Evaluating Attribute-Object Compositionality in Text-Image Multimodal Embeddings

Prasetya Ajie Utama
Brown University
prasetya_utama@brown.edu

Abstract

The infinite possible complex scenes from visual world are composed by finite number of simple concepts i.e. images that depicts human activities could decompose into composition of subject-verb-object triplets; and images of a single salient object decompose into attribute-object pairs. Most of data-driven computer vision approaches do not explicitly model compositionality and instead rely on large number of examples of varying complex visual concepts. On the other hand, visual recognition tasks starts leveraging human language to annotate complex concept within a scene, and consequently learning an image-text multimodal representation using so-called visual-semantic models become increasingly important. However, despite their success in tasks such as cross-modalities retrieval or image captioning, the compositionality aspect of these models has yet to be evaluated explicitly. Our goal in this work is measure the extent of a state-of-the-art visual semantic model called VSE++ [Faghri et al., 2017] in generalizing to unseen attribute-object composition. We compare its performance with models that are trained explicitly to compose multiple classifiers proposed by [Misra et al., 2017]. Despite its advantage of having only a single model for arbitrary number of primitive visual concepts (e.g. subjects, verb, attribute, or object), we show that the visual-semantic model can perform competitively.

1. Introduction

1.1. Visual-Semantic Embedding

The holy grail of high level computer vision is to have a model with holistic understanding of visual scene. Namely, the model should be able to recognize and detect objects along with the visual attributes, and also describe their interaction and spatial relationship. This ability would allow various downstream perceptual tasks which require semantic interpretation such as image retrieval, caption generation, visual question-answering and human-robot interaction. Recent advances towards this direction are largely made possible by the grounding of visual perception into its semantic. In particular, given large scale examples of images paired with their textual annotations e.g. captions; a visual-semantic model is trained to project input images and text into a shared representation space in which semantically related inputs are located nearby. The compact vector representation within this shared space is already useful for several tasks such as image classification and its zero-shot variant [Frome et al., 2013], and cross-modalities retrieval [Kiros et al., 2014, Wang et al., 2016, Faghri et al., 2017]. Additionally, several image captioning works [Kiros et al., 2014, Karpathy and Fei-Fei, 2015] show that significant caption quality improvement could be achieved by projecting images into this space to retrieve its intermediate representation and later decode them using additional recurrent network to output captions.

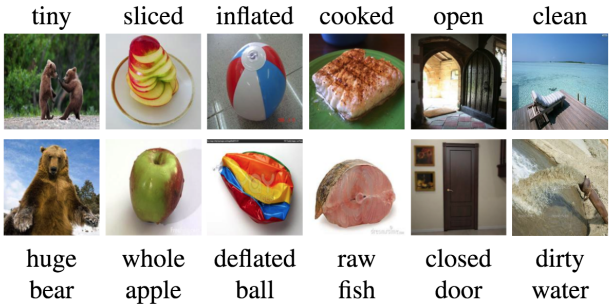

Figure 1: We are interested in the problem of attribute-object compositionality in the visual domain. As can be seen above, applying an attribute (adjective) to objects could transform or modify their appearance very differently.

Despite the consensus regarding the importance of multimodal image-text embedding and how to learn them, it is not clear whether a representation in this space respect the complexity of real world visual scene which possibly contains exponential number of composition of objects, attributes, and relationship. That is, having a visual and linguistic model that produce a distributed compositional se-

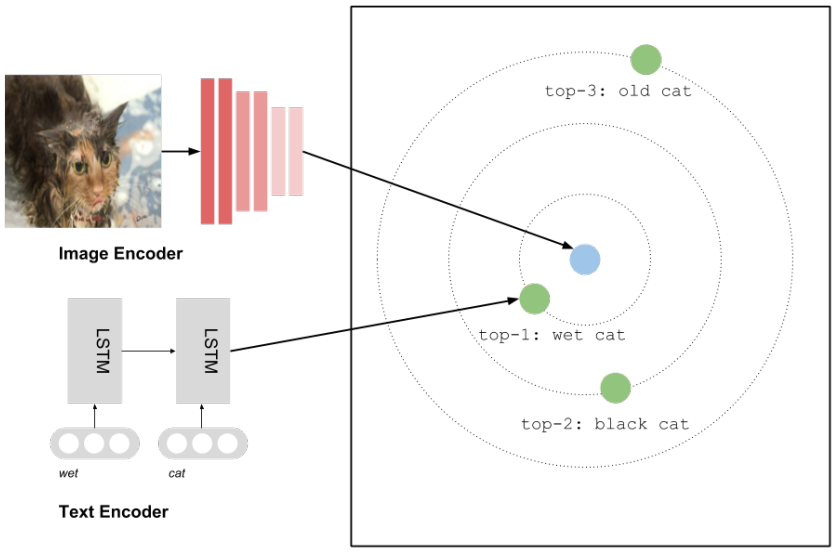

Figure 2: The visual-semantic model is used to perform attribute-object recognition by the following procedure: all possible combination of attributes and objects are encoded into the space in the initial phase and cached for the test time. A queried image is later projected to the same space and top-k nearest attribute-object phrases will be taken as its label.

mantic still remains an open question to be solved. Ideally, the linguistic model should be able to capture the relationships between words and how their composition results to a new meaning in the visual world. Equivalently, its visual model counterpart should have the abstraction capacity to encode visual concept (e.g. attributes) on its nodes activation and compose them into meaningful representation on each layer. Take for example an image of a white dress: the semantic composition of the word "white" and "dress" needs to be consistent with how the visual model in composing the abstraction of color white and object dress across its layer. Thus, the projection of this image to the shared space should be located very closely to the projection of the phrase text "white dress".

In a visual recognition task, the capacity of a model to compose could be indicated by how well it recognize unseen complex scene that comprises of seen visual concepts such as objects or attributes. It should suggest that the model is able to build up simpler concepts into a complex one. The capacity to generalize beyond seen composition is critical since the number of possible combination of simple visual concepts is intractable. Image examples of obscure combination are scarcely available and it is practically unfeasible to collect them as the training data.

Compositionality is especially challenging as it also has to respect contextuality and common sense, as shown by [Pavlick and Callison-Burch, 2016] in the context of linguistic. Namely, visual meaning modification using the same attribute will result differently on varying object instance, depending on its context. One notable example for this aspect is the composition of visual attribute concept of "ripe" with various fruits such as lemon and coffee: lemon turns into yellow as it ripens whereas ripe coffee is red. Other examples can be seen on Figure 1.

In this work, we are particularly interested in evaluating the compositionality of the state-of-the-art visual-semantic model recently proposed by [Faghri et al., 2017]. We attempt to study the behaviour of the multimodal model as well as the structure of the shared representation space. To this end, we consider a particular task of composing the semantic of attribute and object (adjective and noun from the linguistic perpective) and recognizing them in the visual domain. The images examples is split such that there are no overlapping attribute object pairs during the training and testing phase. However, we make sure that the model is trained on all invidual classes of objects and attributes. We perform quantitative performance comparison with other baseline models which are formulated to explicitly model composition of m numbers of visual primitives. Additionally, we experiment with alternative compositional functions for the text encoder to see whether Recurrent Neural Network (RNN) used in visual-semantic model is the most suitable to compose word meaning. Our quantitative result shows that visual-semantic model perform reasonably well in composing unseen visual primitives. We also analyze the predictions made this model to see its behaviour in deciding which combination of attribute and object to output. Lastly, our inspection suggests that the resulting multimodal representation space posses some convenient arithmetic properties.

2. Approach

Our evaluation task is formally defined as follow: given an input image I, the goal is to predict its object class $o \in O$ along with its salient visual attribute $a \in A$. Model output of a given image is considered as correct only if both the object and attribute prediction match with the label ground truth. We consider the state-of-the-art multimodal image-text representation learning model VSE++ [Faghri et al., 2017] with several modifications. Additionally, we propose different compositional functions for the text encoder to validate if the LSTM-based compositional function in VSE++ is best suited for the evaluation task.

VSE++ grounds visual and text input to a shared semantic space of $\mathcal{D}$ dimensions using a two-way model which consist of a Convolutional Neural Network as the visual encoder and a Recurrent Network as the word sequence encoder. We denote $\phi(i;\theta_\phi) \in \mathcal{R}^{D_\phi}$ as the feature vector representation of image i which is computed by feed-forwarding the image into a CNN parameterized by weights matrices θ_ϕ. The vector representation is retrieved by taking the activation values of the layer before fully-connected layer. The overall model is essentially invariant with the CNN architecture used, but deeper Residual Network [He et al., 2016] has shown to perform best. Next, we denote $\psi(p;\theta_\psi) \in \mathcal{R}^{D_\psi}$ as the distributed compositional representation of a phrase p which consist of *attribute-object* word pair computed by an LSTM [Hochreiter and Schmidhuber, 1997] or GRU [Cho et al., 2014] based Recurrent Neural Network (RNN) encoder. Note that the parameters $\theta\psi$ include word embedding vector for each word in the vocabulary. We take the last hidden state of the RNN cell as the phrase representation.

Both representation is then projected into a shared space by two separate linear functions f and g with $W_f \in \mathcal{R}^{D_\phi}$ and $W_g \in \mathcal{R}^{D_\psi}$ as their parameters respectively. Namely:

$$\begin{aligned} f(i;W_f,\theta_\phi) &= W_f \cdot \phi(i;\theta_\phi) \\ g(i;W_g,\theta_\psi) &= W_g \cdot \psi(i;\theta_\psi) \end{aligned} \tag{1}$$

The metric that is used as the similarity measure between image and phrase representation in this space is cosine distance, i.e.,

$$s(i,p) = \frac{f(i;W_f,\theta_\phi) \cdot g(i;W_g,\theta_\psi)}{\|f(i;W_f,\theta_\phi)\|\|g(i;W_g,\theta_\psi)\|} \tag{2}$$

In practice, we first normalize the vectors into unit norm and compute the similarity by taking their dot product.

In total, we have set of model parameters θ which consist of θ_ψ, W_f, and W_g. The CNN parameters are initialized by weights from pre-training over 1K-ImageNet classification challenge. During the later stage of the training for our task, we fine-tune the weights of CNN and therefore θ_ϕ is included in θ. In addition, we also initialize the word representation weights in θ_ψ using pre-computed word embeddings of dimension $K = 300$ learned using enriched skip-gram model [Bojanowski et al., 2017] on Wikipedia corpus.

VSE++ is trained to optimize a triplet loss variant which put emphasis on hard negative example. Namely, let $s(i,p)$ as the similarity measure between a pair of an image and its corresponding attribute-object phrase description (*positive example*). Next, let $s(i,p_{neg})$ as the similarity between an image with a randomly sampled contrastive or non-descriptive phrase and $s(p,i_{neg})$ to be vice-versa (we refer them as *negative examples*). As can be seen on figure 2, once the images and their phrases get projected to the shared space, we take the negative examples of each image and phrase which are located the closest and push them further away by certain margin. These nearest negative examples of each image and phrase are referred as hard negative and we denote them as $s(i,p^+_{neg})$ and $s(p,i^+_{neg})$ respectively.

The triplet loss function for each image-text pair example is formally defined as follow:

$$\begin{aligned} loss^+_{neg}(i,p) = max\{0, \alpha + s(i,p^+_{neg}) - s(i,p)\} + \\ max\{0, \alpha + s(p,i^+_{neg}) - s(i,p)\} \end{aligned} \tag{3}$$

The function comprises of two hinge function terms with the margin distance around the positive image or phrase set by constant α. The training minimizes the empirical risk $\mathcal{E}$ with respect to parameters θ which is defined by the mean of Sum of Max of Hinges over the training set $\mathcal{S} = \{(i_n,p_n)\}^N_{n=1}$, namely:

$$\mathcal{E}(\theta;\mathcal{S}) = \frac{1}{N}\sum_{n=1}^{N} loss^+_{neg}(i_n,p_n)$$

In practice, the negative phrase examples is sampled over the entire training data by treating all phrases paired with other images **within the mini-batch** as negative examples. Similarly, for every phrase description within a mini-batch, all images other than its pair are counted as the negative examples. Note that, although the chances are low, a mini-batch might contain two or more distinct images with the same phrase description. In order to alleviate any optimization issue because of this outliers, we randomly pick only one example of pairs that belong to the same phrase.

During the test stage, the attribute and object of an image is predicted by taking the top-k nearest phrase located around its projection. Phrases representations are computed only once in the initial step by encoding all possible combination of attribute and object labels in the dataset using the trained Recurrent Network. The image representation in the share space is then computed by the CNN image encoder.

2.1. Tensor-based Composition Function

We describe an alternative text encoder Ψ as the alternative to the Recurrent Neural Network encoder. Inspired by [Socher et al., 2013], which uses a tensor-based function in the context of Recursive Tree Network, we implement and experiment with a similar function to compose the distributional semantic representation of attribute and object words before it is being projected to the shared embedding space. The function is formally define as follow: let $h \in \mathcal{R}^d$ as the tensor product, and let w_{att} and w_{obj} as the word embedding of attribute and object respectively with d dimensionality. Let $V \in \mathcal{R}^{2d \times 2d \times d}$ be the tensor product operator that compose w_{att} and w_{obj} into h. For brevity, we denote the i-th slice of V as $V^{[i]} \in \mathcal{R}^{2d \times 2d}$, and this slice will correspond to the i-th element of vector h, i.e., $h^{[i]}$. The computation of $h^{[i]}$ is defined as follow:

$$h^{[i]} = \begin{bmatrix} w_{att} \\ w_{obj} \end{bmatrix}^T \cdot V^{[i]} \cdot \begin{bmatrix} w_{att} \\ w_{obj} \end{bmatrix} \tag{4}$$

Finally, the phrase representation $\Psi(p; \theta_\Psi) \in \mathcal{R}^{D_\Psi}$ is computed by the following operation:

$$\begin{gathered} h = \begin{bmatrix} w_{att} \\ w_{obj} \end{bmatrix}^T \cdot V \cdot \begin{bmatrix} w_{att} \\ w_{obj} \end{bmatrix} \\ \Psi(p; \theta_\Psi) = W_\Psi \cdot f\left(h + W_\vartheta \begin{bmatrix} w_{att} \\ w_{obj} \end{bmatrix} \right) \end{gathered} \tag{5}$$

where f is a leaky RELU function and W_Ψ is a the weights of a linear model that projects tensor product to the shared embedding space. In summary, the tensor model Ψ is parameterized by $\theta_\Psi = \{V^{[1:d]}, W_\Psi, W_\vartheta, w_{att}, w_{obj}\}$.

2.2. Multilayer Perceptron Composition Function

We describe another text encoder denoted as $\mathcal{T}$. This text encoder consist of similar composition function as one defined in [Misra et al., 2017]. It is essentially a feedforward neural network which consist of three fully connected layers with a LeakyRELU [He et al., 2015] non-linearity operator in between. Similar to the setup of section 2.2, we define $w_{att}, w_{obj} \in \mathcal{R}^d$ as the word embeddings for attribute and object. Let fc_j be the intermediate representation computed by j-th fully connected layer. The phrase representation $\mathcal{T}(p; \theta_\mathcal{T})$ is then computed as follow:

$$\begin{gathered} fc_1 = f\left(\begin{bmatrix} w_{att} \\ w_{obj} \end{bmatrix}^T \cdot W_1 \right); fc_2 = f\left(fc_1^T \cdot W_2 \right) \\ fc_3 = f\left(fc_2^T \cdot W_3 \right); \mathcal{T}(p; \theta_\mathcal{T}) = fc_3 \cdot W_{proj} \end{gathered} \tag{6}$$

where f is the non-linearity Leaky RELU function and the dimensions of each linear model weights matrix are as follow: $W_1 \in \mathcal{R}^{2d \times 3d}$, $W_2 \in \mathcal{R}^{3d \times (\frac{3}{2}d)}$, $W_3 \in \mathcal{R}^{(\frac{3}{2}d) \times d}$, $W_{proj} \in \mathcal{R}^{d \times \mathcal{D}}$.

Additionally, we describe a slight modification of the fully connected model which allow the vector representation of attribute and object word to interact more explicitly. Similar to [Conneau et al., 2017], the input to the first layer is extended from just a concatenation of w_{att} and w_{obj} with other vectors consisting of: (i) element-wise product $w_{att} * w_{obj}$, (ii) absolute element-wise difference $|w_{att} - w_{obj}|$. Formally, the fc_1 will be then computed by the following:

$$fc_1 = f\left(\begin{bmatrix} w_{att} \\ w_{obj} \\ w_{att} * w_{obj} \\ |w_{att} - w_{obj}| \end{bmatrix}^T \cdot W_1 \right)$$

where the dimension of $W_1, W_2, W_3$and W_{proj} adjusted accordingly. The explicit interaction allowed by this function is more restrictive than the tensor-based function proposed from the previous section. We argue that this vector interaction might already be sufficient while maintaining the number of parameters to be small and thus less prone to overfitting over training data.

2.3. Weak Baseline: CRF-based Classifier

As a baseline, we would like to have a single visual model that predicts multiple output in a less naive way then predicting all possible combination of attributes and objects as a separate classes. We describe a structured prediction model based on a Conditional Random Field for the baseline. The CRF model, given an image i will predict the set of output $S = \{attribute, object\}$. We model the distribution of the output by the following:

$$\begin{aligned} P(att, obj|i; \theta) &= P(obj|i; \theta) P(att|obj; i; \theta) \\ &\equiv \psi_{obj}(obj, i; \theta) \times \psi_{att}(att, obj, i; \theta) \end{aligned} \tag{7}$$

where potential function ψ_{obj} and ψ_{att} are both a log linear:

$$\begin{aligned} \psi_{obj}(obj, i; \theta) &= e^{\phi_{obj}(obj, i, \theta)} \\ \psi_{att}(att, obj, i; \theta) &= e^{\phi_{att}(att, obj, i, \theta)} \end{aligned} \tag{8}$$

These function are computed by feedforward multilayer perceptron $\phi_{obj}(obj, i, \theta)$ and $\phi_{att}(att, obj, i, \theta)$ which take as input an image vector representation given by a pretrained CNN e.g. ResNet152 [He et al., 2016] or VGG network [Simonyan and Zisserman, 2014].

The model parameters θ is trained to optimize the sum of log-likelihood of observing the correct pair of attribute and object over the training example set $\mathcal{M}$, i.e.:

$$\theta = \arg\max_\theta \sum_{i \in \mathcal{M}} \log \left(P(att, obj|i; \theta) \right) \tag{9}$$

		Top-k accuracy			Average Accuracy
		k ↪ 1	2	3	
Chance		0.14	0.28	0.42	0.28
[Misra et al., 2017]	Visual Product	9.8	16.1	20.6	15.5
	Label Embedding (LE)	11.2	17.6	22.4	17.06
	LE Only Regression (LEOR)	4.5	6.2	11.8	7.5
	LE+Reg (LE+R)	9.3	16.3	20.8	15.46
	Composition Func.	**13.1**	**21.2**	**27.6**	**20.63**
Ours	Tensor Text Encoder	5.2	11.1	15.9	10.7
	MLP Text Encoder	6.53	13.215	19.431	13.05
	MLP+Explicit Interaction Text Encoder	8.4	16.44	22.813	15.88
	RNN Text Encoder Resnet 101	9.129	16.45	22.479	16.01
	RNN Text Encoder Resnet 101+finetune	9.83	17.99	24.27	17.36
	RNN Text Encoder Resnet 152	9.47	17.04	22.85	16.45
	RNN Text Encoder Resnet 152+finetune	N/A	N/A	N/A	N/A

Table 1: Evaluation result on unseen pair of attribute and object on MITStates dataset [Isola et al., 2015]. We evaluate on 700 unseen pairs which consist of around 19k images.

The inference of this model is done via a max-marginal approach. Namely, we first take the object which has the highest marginal score over the attributes:

$$obj = \arg\max_{obj} P(obj|i;\theta) + \sum_{att \in A} P(att|i; obj\theta)$$

Given the highest scoring object obj, we take the attribute of this obj which gives the highest potential.

2.4. Strong Baseline: Multi-classifiers Composition

[Misra et al., 2017] addresses the visual composition problem by introducing a transformation network that compose the output of multiple classification models trained to recognize primitive visual concepts e.g. attributes, and objects. The method first assumes access to limited set of primitive concept combinations (e.g. *red-wine*), and train an individual model on each primitive concept. These classifiers is then taken as inputs to the transformation network that perform a non-linear mapping to produce a compositional representation. Lastly, they take the dot product of this compositional representation with image representation from a pretrained CNN to compute a log-likehood for each *attribute-object* combination. **Note** that it is not clear how the inference is done during the test time. We presume that, given an image, the maximum log-likehood is found by computing the score all possible combination.

3. Experiments

3.1. Implementation Details

The Convolutional Neural Network (CNN) architecture of our choice is 101 layers Residual Network [He et al., 2016] (except for the last experiment where we use 152 layers to push the accuracy result). The Recurrent Network text encoder uses LSTM cell to compose the word sequence representation. We train all model using Adam solver [Kingma and Ba, 2014] with a learning rate of $2e^{-4}$ for 20 epochs. We increase the exploitation during the training by decaying the learning rate by 10 for every 6 epochs. To avoid the gradient leading the optimization to a steep surface, we perform gradient clipping i.e. the magnitude of the gradient is clipped to 2.0 while the direction is preserved. All CNN models are pre-trained on ImageNet 1K classification task; we first exclude the parameters of the CNN to be optimized. After we finished the first 20 epochs, we finetune the CNN parameters for another 20 epochs, and starts the learning rate from $2e^{-5}$.

3.2. Evaluation Dataset

We perform our compositionality evaluation on MIT States Dataset [Isola et al., 2015] which consist of images labeled with pair of *attribute* and *object*. It comprises of roughly 53k images from 245 classes of objects and 115 attribute classes. The dataset is particularly challenging due to the range of visual attribute classes it covers. The attributes range from simple parametric changes such as color and geometry; to physical transformation with complex semantic meaning e.g. *bending*, *peeled*, *dry*, or *aged*. This dataset also suffers from annotation artifacts where an image might accept multiple correct attribute or object labels. The procedure in how the data is collected also introduce many mislabeling noise.

Similar to [Misra et al., 2017], we randomly split the dataset into training and test set such that there are no *attribute-object* overlap between the two. The training set covers 1292 *attribute-object* pairs consisting of nearly 34k images, while the test set has 700 pairs with around 19k

	a) 	b) 	c) 	d) 	e) 
GT	steaming pot	eroded beach	lighweight coat	young iguana	straight desk
pred@1	steaming bowl	eroded beach	lightweight coat	new iguana	empty table
pred@2	steaming pot	eroded coast	heavy coat	young iguana	empty desk
pred@3	steaming coffee	eroded shore	thick coat	old iguana	large table
pred@4	steaming plate	weathered shore	thin coat	small iguana	small table
pred@5	steaming tea	weathered beach	crinkled jacket	large iguana	straight desk
	f) 	g) 	h) 	i) 	j) 
GT	cut cable	straight blade	coiled gemstone	painted building	diced apple
pred@1	thin cord	large knife	engraved jewelry	old boat	mashed garlic
pred@2	cut cable	engraved knife	engraved necklace	heavy boat	crushed garlic
pred@3	winding cord	curved sword	pierced jewelry	grimy boat	diced garlic
pred@4	thin cable	straight knife	small chains	burnt boat	fresh garlic
pred@5	winding cable	engraved sword	large chains	new boat	mashed banana

Table 2: We show the predictions by VSE++ on images of unseen composition. We observe that the behaviour model depends on the visual properties of the attributes and how they change visually across different objects.

images. Using this setup, we are able to measure how the models perform in *zero-shot* setting.

3.3. Quantitative Results

The accuracy measures on MITStates data of our visual-semantic models compared with [Misra et al., 2017] baselines are summarized in the table 1. The highest performance is shown by RNN-ResNet101 model after it is fine-tuned. Its average accuracy is arguably stronger than most of the baselines used in [Misra et al., 2017] by significant margins and only 3% lower than their best model.

Interesting observations can be seen from the choice of composition function for the text encoder. The tensor-based function surprisingly perform worst with only 5.2% top-1 accuracy. Simpler feedforward fully-connected model can actually give higher accuracy than the tensor-based function. Making the feature representation interaction explicit (although in a limited way) turns out can give a performance boost by nearly 3% of average accuracy.

Lastly, we can see that adding depth to the Convolutional Network can add some degree of improvement. Although insignificant, the result from ResNet152 is higher than ResNet101. Unfortunately, we cannot finetune ResNet152 parameters to our dataset due to the limitation of GPU memory. We expect that the finetuned ResNet152 can give even higher accuracy quantity. The hypothesis is that the deeper network has higher abstraction capacity and can encode more number of visual concepts, ranging from simple to complex ones. Note that the CRF model almost cannot recognize unseen composition during the test time. Thus, we exclude its measures from the table.

3.4. Qualitative Results

Figures in table 2 shows a lot about how the visual-semantic model behave on the *attribute-object* recognition. We can see that the model perform really well on attributes that are visually salient in the image. In example (a), the steam is very obvious to recognize and therefore the top-5 predictions include "*steaming*" as their attributes. The visual appearance of other attributes such as "eroded" from example (b) are not too different across varying objects and so it is easily recognize by the model.

It is also interesting to see how our model actually makes a correct prediction on example (e) and (i) but counted as incorrect due to inherent ambiguity of the images: the desk in example (e) could be either described as empty and straight; while image (i) contains both building and boat. Predictions on images (f), (g), and (h) are incorrect but they are seman-

Table 3: We show the arithmatic properties of the multimodal representation space. The resulting vector from the middle column are used to retrieve the labels on the right-most column.

Ground Truth	Operations	Result (ordered)
crinkled paper	$F_c\Big($  $\Big) - F_p\Big($"$paper$"$\Big) + F_p\Big($"$fabric$"$\Big)$	wrinkled fabric, crinkled fabric, folded fabric, frayed fabric, ruffled fabric
peeled banana	$F_c\Big($  $\Big) - F_p\Big($"$banana$"$\Big) + F_p\Big($"$apple$"$\Big)$	ripe apple, sliced apple, peeled apple, cored apple, pureed apple
huge kitchen	$F_c\Big($  $\Big) - F_p\Big($"$kitchen$"$\Big) + F_p\Big($"$bathroom$"$\Big)$	large bathroom, wide bathroom, full bathroom, huge bathroom, small bathroom

(a) object modification

Ground Truth	Operations	Result (ordered)
ruffled cake	$F_c\Big($  $\Big) - F_p\Big($"$ruffled$"$\Big) + F_p\Big($"$sliced$"$\Big)$	sliced cake, sliced apple, sliced cheese, sliced bread, sliced meat
large bowl	$F_c\Big($  $\Big) - F_p\Big($"$large$"$\Big) + F_p\Big($"$steaming$"$\Big)$	steaming bowl, steaming pot, steaming soup, steaming water, steaming bubble
coiled hose	$F_c\Big($  $\Big) - F_p\Big($"$coiled$"$\Big) + F_p\Big($"$straight$"$\Big)$	straight tube, straight screw, straight hose, straight blade, straight sword

(b) attribute modification

tically very related to the ground truth. Example (d) and (c) could actually show the drawback of using distributional semantic for performing inference. Notice that on image (c), the second highest prediction is actually the contradiction of the first prediction, but since they are contextually similar, their projection on the embedding space located closely. Furthermore, the attributes of top two predictions are semantically related, but "new" is not a suitable modifier to "iguana" according to human common sense.

3.5. Exploring Embedding Space Regularities

Word vector representation learned using skip-gram model in [Mikolov et al., 2013] has shown to posses a linguistic regularities which could be used to perform analogical reasoning simply by using vector arithmetic. The prominent example of this word embedding property include: $emb(\text{"}king\text{"}) - emb(\text{"}man\text{"}) = emb(\text{"}queen\text{"}) - emb(\text{"}woman\text{"})$. Here we are interested to see whether

similar phenomenon emerge in the multimodal space.

Table 3b shows some convenient results from performing vector addition and subtraction in the shared space. We first encode the image into the space using learned CNN to retreive its representation. We then compute the projection of a single word (either object and attribute) using the text encoder. The word representation in this space is then subtracted and added to the image representation. The resulting vector is used to retrieved nearest k labels.

4. Conclusion

We presented an evaluation of visual-semantic model VSE++ on the task of predicting unseen composition of attribute and object from open world images. Our experiments on MIT States dataset shows that this model posses some degree of compositionality considering how it performs competitively with compositional model baselines which are trained explicitly to compose multiple classifiers of visual primitives. Our qualitative results indicate that our visual-semantic model performs better than what the quantitative result suggest by making semantically reasonable predictions. We also show that the resulting space has some regularities which could be useful for many task including images retrieval.

References

[Bojanowski et al., 2017] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

[Cho et al., 2014] Cho, K., van Merrienboer, B., aglar Gülehre, Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

[Conneau et al., 2017] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

[Faghri et al., 2017] Faghri, F., Fleet, D. J., Kiros, J. R., and Fidler, S. (2017). Vse++: Improving visual-semantic embeddings with hard negatives.

[Frome et al., 2013] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. In *NIPS*.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. In *ECCV*.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9 8:1735–80.

[Isola et al., 2015] Isola, P., Lim, J. J., and Adelson, E. H. (2015). Discovering states and transformations in image collections. In *CVPR*.

[Karpathy and Fei-Fei, 2015] Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:664–676.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

[Kiros et al., 2014] Kiros, R., Salakhutdinov, R., and Zemel, R. S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

[Misra et al., 2017] Misra, I., Gupta, A., and Hebert, M. (2017). From red wine to red tomato: Composition with context. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1160–1169.

[Pavlick and Callison-Burch, 2016] Pavlick, E. and Callison-Burch, C. (2016). Most "babies" are "little" and most "problems" are "huge": Compositional entailment in adjective-nouns. In *ACL*.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

[Socher et al., 2013] Socher, R., Perelygin, A. V., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank.

[Wang et al., 2016] Wang, L., Li, Y., and Lazebnik, S. (2016). Learning deep structure-preserving image-text embeddings. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5005–5013.