A Linguistic Approach to Categorical Color Assignment for Data Visualization

Vidya Setlur, *Member, IEEE*, Maureen C. Stone, *Member, IEEE*

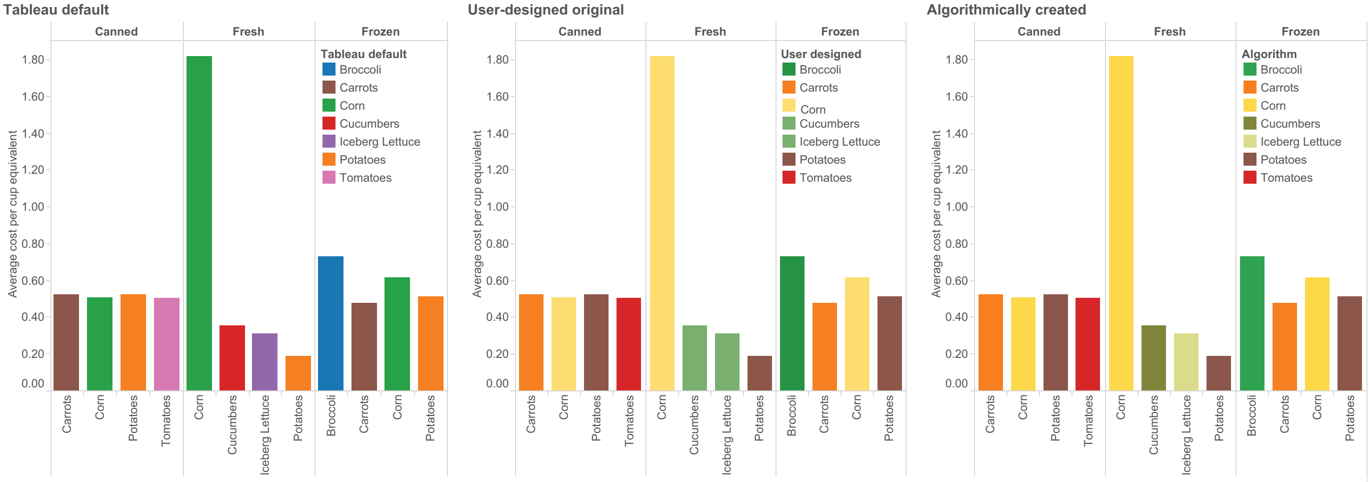


Fig. 1. This visualization was taken from a Tableau Public workbook [11] to illustrate the value of semantic color encoding. Left: The Tableau default colors are perceptually legible, but conflict with the data semantics ('Tomatoes' are pink, 'Corn' is green). Center: The Tableau author matched the colors to the data semantics (red for 'Tomatoes', yellow for 'Corn'), which makes it easier to identify the different types of vegetables in the graph. Right: Our algorithm automatically created a similarly effective result.

Abstract—When data categories have strong color associations, it is useful to use these semantically meaningful concept-color associations in data visualizations. In this paper, we explore how linguistic information about the terms defining the data can be used to generate semantically meaningful colors. To do this effectively, we need first to establish that a term has a strong semantic color association, then discover which color or colors express it. Using co-occurrence measures of color name frequencies from Google n-grams, we define a measure for colorability that describes how strongly associated a given term is to any of a set of basic color terms. We then show how this colorability score can be used with additional semantic analysis to rank and retrieve a representative color from Google Images. Alternatively, we use symbolic relationships defined by WordNet to select identity colors for categories such as countries or brands. To create visually distinct color palettes, we use k-means clustering to create visually distinct sets, iteratively reassigning terms with multiple basic color associations as needed. This can be additionally constrained to use colors only in a predefined palette.

Index Terms— linguistics, natural language processing, semantics, color names, categorical color, Google n-grams, WordNet, XKCD

1 Introduction

Consider the barchart in Figure 1, where the colors label different types of vegetables. The first coloring is a well-designed default categorical palette, with colors that are optimized for legibility and mapped to basic color names. While perceptually legible, there is no semantic relationship between the colors used in the visualization and those commonly associated with these data. To find the bars associated with 'Corn', the viewer needs to first find 'Corn' in the legend, discover that it is green, then remember this while looking at the visualization. In contrast, the other two colorings apply a semantic coloring defined by typical colors for these data items. 'Corn' is yellow, 'Tomatoes' are red, etc., which makes the encoding easier to discover and remember. One example was designed by the author of the visualization, who felt strongly enough that this was important to hand-color 63 different vegetables in the associated dataset. The other was automatically generated by the algorithms described in this paper. The goal of this research is to aid in the semantic mapping of coloring to data, both by presenting a specific technique and by discussing the challenges and trade-offs discovered in this work.

The importance of semantic coloring depends on the domain. A visualization of Crayola colors [3], for example, needs the color names and color values to match. To do otherwise would be very confusing, potentially creating cognitive interference similar to the Stroup Effect [33]. Objects with strongly associated colors, like fruits, vegetables, political parties, and brands also benefit from semantic coloring. In contrast, a chart of sales performance colored by region or sales team has no inherent semantic coloring. An automatic system for assigning semantic coloring, therefore, must first determine the colorability of the objects in the category, then determine the appropriate coloring. In this paper, we focus on coloring for categorical labeling so we must also consider the problem of creating a palette of distinctly different colors for a set of objects, not just colors for individual terms. Finally, semantic coloring is often defined by the context. For example, 'apple' as a fruit is red or green, but 'apple' as a

- *Vidya Setlur is with Tableau Research. E-mail: vsetlur@tableau.com.*
- *Maureen Stone is with Tableau Research. E-mail: mstone@tableau.com*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx Aug. 2015; date of current version 25 Oct. 2015. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

brand is white or silver gray. Our approach, which is based on natural language processing (NLP), provides a way to cleanly discriminate these cases.

This work makes the following contributions:

1. We define whether a term is colorable by modeling its semantic co-occurrence with the Berlin & Kay basic color names using the Google n-grams corpus [6, 34]. This extensive corpus contains English word n-grams (from uni-grams to 5-grams) and their observed frequencies calculated over one trillion words from web page text, creating a rich vocabulary for harnessing language usage of words that could be associated with color. The results of this analysis are a colorability score and a set of basic color names strongly associated with the term.

2. We describe a new method for finding an appropriate color value by querying and analyzing Google images. Unlike prior work, this method uses the basic colors associated with the term along with further linguistic analysis of the term to refine the query.

3. We present an alternative query method that uses semantic context such as 'brand' or 'country' to refine the image search. This makes 'apple' red or silver depending on its context.

4. For palette generation, we use *k*-means clustering in CIELAB colorspace to define a set of visually distinct colors. The result is a set of colors returned by the image search and modulated by the clustering. Similar to other work in data palette generation [29], we can leverage the fact that many terms are associated with multiple basic colors to avoid assigning similar colors to different terms. For example, since 'apple' is associated with red, yellow and green, we can assign green to apple to avoid assigning red to both 'apple' and 'cherry'. We can also constrain the clustering algorithm to match a pre-defined palette.

We start with a discussion of previous work, with special attention to the work by Lin *et al.* [29]. While there is extensive literature on color names and semantic color association, this is the work most similar to ours because it is specifically about categorical palettes for visualization. We then discuss the various steps in our process: The colorability metric, mapping names to colors, both with and without semantic context, and palette generation. We finish with a discussion and future work.

One challenge in this type of work is validation. There is no single answer for the 'best' semantic color, as both language and color are highly contextual. We validate our results both by direct comparison to Lin *et al.*'s work and by supplying visualization examples from the Tableau Public website [11]. In addition, we analyze a large crowd-sourced color name data set that maps names to colors, which is publicly available on the XKCD website [37].

2 Related Work

2.1 Color Names and Cognition

Strong links between color and language are well established, and there are literally thousands of research papers on the topic. One vector of research focuses on basic color names, studying how these names are represented in different languages and cultures. Berlin & Kay's studies on basic color terms are considered foundational in this area [13], and their extension to the World Color Survey [26]. Of relevance to this paper, it establishes that for English (and many similar European languages), there are 11 basic color names: red, green, blue, yellow, orange, purple, pink, brown, black, white and gray. Furthermore, there is strong agreement on the color stimuli, in this case, defined by a set of Munsell color chips that are exemplars of these names.

That there is a fundamental link between language and color cognition is demonstrated by the Stroop effect [33]. When the color of the letters is in conflict with the color described by the word (for example, the word 'blue' is written in red letters), subjects find it harder to name the color of the letters. A similar but smaller effect was found for words with strong color associations, such as 'blood' and 'grass' [19]. This is recognized as interference between the color term evoked by reading (language) and the color name associated with the perceived color.

In visualization, the work by Lin, Fortuna, Kulkarni, Stone and Heer [29] experimentally demonstrated that semantically naming the categorical colors used in a visualization improved cognitive performance. They hypothesize that at the minimum, viewers can more easily memorize the categories, reducing the need to access the legend. We use this work as the most concrete justification for working on semantically-resonant coloring for visualization. To run their experiments, they selected 40 categorical value sets and evaluated their colorability using crowd-sourced experiments. We use their same datasets as validation for our colorability algorithm. They also asked their subjects to create an appropriate palette for these datasets, choosing only colors from a fixed set of expertly designed colors (Tableau 20). In some cases, they also asked an expert to create an appropriate palette. We applied our palette generation algorithms to the same data and find that our results compare favorably with both the crowd-sourced and expert results.

2.2 Color and Language

Language has a long history of resources for describing color. Firstly, a word has a strong association with color, especially when color is a salient feature of the concept it refers to (for example, 'sky' (blue), 'lemon' (yellow)) [18, 27]. Secondly, color names pertaining to pigments and dyes are often derived from the source, such as 'indigo', from the Indigofera Tinctoria plant. Thirdly, many languages have morphological and syntactic processes that create complex color terms out of simple color terms (for example, 'blue-green', 'yellowish', and 'pale pinkish purple'). Finally, many linguistically simple terms that denote subtypes or 'shades' of colors are denoted by other terms. For example, 'scarlet', 'crimson', 'vermillion', 'puce', 'burgundy', and 'maroon' are among the more commonly named shades of red [14]. This suggests that linguistic data sources that consider the semantics of color names might provide for better reference, selection and retrieval of colors for various tasks, including for categorical palettes in data visualizations.

In the 1950s, the U.S. National Bureau of Standards created a color naming dictionary [28] to both define a standard set of color names by partitioning the Munsell color solid [38] and to create a dictionary of commercial color names, defined in terms of their Munsell specification and their standard name. While this standardization effort was not widely adopted, the use of color names to represent colors in commerce continues today. Go to any paint store and the color samples are labeled with both a technical code for constructing the paint and a semantic name, designed for some combination of descriptiveness and memorability. The Sherwin-Williams red collection provides examples such as 'Positive Red', 'Eros Pink', 'Radish', 'Brick' and 'Coral Bells' [10]. HTML 5 supplies 140 color names, from the basic (Blue, Red DarkGray) to the exotic (Chocolate, Chartreuse, BurlyWood) [8]. The power of these names is not their accuracy, but their memorability and ease of use. It is literally impossible to discuss color without naming it in some way. 'DarkBrown' or even 'Chocolate' is easier to remember and specify than the associated hex code.

Havasi, Speer and Holmgren [21] used a crowd-sourced semantic net called ConceptNet to associate words and colors, both for objects and concepts. Like our work, they link names with colors using association with the Berlin & Kay basic names and perform validation with the XKCD database. While similar conceptually, their technology is quite different. While we use n-gram analysis to determine the relationship between terms and basic colors, they use the links provided in ConceptNet. They do not use images to determine colors, but depend instead on finding color centroids semantically. In addition, their focus is on returning one ideal color, not a set of color options. Therefore, for 'apple' they would return only red.

Lindner, Bonnier, and Süsstrunk [30] automatically map semantic expressions to colors, including a colorability score they call a "significance metric.” They use annotated images from the Flickr online image sharing community to build the semantic relationship, mining the annotations to find images that correspond to the expression to be colored. They then generate a 3-dimensional histogram of the colors in the selected images These are constructed such that peaks correspond to image sets with a strong semantic coloring, which can be used to define the related color. Our algorithm uses Google n-grams and basic names to establish colorability, then searches images using the associated basic names plus additional semantic information to find appropriate coloring.

In Lindner and Süsstrunk [32] the authors use Google n-grams to generate a large vocabulary of frequently-used words, then download 60 related images using Google Image Search. They apply the same image analysis as their previous work to create characteristic colors for the term. Using “hue templates” they create color palettes associated with the term, which they compare to those found in Adobe Kuler [1]. These are not data palettes, with a semantic coloring for each category, but rather design palettes evoked by the specified term.

2.3 Mapping Names to Colors

A common way to map names to colors is to fit statistical models to human judgements of color-name associations. The data for these models is created by showing people color patches and asked to name them, sometimes freely, and sometimes with a constrained vocabulary. The oldest and most extensive online color naming survey was established by Nathan Moroney in 2002, and is still collecting data [36, 35]. Our work uses the model created by Heer & Stone [22], which uses the data created by XKCD author Randall Monroe [37]. The problem with such databases for categorical color association is that the terms in the database are predominantly descriptors of color (for example, ‘light blue-green’) rather than terms common to categorical data. Categorical data typically does not directly describe colors, but are rather concepts with a strong color association. Therefore, we need a way to map an arbitrary word or phrase to a color.

Our work and Lin *et al.* address this problem by searching Google images. Both algorithms leverage the simpler and more semantic coloring of clipart. Our algorithm, however, uses the basic color names returned by the colorability score in the query and query expansion of the term to restrict our search to images that are relevant and already have the color of interest. Lin *et al.* map directly from the term to images, and must do further processing to cull out irrelevant images and find the dominant color. We further refine the query with additional semantic information to define the context. The value of our NLP approach makes these sort of refinements straightforward.

Work by Lindner *et al.* [31] uses Google Image search to generate a multi-lingual color thesaurus. They start with a set of color terms from the XKCD dataset translated into multiple languages, but use Google Image Search to define the associated colors. They use the annotations on the images to create the linguistic link, and apply an additional restrictions to match the different languages. They use a 3D histogram to generate the key color for each image. One problem with their approach is that a word may have several different colors associated with it. Instead of distinguishing these cases, they combine them to get a sometimes surprising result.

In computer vision, Van De Weijer *et al.* [43] use the Berlin & Kay basic terms with Google images to find representative pictures of colored objects. The work of Schauerte & Stiefelhagen [41] expands on their results. Their goal is to identify colored objects in an image, such a ‘red car’, to help with search and object identification. This requires finding pixel colors that are most likely to represent a red car in an image, which is quite different from using images to generate a good categorical representation. Instead of clipart with representative colors, their work finds natural images more useful.

3 N-Gram Colorability Measure

To provide a color for a categorical value, we need a way to determine how strong the value’s association is to a color or a set of colors. For example, terms such as ‘mint’ and ‘rose’ are more strongly associated with colors than school subjects such as ‘math’ and ‘science.’ In addition, the same term may have multiple color values such as ‘red’ and ‘green’ for Christmas. Therefore, we need to determine whether a semantically meaningful color can be assigned to a term, and if so, what the basic color could be.

Using Google n-gram co-occurrence of the data term with the basic color name, we compute a *colorability score* that reflects the probability that a basic color described by Berlin and Kay [13], is associated with the term. We validate this colorability score by applying the score to analyze the XKCD dataset.

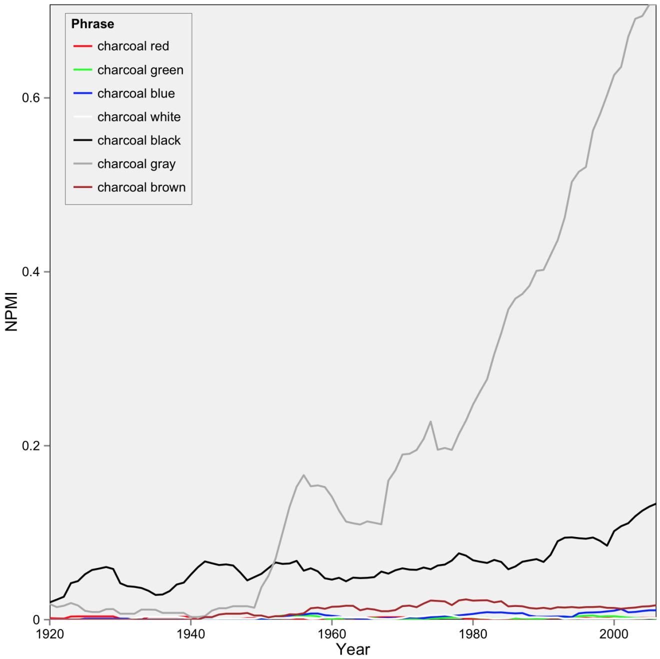


Fig. 2. Normalized NPMI values for ‘charcoal’ co-occurring with each of the Berlin & Kay terms between the years 1920 and 2006. We consider NPMI scores ≥ 0.5 for computing the final colorability score of a term. Here, you can observe that ‘charcoal’ and ‘gray’ co-occur most frequently, with an overall colorability score $S_{NPMI}(charcoal) = 0.7512$.

3.1 Constructing N-Gram Terms and Phrases

An n-gram is a contiguous sequence of *n* items from a given sequence of text or speech. Searching a corpus of text for the n-gram returns a number that indicates how often that sequence appears in the corpus. We apply an unsupervised corpus-based approach for computing color relatedness [23, 24]. To have sufficient coverage of co-occurrence of term, we use an extensive corpus with co-occurrence statistics called Google n-grams [34]. The Google n-grams dataset is a publicly available corpus with co-occurrence statistics of a large volume of web text containing n-grams (from uni-grams to 5-grams) and their observed frequencies calculated over one trillion words from web page text.

The data terms in the XKCD database vary from single words (‘blue’, ‘green’, ‘indigo’, ‘ochre’) to multi-word phrases (‘sky blue’, ‘light yellow ochre’, ‘faded indigo’). Each term is combined with each of the 11 basic color names in turn, where the names are: red, green, blue, yellow, black, white, gray, orange, purple, pink, brown. (For example, ‘faded indigo red’, ‘faded indigo green’, and so on.) The basic n-gram analysis simply returns a value proportional to the number of occurrences. Terms that are strongly associated with colors return a higher score than those that are not. An example is shown in Figure 2, ‘charcoal gray’ is most common, followed by ‘charcoal black’, but ‘charcoal yellow’ is not found.

There are some special cases included in the XKCD analysis. If the phrase ends in a basic term (For example, ‘charcoal gray’), the term is not repeated. That is, the values analyzed will be ‘charcoal gray’ not ‘charcoal gray gray’. However, the test with red will be ‘charcoal gray red’. Many of the XKCD terms are basic color terms like ‘blue’ and ‘red’, derivatives of basic color terms with descriptors such as ‘dark green’ or ‘pale pink’, or compound terms such as ‘purplish pink.’ However, there are also entities in the dataset that elicit strong color associations such as ‘lemon’, ‘cornflower’, ‘blood’ and ‘chocolate.’ If a part-of-speech analysis of the phrase can identify an object, and that object does not already exist in the database, that object is also paired with each of the basic color terms. For example, ‘charcoal gray’ has the object ‘charcoal’ so both ‘charcoal gray blue’ and ‘charcoal blue’ are tested, and charcoal is added as term to the database.

We then compute a colorability score on the n-grams from the corpus, to form probability estimates as described in the following section.

3.2 Measuring Term Color Co-Occurrence

We employ a Pointwise Mutual Information Measure (PMI) [17], an information-theoretic measure that quantifies the probability of how tightly occurring a given term and a Berlin & Kay color term are to the probability of observing the terms independently.

The PMI of a term t with one of the Berlin & Kay basic color terms t_{color}, is defined as:

$$PMI(t, t_{color}) = log \frac{p(t, t_{color})}{p(t)p(t_{color})} \quad (1)$$

It should be noted that $PMI(t, t_{color})$ is not symmetric ($PMI(t, t_{color}) \neq PMI(t_{color}, t)$) as $PMI(t, t_{color})$ and $PMI(t_{color}, t)$ represent two different n-gram co-occurrence events.

To measure the strength of the association, we compute the PMI over a localized context (PMI_{local}) amongst all the terms occurring in the book corpora in the Google n-grams dataset. This localized PMI function is as follows:

$$PMI_{local}(t, t_{color}) = \frac{1}{2C} \sum_{C=1}^{para} PMI(t, t_{color}) \quad (2)$$

where C is the contextual window size. C is computed by a distance vector measuring the number of salient terms (not stop words) between the origin (original term t) and the other term t_{color} [39]. For example, $C = 1$ for t and t_{color} being adjacent to each other. We constrained the maximum context window to be a paragraph of text in the n-gram corpus, *i.e.* we ignored any co-occurrences of the two terms that are occurred outside a single paragraph context. This tends to minimize the number of false positives and the sensitivity of the PMI score w.r.t. larger context window sizes.

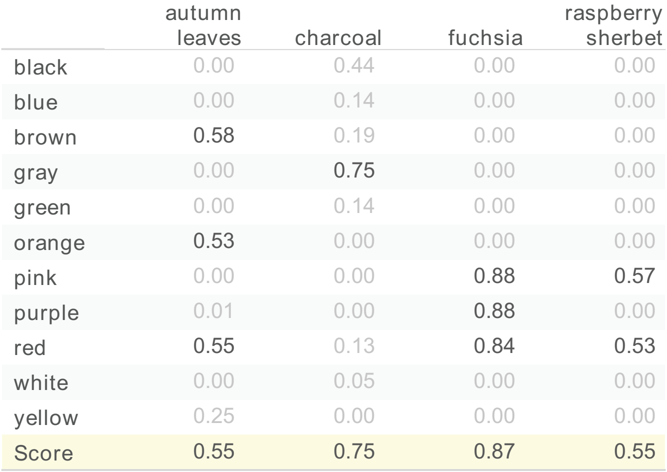
We then normalized each localized PMI score as follows:

$$NPMI(t, t_{color}) = \frac{PMI_{local}(t, t_{color})}{log[p(t, t_{color})]} \quad (3)$$

This results in a probability score between 1 and 0, with 0 for never occurring together and 1 for complete co-occurrence. Figure 2 shows the individual $NPMI$ scores for ‘charcoal’ plus the 11 Berlin & Kay color terms.

The causal link between language use and the statistical patterns of co-occurrence is not necessarily linear, and the Google n-gram corpus often reflects shifts in human language and cultural usage over time [34]. For example, many of the art pigments such as ‘Geranium Lake’ were popular in the late 1800s and early 1900s, but then rapidly declined in usage in recent years. However, the term can still be identified to be associated with color. Since we are more interested in how colorable a given term is and less about its lexical usage at any given time, rather than averaging the PMI scores of an n-gram over time, we ignore any low co-occurrence values, and preserve peaks of high values of co-occurrence in any given year. In practice, we found that a threshold of 0.5 tends to minimize any co-occurrence noise with the term and preserve the more salient color associations.

Once we compute the individual $NPMI$ values for a term with every one of 11 Berlin & Kay basic color terms, we compute the overall colorability score S_{NPMI} of that term t as follows:



	autumn leaves	charcoal	fuchsia	raspberry sherbet
black	0.00	0.44	0.00	0.00
blue	0.00	0.14	0.00	0.00
brown	0.58	0.19	0.00	0.00
gray	0.00	0.75	0.00	0.00
green	0.00	0.14	0.00	0.00
orange	0.53	0.00	0.00	0.00
pink	0.00	0.00	0.88	0.57
purple	0.01	0.00	0.88	0.00
red	0.55	0.13	0.84	0.53
white	0.00	0.05	0.00	0.00
yellow	0.25	0.00	0.00	0.00
Score	0.55	0.75	0.87	0.55

Fig. 3. Table showing average $NPMI$ scores for a word or phrase with each of the 11 Berlin & Kay basic color terms. The final colorability score is an average of the individual $NPMI$ values $>= 0.5$. We observe that ‘charcoal’ has a high association with gray. Similarly, red, pink and purple for ‘fuchsia’; red, orange and brown for ‘autumn leaves’; red and pink for ‘raspberry sherbet.’

$$S_{NPMI}(t) = \frac{1}{n} \sum_{i=1}^{n} NPMI(t, t_{color}) \quad (4)$$

where n is the total number of $NPMI$ values ≥ 0.5. The table in Figure 3 shows the individual and combined scores for the terms ‘fuchsia’, ‘charcoal’, ‘autumn leaves’ and ‘raspberry sherbet’.

3.3 Visualizing the Colorability Score

The XKCD color naming dataset is a set of name-color pairs collected online from 152,401 users (103,430 self-reported males, 41,464 females, and 7,507 declined to state). Filtering for spam responses creates 3,252,134 color-name pairs spanning 2,956,183 unique RGB triples and 132,259 unique color names. Previous work by Heer & Stone [22] reduced this set to 146 color terms and provided a model for transforming names into colors, and colors into names. We used this XKCD corpus of color names to validate our colorability score.

We started by lightly pruning the spam-filtered database by removing non-alphabetic characters, stop words, performing spelling correction, and stemming plurals. We then applied our algorithm to the remaining 114,369 terms and created a dataset of all terms whose score was greater than or equal to 0.5. This dataset has 6,540 unique color terms in it, their n-gram score, and the individual scores for each basic name. In addition, it indicates which terms were retained in Heer & Stone’s reduction of this same data. We analyzed and visualized this data using Tableau, which made it possible to inspect the results of the scoring for all the retained terms. We looked to see that the colors selected by Heer & Stone’s analysis got high scores, that the top scoring names were plausibly highly colorable, and that the basic colors ranked highly for each term seemed appropriate.

We created plots mapping terms to the n-gram score, as shown in Figure 4. As expected, there are a small number of highly colorable terms and a long tail of moderately colorable ones. In this plot, the minimum n-gram score is 0.6, which gives us a plot of 900 rows. The terms selected by Heer & Stone are highlighted (orange marks), and we can see them to be scored highly colorable. The lowest scored term in Heer & Stone is ‘bright yellow’ with a colorability score of 0.723. The highest scoring name not in their dataset is ‘olive green’, with a score of 0.854.

Figure 5 shows the top 20 colors with their scores broken down by each of the 11 basic color names. That is, each bar indicates the strength of the association between the term and a specific basic color name. The color of the bar matches the basic name, its length and

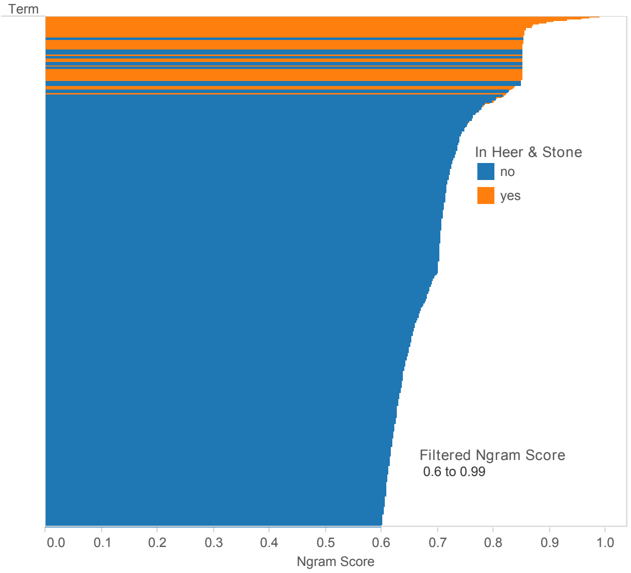


Fig. 4. The top 900 terms in the XKCD dataset (vertical axis) vs. their colorability score. At this scale, the term names are too small to display. Orange indicates that the term was in the Heer & Stone set.

label indicate the score for that term. For each term, all basic names with scores over 0.5 are shown. As expected, most of the basic names are included here and associate only with their own color name, *i.e.* blue with blue, green with green, etc. The terms 'turquoise' and 'teal', however, are associated with both blue and green, 'lime green' with yellow and green, and 'fuchsia' with red, green and pink.

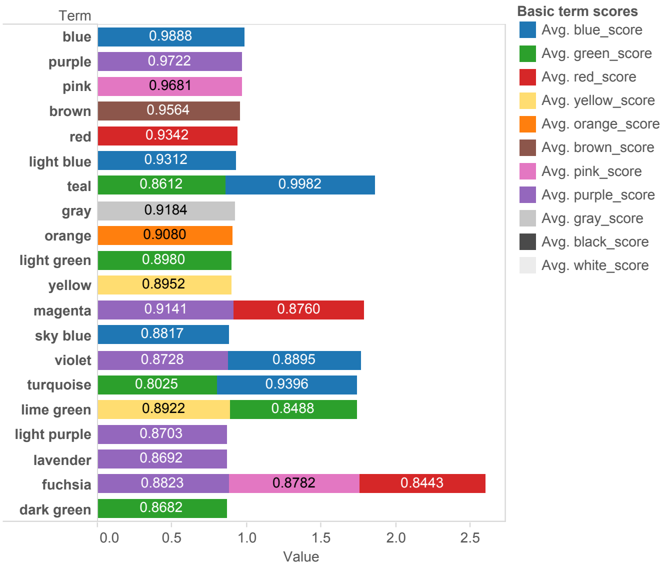


Fig. 5. Top 20 colors and the scores of the associated basic name. The color of the bar indicates the name, the length and label are the colorability with respect to that name.

We found it useful to categorize the terms by their terminal word, which was typically either a simple color term or an object. This additional structure makes it easy to see, for example, all the phrases that end in 'apple', then to evaluate how well the n-gram algorithm has identified the corresponding basic colors. Figure 6 was chosen to show different types of results. Some terms have a single corresponding color ('amethyst', 'anthracite', 'anger'). 'Amber' has multiple related colors, all of which are somewhat similar to its true color. The colors for 'apple', however, (red, yellow and green) are more distinctly different types of apples. The pink term in 'apricot' is somewhat unexpected, but the term 'apricot pink' really is more common than 'apricot orange' in the Google corpus.

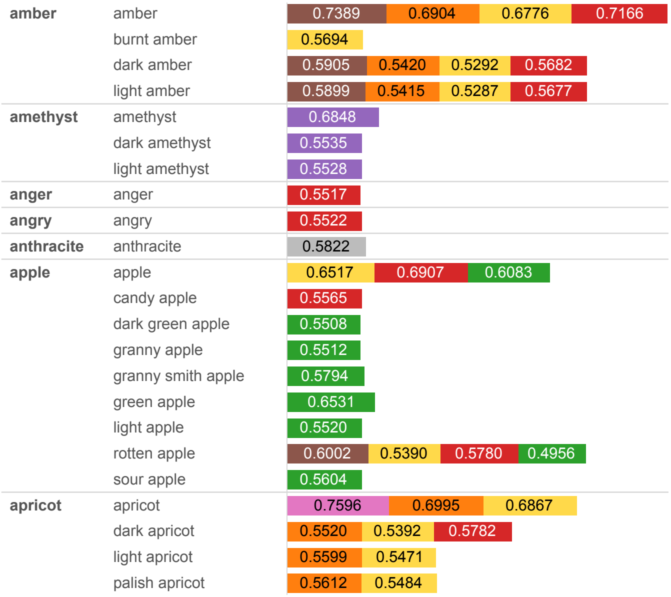


Fig. 6. A typical sampling of color categories defined by the terminal word in the phrase and the associated basic color terms.

Using the terminal terms to structure the visualization, it was possible to inspect and qualitatively evaluate the n-gram algorithm for the entire 6,540 entry dataset. It was easy to verify that within a category defined by a terminal term the associated basic colors were similar and plausible. A common pattern is to have one or two basic colors common to all phrases in the set, with the addition of additional terms defined by the modifiers. For example, the all phrases ending in 'lavender' include a strong purple component but 'grayish lavender' includes a gray component as well.

Splitting off the terminal term revealed some interesting structure in this dataset. There are only 907 unique terminal terms. Phrases ending in a basic name (For example, 'lime green', 'sky blue', 'light gray') account for over half of the dataset (3,830 rows, or nearly 59%). The distribution within these names is highly skewed. Phrases ending in the term 'green' dominate, making up 17% of the database. At the other end of the distribution, 615 terminal terms occur only once, and 111 only twice.

4 Mapping Names to Colors

Given a colorable term, we need to find its semantically resonant color. Typical color naming data, however, is created by asking people to describe a displayed color in language. As we have just discussed in the previous section, the result is primarily words associated with describing color, such as 'light sky blue'. For categorical color assignment, what we need is a way to map data categories (which are rarely names describing colors) to an appropriate color value.

We now describe a general purpose algorithm that combines semantic analysis including the basic name to search Google images and returns a representative color. We validate this approach by comparison to the previous work by Lin *et al.* and to a selection of colors in the XKCD database.

4.1 Search Parameters

For any word or phrase, the Google n-gram analysis provides a colorability score and a list of basic colors that are strongly associated with that phrase. This information is used to determine if a given term is colorable, and if so, the list of basic colors is used as a query parameter to obtain images with those dominant colors.

Similar to Lin *et al.* [29], we employ Google's Image Search API [5] to retrieve images based on the input term. We also add an additional parameter 'clipart' to the search to rank clipart images higher than photos, as the canonical colors representing clipart imagery tend to more directly correspond to the basic color options. However, to retrieve images that correlate more closely with the basic colors from Google n-grams, we use a color based feature in Google image search, called a dominant color filter that provides 12 common color options for mapping to common color names. These include the 11 Berlin & Kay basic color terms and an additional option 'teal.' The search API allows a dominant filter parameter to be set with one or more of these colors and returns the images in which the selected color is one of the dominant colors based on a frequency feature of interested colors to rank the images [25]. As parameters to this dominant filter, we assign the basic color list obtained from Google n-grams as well as an option for 'teal' if both 'blue' and 'green' color names exist with the n-gram colorability score.

4.2 Query Expansion and Image Retrieval

While the term and the basic color names can be used as input queries to an image search engine, the query words may be different than the ones used in the metadata describing the semantics of the imagery. That means a gap exists between the query space and document representation space, resulting in lower precision and recall of queries. We use query expansion to augment related terms to each of the queries for improving search precision and recall.

For additional query terms, we use Wordnet's ontology to obtain the Least Common Subsumer (LCS) of the term's synset and the color synset (*color.noun.synset.01*) if it exists [40, 42]. The LCS of two synsets A and B, is the most specific concept which is an ancestor of both A and B, where the concept tree is defined by the is-a relation. By computing the LCS between the term and the color synset, we get the most generalizable entity synset that has a color association to it. For example, the LCS of 'taxi' , 'rose' and 'turmeric' are 'automobile,' 'flower' and 'spice' respectively.

Once the query is executed, the result set of images returned are accompanied by a normalized confidence measure from the Google search engine. The confidence measure is a descriptor that represents two aspects of relevancy - first being the relevancy of the image's metadata to the input query, and second how close the dominant region color of the image matches the dominant color filters specified, and if so, which of those color filters does the dominant region correspond to. We use a subset of the image results with a confidence score $>= 0.65$.

Since the images in the search results with higher confidence scores already contain dominant color regions matching the dominant color filter specified, we do not have the overhead of performing background detection for improving pixel selection as described in [29]. However, the search service does not provide the actual R,G,B color values of these dominant color regions. We apply clustering on the images to determine the actual color of that dominant color region. The images are clustered using a hierarchical agglomerative clustering algorithm to cluster large homogenous color regions without defining the number of clusters ahead of time [45]. We then iteratively proceed in decreasing order of cluster size to determine the largest common set of clusters across the images.

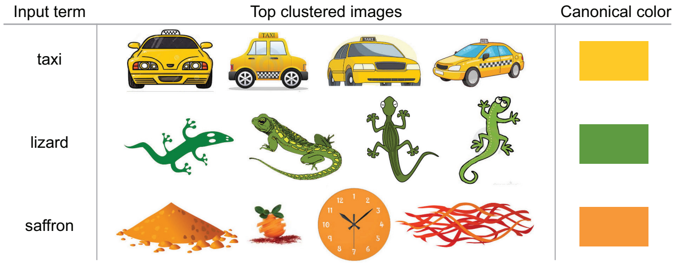


Fig. 7. From top to bottom: Canonical colors retrieved from Google Images using dominant filtering for 'taxi', 'lizard' and 'saffron.'

This is illustrated in Figure 7, which shows the highest scored images returned for each of three terms. The taxi example shows the images returned for the dominant filter yellow. For each image, neighboring pixels are iteratively merged to find the largest homogenous region which is some shade of yellow. The final canonical color shown is an average of the color values of each of the most dominant color regions of the images. Similarly, results show green for 'lizard' and orange for 'saffron.'

4.3 Validation

For our first validation, we algorithmically selected 36 colorable objects from the XKCD database, colored them, and visually compared them to the values returned by Heer & Stone. In addition, we computed the CIEDE2000 color difference between the colorings. The results, which range from 22.1 to 0.9 are shown in Figure 8. While visibly different, our algorithm returned colors that are a reasonable match to the color term.

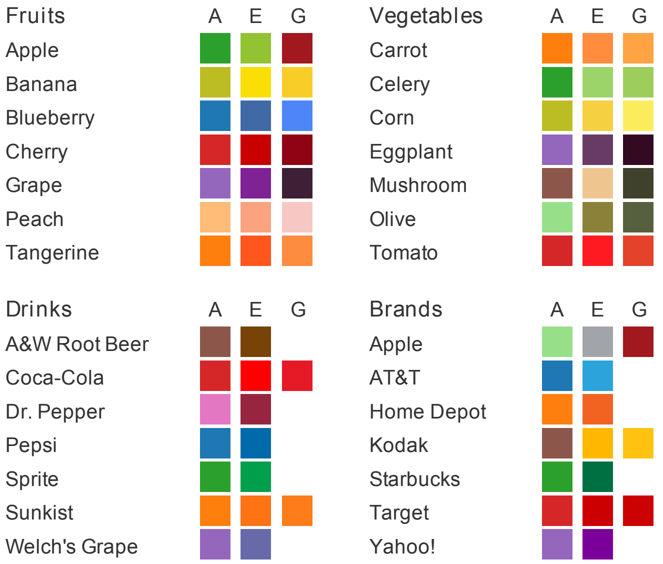
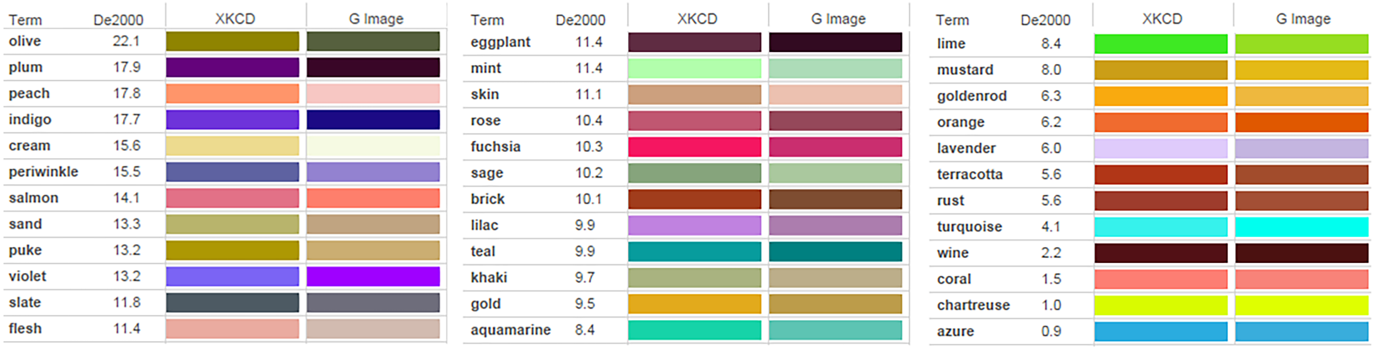


Fig. 9. Results for Lin's algorithm (A), expert (E) and our algorithm (G).

We then compare our results to Lin *et al.*'s work. In their semantic coloring paper [29], they created a set of four palettes and applied both their algorithm and asked an expert to color the same categories. We took the collection of terms from these palette and colored them using the method described in the previous sections. The results for terms we found colorable are shown in Figure 9. It can be seen that the correspondence is generally very good, though some of our colors are darker than ideal. These results illustrate several differences between our approach and Lin *et al.*'s. Firstly, we color only a subset of the terms as we are culling based on our colorability score. Secondly, our results treat each term independently so both 'apple' and 'cherry' are red in the Fruits Palette, and do not distinguish between 'apple' the fruit and 'apple' the brand. Finally, both Lin *et al.*'s algorithm, which returned only colors from the Tableau 20 default palette, and their expert carefully used colors that worked well as categorical coloring. Our results have no such constraint on them, resulting in several that are too dark and one that is too light ('corn').

In their work, Lin *et al.* collected 40 palettes, ranked them by colorability, then selected 10. Each of these palettes was colored both using their algorithm and by colors selected in a Mechanical Turk study. All colors in these palettes were selected from the Tableau 20. Of the 57 terms in these 10 palettes, our algorithm successfully colored 39 of them, with results comparable to the example above. In addition, we applied our algorithm to all 246 terms in the the full set of 40 palettes and successfully colored 114 of them, again with results comparable

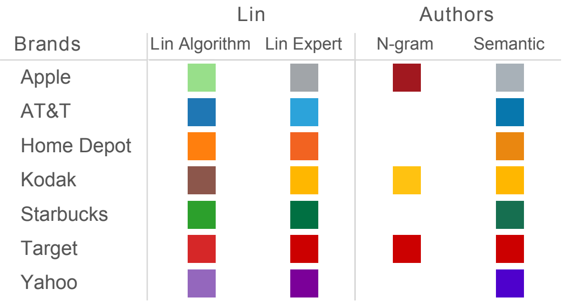


Term	De2000	XKCD	G Image
olive	22.1	🟨	🟩
plum	17.9	🟪	⬛
peach	17.8	🟧	🟥
indigo	17.7	🟪	🟦
cream	15.6	🟨	⬜
periwinkle	15.5	🟦	🟪
salmon	14.1	🟥	🟥
sand	13.3	🟨	🟫
puke	13.2	🟨	🟨
violet	13.2	🟪	🟪
slate	11.8	⬛	⬛
flesh	11.4	🟥	🟫
eggplant	11.4	🟪	⬛
mint	11.4	🟩	🟩
skin	11.1	🟫	🟫
rose	10.4	🟥	🟥
fuchsia	10.3	🟥	🟥
sage	10.2	🟩	🟩
brick	10.1	🟥	🟫
lilac	9.9	🟪	🟪
teal	9.9	🟦	🟦
khaki	9.7	🟩	🟫
gold	9.5	🟨	🟨
aquamarine	8.4	🟩	🟩
lime	8.4	🟩	🟩
mustard	8.0	🟨	🟨
goldenrod	6.3	🟧	🟨
orange	6.2	🟧	🟧
lavender	6.0	🟪	🟪
terracotta	5.6	🟥	🟫
rust	5.6	🟥	🟫
turquoise	4.1	🟦	🟦
wine	2.2	🟫	🟫
coral	1.5	🟥	🟥
chartreuse	1.0	🟨	🟨
azure	0.9	🟦	🟦

Fig. 8. Comparison of 36 colorable object terms showing the XKCD colors determined by Heer & Stone, the colors from our algorithm obtained through Google Image Search, and the CIEDE2000 distance between them.

to the examples above. Additional examples are shown in Figures 10 and 11.

4.4 Expanding Semantic Context for Color Assignment



	Lin		Authors	
Brands	Lin Algorithm	Lin Expert	N-gram	Semantic
Apple	🟩	⬜	🟥	⬜
AT&T	🟦	🟦		🟦
Home Depot	🟧	🟧		🟧
Kodak	🟫	🟨	🟨	🟨
Starbucks	🟩	🟩		🟩
Target	🟥	🟥	🟥	🟥
Yahoo	🟪	🟪		🟪

Fig. 10. A comparison of Lin *et al.*'s algorithm and their expert's selection, with our algorithm's color assignment using n-gram analysis and using semantic context. Here, the category field 'Brands' from Lin *et al.*'s dataset is used to obtain symbolic logo color representations for the terms in the semantic mapping.

The current algorithm depends on the Google n-grams corpus to determine the colorability of a term. While the corpus is extensive, it is definitely not exhaustive, and may not contain co-occurrence trends for certain terms that are associated with visual symbols, particularly Company and Brand names. In Figure 10 for the 'Brands' category, Google n-grams does not return any value for 'AT&T', 'Home Depot', 'Starbucks' and 'Yahoo!', and the color for 'Apple' is that of a fruit rather than the company 'Apple'. However, we can use additional semantic context to find colors for these terms. This context could be supplied from the name of the data field, provided as input by the user, or by additional semantic analysis.

Given a word or phrase that describes the category, we use an approach similar to the one Setlur & Mackinlay [42] described to find icons associated with data. We first compute the semantic relatedness of the category description to the symbol synset (*symbol.noun.synset.02*) in Wordnet. Similar to the query expansion process in subsection 4.2, we find the Least Common Subsumer (LCS) for the category. For example, for the category 'Countries', the LCS with the symbol synset is 'flag' while 'Brands' and 'Companies' are associated with 'logo'. This highest scoring symbolic word is then used to create a query to retrieve symbolic clipart imagery from Google images.

From the highest scored images for each term, the dominant colors are extracted as described in the image retrieval process of subsection 4.2. The final canonical color shown is an average of the color values of each of the most dominant color regions of the images, as shown in Figure 10.

5 Palette Generation

We have shown that given a colorable term, we can determine the most likely color(s). However, color-encoding for data visualizations tends to be for a set of terms, as opposed to a single one. Effective categorical color palettes encode visually distinct colors to different categorical values for distinction, in addition to the color being as semantically meaningful to the corresponding value.

5.1 Color Palette Assignment for a Set of Terms

Figure 9 shows our algorithm picking individual colors for each term. Without any context of the other color assignments in the set of terms, similar colors may be assigned as seen for terms 'Apple' and 'Cherry.' This is not optimal for generating a visually discernible palette of colors. To determine if there are collisions in colors for a set of terms, and if so, search for alternative color assignments, we perform clustering on the color values in the set.

We apply the *k*-means clustering algorithm to color quantize the input set of colors into visually discriminable clusters [20]. The algorithm tries to minimize total intra-cluster variance,

$$V = \sum_{i=1}^{k} \sum_{x_j} \in S_i Dist(x_j, \mu_i) \tag{5}$$

where S_i represents k clusters, $i = 1,2..k$ at a given iteration and μ_i represents centroids of all the points $x_j \in S_i$, and $Dist(x_j, \mu_i)$ is the Euclidean distance measured in CIELAB.

Since we ideally want each color value to represent one color palette item, we partition the k input color terms into k initial singleton sets, where k represents the total number of items in the palette and each centroid directly corresponds to the single color value in each singleton set. As the number of sets is equal to the number of partitions, we do not encounter the problem of randomly picking a number of sets for convergence that is typical of the *k*-means algorithm.

Every color term has one or more associated canonical color values. For example, 'Apple' has canonical color values of red and green, while 'Cherry' has one canonical color value of red. However, for terms with multiple canonical colors, we initiate the clustering starting with the highest scored canonical color for each term. So, for the term 'Apple' we start with the red value since it is scored higher than that of green.

The centroid μ_i of each set is computed and constructs a new partition by associating each point with its closest centroid. After this association, the centroids are recalculated to form new clusters. We apply a constraint to the algorithm such that if the size of each new set is greater than 1, *i.e.* two or more colors are similar in value, we check if the terms representing those colors have alternative canonical color values. If so, the color is replaced by the next highest ranked canonical color and the algorithm repeats these two steps until convergence, *i.e.* when no data points switch clusters or visually discriminative singleton clusters are generated. For example, the canonical red colors of

‘Apple’ and ‘Cherry’ collide into one cluster, and for the next iteration, the canonical green color for ‘Apple’ is considered instead of the red to separate the singleton clusters apart, thus minimizing V from Equation 5.

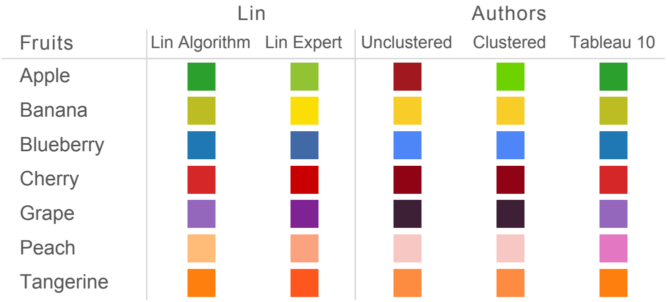


Fig. 11. A comparison of Lin *et al.*’s algorithm and their expert’s selection with our algorithm’s color assignment without and with clustering. With clustering, the red color value for ‘Apple’ is replaced with a more discriminable green color. This clustered set of results can also be assigned to the closest set of colors from Tableau 10, as shown.

Figure 11 shows a comparison of Lin *et al.*’s algorithm and expert selection with our algorithm’s color assignment without and with clustering. Here, the red color value for ‘Apple’ is replaced with a more discriminable green color. This clustered set of results is then assigned to the closest set of colors from Tableau 10.

When the color palette is fixed, the k-means clustering algorithm is more of a color quantization algorithm, which simply takes each color value and finds the closest palette entry, minimizing the Euclidean distance.

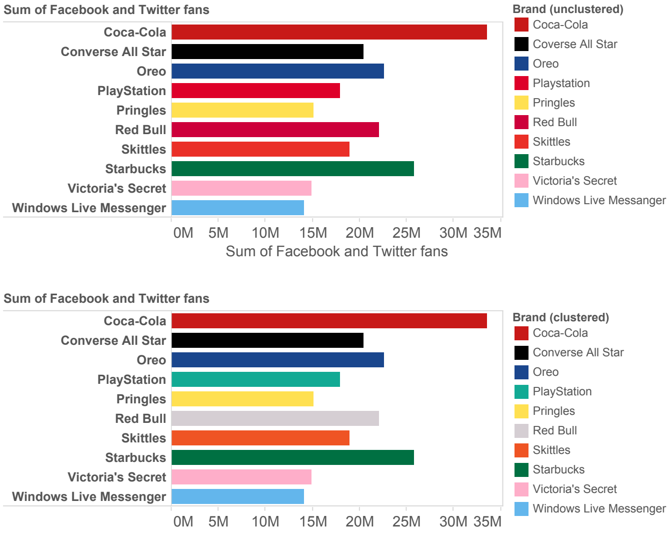


Fig. 12. Top row: A Tableau Public visualization [11] semantically color encoded with our algorithm before clustering is applied. One can observe several brands colored with shades of red. Bottom row: Once clustering is applied to the set of colors, some of the reds are replaced by alternate canonical colors obtained from the corresponding logo images.

Figure 12 shows the semantic colors generated from each of the company logos, with and without clustering. Notice that in the clustered example, the colors assigned for ‘Red Bull’ and ‘PlayStation’ are lesser known colors gray and teal picked up from their respective logo images as opposed to red and black. This is because the colors red and black are already assigned to ‘Coca-Cola’ and ‘Converse All-Star’, since they were the sole dominant colors in their logos.

6 Discussion and Future Work

For a set of data categories, our techniques can provide two useful pieces of information - First, whether the terms have strong associations to color, and second, if there are such color associations, what are the corresponding semantic colors. In addition, we can construct palettes that contain distinctly different colors, either as defined in CIELAB space or by mapping to a pre-defined palette. In some cases, the results are adequate as returned. In others, there needs to be additional refinement.

Consider the example in Figure 13. This is another example from Tableau Public with a custom set of colors created by the creator of the workbook (original). Doing this in the current system requires hand-editing each color with a conventional color picker. We generated semantic coloring (based on the n-gram algorithm) and applied clustering to get the second image (semantic). The colors are similar, but not as distinct as those created by hand. Mapping to a predefined palette (fixed palette), the colors become more distinct. but we lose the lovely cream color mapped to ‘Vanilla’, replacing with with light orange. There was no similar color in the palette. But either of these automatically generated palettes are a good starting point for further refining the colors, either by editing individual colors by hand with a color picker, or by choosing an alternative or augmented palette to use, then re-running the clustering.

We could improve our algorithm in a number of ways.

Perceptual constraints: There are known legibility constraints on colors suitable for visualization, especially with respect to contrast. We are interested in exploring modifying the returned set of colors to match these constraints.

Palette recommendations: In our paper, our algorithm generates a palette of colors from the ground up or assigns terms to an existing palette based on color distance. While we use color discriminability as one measure, other parameters are worth looking into for either color reassignment or recommending a suitable palette. For example, Likert terms may benefit from a red-green divergent ramp as opposed to a traditional categorical palette for values with positive and negative connotations.

Categorical terms may not have visually discriminable colors, such as a set of metals that are all various shades of gray. Perhaps using a categorical set of colors may not be an optimal design choice in such cases. Also, logo colors for brands may not have one dominant color; Take the FedEx logo with purple and orange hues or the Google logo with multiple primary colors. In such cases, it may be worth exploring alternative encodings such as multi-color glyphs. Researching ways for using color co-occurrence information in visualization interfaces for helping users with optimal encoding design choices such as logos or glyphs, is yet another promising direction.

Exploring additional linguistic corpora and semantic trends: We used Google n-grams as the corpus for determining term colorability for single and multi-word terms. For terms that were not necessarily colorable, such as product brands, we used symbolic synsets from Wordnet. However, such domain specific terms may not be present in the corpora we used. While a few abstract concepts such as ‘anger’ and ‘happy’ were identified with colors red and yellow respectively in the Google n-grams corpus, the coverage for other such abstract terms is limited.

Another Tableau Public example is a visualization showing the timeline of when various Crayola colors were introduced [7]. While our algorithm generates plausible color values for each term, they are different from the user generated ones. The author of this workbook used the actual Crayola color values, whereas our algorithm used Google Image Search. While some of the Crayola colors such as for ‘Bittersweet’ and ‘Sunglow’ were found through Google Image Search, many of the colors are not true Crayola ones.

An extension of this method is to consider additional structured and semi-structured data sources for increased domain-specific coverage

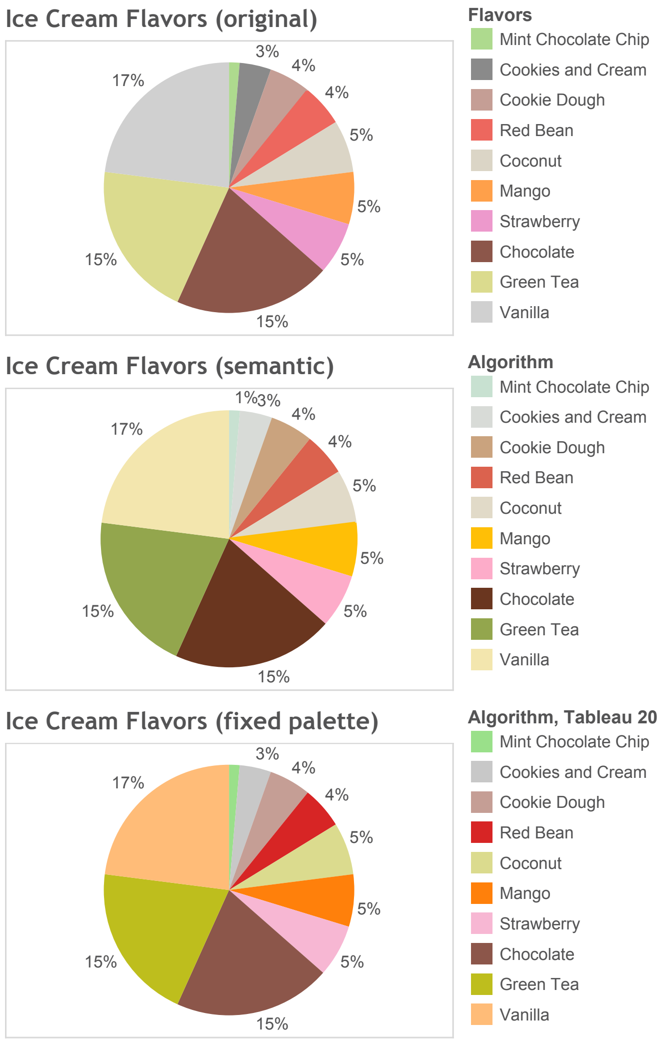


Fig. 13. Top row: Tableau workbook author's custom color assignment to various ice cream flavors. Middle row: Color palette generated automatically by our algorithm. Bottom row: Color palette created from a combination of colors from our algorithm and the Tableau 20 color palette.

such as color name datasets like Crayola [3] and Pantone [9], and concept knowledge bases such as DBpedia [4] and ConceptNet [2].

The *PMI* colorability score does not capture the entire spectrum of meanings the word associates with. Books in the Google n-grams dataset from distant past can contain diverse surface forms of the same entities. For example, terms 'cinnabar' and 'vermilion' were used interchangeably until around the 17th century, and then 'vermilion' became the more common name. By tracking the temporal co-occurrence aspects of terms based on the life cycle of keywords, the colorability score could be improved to better reflect semantic trends in the corpus.

Extending semantic context to other categorical domains: In Section 5.2, we showed how semantic context can be explored for a set of categorical terms to retrieve color values. While companies, brands and sports team are common categories, there are other categories such as Likert terms that do not have a strong symbolic connection. Alternative forms of color co-occurrence algorithms such as sentiment analysis to determine color associations for Likert terms and emotions could be an interesting research direction.

Visual consistency and harmony in color assignment: Our current algorithm constrains the image search query by using dominant colors as the main visual feature. The granularity of color could be somewhat coarse, as the search confidence score is merely matching the general hues. This could lead to certain colors being either too dark or too light as seen in colors like 'grape' in Figure 11. An extension of this method may include other visual features such as saturation and lightness to make the set of colors more consistent with one another. Color descriptors such as 'light', 'pale', 'dark', 'deep', and 'milky' associated with the terms could be used to create these visual constraints. Also, evaluating color harmony algorithms by assigning colors with equal or similar color saturation while maintaining hue and value for example, could help improve the color assignment algorithm [16, 15].

7 Conclusion

In this paper, we have shown two ways to leverage natural language techniques to map category names to semantically appropriate colors. The first uses n-gram analysis with respect to basic color names, the second semantic context for the data category, to find identity colors from visual symbols such as logos. Both return scores that can be used to determine if a data term has a semantic coloring. We then generate visually distinct palettes. In data visualization tools such as Tableau [12], such an algorithm can complement the color picking and palette selection features that already exist in those tools. By providing semantically meaningful colors as a reasonable default, our algorithm can guide such tools to match the terms to predefined palettes that are designed to be visually discriminable and aesthetic for visual analysis. A practical application for our algorithm would be to build it into an online service to generate palettes for such tools. Alternatively, it could be used to create a large dictionary of highly-colorable terms for more efficient use.

Linguistic corpora provide rich resources for denoting various color sensations and expressions for describing and understanding color. Our results show that a natural language approach to harnessing patterns of color co-occurrence with both text and images from large-scale linguistic corpora such as Google n-grams and Google Images, is a promising approach and can be improved in many ways. Our algorithm does not have to explicitly contain a model for color as it obtains much of the knowledge implicitly from the data, often reflecting how humans have chosen to talk about color. As more data is available online, and as computing capacity increases, we believe that a data-driven methodology could be a promising approach to helping with tasks in visualization beyond categorical color assignment.

As this verse by William Wordsworth in the poem titled 'The Thorn' [44], available in digitalized form states [34] -

> Ah me! what lovely tints are there
> Of olive green and scarlet bright,
> In spikes, in branches, and in stars,
> Green, red, and pearly white!

Acknowledgments

The authors wish to thank Michelle Gumport, Leland Wilkinson and the anonymous reviewers for their helpful feedback.

REFERENCES

[1] Adobe Kuler. `https://color.adobe.com/explore/newest/`.

[2] ConceptNet. `http://conceptnet5.media.mit.edu/`.

[3] Crayola. `http://www.crayola.com`.

[4] DBpedia. `http://dbpedia.org`.

[5] Google Image Search API. `https://developers.google.com/custom-search/`.

[6] Google n-grams. `https://books.google.com/ngrams`.

[7] History of Crayola Colors. `https://public.tableau.com/views/HistoryofCrayolaColors/HistoryofCrayolaColors`.

[8] HTML Color Names. `http://www.w3schools.com/html/html_colornames.asp`.

[9] Pantone. `http://www.pantone.com/`.

[10] Sherwin-Williams. `http://www.sherwin-williams.com/homeowners/color/find-and-explore-colors/paint-colors-by-family/family/red/`.

[11] Tableau Public. `https://public.tableau.com`.

[12] Tableau Software. `http://www.tableau.com`.

[13] B. Berlin and P. Kay. *Basic Color Terms: their Universality and Evolution*. University of California Press, 1969.

[14] C. Biggam and C. Biggam. *The Semantics of Colour: A Historical Approach*. The Semantics of Colour: A Historical Approach. Cambridge University Press, 2012.

[15] F. Birren and T. Cleland. *A grammar of color: a basic treatise on the color system of Albert H. Munsell*. Van Nostrand Reinhold Co., 1969.

[16] M. Chevreul and C. Martel. *The Principles of Harmony and Contrast of Colours, and Their Applications to the Arts*. Collections spéciales. Longman, Brown, Green, and Longmans, 1855.

[17] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Journal of Computational Linguistics*, 16(1):22–29, Mar. 1990.

[18] H. C. Conklin. Hanunóo color categories. *Southwestern Journal of Anthropology*, 11:339–344, 1955.

[19] E. Dalrymple-Alford. Associative facilitation and interference in the stroop color-word task. *Perception & Psychophysics*, 11(4):274–276, 1972.

[20] J. A. Hartigan and M. A. Wong. A K-means clustering algorithm. *Journal of Applied Statistics*, 28:100–108, 1979.

[21] C. Havasi, R. Speer, and J. Holmgren. Automated color selection using semantic knowledge. In *AAAI Fall Symposium: Commonsense Knowledge*, 2010.

[22] J. Heer and M. Stone. Color naming models for color selection, image editing and palette design. In *ACM Human Factors in Computing Systems (CHI)*, 2012.

[23] E. Iosif and A. Potamianos. Unsupervised semantic similarity computation between terms using web documents. *IEEE Trans. on Knowl. and Data Eng.*, 22(11):1637–1647, Nov. 2010.

[24] A. Islam and D. Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(2):10, 2008.

[25] Y. Jing, S. Baluja, and H. Rowley. Canonical image selection from the web. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pages 280–287, New York, NY, USA, 2007. ACM.

[26] P. Kay, B. Berlin, and L. Maffi. *The World Color Survey*. CSLI lecture notes. Center for the Study of Language and Information Publications, 2011.

[27] P. Kay, N. J. Smelser, P. B. Baltes, and B. Comrie. The linguistics of color terms. *International Encyclopedia of the Social and Behavioral Sciences*, 2001.

[28] K. L. Kelly. ISCC-NBS method of designating colors and a dictionary of color names. 1955.

[29] S. Lin, J. Fortuna, C. Kulkarni, M. Stone, and J. Heer. Selecting semantically-resonant colors for data visualization. *Computer Graphics Forum (Proc. EuroVis)*, 2013.

[30] A. Lindner, N. Bonnier, and S. Süsstrunk. What is the color of chocolate?–extracting color values of semantic expressions. In *Conference on Colour in Graphics, Imaging, and Vision*, volume 2012, pages 355–361. Society for Imaging Science and Technology, 2012.

[31] A. Lindner, B. Z. Li, N. Bonnier, and S. Süsstrunk. A large-scale multi-lingual color thesaurus. In *Color and Imaging Conference*, volume 2012, pages 30–35. Society for Imaging Science and Technology, 2012.

[32] A. Lindner and S. Süsstrunk. Automatic color palette creation from words. In *Color and Imaging Conference*, volume 2013, pages 69–74. Society for Imaging Science and Technology, 2013.

[33] C. M. MacLeod. Half a century of research on the stroop effect: an integrative review. *Psychological bulletin*, 109(2):163, 1991.

[34] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, T. G. B. Team, J. P. Pickett, D. Holberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 2010.

[35] N. Moroney. Color naming experiment. `http://www.hpl.hp.com/personal/Nathan_Moroney/color-name-hpl.html`, 2002.

[36] N. Moroney. Unconstrained web-based color naming experiment. In *Electronic Imaging 2003*, pages 36–46. International Society for Optics and Photonics, 2003.

[37] R. Munroe. Color survey results. 2010.

[38] A. H. Munsell. *A color notation*. Munsell color company, 1919.

[39] Y. Niwa and Y. Nitta. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, COLING '94, pages 304–309, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[40] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[41] B. Schauerte and R. Stiefelhagen. Learning robust color name models from web images. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3598–3601. IEEE, 2012.

[42] V. Setlur and J. D. Mackinlay. Automatic generation of semantic icon encodings for visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 541–550, New York, NY, USA, 2014. ACM.

[43] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009.

[44] W. Wordsworth and S. T. Coleridge. *Lyrical Ballads with a Few Other Poems*. J. and A. Arch, London, 1798.

[45] Z. Xiang and G. Joy. Color image quantization by agglomerative clustering. *IEEE Computer Graphics & Applications*, 14(3):44–48, May 1994.