Applying New Model Architectures to the Color Context Problem

Amber Yang
Stanford University
yanga@stanford.edu

Kasha Akrami
Stanford University
kakrami@stanford.edu

Vivek Kumar
Stanford University
vivkumar@stanford.edu

1 Hypotheses

Drawing from the pragmatic color descriptions problem first defined by (Monroe et al., 2016), the core hypothesis of this project is that the results of the color context problem can be improved upon by modifying the model architecture used. In the original color context model provided as an assignment, a Gated Recurrent Unit (GRU) is used as the basis for the encoder and decoder. We seek to test out the following hypotheses for this project:

1. The model performance can be improved by using a bidirectional LSTM for the encoder-decoder model.

2. Making changes, such as enhancing the tokenization scheme for the dataset used can improve model performance.

3. Applying transformer-based architectures, whether original or by leveraging existing, pre-trained transformer models like BERT can improve model performance.

2 Data

The dataset we use follows a standard structure that was overviewed in (Monroe et al., 2016). It is based on the Standard English Colors in Context Corpus (SCC). The corpus itself is based on a two-player interactive game, where a speaker is privately assigned a target color and asked to produce a description, and a listener aims to identify the speaker's target. In the data itself, each row is composed of three colors, with the target color being boxed and the other two colors being distractors. Next, each row of data could follow one of three conditions: far, split, close. In the far case, all three colors are far apart in color space. In the split case, the target color is close to one of the distractors. In the close case, the target color is similar to both distractors. As we did not want to pull all the colors from the SCC corpus, we used a provided module called "$torch\_color\_describer.py$ to create small, easy datasets.

3 Metrics

At its core, the color context problem is a classification problem. There is a speaker and a listener, and the listener is shown three different boxes of colors. The speaker preemptively knows what the target color is for the specific task and provides the listener with a textual description of what the target color looks like. The job of the listener is to then identify/classify what they think the target color is.

Taking the approach from (Monroe et al., 2016) to ensure a range of difficulty, we will randomly integrate an equivalent number of trials from the following three conditions: (1) *close*, where the three colors presented are within a distance θ from each other but perceptibly different, (2) *split*, where one color is a distance θ from the target color and the third color in the set is a distance farther than θ from the target, (3) *far*, where all three colors are farther than distance θ from each other.

The key metric that we will use to compare the various model architectures is the accuracy of the listener neural network since that represents the ultimate classification task. Since this is a classification task, we will create a confusion matrix for the listener neural network for each iteration of our model. This will provide us with a more holistic comparison of how the observed/gold labels compare to the labels predicted by the listener classifier.

We also plan to calculate the precision that can give us more information on a per-class basis. It is a straight-forward extension to then take the accuracy and precision to calculate the F score

(another per-class metric). It will also be insightful to calculate an overall accuracy, precision, and F score, but also those three metrics for each of the three trial conditions mentioned earlier. This way, we can analyze which conditions the model struggles more with, if any.

Furthermore, we plan to include a BLEU score to ensure that the system is speaking English.

4 Models

The heart of the model we are using to solve this problem is a standard encoder–decoder model. Encoder processes the color contexts as a sequence where as Decoder is a neural language model whose initial hidden representation is the final hidden representation of the Encoder.

For the baseline, we are using GRU based encoder decoder as defined in the *colors_overview.ipynb*. The baseline Encoder and Decoder have only one hidden layer. As a first next step, we propose to make it a multi layer model for both Encoder and Decoder of the same depth. The next set of models in our exploration involves using multi layer, bidirectional LSTM instead of GRU. We expect to benefit from bidirectional nature of model to capture information from both sides: left to right and right to left. At the same time, by adding more depth to layers, we want to allow the model to fit to more complex functions. On top of bidirectional LSTM, We plan to experiment with multiple parametrs of model like embedding size, hidden size, and number of hidden layers driven by desire to achieve better score and determine optimal model complexity for the SCC dataset.

Additional model that we will be investigating is transformer based pretrained model like BERT. The focus of our work is to understand if it is possible to adapt these models with fine tuning and whether or not it performs better than sequence to sequence models like LSTM and GRU for the task at hand. One more experiment involves analyzing the effect of using only [CLS] output token or mean of all the tokens in final output layer.

5 General Reasoning

First step is to tokenize the input by converting it to lowercase, followed by removal of whitespace, punctuations and performing sub-word level segmentation at the end. For eg: an input string *'Darker bluish!'* tokenize to {*'dark', 'er', 'blu', 'ish', '!'*}. Tokenized input is also added with tokens of start and end symbol.

Further, we improve the color representation by first transforming HLS format to HSV format and apply Fourier transform method of (Monroe et al., 2016) and (Monroe et al., 2017). We create a GloVe embedding based on the model vocabulary on a sub-word level. Using Glove embeddings, encoder generates the output and hidden states for input color_seqs and word_seqs. For decoder, the input vector to the model at each timestep is concatenation of input token representation with the representation of the target color. Encoder extracts the target colors from color_seqs and feed them to the decoder to predict new sequences based on the color contexts in the input color_seqs. Evaluation metric of listener accuracy and BLEU score is evaluated on a dev dataset to determine the course of model improvement.

6 Summary of Progress

So far, we have been refining our previous implementation of the Colors HW problem by figuring out the problem make and usage of the bidirectional LSTM. As we had known, this was to be the major obstacle for our progress. The bidirectional encoder inputs and outputs a Numpy array that is not intuitive in size. Consequently, our team has spent a great deal of time going through the math of figuring what our input size to both bidirectional LSTMs would be and what the consequent output/size of the output would look like.

Another element of our progress has been fine tuning our hyperparameters. Thankfully on this front, implementation has been less mathematically intense and instead followed trial and error testing. For this, we are following a set of strong literature to assist in our fine-tuning. This includes (Reimers and Gurevych, 2017) paper. Notably, the most useful hyperparameters we have experimented with so far include the "dropout" and "num_layers" of LSTM.

One obstacle we initially worried about but were able to address early on was ambiguity with what our data and how this would impact our inputs to LSTM. We were purposeful in using the same dataset that was given in the original Colors HW assignment. Not only were we familiar with

this dataset, but we it's composition was one that made sense to us (again, our reasoning for this is explained in section 2. Data).

References

Will Monroe, Noah D. Goodman, and Christopher Potts. 2016. Learning to generate compositional color descriptions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2243–2248, Austin, Texas. Association for Computational Linguistics.

Will Monroe, Robert Hawkins, Noah Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5(0):325–338.

Nils Reimers and Iryna Gurevych. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks.