Learning to Generate Compositional Color Descriptions

Will Monroe,1 Noah D. Goodman,2 and Christopher Potts3
Departments of 1Computer Science, 2Psychology, and 3Linguistics
Stanford University, Stanford, CA 94305
wmonroe4@cs.stanford.edu, {ngoodman, cgpotts}@stanford.edu

Abstract
The production of color language is essential for grounded language generation. Color descriptions have many challenging properties: they can be vague, compositionally complex, and denotationally rich. We present an effective approach to generating color descriptions using recurrent neural networks and a Fourier-transformed color representation. Our model outperforms previous work on a conditional language modeling task over a large corpus of naturalistic color descriptions. In addition, probing the model’s output reveals that it can accurately produce not only basic color terms but also descriptors with non-convex denotations (“greenish”), bare modifiers (“bright”, “dull”), and compositional phrases (“faded teal”) not seen in training.

1 Introduction
Color descriptions represent a microcosm of grounded language semantics. Basic color terms like “red” and “blue” provide a rich set of semantic building blocks in a continuous meaning space; in addition, people employ compositional color descriptions to express meanings not covered by basic terms, such as “greenish blue” or “the color of the rust on my aunt’s old Chevrolet” (Berlin and Kay, 1991). The production of color language is essential for referring expression generation (Krahmer and Van Deemter, 2012) and image captioning (Kulkarni et al., 2011; Mitchell et al., 2012), among other grounded language generation problems.

We consider color description generation as a grounded language modeling problem. We present an effective new model for this task that uses a long short-term memory (LSTM) recurrent neural network (Hochreiter and Schmidhuber, 1997; Graves, 2013) and a Fourier-basis color representation inspired by feature representations in computer vision.

We compare our model with LUX (McMahan and Stone, 2015), a Bayesian generative model of color semantics. Our model improves on their approach in several respects, which we demonstrate by examining the meanings it assigns to various unusual descriptions: (1) it can generate compositional color descriptions not observed in training (Fig. 3); (2) it learns correct denotations for underspecified modifiers, which name a variety of colors (“dark”, “dull”); (3) it can model non-convex denotations, such as that of “greenish”, which includes both greenish yellows and blues (Fig. 4). As a result, our model also produces significant improvements on several grounded language modeling metrics.

2 Model formulation
Formally, a model of color description generation is a probability distribution \( S(d \mid c) \) over sequences of
tokens $d$ conditioned on a color $c$, where $c$ is represented as a 3-dimensional real vector in HSV space.\(^1\)

**Architecture** Our main model is a recurrent neural network sequence decoder (Fig. 1, left panel). An input color $c = (h, s, v)$ is mapped to a representation $f$ (see Color features, below). At each time step, the model takes in a concatenation of $f$ and an embedding for the previous output token $d_i$, starting with the start token $d_0 = \langle s \rangle$. This concatenated vector is passed through an LSTM layer, using the formulation of Graves (2013). The output of the LSTM at each step is passed through a fully-connected layer, and a softmax nonlinearity is applied to produce a probability distribution for the following token.\(^2\) The probability of a sequence is the product of probabilities of the output tokens up to and including the end token $\langle / s \rangle$. Note that the same color representation $f$ is input to the model at every time step in decoding.

We also implemented a simple feed-forward neural network for comparison. This architecture (atomic; Fig. 1, right panel) consists of two fully-connected hidden layers and a softmax output over all color descriptions in the dataset, treating the descriptions as atomic symbols rather than sequences.

**Color features** We compare three representations:

- **Raw**: The original 3-dimensional color vectors, in HSV space.
- **Buckets**: A discretized representation, dividing HSV space into rectangular regions at three resolutions ($90 \times 10 \times 10$, $45 \times 5 \times 5$, $1 \times 1 \times 1$) and assigning a separate embedding to each region.
- **Fourier**: Transformation of HSV vectors into a Fourier basis representation. Specifically, the representation $f$ of a color $(h, s, v)$ is given by

$$f = \left[ \text{Re}\{\hat{f}\}, \text{Im}\{\hat{f}\} \right], \quad j, k, \ell = 0.2$$

where $(h^*, s^*, v^*) = (h/360, s/200, v/200)$. This representation is inspired by the use of Fourier feature descriptions in computer vision applications (Zhang and Lu, 2002).

**Training** We train using Adagrad (Duchi et al., 2011) with initial learning rate $\eta = 0.1$, hidden layer size and cell size 20, and dropout (Hinton et al., 2012) with a rate of 0.2 on the output of the LSTM and each fully-connected layer. We identified these hyperparameters with random search, evaluating on a held-out subset of the training data.

We use random normally-distributed initialization for embeddings ($\sigma = 0.01$) and LSTM weights ($\sigma = 0.1$), except for forget gates, which are initialized to a constant value of 5. Dense weights use normalized uniform initialization (Glorot and Bengio, 2010).

**3 Experiments**

We demonstrate the effectiveness of our model using the same data and statistical modeling metrics as McMahan and Stone (2015).

**Data** The dataset used to train and evaluate our model consists of pairs of colors and descriptions collected in an open online survey (Munroe, 2010). Participants were shown a square of color and asked to write a free-form description of the color in a text box. McMahan and Stone filtered the responses to normalize spelling differences and exclude spam responses and descriptions that occurred very rarely. The resulting dataset contains 2,176,417 pairs divided into training (1,523,108), development (108,545), and test (544,764) sets.

**Metrics** We quantify model effectiveness with the following evaluation metrics:

- **Perplexity**: The geometric mean of the reciprocal probability assigned by the model to the
Table 2: Experimental results. Top: development set; bottom: test set. AIC is not comparable between the two splits. HM and LUX are from McMahan and Stone (2015). We reimplemented HM and re-ran LUX from publicly available code, confirming all results to the reported precision except perplexity of LUX, for which we obtained a figure of 13.72.

| Model    | Feats. | Perp.       | AIC          | Acc.       |
|----------|--------|-------------|--------------|------------|
| atomic   | raw    | 28.31       | 1.08×10^6    | 28.75%     |
| atomic   | buckets| 16.01       | 1.31×10^6    | 38.59%     |
| atomic   | Fourier| 15.05       | 8.86×10^5    | 38.97%     |
| RNN      | raw    | 13.27       | 8.40×10^5    | 40.11%     |
| RNN      | buckets| 13.03       | 1.26×10^6    | 39.94%     |
| RNN      | Fourier| 12.35       | 8.33×10^5    | 40.40%     |
| HM       | buckets| 14.41       | 4.82×10^6    | 39.40%     |
| LUX      | raw    | 13.61       | 4.13×10^6    | 39.55%     |
| RNN      | Fourier| 12.58       | 4.03×10^6    | 40.22%     |

Results The top section of Table 2 shows development set results comparing modeling effectiveness for atomic and sequence model architectures and different features. The Fourier feature transformation generally improves on raw HSV vectors and discretized embeddings. The value of modeling descriptions as sequences can also be observed in these results; the LSTM models consistently outperform their atomic counterparts.

Test set results appear in the bottom section. Our best model outperforms both the histogram baseline (HM) and the improved LUX model of McMahan and Stone (2015), obtaining state-of-the-art results on this task. Improvements are highly significant on all metrics (p < 0.001, approximate permutation test, R = 10,000 samples; Padó, 2006).

4 Analysis

Given the general success of LSTM-based models at generation tasks, it is perhaps not surprising that they yield good raw performance when applied to color description. The color domain, however, has the advantage of admitting faithful visualization of descriptions’ semantics. We exploit this to highlight three specific improvements our model realizes over previous ones. Visualizations are made by querying the model for the probability of the same description for each color in a uniform grid, summing the probabilities over the hue dimension (left cross-section) and the saturation dimension (right cross-section), normalizing them to sum to 1, and plotting the log of the resulting values as a grayscale image.

Learning modifiers Our model learns accurate meanings of adjectival modifiers apart from the full descriptions that contain them. We examine this in Fig. 2, by plotting the probabilities assigned to the bare modifiers “light”, “bright”, “dark”, and “dull”. “Light” and “dark” unsurprisingly denote high and low lightness, respectively. Less obviously, they also exclude high-saturation colors. “Bright”, on the other hand, features both high-lightness colors and
saturated colors—“bright yellow” can refer to the prototypical yellow, whereas “light yellow” cannot. Finally, “dull” denotes unsaturated colors in a variety of lightnesses.

**Compositionality** Our model generalizes to compositional descriptions not found in the training set. Fig. 3 visualizes the probability assigned to the novel utterance “faded teal”, along with “faded” and “teal” individually. The meaning of “faded teal” is intersective: “faded” colors are lower in saturation, excluding the colors of the rainbow (the V on the right side of the left panel); and “teal” denotes colors with a hue near 180° (center of the right panel).

**Non-convex denotations** The Fourier feature transformation and the nonlinearities in the model allow it to capture a rich set of denotations. In particular, our model addresses the shortcoming identified by McMahan and Stone (2015) that their model cannot capture non-convex denotations. The description “greenish” (Fig. 4) has such a denotation: “greenish” specifies a region of color space surrounding, but not including, true greens.

**Error analysis** Table 3 shows some examples of errors found in samples taken from the model. The main type of error the system makes is ungrammatical descriptions, particularly fragments lacking a basic color term (e.g., “robin’s”). Rarer are grammatical but meaningless compositions (“reddish green”) and false descriptions. When queried for its single most likely prediction, \( \arg \max_d S(d \mid c) \), the result is nearly always an acceptable, “safe” description—manual inspection of 200 such top-1 predictions did not identify any errors.

### Table 3: Error analysis: some color descriptions sampled from our model that are incorrect or incomplete.

| Color           | Top-1         | Sample               |
|-----------------|---------------|----------------------|
| orange          | “ugly”        | (36, 86, 63) “orange” “ugly” |
| teal            | “robin’s”     | (177, 85, 26) “teal” “robin’s” |
| tan             | “reddish green”| (29, 45, 71) “tan” “reddish green” |
| grey            | “baby royal”  | (196, 27, 71) “grey” “baby royal” |
| grey            | “baby royal”  | (196, 27, 71) “grey” “baby royal” |

### Figure 4: Conditional likelihood of “greenish” as a function of color. The distribution is bimodal, including greenish yellows and blues but not true greens. Top: LUX; bottom: our model.

5 **Conclusion and future work**

We presented a model for generating compositional color descriptions that is capable of producing novel descriptions not seen in training and significantly outperforms prior work at conditional language modeling.\(^3\) Natural extensions include character-level sequence modeling to capture complex morphology (e.g., “-ish” in “greenish”) and contextual modeling to capture how people describe colors differently to contrast them with other colors via pragmatic reasoning (DeVault and Stone, 2007; Golland et al., 2010; Monroe and Potts, 2015).

\(^3\)We release our code at [https://github.com/stanfordnlp/color-describer](https://github.com/stanfordnlp/color-describer).
Acknowledgments

We thank Jiwei Li, Jian Zhang, and Anusha Balakrishnan for valuable advice. This research was supported in part by the Stanford Data Science Initiative, NSF BCS 1456077, and NSF IIS 1159679.

References

Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Rami Al-Rfou, Guillaume Alain, Anjd Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, et al. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*.

Brent Berlin and Paul Kay. 1991. *Basic color terms: Their universality and evolution*. University of California Press.

David DeVault and Matthew Stone. 2007. Managing ambiguities across utterances in dialogue. In Ron Artstein and Laure Vieu, editors, *Proceedings of DECALOG 2007: Workshop on the Semantics and Pragmatics of Dialogue*.

Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, et al. 2015. Lasagne: First release.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*.

Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *EMNLP*.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Emiel Krahmer and Kees Van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, et al. 2011. Baby talk: Understanding and generating image descriptions. In *CVPR*.

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

Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, et al. 2012. Midge: Generating image descriptions from computer vision detections. In *EACL*.

Will Monroe and Christopher Potts. 2015. Learning in the Rational Speech Acts model. In *Proceedings of the 20th Amsterdam Colloquium*.

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

Sebastian Padó. 2006. User’s guide to *sigf*: Significance testing by approximate randomisation. http://www.nlpado.de/~sebastian/software/sigf.shtml.

Dengsheng Zhang and Guojun Lu. 2002. Shape-based image retrieval using generic Fourier descriptor. *Signal Processing: Image Communication*, 17(10):825–848.