Grounded Learning of Color Semantics with Autoencoders

Dev Bhargava
Department of Computer Science
Stanford University
Stanford, CA 94305
devb@stanford.edu

Gabriel Vega
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
gavega@stanford.edu

Blue Sheffer
Symbolic Systems Program
Department of Computer Science
Stanford University
Stanford, CA 94305
bsheffer@stanford.edu

Abstract

Humans learn language by grounding word meaning in the physical world. Recent efforts in natural language processing have attempted to model such multimodal learning by incorporating visual information into word and sentence representations. Here, we explore the task of grounding lexical color descriptions in their visual referents. We propose an RNN-based autoencoder model to learn vector representations of sentences that reflect their associated color values. Our model effectively learns a joint visual-lexical space that demonstrates compositionality and generalizes to unseen color names. As a demonstration of such a space learned, we show that our model can predict captions from color representations and color representations from captions. In addition to successfully modeling color language, this work provides a novel framework for grounded language learning.

1 Introduction

Natural language is grounded in perception. Whereas humans learn language incrementally, associating words with their experiential referents, current state-of-the-art natural language processing (NLP) techniques rely on learning from massive text corpora. Grounded language learning is a technique that uses a multimodal set of inputs for language acquisition rather than simply a set of words or symbols. The study of grounded language learning has the potential to give insight into how humans learn language as well as make NLP systems more human-like.

We are interested in the grounded semantics of color. Grounding color descriptions in their physical representations serves as a proof of concept of grounding language in the visual world. We propose a model for learning joint visual-lexical representations of colors via autoencoders.

2 Related Work and Background

Color naming and color generation serve as ideal benchmark tasks for models of grounded language learning. There is a growing body of work that uses color descriptions as a case study in grounded

language modeling. The LUX model introduced by McMahan and Stone (2015) uses Bayesian generative modeling for color naming. Monroe et al. (2016) improved on this model by using a Fourier basis representation of color in conjunction with a long short-term memory (LSTM) recurrent neural network. A character-level model for color naming and color generation has also recently been reported by Kawakami et al. (2016).

We aim to create "grounded" vector representations of color descriptions using autoencoders. An autoencoder is a neural network that attempts to copy its input to its output, usually with some constraint on the internal representation of its input (e.g. undercompleteness, sparsity) (Goodfellow et al. 2016). RNN-based autoencoders have recently been explored as a way of learning fixed-length vector representations of sentences (Oshri and Khandwala, 2016).

3 Approach

As outlined in Figure 1, our model takes as inputs lexical captions and attempts to reconstruct them. Each caption has an associated color represented by a three dimensional normalized HSV vector. The HSV color model represents a given color by providing three components: *hue*, which is a circular measure for the primary color component, *saturation*, capturing the amount of whiteness, and *value*, which holds the amount of lightness or brightness. While LUX (McMahan and Stone, 2015) and Monroe et al. (2016) utilized HSV color representations, we opted to use normalized RGB values to avoid the complications introduced by having a circular variable. RGB is an additive color space where a given color is defined by its relative intensities of red, green, and blue.

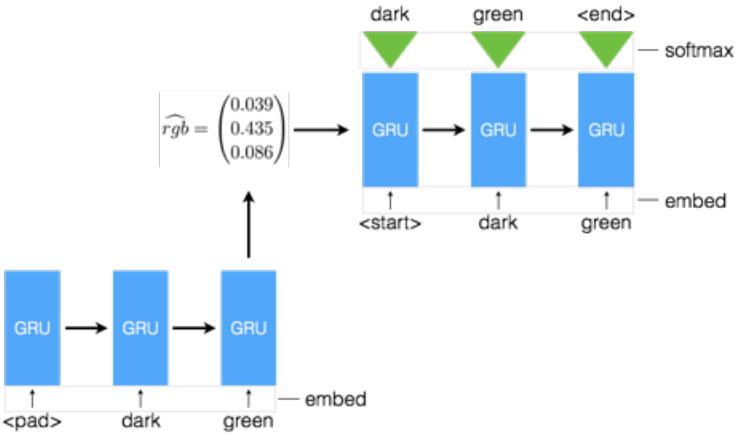


Figure 1: RNN autoencoder for color description reconstruction

The architecture contains two connected sub-modules: an encoder that maps captions into a low-dimensional latent space, and a decoder which reconstructs the encoded representation back to a caption. In our encoder, we utilized an RNN iterating over word embeddings to produce a caption representation. We standardized the length of the RNN inputs by introducing a special 'pad' token *at the beginning* of captions where necessary (to ensure that the non-'pad' tokens are closest the final output). We use GloVe word embeddings (Pennington et al. 2014) pre-trained on Wikipedia 2014 and GigaWord 5 corpora. We allowed the embeddings to be updated by backpropogation (declared them as variables) to incorporate the differences in semantics in the context of color description.

The output of the encoder RNN was then transformed from the caption space to our latent space as follows:

$$enc(x) = \sigma(RNN(caption)W_{cl} + b_l)$$

Note that the sigmoid non-linearity ensures that each value is between zero and one (the range of the normalized RGB values).

In the decoder submodule, we first project our latent representation into the caption space via an affine transformation and then feed that through an RNN to produce a softmax distribution across our vocabulary from which the token with highest probability is included into our predicted caption.

To each cell of the RNN we also input the latent representation such that the latent representation could play a more direct role in token selection throughout the caption. A loss contribution for this sub-module compared predicted tokens to processed true captions that included an 'end' token after the last word and 'pad' tokens following it to standardize length but the penalty for not predicting the pad token was excluded (via masking). The reconstruction error was applied to all tokens up to and including the 'end' token, allowing our model to predict variable-length captions.

3.1 Optimization

Our loss is composed of penalties incurred from the encoder and decoder portions:

$$\mathcal{L}(\theta) = \mathcal{L}_{reconstruction}(\theta) + \mu\mathcal{L}_{visual}(\theta)$$

The first term, the decoder penalty, is the cross-entropy loss for the reconstruction of the captions:

$$\mathcal{L}_{reconstruction}(\theta) = \sum_{t=1}^{T} CE(x^{(t)}, \hat{x}^{(t)})$$

The second term, the encoder loss, penalizes the latent representations of the captions, $enc(x)$, that are far from their corresponding visual representations ($v(x)$):

$$\mathcal{L}_{visual}(\theta) = D(enc(x), v(x))$$

The motivation behind the inclusion of this distance penalty was two-fold. First, in recent work in grounded learning (Lazaridou et al. 2016), multimodal word embeddings were learned by imposing a similar "visual similarity" penalty on word vectors. Second, as in variational autoencoders (Kingma and Wellington, 2013), we aimed to shape the distribution of the latent vectors to reflect a known distribution such that we could use the decoder as a generative model.

The choice of D is a hyperparameter of our model. Our experiments include using $\ell 2$ distance, an "RGB distance", and a max margin loss. The gain parameter μ controls the tradeoff between the two losses, as is shown in Figure 2.

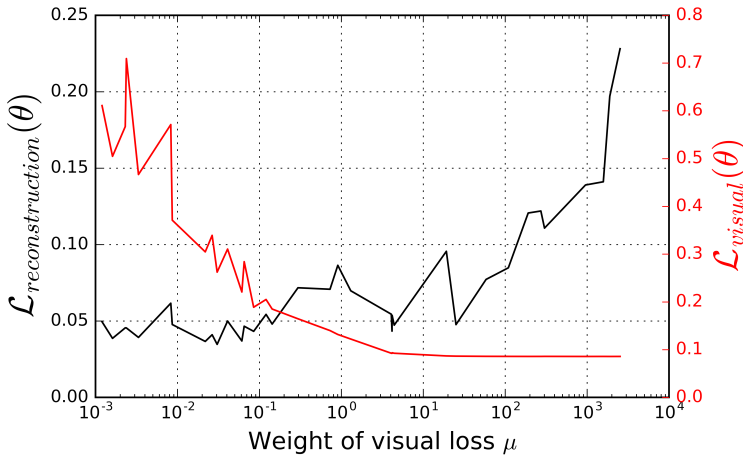


Figure 2: Tradeoff between reconstruction and visual losses for different values of μ

4 Experiments

4.1 Dataset

We follow McMahan and Stone (2015) and Monroe et al. (2016) in using a dataset generated from an online survey where human users were presented colors and their descriptions were recorded

(Monroe, 2010). We used a subset of these color-description pairs provided by McMahan and Stone, (2015). This subset included those pairs generated only from non-colorblind, English speakers and that were reported at least one hundred times. Additional processing was performed by McMahan and Stone (2015) to normalize spelling and remove high-frequency spam labels. As such, the dataset includes 2,176,417 unique color -description pairs pre-separated into training, development, and test sets of sizes 1,523,108, 108,545, and 544,764 respectively. The dataset includes only 829 unique captions but each caption can be paired with multiple HSV/RGB color values and vice-versa.

4.2 Model Considerations

4.2.1 HSV vs RGB

In our final model, we opted to size our latent space to the same as RGB space ($\mathcal{R}^3$). We did not see any improvements in performance with increasing the size of the latent space (and in turn randomly projecting RGB values into that space with a pseudo-invertible transformation by a semi-orthogonal tensor). Restricting the dimensionality of the latent space also served the purpose of helping prevent overfitting. We opted to use RGB representations instead of HSV to avoid complications introduced by the circular nature of the "hue" measure in HSV while capturing differences in colors.

4.2.2 Choice of Distance Function

We considered several distance functions D for the visual loss $\mathcal{L}_{visual}(\theta)$:

$$D_{\ell 2}(enc(x), v(x)) = \|enc(x) - v(x)\|^2$$
$$D_{RGB}(enc(x), v(x)) = \|enc(x)^2 - v(x)^2\|^2$$
$$D_{maxmargin} = \sum_{c' \sim P_n(c)} \max\{0, \gamma - \cos(enc(x_c), v(x_c)) + \cos(enc(x_c), v(x_{c'}))\}$$

Our max margin distance was adapted from Lazaridou et al. (2016). Following the max margin formulation, our model optimizes cosine similarities between caption latent and true RGB representations against negative samples. We formulated the D_{RGB} loss as a means to account for the gamma compression in storing luminescence information. In our evaluations, we opted for the D_{RGB} metric because it resulted in latent encodings that were most visually similar to the colors they represent.

4.2.3 Minor Improvements

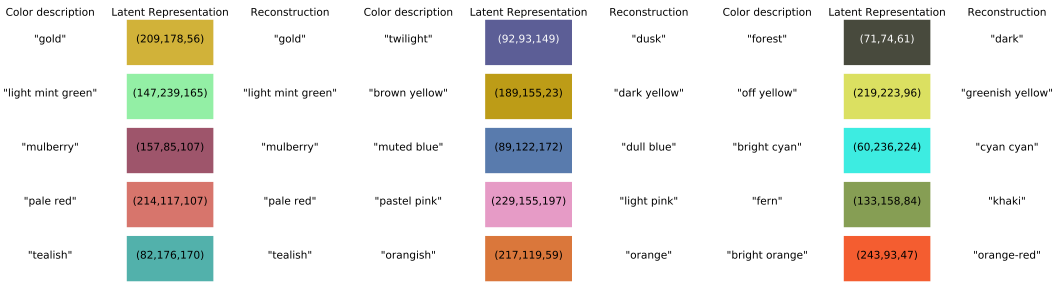
Additionally, although our task involved short captions (no more than three words), GRU cells outperformed basic RNN cells in terms of total as well as visual and reconstruction losses. Although pre-trained semantic distributional representations may not contribute to color meaning in a standard or easily learnable way, we also noticed slight improvements in loss, sample reconstruction and caption generation using pre-trained GloVe as compared to randomly initialized embeddings. Perhaps the improvements in using pre-trained embeddings would be more pronounced in a setting with a larger data set and more unique words.

4.3 Results

4.3.1 Color Description Reconstruction

We first report the ability of our model to reconstruct color descriptions. Examples of colors, their latent representations (and corresponding RGB values), and reconstructions can be seen in Figure 3. In panel (a), we show examples of exact reconstructions.

In panel (b) are cases where the latent representations faithfully represent their corresponding color descriptions, but the decoder's reconstructions do not exactly match the encoder inputs. There is a trend of over generalization in these errors. For example, the model opts to output the more often seen "orange" than the rarer "orangish". The panel also illustrates the error of slightly misplaced modifiers.



(a) Exact reconstructions (b) Inexact, valid reconstruction (c) Incorrect reconstruction

Figure 3: Caption latent representations and reconstruction

Panel (c) then illustrates some gross errors of incorrectly applying modifiers (“greenish”), repeating color names in place of modifiers (“cyan cyan”), and even predicting lone-standing modifiers (“dark”).

4.3.2 Compositionality

Figure 4: Learned compositionality of color descriptors (unseen colors names in bold)

Monroe et al. (2016) achieved generalization to compositional descriptions not found in the training set, but was only able to show this via the conditional likelihood of color descriptions. When presented with unseen captions, our generative model provides color representations that reflect compositional understanding. As illustrated in Figure 4, our model is able to generalize how modifiers in the caption space, for example “light”, “dark”, and “very”, extend to the joint visual space.

In addition to learning the basic scalar modifiers presented in Figure 4, our model learns more complex transformations of color space, such as “neon”, “ugly”, and “vibrant”. We show these modifiers applied to the basic color spectrum in Figure 5.

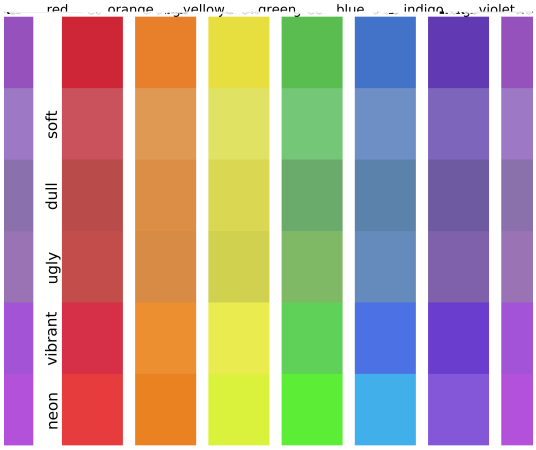


Figure 5: Various modifiers applied to spectrum of base colors

4.3.3 Sampling the Latent Space

The decoder of our model learns to map latent representations of color descriptions back to color descriptions. Because the encoder has been penalized to produce encodings that reflect the RGB values associated with their color name, we can decode not just these latent representations but also any RGB values. We show examples of sampling the latent space in Figure 6.

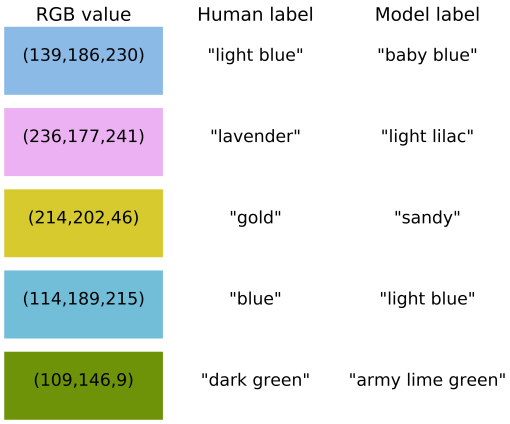


Figure 6: Color naming. Left: the sampled RGB value. Middle: an example of human label for that RGB value. Right: Output color name from our model's decoder.

Even though our model has not been directly optimized for the task of color naming as in McMahan and Stone (2015) and Monroe et al. (2016), it still produces accurate descriptions of color. In fact, there are many cases where the model produces more precise descriptions of colors than humans (as in Figure 6, "blue" v.s. "light blue"). However, our "test accuracy" (the percentage of RGB values for which our decoder's output exactly matched human given labels) is only 7.7%. This is unsurprising, as our model was not optimized to reproduce human given labels for specific RGB values, and there is considerable variability in human labels for a given RGB value (McMahan and Stone, 2015).

5 Conclusion

We present a model that learned a joint color-caption space and is capable of generating both color descriptions from captions and captions from color descriptions. We learn compositionality in a generative fashion rather than by the probabilistic method in Monroe et al. (2016).

For future work, we would look to improve the evaluation metrics used for the task. A perfect match accuracy over penalizes model generated captions that are humanly viable (e.g. those in Figure 3 (b)). We could achieve this by incorporating cosine similarity of output tokens compared to ground truths or by expanding the sets of correct solutions considered per color value and/or caption. We would also hope to generalize such a model to unseen captions and words by incorporating, for example, a more effective mapping from general to color-context semantics.

Ultimately we would extend our framework to learning more complex multi-modal spaces. As an example, we suggest applying this model to perform tasks such as captioning on real world images represented by latent convolutional codes.

Acknowledgements

We would like to thank Jon Gauthier and the CS224N teaching staff for guidance, and Microsoft Azure for computing resource.

References

Andreas, Jacob, and Dan Klein. "Grounding Language with Points and Paths in Continuous Spaces." CoNLL. 2014.

Lazaridou, Angeliki, et al. "Multimodal Word Meaning Induction From Minimal Exposure to Natural Text." Cognitive Science. 2016.

Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep Learning. Cambridge (EE.UU.): MIT, 2016. Print.

Kawakami, Kazuya, et al. 2016. "Character Sequence Models for ColorfulWords." arXiv preprint arXiv:1609.08777 (2016).

Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

McMahan, Brian, and Matthew Stone. "A Bayesian model of grounded color semantics." Transactions of the Association for Computational Linguistics 3 (2015): 103-115.

Munroe, Randall. 2010. Color survey results. Online at http://blog.xkcd.com/2010/05/03/color-surveyresults.

Mordatch, Igor, et al. "Learning to communicate". OpenAI Blog Post. Online at https://openai.com/blog/learning-to-communicate

Monroe, Will, Noah D. Goodman, and Christopher Potts. "Learning to generate compositional color descriptions." arXiv preprint arXiv:1606.03821 (2016).

Oshri, Barak and Nishith Khandwala. 2016. There and Back Again: Autoencoders for Textual Reconstruction. Stanford CS224D Final Project.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.