Learning Distributions of Meant Color

Anonymous ACL submission

Abstract

When a speaker says the name of a color, the color that they picture is not necessarily the same as the listener imagines. Color is a grounded semantic task, but that grounding is not a mapping of a single word (or phrase) to a single point in color-space. Proper understanding of color language requires the capacity to map a sequence of words to a probability distribution in color-space. A distribution is required as there is no clear agreement between people as to what a particular color describes – different people have a different idea of what it means to be "very dark orange". We propose a novel GRU-based model to handle this case. Learning how each word in a color name contributes to the color described, allows for knowledge sharing between uses of the words in different color names. This knowledge sharing significantly improves predicative capacity for color names with sparse training data. The extreme case of this challenge in data sparsity is for color names without any direct training data. Our model is able to predict reasonable distributions for these cases, as evaluated on a held-out dataset consisting only of such terms.

1 Introduction

Color understanding is an important subtask in natural language understanding. It is a challenging domain, due to ambiguity, multiple roles taken by the same words, the many modifiers, and shades of meaning. Due to its difficulty, texts containing color descriptions such as `the flower has petals that are bright pinkish purple with white stigma` are used as demonstrations for state of the art image generation systems (Reed et al., 2016; Mansimov et al., 2015). The core focus of the work we present here is addressing these linguistic phenomena around the descriptions of the color, in a single patch, as represented in a color-space such as HSV (Smith, 1978). Issues of illumination and perceived color based on visual context are considered out of the scope.

Consider that the word `tan` may mean one of many colors for different people in different circumstances: ranging from the bronze of a tanned sunbather, to the brown of tanned leather; `green` may mean anything from `aquamarine` to `forest green`; and even `forest green` may mean the rich shades of a rain-forest, or the near grey of the Australian bush. Thus the color intended cannot be uniquely inferred from the color name. Without further context, it does nevertheless remain possible to estimate likelihoods of which colors are be intended based on the population's use of the words. The primary aim of this work is to map a sequence of color description words to a probability distribution over a color-space. This is required for a proper understanding of color language.

Estimating color probabilities has a clear use as a subsystem in many systems. For example, in a human-interfacing system, when asked to select the `dark bluish green` object, each object can be ranked based on how likely its color is according to the distribution. This way if extra information eliminates the most-likely object, the second most likely object can immediately be determined. Further, if the probability of the color of the object being described by the user input is known, a threshold can be set to report that no object is found, or to ask for additional information. More generally, the distribution based on the color name alone can be used as a prior probability and combined with additional context information to yield better predictions.

Proper understanding requires considering *the color intended* as a random variable. In other words, a color name should map to a distribution, not just a single point or region. For a given color name, any number of points in the color-space could be intended, with some being more or less likely than others. Or equivalently, up to interpretation, it may intend a region but the likelihood of what points are covered is variable and uncertain. This distribution is often multimodal and has high and asymmetrical variance, which further renders regression to a single point, as has been done by Kawakami et al. (2016), unsuitable. We estimate a probability distribution over the color-space. To qualify our estimate of the distribution we discretize the space into a large number of patches, and produce an output much like a histogram. This allows us to take advantage of the well-known softmax based methods for estimating a probability mass distribution using a neural network.

This understanding of color language also requires a model capable of understanding linguistic compositionality. It must understand how modifiers such as `dark` modify basic colors; and how other modifiers such as `very` would interact as a modifier to modifiers. It also must understand the functioning of affixes such as `-ish` in `greenish`. This compositional understanding is needed both as a point of theory and practically. Practically, the generalisation ability from a compositional model allows it to handle color descriptions not seen in training. Due to the combinatorial nature of language, a data-sparsity problem exists: that for a large number of word combinations there are few examples in any given corpus. This is a well-known issue in n-gram language modelling (Kneser and Ney, 1995; Chen and Goodman, 1996; Rosenfeld, 2000). To handle this we take inspiration from a solution used in that area: the use of a recurrent neural network to process each color description as a compositional sequence of tokens (Mikolov et al., 2011). Processing per token allows for knowledge sharing between uses of the tokens in different terms, thus overcoming data sparsity programs (Bengio et al., 2003). Including, the extreme case of there being no direct training data at all.

The core contribution of this work is a novel method for estimating probability distributions over color-space for a color name, which is able to generalise to estimate distributions for color descriptions which are never seen during training. To handle distribution estimation we employ a discretization and blurring procedure. We define a GRU-based neural network to learn the compositional relationship from the term sequences describing the colors, overcoming the data-sparsity problem. We call this model the Color Distribution Estimation from Sequences of Terms (CDEST) model. As, to our knowledge, there is no existing work on estimating distributions from color-names, in order to evaluate the CDEST model we also define a histogram-based baseline method, which while lacking the generalisation capacity, more directly extracts the information from the training data.

2 Related Work

The understanding of color names has long been a concern of psycholinguistics and anthropology (Berlin and Kay, 1969; Heider, 1972; Heider and Olivier, 1972; Mylonas et al., 2015). It is thus no surprise that there should be a corresponds field of research in natural language processing.

The earliest works revolve around explicit color dictionaries. This includes the ISCC-NBS color system (Kelly et al., 1955) of 26 words, including modifiers, that are composed according to a context free grammar such that phrases are mapped to single points in the color-space; and the simpler, non-compositional, 11 basic colors of Berlin and Kay (1969). Works including Berk et al. (1982); Conway (1992); ele Lammens (1994); Mojsilovic (2005); Menegaz et al. (2007); Van De Weijer et al. (2009) which propose methods for the automatic mapping of colors to and from these small manually defined sets of colors. We note that Menegaz et al. (2007); Van De Weijer et al. (2009) both propose systems that discretize the color-space, though to a much courser level than we consider in this work.

More recent works, including the work presented here, function with much larger number of colors, larger vocabularies, and larger pools of respondents. In particular making uses of the large Munroe dataset Munroe (2010), as we do here. This allows a data driven approach towards the modelling.

McMahan and Stone (2015) and Meo et al. (2014) present color naming methods, mapping from colors to to their names, the reverse of our task. These works are based on defining fuzzy rectangular distributions in the color-space to cover the distribution estimated from the data, which are used in a Bayesian system to non-compositionally determine the color name. Monroe et al. (2016) maps a point in the color-space, to a sequence of distributions over color terms. They extends beyond, all prior color naming systems to produce a compositional color namer based on the Munroe dataset. Their method uses a recurrent neural network (RNN), which takes as input a color-space point, and the previous output word, and gives a probability of the next word to be output – this is a conditional language model. Our proposed CDEST model is the direct inverse of their conditional language model, CDEST use a RNN to map a sequence of color terms to a distribution over colors.

Kawakami et al. (2016) also propose a compositional color naming model. They use a per-character RNN and a variational autoencoder approach. It is in principle very similar to Monroe et al. (2016), but functioning on a character, rather than a word level. The work by Kawakami et al. also includes a method for generating colors. However it generates just single points, rather than distributions. This has significant limitations as discussed in Section 1, which our work attempts to overcome by modeling the distributions.

Monroe et al. (2017) presents a neural network solution to a communication game, where a speaker is presented with three colors and asked to describe one of them, and the listener is to work out which is being described. Speaker and listener models are trained, using LSTM-based decoders and encoders respectively. The final time-step of their model produces a 100 dimensional representation of the description provided. From this, a Gaussian distributed score function is calculated, over a high dimensional color-space from Monroe et al. (2016), which is then used to score each of the three options. While this method does work with a probability distribution, as a step in its goal, this distribution is always both symmetric and unimodal – albeit in a high-dimensional color-space. To the best of our knowledge no current work proposes as a distribution estimation system such as we describe in this paper.

3 Color Distribution Estimation Framework

We define two models for the estimation of colors from textual descriptions. A baseline histogram-based model and the GRU-based CDEST model. The baseline model estimates the distribution based on averaging the discretized observations of colors in the training set for each input color description. It cannot handle combinations of terms not seen during training as there is no data to average. The CDEST model relies on using machine learning to learn the relationship between words and the color distribution; and is trained on the same observations used in the baseline model. As it is learning a relationship between words and the color-space probability output, it can handle inputs made up of any words that were seen during training, even if the whole color description has never been used before. Both models rely on the same assumption of conditional independence, and the same method for discretization.

3.1 Conditional Independence Assumption

We make the assumption that given the name of the color, the distribution of the H, S and V channels are independent. That is to say, it is assumed if the color name is known, then knowing the value of one channel would not provide any additional information as to the value of the other two channels. The same assumption is made, though not remarked upon, in Meo et al. (2014) and McMahan and Stone (2015). This assumption of conditional independence allows considerable saving in computational resources. Approximating the 3D joint distribution as the product of three 1D distributions decreases the space complexity from $O(n^3)$ to $O(n)$ in the discretized step that follows.

Superficial checks were carried out on the accuracy of this assumption. Spearman's correlation on the training data suggests that for over three quarters of all color names, there is only weak correlation between the channels (Q3 = 0.187). However, this measure underestimates correlation for values that have circular relative value, such as hue. HSV had the lowest correlation by a large margin of the 16 color-spaces evaluated. Full details, including the table of correlations, are available in supplementary materials. These results are suggestive, rather than solidly indicative, on the degree of correctness of the conditional independence assumption. We consider the assumption sufficient for this investigation.

3.2 Discretization and Blurring

The core problem is to estimate a continuous probability distributions, conditional on the color name. Estimating a discrete conditional distributions is a significantly more studied application of neural networks – this is the basic function of any softmax classifier. To simplify the problem, we therefore transform it to be a discrete distribution estimation task, by discretizing the color-space. Discretization to a resolution of 64 and 256 bins per channel is considered.

Discretization to resolution n is the process by which a scalar observation[1] x from one of the continuous color channels (hue, saturation or value) is converted into an n-vector with the properties expected of a probability mass function. A naïve

[1] In the Munroe dataset, the provided HSV values are scaled to between 0 and 1 in all channels. We make use of this convention throughout this paper, and in our implementation.

approach is one-hot binning:

$$\Omega_n^{1hot}(x)=\left(\begin{cases}1 & \text{if } \frac{i-1}{n}<x\leq\frac{i}{n}\\ 0 & \text{otherwise}\end{cases}\right)_{i=1}^{i=n}$$

This gives an n-vector that is zero everywhere, except for the element corresponding to the patch of color-space that the value x lies within. Discretization in this way loses all notion of continuousness of the color-space. In truth the distribution in color-space is intrinsically continuous – this comes as a logical consequence of human color sensitivity being continuous (Stockman et al., 1999). Points near each other in the color-space should have similar probabilities of being the intended color for a color name. While discretization inevitably renders the space discrete, it is desirable to bring back this notion of smoothness as prior knowledge.

We enhance the training data by adding a blur during discretization. Consider $\mathcal{D}(\mu,\,\sigma^2)$ some unimodal distribution, characterised by having an expected value μ and a variance parameter σ^2. For saturation and value this is a truncated Gaussian. Hue can elegantly be handled using a wrap-around Gaussian. We write $P_{\mathcal{D}}(y_1{<}Y{\leq}y_2\,|\,M{=}\mu,\Sigma{=}\sigma)$ to mean the probability of a value distributed according to $\mathcal{D}(\mu,\sigma^2)$ being in the patch bordered by y_1 and y_2. Using this, the blurred-binning function is defined:

$$\Omega_n^{blur}(x,\mathcal{D},\sigma)=\left(P_{\mathcal{D}}\left(\frac{i-1}{n}<Y\leq\frac{i}{n}\,|\,M=x,\Sigma=\sigma\right)\right)_{i=1}^{i=n}$$

This function maps points x in the continuous color-space, to probability mass vectors of length n. The majority of the mass will be in the bin that the value x would be in, but some will be shared with the bins either side, and further.

By applying more or less blurring to the training data, the priority of smoothness v.s. exact matching is controlled. Considering the limits: for all $\mathcal{D}$ and values x: $\lim_{\sigma\to 0}\Omega_n^{blur}(x,\mathcal{D},\sigma)=\Omega_n^{1hot}(x)$, and $\lim_{\sigma\to\infty}\Omega_n^{blur}(x,\mathcal{D},\sigma)=\left(\frac{1}{n}\right)_{i=1}^{i=n}$ (uniform). A coarse parameter sweep on the value of σ was carried out using the development portion of the dataset (see Section 4.1). Best results were found for $\sigma=\frac{1}{2n}$. For a training point that would be at the center of a bin, this roughly corresponds to 68.3% of the probably mass assigned to the central bin, 15.7% assigned to adjacent bins, and the remaining 0.3% distributed to the remaining bins. All results presented here are for this level of blurring.

Discretizing the data is is a useful solution used in several other machine learning systems. Oord et al. (2016); van den Oord et al. (2016) apply a similar discretization step and found their method to outperforming the more complex continuous distribution outputs. These works did not employ a blurring-step. We found the blurring step to consistently improve results for all models during preliminary investigation using the development dataset. This is expected as a blurred discrete distribution captures some of the notions of continuity that a truly continuous output distribution would intrinsically feature.

We note that a truly continuous output is pragmatically unnecessary as 24-bit color (as was used in the survey) can have all information captured by a 256 bin quantization per channel. 24 bit color allows for a total of 2^{24} colors to be represented, and even 1 hot encoding for each of the 256 bin quantized channels allows for the same.

3.3 Baseline Model

While the main interest in this work is in compositionally modelling the color language, we also define a non-compositional baseline model to allow for comparison. This model loosely resembles the histogram model discussed in Meo et al. (2014) and McMahan and Stone (2015). Existing works do not aim to estimate a general distribution, and they are therefore unsuitable for comparison. Our baseline must be able to estimate multimodal and asymmetric color distributions.

The baseline is defined using the the element-wise mean of discretized training observations, with add-one smoothing. During our investigations we found that without the add-one smoothing the baseline would predict a probability of zero for some observations in the development dataset. Applying add-one smoothing to each output distribution solves this.

For the training data $V\subset[0,1]^3\times T$, where $[0,1]^3\subset\mathbb{R}^3$ is the scaled HSV color-space, and T is the natural language space. The subset of the training data for the description $t\in T$ is given by $V_{|t}=\{\tilde{v}_i\;|\;(\tilde{v}_i,\,t_i)\in V\wedge t_i=t\}$. Per channel $c{\in}\{H,S,V\}$ the baseline model is defined by:

$$q_c(x_c\,|\,t)=\frac{\sum\limits_{\forall(v_H,v_S,v_V)\in V_{|t}}\Omega_n^{blur}(v_c,\mathcal{D}_c,\sigma)\cdot\Omega_n^{1hot}(x_c)+1}{\left|V_{|t}\right|+n}$$

In this equation taking the dot-product with $\Omega_n^{1hot}(x_c)$ is selecting the bin containing x_c. Note the distinction between x_c and v_c: x_c is the point being queried, whereas v_c is a point from the training set. By the conditional independence assumption the overall baseline model is given by: $q(x_H,x_S,x_V\,|\,t)=\prod_{c\in\{H,S,V\}}q_c(x_c\,|\,t)$

The baseline model can be used to predict distribu-

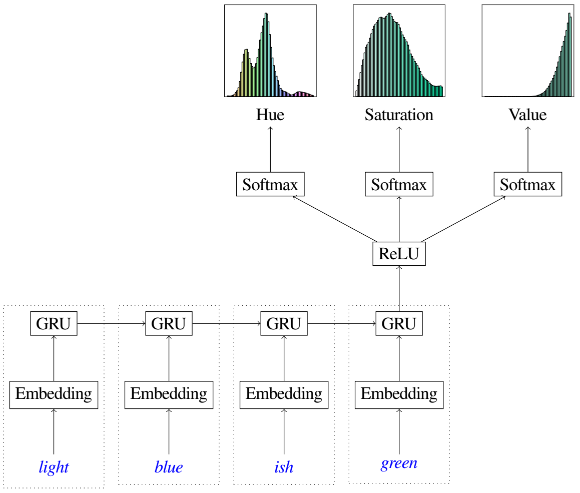

Figure 1: The CDEST model for predicting the color-space probability distributions of color. The section in the dotted-boxes is repeated for each time step.

tions for all color descriptions in the training set. This is inferior in generalisability to the CDEST model, which can handle any combination of tokens from the training set. We suggest that the baseline model is strong and reasonable. It is a much simpler modelling problem as it does not have a requirement to learn the how the multiple terms in the color name are compositionally combined. It directly captures the information from the training set. If the CDEST model can match its performance, that would at least show that it was capturing the information from the training data. If it can also have similar performance for cases that do require compositional understanding (see Section 4.2), that would show that it is indeed achieving the goal of properly modelling the language use.

3.4 CDEST Model

The CDEST model is an RNN which learns the compositional interactions of the terms making up a color description, to output a distribution estimate in color-space. The general structure of this network, shown in Figure 1 is similar to Monroe et al. (2016), or indeed to most other word sequence learning models. Each word is first transformed to an embedding representation. This representation is trained with the rest of the network allowing per word information to be efficiently learned. The embedding is used as the input for a Gated Recurrent Unit (GRU) (Cho et al., 2014). The output of the last time-step is fed to a Rectified Linear Unit (ReLU) (Dahl et al., 2013). Finally, the output of the ReLU is the shared input for three distinct softmax output layers – one for each of hue, saturation and value. These outputs are vectors $\hat{y}_H(t)$, $\hat{y}_s(t)$, and $\hat{y}_V(t)$. Using the conditional independence assumption the probability estimate is given by:

$$\hat{p}(x_H,x_S,x_V \mid t) = \prod_{c\in\{H,S,V\}} \hat{y}_c(t) \cdot \Omega_n^{1hot}(x_c))$$

As in the baseline model, the dot-product with $\Omega_n^{1hot}(x_c)$ serves to select the bin containing x_c.

The distinguishing features of this model compared to other word sequence learning models, is the use of GRU, rather than Long Short Term Memory (LSTM), and the three output layers.

We chose GRU as the basis of our reused structure in the recurrent network because it has fewer parameters to learn than the more established LSTM. It has generally been found to preform similarly well to LSTM (Chung et al., 2014); including on the color naming problem (Monroe et al., 2016). A component for processing per-term such as the GRU, is essential in allowing the model to learn the compositional function of each term, and thus to learn to handle color descriptions from outside the training set.

The three output layers are used to predict the discretized distributions for the three channels. Separating them like this requires a conditional independence assumption (see Section 3.1). The network is trained to minimize the sum of the three cross-entropy losses for these output layers. Similar multiple output layers as used in multitask learning (Caruana, 1997; Collobert and Weston, 2008). The layers prior to the output are shared, allowing common knowledge to be shared.

4 Experimental Setup

4.1 Data Preparation and Tokenization

We make use of the Munroe dataset as prepared by McMahan and Stone (2015) from the results of the XKCD color survey. The XKCD color survey (Munroe, 2010), collected over 3.4 million observations from over 222,500 respondents. McMahan and Stone take a subset from Munroe's full survey, by restricting it to the responses from native English speakers, and removing rare color names with less than 100 uses. This gives a total of 2,176,417 observations and 829 color names. They also define a standard test, development and train split.

In the dataset each observation is a textual color description, paired with a point in HSV color-space. We tokenized the textual color descriptions into separate words and affixes, using a short list of word replacement rules. Beyond simply breaking up a description `greenish blue` into words: `greenish` and `blue`, the suffixes `-ish` and

-y are also separated into their own tokens: green, ish, blue. Hyphens are also treated as their own tokens: blue-green becomes blue, -, green. The beginning and end of the color description is not demarcated with any form of marker token. Using this tokenization, each description is split into up to four tokens. This results in a total of 311 unique tokens used by the CDEST model. The baseline model does not function per token, and so uses the original 829 descriptions directly.

4.2 Extrapolation Sub-Dataset

The primary goal in constructing the CDEST model was for it to be able to to predict the distribution for never before seen descriptions of colors. For example, based on the learned understanding of salmon and of bright, from examples like bright green and bright red, our system can suggest the distribution in the color-space of bright salmon, even though that description never occurs in the training data. This would demonstrating proper compositional learning. To evaluate this generalisation capacity, we define an extrapolation sub-dataset. This is defined by selecting the rarest 100 color descriptions from the dataset, with the restriction that every token in a selected description must still have at least 8 uses in other descriptions. The selected examples include multi-token descriptions such as: bright yellow green and also single tokens that occur more commonly as modifiers than as stand-alone descriptions such as pale. The test and development datasets are restricted to contain only observations of these selected color descriptions. Conversely, the training set has no observations of these color descriptions. This produces a dataset suitable for evaluating the capacity of our model to estimate the distributions for color descriptions not seen in training. A similar approach was used in Atzmon et al. (2016).

4.3 CDEST Model Parameters

All hidden layers have width 128, except the embedding layer which has width 16. These values were found by a coarse search of the hyper-parameters using the development dataset with the output resolution being 64 bins. These parameters were also used for the 256 bin output resolution, though we suggest increasing the hidden layer size would give additional benefit for the higher output resolution case. During the hyper-parameter search, it was noted that the accuracy continued to improve as the hidden layer width was increased. However significantly diminishing returns in terms of training time v.s. accuracy lead us to limit the hidden layer sizes. Dropout (Srivastava et al., 2014) with a probability of 0.5 was used during training, on all hidden layers, except the embedding layer.

4.4 Perplexity in Color-Space

The perplexity allows us to evaluate how well our estimated distribution matches the distribution of the observations in the test set. Perplexity is commonly used for evaluating language models. However here it is being used to evaluate the discretized distribution estimate. It can loosely be thought of as to how well the model's distribution does in terms of the size of an equivalent uniform distribution. Note that this metrics does not assume conditional independence of the color channels.

Here τ is the test-set made up of pairs consisting of a color name t, and color-space point $\tilde{x}$; and $p(\tilde{x} \,|\, t)$ the output of the evaluated model. Perplexity is defined:

$$PP(\tau) = \exp_2\left(\frac{-1}{|\tau|} \sum_{\forall (t,(\tilde{x})) \in \tau} \log_2 p(\tilde{x} \,|\, t) \right)$$

As this varies depending on the output resolution, we define a standardized perplexity $\frac{PP(\tau)}{n^3}$, where n is the per channel output resolution of the model. The standardised perplexity allows us to compare models of different output resolutions. It is equivalent to comparing the relative performance of the model to that of a uniform distribution $PP_{uniform} = n^3$. Perplexity is a measure of how well the distribution, estimated by the model, matches reality according to the observations in the test set.

4.5 Implementation

The implementation of the CDEST and baseline models was in the Julia programming language (Bezanson et al., 2014). The full implementation is included in the supplementary materials.

5 Results and Discussion

5.1 Qualitative Comparison of the Distribution

Shown in Figures 2 to 4 are side-by-side comparisons of the output of the CDEST and the baseline models. Overall, it can be seen that the baseline model is has a lot more spikes, whereas the CDEST model tends to be much smoother, even though both use the same blurring during discretization. This smoothness is in line with the desired results from the model. As the bin boundaries are artificial and very narrow, it is not reasonable to expect that in reality viewers have such

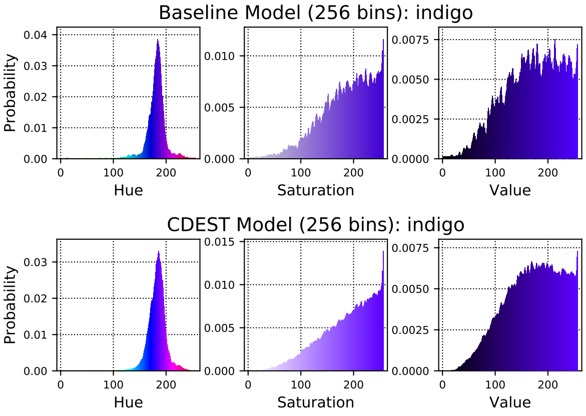

Figure 2: Distribution estimate for `indigo`

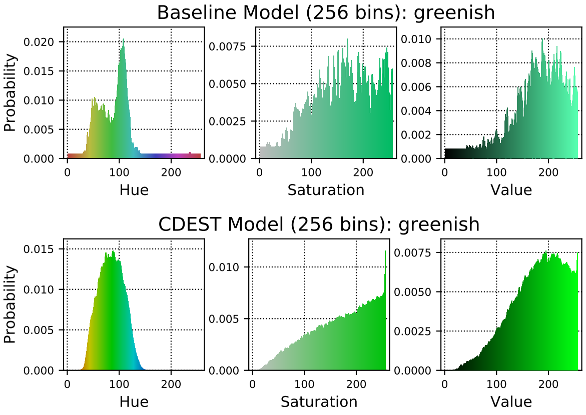

Figure 3: Distribution estimate for `greenish`

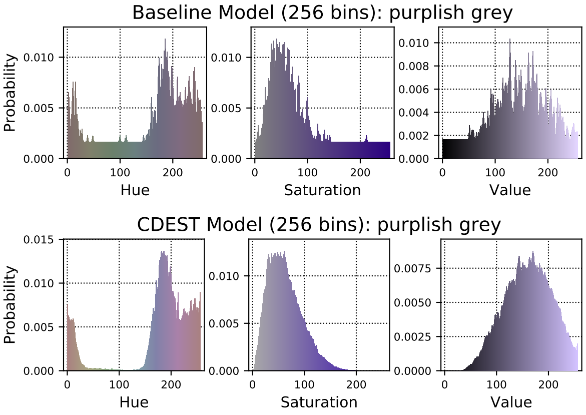

Figure 4: Distribution estimate for `purplish grey`

bands of colors that they think of as more connected to the color name than their neighbours. We expect (as discussed in Section 3.2) continuity, where adjacent points in color-space have similar probability values.

This smoothness can be taken too far, however, when it results in the filling-in of the area between peaks for multimodal colors. It can be seen that the CDEST model fails for some multimodal colors – such as the hue `greenish` (Figure 3 Hue) where the concave section is filled in; but succeeds for others such as `purplish grey` (Figure 4). We attribute this to the particular difficulty of `greenish` which functions very differently as a modifier vs as a standalone color, and suggest future models may benefit from tagging modifiers distinctly from the head-terms during preprocessing.

model	n	PP	$\frac{PP}{n^3}$
CDEST	64	20,200	0.077
Baseline	64	20,100	0.077
CDEST	256	1,210,000	**0.072**
Baseline	256	1,330,000	0.079

Table 1: The results of evaluation on the full Munroe dataset. Smaller $\frac{PP}{n^3}$ is better.

The horizontal bands in the baseline model outputs are the result of the add-one smoothing process. Notice that they are larger for colors with fewer examples – such as `purplish grey`. In the seminal work of Bengio et al. (2003) one of the motivations for employing neural networks in natural language processing was to better handle cases that do not occur in the training data, by sharing information between terms. CDEST efficiently applies the same core idea here for distribution estimation. The neural model of CDEST can, by knowledge sharing, better estimate the values for the unseen points in color-space, as compared to using smoothing. This is distinct from, but related to, its key capacity as a compositional model to handle unseen cases in the natural language space.

5.2 Direct Distribution Estimation

We first test the capacity of the model to estimate the distributions on the standard test dataset, using the standard training dataset. We perform this evaluation before the more difficult (and important) evaluation on the extrapolation task, to confirm that the models are capable of estimating distributions. The results are shown in Table 1. It can be seen that all models perform similarly. The CDEST model based on sequence of color tokens, reflects the real use of the color descriptions in the test set just well as the non-compositional baseline, which counts the exact uses of whole descriptions. This confirms that the CDEST model is able to learn to estimate a color distribution, and that the tokenization and sequential processing did not reduce the mapping ability of the model.

The CDEST model matches baseline performance, when trained on a full set of color terms with all com-

model	n	PP	$\frac{PP}{n^3}$
Extrapolating CDEST	64	20,400	0.078
Non-extrapolating CDEST	64	15,200	0.058
Non-extrapolating Baseline	64	18,100	0.069
Extrapolating CDEST	256	1,290,000	0.077
Non-extrapolating CDEST	256	851,000	0.051
Non-extrapolating Baseline	256	2,140,000	0.128

Table 2: The results of evaluation on the extrapolation sub-dataset. Smaller $\frac{PP}{n^3}$ is better.

binations of terms present in the training data. It seems there is little reason to use the CDEST model in this case, since the baseline model is simpler. However, the key advantage of the CDEST model is its ability to predict a distribution for an unseen combination of colors. This is evaluated using the extrapolation task.

5.3 Extrapolation to Unseen Color Names

A core motivation of using the CDEST model, is its ability to learn to combine tokens in a description in ways not seen in training. This demonstrates that the model is capable of learning the compositional effects of the tokens in the color name. That is to say learning how each token influences the final distribution – rather than simply memorising the training data, as is done in the case of the baseline.

When it comes to the extrapolation task, the best the baseline model can do is an uniform distribution as the color descriptions in the test set do not occur in the training set. This is an uninteresting comparison as it is always $\frac{PP}{n^3} = 1.0$ (and as such is not included in Table 2). Thus we look to comparing the results for extrapolation to the models when they are trained without the need for extrapolation.

We compare a CDEST model trained on the extrapolation sub-dataset, to the models trained on the full dataset. Both the non-extrapolating, and extrapolating models are evaluated on the same test set of rare color descriptions, but the non-extrapolating models are also shown these rare descriptions during training. The non-extrapolating models are expected to perform better given they have direct information on the rare color descriptions' distributions. The extrapolating model must use the knowledge of how those color terms influence the color distribution without direct training.

The results for this evaluation are shown in Table 2. As expected, the non-extrapolating CDEST outperforms the extrapolating CDEST. However, the decrease in performance when forced to extrapolate is relatively small. The extrapolation results are similar to the overall results from Table 1. These are good results, indicative that the model has learnt how the terms interact to define the color distribution. By training on uses of color terms in other descriptions the model learns these useful relationships and encodes them into the networks weights, such that when the terms are used new descriptions, the network can still estimate the distribution. This kind of learning allows knowledge sharing between color descriptions.

The non-extrapolating CDEST also benefits from the same knowledge sharing that enables the extrapolating CDEST model to function. This knowledge sharing allows it to outperform the baseline model, as the relationship between terms provides extra-data to better estimate the shape of the low-data curves. The baseline model does not have such knowledge sharing, thus has difficulties in estimating the curve of these rare descriptions. This is notable in the high resolution case (256 bin), where the sparsity of the training data is high enough to demonstrate the benefits of the know-sharing as shown by the extrapolating CDEST model outperforming the non-extrapolating baseline.

6 Conclusion

We have presented the CDEST model for estimating the probably distribution of colors that may be ascribed to an input name. For each input color name our model outputs a probability distribution over discrete regions of the color-space. Outputting a probability distribution, rather than a single point, allows for better handling of colors with observed distributions that are asymmetric, with high variance or which are multimodal in the color-space – which is the case for most colors.

The CDEST model learns the compositional structure of a color name, which allows it to predict distributions for color names which are not seen during training. As the it learns how each term influences the shape of the distribution, it can thus estimate a distribution for arbitrary compound color names, based on the learnt understanding of the individual terms. This allows it to excel when the sparsity of training data is high.

We find that the discretization process for representing the continuous probability distribution is pragmatically effective, but unsatisfying. While it is possible to simply fit a GMM or other continuous model to the final discretized output; in future work would investigate the extensions of works such as Magdon-Ismail and Atiya (1998); Likas (2001); Ambrogioni et al. (2017)..

References

L. Ambrogioni, U. Güçlü, M. A. J. van Gerven, and E. Maris. 2017. The Kernel Mixture Network: A Nonparametric Method for Conditional Density Estimation of Continuous Random Variables. *ArXiv e-prints*.

Yuval Atzmon, Jonathan Berant, Vahid Kezami, Amir Globerson, and Gal Chechik. 2016. Learning to generalize to new compositions in image understanding. *CoRR*, abs/1608.07639.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, pages 137–186.

Toby Berk, Arie Kaufman, and Lee Brownston. 1982. A human factors study of color notation systems for computer graphics. *Commun. ACM*, 25(8):547–550.

Brent Berlin and Paul Kay. 1969. *Basic color terms: Their university and evolution*. California UP.

Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. 2014. Julia: A fresh approach to numerical computing.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Stanley F Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Damian Conway. 1992. An experimental comparison of three natural language colour naming models. In *Proc. east-west int. conf. on human-computer interaction*, pages 328–339.

George E Dahl, Tara N Sainath, and Geoffrey E Hinton. 2013. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8609–8613. IEEE.

Eleanor R Heider. 1972. Universals in color naming and memory. *Journal of experimental psychology*, 93(1):10.

Eleanor Rosch Heider and Donald C. Olivier. 1972. The structure of the color space in naming and memory for two languages. *Cognitive Psychology*, 3(2):337 – 354.

Kazuya Kawakami, Chris Dyer, Bryan R. Routledge, and Noah A. Smith. 2016. Character sequence models for colorfulwords. *CoRR*, abs/1609.08777.

Kenneth Low Kelly et al. 1955. Iscc-nbs method of designating colors and a dictionary of color names.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

Johan Maurice Gis ele Lammens. 1994. *A Computational Model of Color Perception and Color Naming*. Ph.D. thesis, State University of New York.

Aristidis Likas. 2001. Probability density estimation using artificial neural networks. *Computer physics communications*, 135(2):167–175.

Malik Magdon-Ismail and Amir Atiya. 1998. Neural networks for density estimation. In *NIPS*, pages 522–528.

E. Mansimov, E. Parisotto, J. Lei Ba, and R. Salakhutdinov. 2015. Generating Images from Captions with Attention. *ArXiv e-prints*.

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

Gloria Menegaz, Arnaud Le Troter, Jean Sequeira, and Jean-Marc Boi. 2007. A discrete model for color naming. *EURASIP Journal on Applied Signal Processing*, 2007(1):113–113.

T. Meo, B. McMahan, and M. Stone. 2014. Generating and resolving vague color reference. *Proc. 18th Workshop Semantics and Pragmatics of Dialogue (SemDial)*.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan H Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

Aleksandra Mojsilovic. 2005. A computational model for color naming and describing color composition of images. *IEEE Transactions on Image Processing*, 14(5):690–699.

W. Monroe, N. D. Goodman, and C. Potts. 2016. Learning to Generate Compositional Color Descriptions. *ArXiv e-prints*.

Will Monroe, Robert X. D. Hawkins, Noah D. Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *CoRR*, abs/1703.10186.

Randall Munroe. 2010. Xkcd: Color survey results.

Dimitris Mylonas, Matthew Purver, Mehrnoosh Sadrzadeh, Lindsay MacDonald, and Lewis Griffin. 2015. The use of english colour terms in big data. The Color Science Association of Japan.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 3.

Ronald Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278.

Alvy Ray Smith. 1978. Color gamut transform pairs. *ACM Siggraph Computer Graphics*, 12(3):12–19.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Andrew Stockman, Lindsay T. Sharpe, and Clemens Fach. 1999. The spectral sensitivity of the human short-wavelength sensitive cones derived from thresholds and color matches. *Vision Research*, 39(17):2901 – 2927.

Joost Van De Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. 2009. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523.