Deep Learning Final Project: Comparative Color Representation

Ryan Culkin[*] and **Arya D. McCarthy**[*] and **Elias Stengel-Eskin**[*]
Department of Computer Science
Johns Hopkins University

Abstract

We sought to replicate and expand upon the work of Winn and Muresan (2018)—an examination of compositional color terms. The task involved predicting vectors in RGB space from a reference point and comparative description (e.g., "more electric"). We find that with *radically* fewer training epochs, we achieve comparable performance. Further, we explored a number of data preparation methods and architectures for the task; we found that a variant of the Winn and Muresan (2018) architecture was most successful.

1 Introduction

How many colors are there in the rainbow? An infinite number, to be sure, but each language divides up perceptual space into a *finite* number of categories by giving names to colors. These color words can then be used to identify their denotata for recognition tasks, as with the classic SHRDLU system (Winograd, 1971). To perform *fine-grained* object recognition, though, we may rely on not only objects' absolute properties, but also on their *relative* ones—the gradations in degree of a property among the objects. In this case, we will focus on the comparative relationships between colors, e.g., between "blue", "lighter blue", and "lighter".

While it is natural to ground a color representation in a region of color space (possibly as a fuzzy set (Kay and McDaniel, 1978)), the representation of a *relative* description is less straightforward. To wit, the representation of "light", "lighter", "lighter blue", and even "blue-green" are each tempered by the reference point from which they illustrate a contrast.

2 Grounding Comparative Relationships Between Color Words

We take a quantitative, vector-space approach to predicting the referents of comparative color descriptions. Here, the comparative term implies a locus of points with some position relative to the reference color.

Notation First, we define the notations we use in the paper. Let $\mathcal{R}$, $\mathcal{C}$, and $\mathcal{T}$ denote the sets of reference colors, comparative descriptions, and target colors, respectively. The sets $\mathcal{R}$ and $\mathcal{T}$ are each subsets of some space $\mathcal{S}$ containing all colors to model, which we take to be the standard RGB space $[0, 256]^3$. Likewise, $\mathcal{C}$ is a subset of $\mathcal{L}^k$, where $\mathcal{L} = \mathbb{R}^d$ embeds our lexicon into d-dimensional vector space. We assume, thus, that the length of a comparative description will be padded to length k. A problem instance will be of the form $(x, y) = ((r, c), t) \in \mathcal{R} \times \mathcal{C} \times \mathcal{T}$. For convenience, we also define the space $\mathcal{X} = \mathcal{R} \times \mathcal{C}$.

Comparative descriptions as vectors

3 Goals of the Project

At the outset, we claimed that we would lend more credence to the original results through a handful of search strategies:

1. Altering the network architecture
2. Hyper-parameter search
3. Exploring different word embeddings
4. Comparing between adjustable and fixed embeddings during training

In the end, the authors' obtuse data format led us to ask the question, "What is the right way to handle the data for this task?" We supplanted the fourth goal above with a new goal of exploring whether

[*]Equal contribution; sorted alphabetically

Data split	Tuples	Data points
Training	62	15.4
Validation	6	0.86
Test (seen (r, c))	62	2.40
Test (unseen (r, c))	11	0.24
Test (unseen r)	10	2.03
Test (unseeen c)	22	0.39
Test (fully unseen)	5	0.87

Table 1: Our data splits slightly differ from those of Winn and Muresan (2018), due to preprocessing decisions that were not spelled out in the paper. Data point counts are given in millions.

augmentation (or reduction!) of the dataset were sufficient. We achieved the first three preliminary goals of our report, plus this new fourth. We were able to achieve comparable performance in only 5% of the number of training epochs.

4 Data and Experiments

Data provenance Evaluation in our context is tricky; to date, there is only one major dataset of comparatively named colors. We use the English-language color naming data from the xkcd Color Survey (Munroe, 2010), as cleaned by McMahan and Stone (2015). This data was collected through a survey which asked participants to provide labels to a variety of RGB values, and leverages the fact that some color names are modified names of other reference colors – for example, "light blue" can be roughly interpreted as a lighter shade of some reference "blue" color.

Data preprocessing We define a reference color as any color that appears both on its own (e.g. "blue") and with a modifier (e.g. "light blue"). These reference colors can then be combined with a comparative (e.g. "lighter") to yield a reference-comparative pair (e.g. "lighter blue"). Winn and Muresan (2018) chose to batch their data by reference color – this leads to uneven batch sizes and forces the network to take steps in the gradient manifold that optimize for a particular color at at time, which may be suboptimal. In contrast to this, we choose instead to use a fixed batch size (2048). We also group the tuples differently, using the implicit encoding represented by the filenames in the original dataset (e.g. files ending in "test" are placed in the test split, etc.). Following Winn and Muresan (2018), we augment the data by treating each target RGB triple as its own datapoint, meaning that for every reference-comparative pair in the dataset, there are possibly hundreds of exemplars. This is especially important as not all reference-comparative pairs are found in the dataset.

Critically, we share the modeling assumption of Winn and Muresan (2018): we collapse the locus of points implied by a reference and comparator into the points' centroid. Introducing this makes the relationship between $\mathcal{X}$ and $\mathcal{Y}$ a function.[1] The sizes of our data splits are given in Table 1.

Predicting comparative values As every point in our vector representation of color is interpretable, we can compute meaningful error functions between a target color t and a model's prediction $\hat{t}$. The two quantities of interest are the angle between the prediction and target, and the distance between them. Imagining the trajectory of a description as a direction in vector space, minimizing the angle between prediction and target ensures that the vector's direction is correct. We quantify this loss using the cosine similarity between the two vectors. Additionally, we correct the magnitude of the vector, using the Euclidean distance in RGB space as an additional loss. (This is equivalent to the sum of squared errors, or SSE.) The objectives are weighted by a hyperparameter β, and the model's parameters are optimized with stochastic gradient descent.

Data preparation The original dataset, in a roundabout sense, consists of text representations of colors, coupled with sets of points—a sample from that color's extension. We investigated different aggregation strategies; after all, not all of the "light blue" points would be lighter than all "blue" points. We chose to average together the target points, consistently predicting the average of the target locus. (This also limited a quadratic explosion in training data.) On the source side, though, we experimented with two strategies. The first involved using the centroid for the reference representation; this condensation strategy would mean that we could learn a mapping between the targets and referents in a symmetric, efficient way. In practical experiments, we found that this strategy achieved near-perfect cosine similarity but failed to produce correct vector magnitudes in the space. By contrast, the approach of treating each point in

[1] In fact, a simple set-theoretic definition of a function f is a set of ordered pairs (x_i, y_i) satisfying $(x, y_1) \in f \wedge (x, y_2) \in f \implies y_1 = y_2$.

the reference locus uniquely (coupled with hyperparameter tuning) gave comparable performance to Winn and Muresan (2018) off-the-bat.

Model architecture We experimented with a number of model architectures. For one, we used a multilayer neural network with highway connections to perform prediction; it would be fed two word vectors to encode the color comparison and triplet that conveyed the reference color. We varied the network depth and found that 1 layer actually nearly matched deeper networks for a comparable number of training steps. Further, the strategy of filling in missing word embeddings randomly worked well; adapting these throughout training provided no benefit. (We explored this, but our search was not an exhaustive grid search.)

Weighting our objectives Is it more important to get the direction of a comparative, or its magnitude? How should these objectives be balanced in training? Our system introduces a hyperparameter β, to linearly trade off between the two. As the original authors do not report the tradeoff that they use, we sweep broadly in the range of $[0.30, \ldots, 0.7]$ and tightly in a window of $[0.4, 0.41, \ldots, 0.50]$. This second sweep is informed by the choice of β in the original paper. In actuality, neither of these objectives have perceptual value—we take $\mathcal{S}$ to be continuous RGB space, which does not align well to the just-noticeable differences between colors for humans. For this reason, we'd prefer an exact prediction of the target point—if we can achieve this, we can achieve it in any transformation of $\mathcal{S}$, such as into the CIE LAB coordinate system (Regier et al., 2007).

Other hyperparameters Winn and Muresan (2018) trained for 800 epochs. With fifteen million training records, this was untenable within the time constraints of the project. We run each model's training for this long, but we save periodic checkpoints. The checkpoint after 20 epochs is quite robust.

5 Results and Analysis

Table 2 shows the results of tuning the β hyperparameter, which controls the tradeoff between cosine similarity loss and MSE. We find that $\beta = 0.50$ performs the best of these settings.

Table 3 shows the results of full models trained with the hyperparameter settings used by Winn and Muresan (2018), varying only the batch size (2048 instead of their variable batch size). We see that without adjusting the other hyperparameter settings these networks fall short of their results.

Beta	average cosine
0.7	0.6116
0.6	0.5711
0.5	0.6264
0.48	0.6104
0.46	0.5965
0.44	0.6141
0.42	0.6211
0.4	0.6037
0.3	0.5925

Table 2: Average cosine distance on the test set for different settings of β, trained for 10 epochs with default model settings. For brevity, only the best hyperparameters from the narrow sweep are reported.

Model	average cosine
1-layer	0.5933
3-layer	0.6072

Table 3: Results replicating Winn and Muresan (2018) with 2048 batch size, as well as extending the architecture by adding additional layers.

6 Conclusion, Lessons, and Advice

We conducted an exploration of neural networks as a tool for comparative color grounding. We varied the dataset preparation, the network size, the network architecture, and the hyperparameters. (Granted, any others of these may be considered a hyperparameter in the broad sense!)

While our goal was to reproduce and extend the work of Winn and Muresan (2018), we were unable to train for their (amazingly high) 800 epochs. Shockingly, *we were able to come close to their cosine similarity with only 5% as many training epochs*. It also remains to be seen how effective a multi-layer word representation like ELMo () would be in representing color and yielding predictions—while ELMo has improved performance on a number of NLP tasks, it's possible that it would drown out signal from the reference color vector, which already must be fed through a highway connection.

The key takeaways from our work are familiar to anyone who watched the obliteration of statistical machine translation by neural techniques: (1) The gains come at a cost. Training time for a deep neural model is vastly higher than a statistical model trained with EM. (2) It's a mistake to take results for granted. As Reimers and Gurevych (2017) showed, it is possible for reported scores to be outliers. Further, as Durrett and Klein (2013) (and we) show, easy can be better than complicated—neural models can be overpowered. (3) (And this is a bit facetious—) Be good citizens of your computing cluster. Grids are shared resources, and there are strategies for minimally acquiring locks (familiar to any operating systems student) that can drastically improve latency for all users.

To future students, we encourage the philosophy which we ourselves practiced: Work early, and work often. It's better to make the journey in twenty manageable steps than to frantically cobble together a jury-rigged contraption and putting out seventy fires at once. To the course staff, we must—with all due respect—remind that teaching well is a labor and a duty, not an accident. A carefully structured course with tight feedback cycles will help the vulnerable students for whom this is a first exposure to deep learning. We were fortunate to have hands-on experience from our research; other students lack our advantage. Ignoring this fact exacerbates the incoming imbalance, creating a less inclusive and productive learning environment.

References

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982. Association for Computational Linguistics.

Paul Kay and Chad K McDaniel. 1978. The linguistic significance of the meanings of basic color terms. *Language*, pages 610–646.

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

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

Terry Regier, Paul Kay, and Naveen Khetarpal. 2007. Color naming reflects optimal partitions of color space. *Proceedings of the National Academy of Sciences*, 104(4):1436–1441.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348. Association for Computational Linguistics.

Olivia Winn and Smaranda Muresan. 2018. 'lighter' can still be dark: Modeling comparative color descriptions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 790–795. Association for Computational Linguistics.

Terry Winograd. 1971. Procedures as a representation for data in a computer program for understanding natural language. Technical report, Massachusetts Institute of Technology.