT+hint	0.97 (+0.20)	**1.00** (+0.54)	0.78 (+0.44)	0.84 (+0.44)	0.93 (+0.10)
RoBERTa+hint	0.81 (±0.00)	0.94 (+0.50)	0.78 (+0.38)	0.87 (+0.38)	0.90 (+0.35)
ALBERT+hint	0.88 (±0.00)	0.94 (+0.06)	**0.87** (+0.06)	0.88 (+0.10)	0.66 (+0.20)
XLM+hint	**1.00** (+0.60)	0.97 (+0.51)	0.81 (+0.32)	**0.91** (+0.55)	**0.97** (+0.42)

Table 2: Macroaveraged KNN F1 over transformed object word embedding test sets (mapping computed using attribute/verb embeddings). Numbers in parentheses show performance increase with "hinting." $N = 30$ for all.

other words, if the model already "knows" about the dual aspects of cones and cylinders, it becomes easier to distinguish an abstract concept of *flatness* from *roundness*. Clusters of transformed RoBERTa object word vectors tend not to separate very clearly until explicit hints about them are provided.

Flat/round is the easiest of the attributive or verbal concepts to distinguish, through affine transformations that include information about flat and round objects. *Stable/unstable* is a particularly hard distinction for most model representations, in part because of the low support for these terms in the training corpus but also because in the scenario captured in the simulation data and described in training sentences, the terms refer to properties not of the objects themselves, but of the objects in the context of the stacking task (i.e., spheres are not inherently "unstable" but are if someone attempts to stack them). This suggests data gathered from either stacking more objects, or from tasks involving more complex balancing acts would be useful to learn a robust interpretation of such terms.

Inverse trends are observable in Fig. 6, where we see that when the transformation includes only information about attributes and verbs, transformed BERT and XLM object word vectors for contrasting objects do not meaningfully separate until explicit hints are provided (and even then sometimes they don't separate much). Some of the RoBERTa object word clusters do appear to appreciably separate as more attribute and verb information is added to the transformation, and ALBERT object word clusters, actually at first grow *closer* as related conceptual information is added to the transformation, until suddenly separating at the provision of an explicit hint. This suggests that ALBERT, perhaps due to its smaller training size and architecture, learns vocabulary representations that are more "entangled" or that representations of flat-sided or round object words carry with them a bias toward object-related interpretations of "flat," "round," and associated terms. Meanwhile XLM and other representations of abstract vocabulary are perhaps less correlated with concrete nouns, making them less easy to ground but also in principle more compositional with less bias toward certain interpretations.

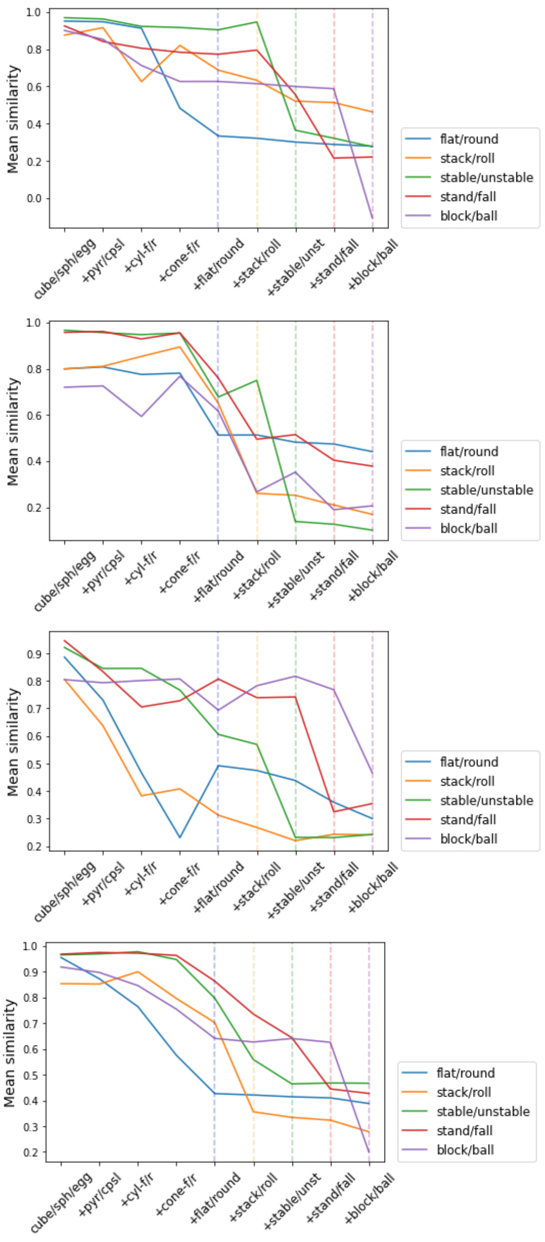


Figure 5: Separation of cluster centers for transformed (in order) BERT, RoBERTa, ALBERT, and XLM embeddings for verb and property concepts, as more information about other concepts is progressively added to compute the transformation. Dashed lines show where a "hint" is given about the concept to be grounded (denoted by the similarly-colored solid line).

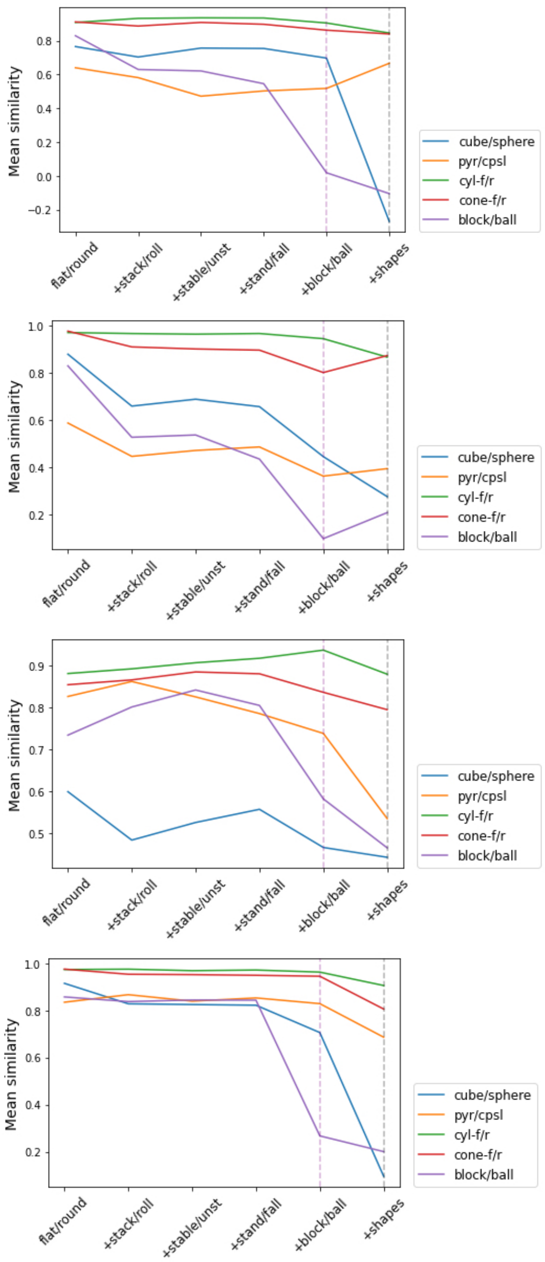


Figure 6: Separation of cluster centers for transformed embeddings for object concepts, as more information about other concepts is progressively added to compute the transformation. Format is identical to Fig. 5. `+shapes` denotes adding information about all objects.

Classification of conceptual terms With hinting, XLM vectors perform best in the term classification task. XLM is the largest of the four models and has the largest embedding size (2,048 where all other models use an embedding size of 768). Hinting typically provides the biggest boost in performance to XLM vectors, both for grounding concrete object and abstract terms. This suggests that the object concepts and the attributive/verbal concepts form distinct and possibly distant regions in the original XLM embedding space, and that an affine transformation into the object space does not always put pairs of contrastive attributes or verbs closer to distinct objects that display those respective properties. Providing hints helps all models achieve high performance by matching objects and related concepts, but the performance boost is particularly high for XLM vectors, which often perform very badly in KNN classification of some concepts (e.g., *stable/unstable*, *stand/fall*) until hints are provided. Hinting is still less helpful for transforming XLM object vectors when only previous information about attributes or verbs is provided.

Interestingly, hinting is least helpful when grounding word vectors from ALBERT, the smallest of the four models. On eight out of ten concept pairs explored, ALBERT vectors perform the best by far in the KNN classifier before any hints are provided, but providing subsequent hints makes only a small difference to classification F1, and sometimes none at all, while boosting the performance of other representation ahead of ALBERT vectors. This suggests that the object and related concept representations already share some level of correlation and possibly overlap in ALBERT embedding space. In turn, these results suggest that larger models like XLM may be better able to represent multiple word senses and figurative, non-physically-grounded usages of terms like these. However, grounding these concepts to a physical environment without some explicit "nudges" may be more challenging for larger pretrained models than smaller ones, in which the abstract concepts may already be biased toward correlations with the associated concrete object concepts. Further discussion is provided in the Appendix.

6 Conclusion and Future Work

In this paper, we have presented evidence that similarity learning over rich object behavior and trajectory data from an embodied simulation environment can create a representation space that not only successfully classifies concrete objects but can make analogical comparisons between them based on abstract properties that inhere across multiple object types. We used the resulting representation to conduct investigations into the properties of token embeddings from different LLMs by mapping them into the object space using a linear ridge regression technique. We found that computing a mapping using representations of objects/object terms correlated with increased ability to distinguish and assign related conceptual vocabulary to the right categories, but that representations from different LLMs behaved quite differently. We also observed that computing mappings using information about abstract properties was less useful for distinguishing and classifying object terms. This reflects earlier arguments from psycholinguistics and analogical reasoning, e.g., Gentner (2006)'s hypothesis that names for concrete objects should be learnable by humans very early but that associated verbs and attributes are harder.

Our approach uses numerical data that situates and embodies an agent's positioning in the environment relative to the objects it interacts with. This method allows us to build a model over rich information without visual artifacts like occlusion or perspective distortion, Prior research, e.g., Krishnaswamy and Pustejovsky (2022); Pustejovsky and Krishnaswamy (2022) has demonstrated that embodiment is also influenced by other factors like events and habitats, and that purely linguistic representations of objects, attributes, and activities may not capture these types of information. In fact, the corpus generated using ChatGPT, an unembodied language model trained solely over text, is likely not entirely representative of these aspects beyond cooccurrences between object terms, habitats, and affordances in the training data. What our embodied approach brings is a way to correlate representations extracted from unembodied models to representations learned from embodied data, and provides evidence that the ability to ground real-world entities, properties, or actions to lexical items could enable LLMs to simulate the human ability to link utterances to specific communicative intents. However, further investigation is necessary.

Since the primary objective of this research is to provide a method that achieves human-like "understanding" of communicative intents, we should note that we do not argue that human learners use the same mathematical transformations we use herein, but just that we can use them to make AI systems behave similarly.

Directions for future work include 1) investigating the effects of intra-class order when grounding tokens, e.g., introducing object concepts to the affine mapping in a different order; 2) using similarity learning over images, or images combined with the embodied data, to create the representation space; 3) using data gathered in other embodied tasks to investigate other concepts like concavity or directedness, that are not captured in this stacking task; 4) evaluating token representations directly from a decoder like a GPT-style model; and 5) directly operationalizing analogical comparisons in a real-time embodied simulation, e.g., by making an agent solve problems using analogical reasoning in a live environment.

Acknowledgments

We would like to thank our reviewers for their helpful suggestions on improving the final copy of this paper. This research was supported in part by a grant from the U.S. Army Research Office (ARO) to Colorado State University, under award #W911NF-23-1-0031. The positions expressed herein do not reflect the official position of the U.S. Department of Defense or the United States government. Any errors or omissions are the responsibility of the authors.

References

Renee Baillargeon. 1987. Object permanence in $3\frac{1}{2}$-and $4\frac{1}{2}$-month-old infants. *Developmental psychology*, 23(5):655.

Sean Bell and Kavita Bala. 2015. Learning visual similarity for product design with convolutional neural networks. *ACM transactions on graphics (TOG)*, 34(4):1–10.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Dylan Ebert, Chen Sun, and Ellie Pavlick. 2022. Do trajectories encode verb meaning? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2860–2871.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Kenneth D. Forbus, Dedre Gentner, and Keith Law. 1995. Mac/fac: A model of similarity-based retrieval. *Cognitive science*, 19(2):141–205.

Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170.

Dedre Gentner. 2006. Why verbs are hard to learn. *Action meets word: How children learn verbs*, pages 544–564.

Sadaf Ghaffari and Nikhil Krishnaswamy. 2022. Detecting and accommodating novel types and concepts in an embodied simulation environment. In *Proceedings of the Tenth Annual Conference on Advances in Cognitive Systems*.

James J. Gibson. 1977. The theory of affordances. *Hilldale, USA*, 1(2):67–82.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. 2017. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*.

Felix Hill, Adam Santoro, David Barrett, Ari Morcos, and Timothy Lillicrap. 2019. Learning to make analogies by contrasting abstract relational structure. In *International Conference on Learning Representations*.

Douglas R. Hofstadter and Emmanuel Sander. 2013. *Surfaces and essences: Analogy as the fuel and fire of thinking*. Basic books.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nikhil Krishnaswamy, William Pickard, Brittany Cates, Nathaniel Blanchard, and James Pustejovsky. 2022. The VoxWorld platform for multimodal embodied agents. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1529–1541, Marseille, France. European Language Resources Association.

Nikhil Krishnaswamy and James Pustejovsky. 2022. Affordance embeddings for situated language understanding. *Frontiers in Artificial Intelligence*, 5.

Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. 2015. Combining language and vision with a multimodal skip-gram model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 153–163, Denver, Colorado. Association for Computational Linguistics.

Andrew Lovett and Kenneth D. Forbus. 2017. Modeling visual problem solving as analogical reasoning. *Psychological review*, 124(1):60.

Mikołaj Małkiński and Jacek Mańdziuk. 2022. Multi-label contrastive learning for abstract visual reasoning. *IEEE Transactions on Neural Networks and Learning Systems*.

Matthew D. McLure, Scott E. Friedman, and Kenneth D. Forbus. 2010. Learning concepts from sketches via analogical generalization and near-misses. In *Proceedings of the annual meeting of the cognitive science society*, volume 32.

David McNeely-White, Ben Sattelberg, Nathaniel Blanchard, and Ross Beveridge. 2022. Canonical face embeddings. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(2):197–209.

Jack Merullo, Dylan Ebert, Carsten Eickhoff, and Ellie Pavlick. 2022. Pretraining on interactions for learning grounded affordance representations. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 258–277.

Abhijnan Nath, Rahul Ghosh, and Nikhil Krishnaswamy. 2022. Phonetic, semantic, and articulatory features in assamese-bengali cognate detection. In *Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 41–53.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. 2016. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012.

Roma Patel and Ellie Pavlick. 2022. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*.

Sandro Pezzelle, Ece Takmaz, and Raquel Fernández. 2021. Word representation learning in multimodal pre-trained transformers: An intrinsic evaluation. *Transactions of the Association for Computational Linguistics*, 9:1563–1579.

James Pustejovsky and Nikhil Krishnaswamy. 2022. Multimodal semantics for affordances and actions. In *Human-Computer Interaction. Theoretical Approaches and Design Methods: Thematic Area, HCI 2022, Held as Part of the 24th HCI International Conference, HCII 2022, Virtual Event, June 26–July 1, 2022, Proceedings, Part I*, pages 137–160. Springer.

James C. Raven. 1936. Mental tests used in genetic studies: The performance of related individuals on tests mainly educative and mainly reproductive. *Unpublished master's thesis, University of London*.

Linsey Smith and Dedre Gentner. 2014. The role of difference-detection in learning contrastive categories. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36.

Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218.

Elizabeth S. Spelke. 1985. Perception of unity, persistence, and identity: Thoughts on infants' conceptions of objects.

Elizabeth S. Spelke. 1990. Principles of object perception. *Cognitive science*, 14(1):29–56.

Elizabeth S. Spelke, Claes von Hofsten, and Roberta Kestenbaum. 1989. Object perception in infancy: Interaction of spatial and kinetic information for object boundaries. *Developmental Psychology*, 25(2):185.

Leonard Talmy. 1975. Semantics and syntax of motion. In *Syntax and Semantics volume 4*, pages 181–238. Brill.

Jesse Thomason, Mohit Shridhar, Yonatan Bisk, Chris Paxton, and Luke Zettlemoyer. 2022. Language grounding with 3d objects. In *Conference on Robot Learning*, pages 1691–1701. PMLR.

Mycal Tucker, Huao Li, Siddharth Agrawal, Dana Hughes, Katia Sycara, Michael Lewis, and Julie A. Shah. 2021. Emergent discrete communication in semantic spaces. *Advances in Neural Information Processing Systems*, 34:10574–10586.

Mycal Tucker, Julie Shah, Roger Levy, and Noga Zaslavsky. 2022. Towards human-agent communication via the information bottleneck principle. *arXiv preprint arXiv:2207.00088*.

Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5022–5030.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742.

Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. 2016. Situation recognition: Visual semantic role labeling for image understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5534–5542.

Tiangang Zhu, Yue Wang, Haoran Li, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Multimodal joint attribute prediction and value extraction for E-commerce product. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2129–2139, Online. Association for Computational Linguistics.

Wanrong Zhu, Xin Wang, Tsu-Jui Fu, An Yan, Pradyumna Narayana, Kazoo Sone, Sugato Basu, and William Yang Wang. 2021. Multimodal text style transfer for outdoor vision-and-language navigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1207–1221, Online. Association for Computational Linguistics.

A Appendix: Additional Results

The following figures are provided for comparison with Fig. 4 and Table 1.

Fig. 7 shows the projection of “flat” and “round” token embeddings from **BERT** into the learned object representation space when the mapping is computed using paired embeddings of objects and object terms, but no explicit hint is provided about the meaning of “flat” and “round.” The two clusters clearly separate from each other but do not map clearly onto flat and round object representations at this stage. Fig. 8 shows the same projection after a 5-sample hint about “flat” and “round” is added to the mapping.

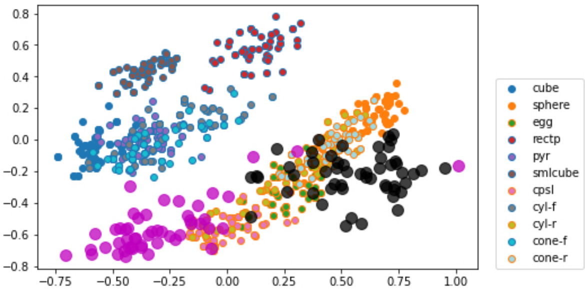


Figure 7: PCA of “flat” (pink) and “round” (black) test embeddings from BERT mapped into object representation space. Mapping is computed using information about all objects but without flat/round hinting.

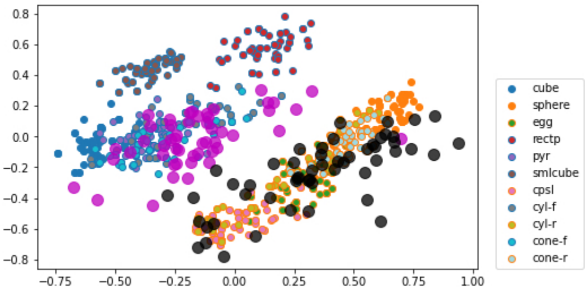


Figure 8: PCA of “flat” (pink) and “round” (black) test embeddings from BERT mapped into object representation space. Mapping is computed using information about all objects and flat/round hinting.

Fig. 9 shows the projection of “flat” and “round” token embeddings from **RoBERTa** into the learned object representation space when the mapping is computed using paired embeddings of objects and object terms, but no explicit hint is provided about the meaning of “flat” and “round.” Again, the two clusters clearly separate from each other at this stage, but the “flat” embeddings are closer to the round objects embeddings in the 64D space while the “round” embeddings are distinct from each object cluster. Fig. 10 shows the same projection after a 5-sample hint about “flat” and “round” is added to the mapping.

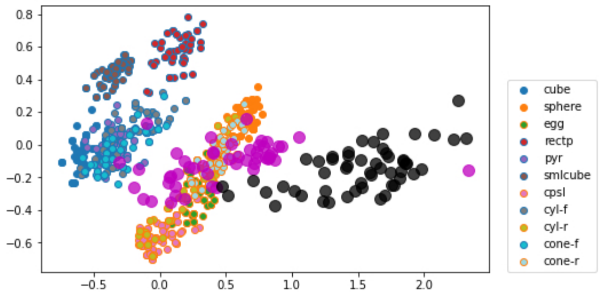


Figure 9: PCA of “flat” (pink) and “round” (black) test embeddings from RoBERTa mapped into object representation space. Mapping is computed using information about all objects but without flat/round hinting.

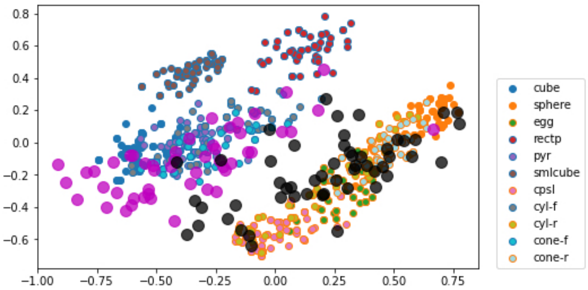


Figure 10: PCA of “flat” (pink) and “round” (black) test embeddings from RoBERTa mapped into object representation space. Mapping is computed using information about all objects with flat/round hinting.

Figs. 11 and 12 show the equivalent using the **ALBERT** “flat”/“round” token embeddings. Here, without hinting, the transformed “flat” embeddings mostly cluster with flat-sided objects and the transformed “round” embeddings mostly cluster with round objects, suggesting that in ALBERT, the representations of “flat”, “round”, and other associated object-related concepts are relatively entangled with the object terms themselves. Hinting solidifies this correlation somewhat but the effect is relatively small, as discussed in Sec. 5.

Fig. 13 shows contextualized token embeddings for *all* vocabulary items from (top to bottom) BERT, RoBERTa, ALBERT, and XLM mapped into the object representation space when the mapping is computed using information about all concepts, including hinting. For all points, the outer color denotes the token it represents and the inner color (blue or orange) indicates whether the transformed embedding clusters with flat-sided or round object representations. Therefore, a black point with an orange center indicates a “round” token embedding

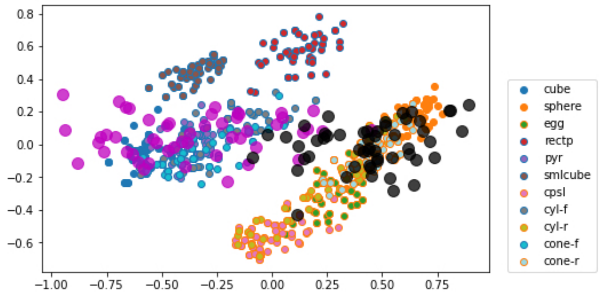


Figure 11: PCA of "flat" (pink) and "round" (black) test embeddings from ALBERT mapped into object representation space. Mapping is computed using information about all objects but without flat/round hinting.

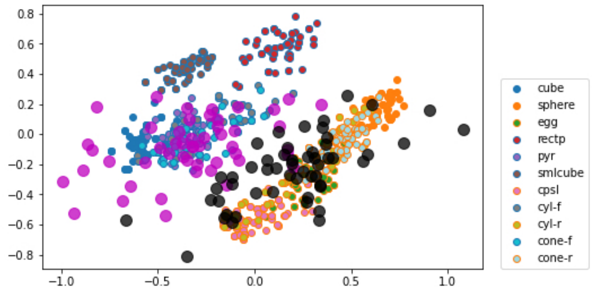


Figure 12: PCA of "flat" (pink) and "round" (black) test embeddings from ALBERT mapped into object representation space. Mapping is computed using information about all objects with flat/round hinting.

that clusters with round objects (correctly), but a black point with a *blue* center indicates one that incorrectly clusters with flat-sided objects. We see that when using the full set of concepts in computing the mapping between spaces, the larger models show the strongest correlations between correctly-mapped token embeddings and the expected set of object representations. Mapped XLM vectors show the strongest separation between the flat-related concepts and round-related concepts, while mapped ALBERT vectors display a fairly significant overlap between those correlated with flat objects and those correlated with round object (this is evident in the center of the plot "between" the two main flat and round clusters). Mapped RoBERTa and to a lesser extent BERT embeddings show a similar overall separation to mapped XLM embeddings, but with a wider dispersion in the distribution of mapped embeddings, where some (particularly in the case of RoBERTa embeddings) have a very high Euclidean distance from the two core object representation clusters to which they are compared.

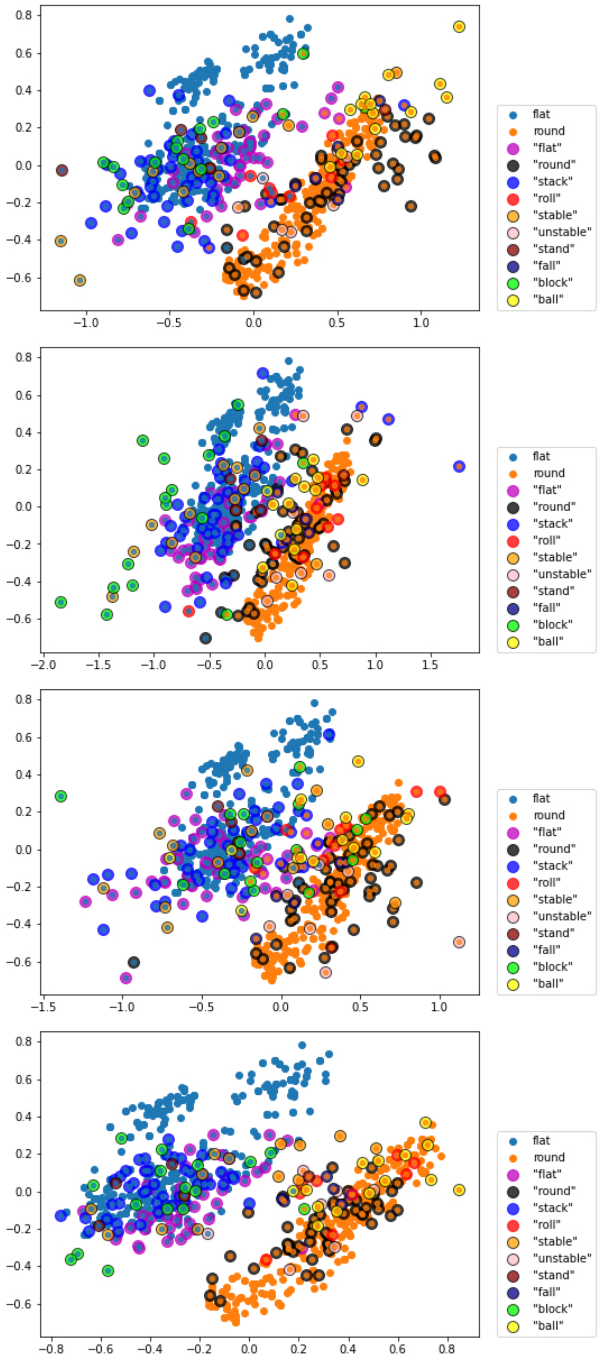


Figure 13: PCA of (top to bottom) BERT, RoBERTa, ALBERT, and XLM test word embeddings for all concepts mapped into object representation space, including hinting in the mapping. Innermost colored point indicates whether that transformed embedding clusters with flat-sided objects or round objects.