Color information in Word Embedding Vector

Michitatsu Sato

December 2022

Abstract

This research aimed to determine whether recent word embedding vectors generated from static or dynamic word embedding models could obtain a visual representation of the color words. Word-embedded vectors are known as their arithmetic operation, which enables representing the semantic relations between the objects of the words being embedded. Since color is a concept that has a strong relative relationship with each other and could be learned from the context of the words, we came up with the idea use of word embedding to obtain a visual representation of the colors. In this research, we first focused on the possibility of the word embedding model to obtain the color representation like RGB, then tried to apply this model to predict the visual representation (RGB) of the non-color concept or unseen words.

1 Introduction

1.1 Word embedding

Word embedding is one of the most important technology for current natural language processing. Recently it is used in the pre-processing process of most of the Natural language processing models. The idea of word embedding was supported under the"Distributional Hypothesis" assumption. The hypothesis claim that the words that occur in the same contexts tend to have similar semantic meaning [5]. Based on this assumption there were many types of word embedding. For example, **one-hot vector embedding** represents each word with a vector whose length is the number of vocabulary, and each entry which contains 1 if the word corresponds to that vector entry exists in the word context otherwise 0. Also, another famous example is **Latent Semantic Indexing** which is also a count-base method for embedding, but it includes the process of vector dimension reduction so that the resultant vector will be much more compact compared to the one-hot vector embedding representation. In the history of word embedding, one of the most famous models known as the breakthrough of natural language processing area is **Word2Vector**. Word2Vector is a word embedding model which uses neural network architecture inside the model. The concept is widely recognized by the research of Mikolov(2013)[10]. This approach's basic idea is to build a model that could infer and predict the gap between words in sentences. The approach for predicting interval words from two side contexts is called the "CBOW" model while predicting context (surrounding) words from a word is called the "skip-gram" model. In addition to the Word2Vector model introduced above, more models can embed the words based on their contexts (for example, Fast-Text, Glove, etc). The most popular model used in the recent model is the dynamic word embedding approach using a Transformer and BERT to embed the word vector. In this research, we evaluated the Word2Vector model and the recent BERT-based word embedding model.

1.2 Color and vector arithmetic of the word embedding model

This research is motivated by the interesting property that word embedding vectors have. According to some research, it is found that the word embedding model could obtain not only similarities between words but also similarities between pairs of words. A famous example of this phenomenon is;

$$Vector(King) - Vector(man) \approx Vector(Queen) - Vector(woman)$$

$$\rightarrow Vector(King) - Vector(man) + Vector(woman) \approx Vector(Queen)$$

This example shows that the semantics of the embedded word vector can be computed in arithmetic operation which is very intuitive to human understanding of the word's concept. This leads to the hypothesis that each entry of the embedded word vector should have some kind of human interpretable meaning (for example in the above example, the arithmetic operation holds because some entries of the word vector represent some kind of loyal and gender information).
At this point, we come up with the question; "Is visual(color) information also included in word embedding vector?". To find out this, we prepared a simple word2vec-based pre-trained model[7] and tried the following operations;

$$\begin{aligned} &Vector(Zebra) - Vector(stripe) + Vector(Brown) \approx \\ &Vector(Giraffe) \\ &expected\ \text{"}Horse\text{"} \end{aligned}$$

$$Vector(BlueBerry) - Vector(Blue) + Vector(Red) \approx Vector(Cherry)$$
$$expected \text{ “}Strawberry\text{”}$$

From these operations and results, it became clear that the word embedding vector is actually able to capture the visual features of the object partially. However, compared to our expectations, it is reasonable to say that the vector did not completely encode color information (It is questionable to argue zebra without a stripe pattern with brown color is a giraffe). This is due to the word embedding vector not separately capturing the color information (if color information is separately captured in a specific vector entry, the second operation only involves color addition and subtraction should not lead to loss of information "berry", although this is based on the stereotype cherry should not be categorized as a berry). We assumed this is caused by too much information other than color words included during the training. So we hypothesized that if we can train or fine-tune the word embedding model with only sentences with color words included, the resultant vector should capture color representation more clearly. Even more, we expected these vectors could capture the physical wavelength information about the color (for example RGB values, etc.) so that calculations like;

$$Vector(Yellow) + Vector(Blue) \approx Vector(Green)$$

could holds. In addition, by finding out the proper mapping from these word embedding vectors to the RGB value, it is possible to predict the color of the non-color concept words or unseen words. In the last part of this experiment, we tried to predict the color of the non-color concept and unseen words and evaluate the results.

2 Related work

There is no previous work that tried to capture the color feature from word embedding representation. However, there are some works that tried to generate color from words. Some works tried to model the probability distribution among colors given some natural language text[9]. Also, other works are done by Iiba et al.[2] for the generation of the color and font from the text. Interesting works done by Sakamoto et al.[11][13][12] tried to capture the similarity between words based on color features in vector spaces. This work is tried to capture objects as 32 dimension vectors each entry corresponds to some color which is a useful insight to prove that it is possible to predict the color of the words if the vector representation is concise. About the relationship between words and color, there is some interesting work[6] which stated machine models could have some color images toward specific alphabets. This could have a possibility to model the human synesthesia[1].

3 Methodology

In this section, we talk about how the experiments were conducted, what kind of preparation was done, and what types of the model is used. We conducted three experiments in general.

3.1 Data

For the training data or data for fine-tuning, we used sentences from the *Gutenberg* corpus included in NLTK python packages. The reason for choosing this corpus is because this corpus generally consists of English literature. We assumed English literature should include more color words for literature expression compared to the sentences used in newspapers or SNS chat. The sentences are extracted as training data if the sentence includes the color words listed in our color list (the color list and its RGB value is generated based on scraping on some specific website[14][4]). We succeeded to extract 329 colors and 6973 sentences for training data.

3.2 Training Word2Vec model for RGB vector

In this section, the details of the experiments using the Word2Vector model are introduced. We used the word2vector model provided by the *gensim* library of the Python package. The goal of this experiment is to construct a word2vector model with color words sentences and train the Feed Forward Neural Networks (FNN) to map the word embedding vector generated by the word2vector model to specific RGB values. For the training mode of the word2vector model, we choose the skip-gram approach. The brief architecture of the model is presented in the following diagram(Figure 1).

3.2.1 Word2Vec with 3dim vector + RGB FNN

The purpose of this experiment is to find out whether the output vector of the word2vec model trained on color words corpus can represent the RGB(or related) color representation. In this experiment, we constrained the embedding dimension of the word2vector model to be 3, slightly expecting that the 3-dimensional output embedding vector directly represents the Red, Green, and Blue values of the RGB color representation. In addition, for the backup plan, we trained FNN to obtain mapping between the word2vector embedding vector and real RGB values in the case of a 3-dimensional embedding vector that did not exactly represent the color representation in RGB or related manner (to be honest, we did not expect the 3-dimensional embedding vector to directly represent RGB color expression since this is a very compact representation and will not be able to capture by word2vector model without any constrain on learning). FNN architecture is the following.

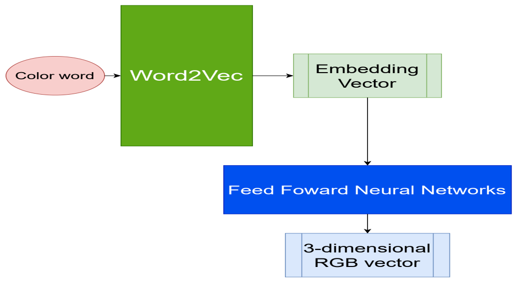


Figure 1: Word2vec+FNN architecture

1. Linear layer with dimensions 3x10
2. Linear layer with dimensions 10x10
3. Linear layer with dimensions 10x3(RGB)

Word2vector model is trained with the above extracted 6973 sentences. After the training, obtained the embedding vectors for each color word in color lists. FNN is trained on these data as a regression problem to predict the proper RGB value of the color from the color's embedding vectors, therefore the input to the model will be the embedding vector of each of the colors and the supervised correct label will be real RGB values of these color scraped from a website.

3.2.2 Word2Vec with 100dim vector + RGB FNN

This experiment is almost the same as the above experiment while this experiment used the default 100 dimensions as the embedding dimension. Different from the above experiments, since the embedding vector will have 100 dimensions instead of 3 dimensions, we can not expect this embedding vector to express RGB representation directly, and therefore we are required to train FNN to map embedding representation to proper color RGB values. The process of training each model is exactly the same as the above experiment, however, the architecture of the FNN is slightly different. Another hidden layer is added since the dimension of the input embedding vector became bigger.

1. Linear layer with dimensions 100x50
2. Linear layer with dimensions 50x30
3. Linear layer with dimensions 30x10
4. Linear layer with dimensions 10x3(RGB)

3.3 Training BERT-based model for RGB vector

In this section, the details of the experiments using the BERT-based word embedding model are introduced. The main idea is almost the same as the above experiments. The only difference is the use of the BERT-based word embedding model instead of the Word2Vector model. As the BERT-based model, a pre-trained BERT model from *huggingface.transformers*[3] python package is used. The model is first trained by our color words sentences for fine-tuning purposes. The architecture is following(Figure 2)

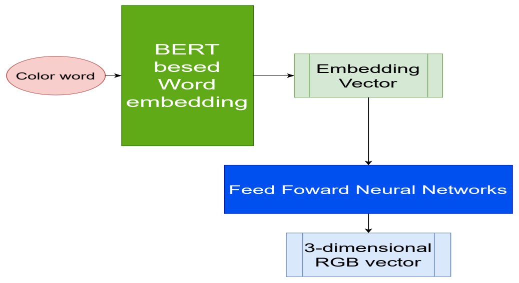


Figure 2: BERT-based word embedding+FNN architecture

3.3.1 768 dim BERT word embedding + RGB FNN

The purpose of this experiment is to obtain the color representation (especially RGB vector) from 768-dimension word embedding vector output by the BERT-based model. As stated above, the pre-trained model is first fine-tuned by our color words corpus. After that, we obtained word embedding vectors for each of the colors in the color list. We first evaluate these vectors to examine whether they captured the color representation or visual feature of the color words. Furthermore, to convert these 768 dimension vectors to RGB vectors, we trained FNN as a regression problem to predict the RGB value of the color from the color's embedding vectors. The architecture of the FNN is the following.

1. Linear layer with dimensions 768x200
2. Linear layer with dimensions 200x200
3. Linear layer with dimensions 200x200
4. Linear layer with dimensions 200x3(RGB)

3.4 Prediction of non-color words on BERT-based model

As we stated in the Introduction section, as a final experiment, we conducted a prediction of RGB color representation for non-color concepts and unseen words. We used the model of BERT-based one since another model Word2Vector is a static word embedding approach and can not apply to unseen words. We prepared 10 words of fruit, 15 words of non-color concepts, characters or digits, and non-existing words. We decided to use fruit for the evaluation of the model since it is simple to relate these concepts with the color concept(for example apple will be red, etc.). The non-color concept and unseen words list are used for the evaluation of the ability of the model to predict and relate the color information of the near concepts with the given word. The list is the following.

- Fruit:
  [apple, grape, strawberry, blueberry, mango, banana, watermelon, lemon, lime, peach]
- Non-color concept:
  [blue-planet, 1, 2, A, Z, dramatic, mathematical elegant, noble, `0range`, `b1ue`, catastrophic, hopeless, blooming, BigBang]

Since it is difficult to evaluate the performance of these abilities of the model by machine-based score, we evaluated the performance by simply looking at the predicted color and the nearest 5 colors suggested for the given words.

3.5 Re-scaling and normalization of the output RGB

Since some of the RGB values vector output from FNN is not well distinguished between color to color due to lack of color data (number of colors) and biased data set(blue and dark colors are dominated more than half of the color list), we introduced some normalization methods to output RGB vector predicted by the FNN model for mapping from word embedding to RGB values. In most of the results, we observed that the RGB value differs only in the non-most significant value(for example, like [152,123,98] and [158, 128, 101] there is only a small value of difference which can distinguish the different color) or even in value after the decimal point. Just plotting these RGB values will result in almost the same color across different color words. To solve this problem, we introduced re-scaling and normalization of the RGB values output by the FNN model. The specific normalization we applied is, **Min-Max normalization** and **Z-score normalization**. We first normalized the list of RGB values of the colors by Z-score normalization concerning each entry R, G, and B. Then we applied Min-Max normalization for re-scaling it between 0 to 1. Finally, we multiply 255 which is the max value of the RGB by these normalized values to obtain the proper RGB values. The normalization process can be expressed by the following formula.

$$Normalized\_RGB = round(255 \times Min\_Max\_Norm(Z\_score\_Norm(original\_RGB)))$$

4 Results

4.1 Evaluation metric

Although we applied FNN to convert the word embedding vector to an RGB vector as near as possible, it does not guarantees the output RGB values will be the same as the original RGB of the given color. As stated above, our FNN model is not well-trained to predict the exact RGB values of the given color(we are required to apply a normalization process to obtain valid and proper color representation). In addition, since the raw word embedding before converted by FNN could have dimensions more than 3, it is impossible to directly relate these to real RGB values. Due to these situations, we need to find out the evaluation metric which will not calculate the performance by directly comparing real RGB values and the output vector of given color words. As such methods, we proposed three metrics to evaluate the relationship between the embedded vector (or converted RGB vector) and real RGB values of the colors.

- Color adding agreements rate
- Nearest color overlapping agreements rate
- Nearest adding color overlapping agreements rate

4.2 Color-adding agreement rate

The score is calculated by counting the number of agreements between colors adding the real RGB values and embedding vectors. The idea comes from the assumption that if the vector is able to capture the physics aspect of color representation (for example wavelength of the color or RGB, etc), the addition of the specific two colors should lead to the same color even if the vector did not represent RGB values. For example, if the addition of yellow and blue leads to blue in real RGB value, the most similar color represented by the vector obtained by adding an embedding vector corresponding to yellow and an embedding vector corresponding to blue should

Table 1: Color-adding agreement rate

Embedding vectors	color-adding agreement rate
Word2Vec Raw 3dim	0.015
Word2Vec Raw 100dim	0.061
Word2Vec 3dim converted	0.027
Word2Vec 100dim converted	0.022
BERT Raw 768dim	0.014
BERT 768dim converted	0.003
BERT 768dim converted normalized	0.003

be green if the embedding vector capturing some kind of RGB related features. The exact formula to calculate the score is presented following.

$$Score = \frac{length(Agreements)}{length([(x,y)|x \in Colors, y \in Colors])} \text{ where;}$$

$$Agreements = [(x,y)|x \in Colors, y \in Colors; Most\_similar(add(RGB(x), RGB(y))) == Most\_similar(Vec(x), Vec(y))]$$

RGB(x) = real RGB vector of color x
Vec(x) = word embedding vector of color x

For the calculation of most similar colors(Most_similar), cosine similarity between vectors (or RGB vectors) is used. This score is basically focusing on the relative relationship between each color concept. In this metric, the relative relationship is expressed by adding the results of the color. If there is a high amount of agreement between adding the color between real RGB and the embedding vector, it is reasonable to conclude word embedding vector succeeded to capture the relative relationship(in this case color addition) between each color.
We computed color-adding agreement rate scores for both raw word embedding vectors before conversion by FNN and converted RGB vector by FNN. In addition, we also included the results for normalized word embedding vectors converted to RGB values(table1).

Raw represents embedding vector before converted by FNN (for example Raw 100dim means 100 dimension word embedding vector before converted), converted means embedding vector converted by FNN (for example 100dim converted is 3dim RGB vector converted by FNN and originally it was Raw 100dim embedding vector), normalized means vectors which normalization stated above applied.
One thing that became clear from the above results is that Word2Vec embedding with 3 dimension embedding size will not capture RGB-like representation. Compared to “Word2Vec Raw 100dim" which has 100 dimensions for embedding size, the performance on this task of “Word2Vec Raw 3dim" is very low and it is reasonable to conclude Word2Vec with 3-dimension embedding did not capture the relative relationships between colors whose real RGB value can capture.

In addition, the results showed that the normalization process will not affect the performance of this task. It is a very natural result since the normalization process is just a re-scaling of the vector value. It should not impact the relative relationship between vectors(colors).
It is also important to note that in most cases, FNN conversion contributes to the decrease in the score of this task. Although in Word2Vec with 3 dimension embedding, there is a slight improvement in score, in both Word2Vec with 100 dim and BERT-based model, the score is greatly decreased. From these results, we concluded that FNN conversion is harmful to capture the inter-color relationship. We analyzed the cause of this problem as a loss of information during the conversion. Since the decrease in the score happens when the FNN is converting from a higher dimension to a lower dimension, we suspect that FNN through away some important information that is captured in the original embedding vector. This can be observed from the 2-dimensional plotting of each color embedding vector before and after the conversion of FNN(Figure[?]) The plot shows that FNN lost some specific color distribution modeled by original embedding vectors.

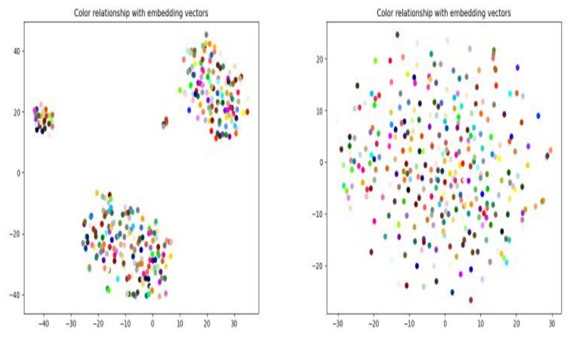


Figure 3: BERT-based model embedding vector plotting in 2D space: left=before conversion, right=after conversion

4.3 Five nearest color overlapping agreements rate

The score is calculated by counting the number of similar color agreements between real RGB values and embedding vectors. The idea comes from the assumption if two colors are visually similar, they should have similar color representation if the vector representation captured the RGB-like color features. For example, if Red and Scarlet is similar color in real RGB, in the vector space of word embedding, two vectors corresponding to these two colors should be similar. In this task, we decided to define the agreement between real RGB and embedding vector if there are more than two color overlaps between the

Table 2: Nearest color overlapping agreements rate

Embedding vectors	nearest color agreement rate
Word2Vec Raw 3dim	0.261
Word2Vec Raw 100dim	0.290
Word2Vec 3dim converted	0.290
Word2Vec 100dim converted	0.304
BERT Raw 768dim	0.037
BERT 768dim converted	0.046
BERT 768dim converted normalized	0.061

five most similar colors computed from a given specific color. The reason for the decision is our biased data set. As stated above, our color list includes many blue colors. Therefore, the computation of the most similar color for the bluish color could be slightly different between RGB and the embedding vector. However, since the purpose of this experiment is the evaluation of the embedding vector's ability to represent the color feature, we will not care if the embedding model tells us the most similar color of the blue is sapphire or turquoise unless it did not say scarlet. However, if we only care about the most similar colors, we will treat the difference between blue and sapphire as the same as the difference between red and green. Since there is more blue color and minor error like blue similar sapphire could easily happen, it is a disadvantage for blue colors to only count the exact match of the most similar color. To remove minor errors like blue similar to sapphire and more focus on serious errors like red similar to green, we introduced overlap between the five most similar colors. The score computation is expressed in the following formula.

$$Score = \frac{length(Agreements)}{length(Colors)} \text{ where;}$$

$$Agreements = [x|x \in Colors, ; 5\_most\_similar(RGB(x)) \cap 5\_most\_similar(Vec(x)) \geq 2]$$

RGB(x) = real RGB vector of color x
Vec(x) = word embedding vector of color x

Again similarities between colors are computed based on cosine similarity between corresponding vector representations. Compared to the previous task, this task is more focused on the relationship between similar colors instead of a more broad sense of relationship like the addition of the different colors. The results are presented in the following(Table2). In this task, it is shown that normalization has an impact on the improvement of the score. This is because, in this task, the difference between colors(vectors) becomes more important than in the previous task, and the normalization process that amplified the difference between colors leads to an easier distinction between different colors.
Different from the previous task, these results showed that FNN conversion leads to better performance(for all of the models, the performance is better for the FNN-converted vector). It is difficult to provide a clear explanation for this phenomenon, however, we thought it is reasonable to say that for the task of finding similar colors, a wrong cluster of vectors with different colors(Figure3 left) will harm the result compared to relatively uniformly distributed vectors(Figure3 right).

4.4 Nearest adding color overlapping agreements rate

This metric can be considered as a weakly constrained version of the "color-adding agreement rate". Instead of counting the exact match of added colors, this score counts overlaps between the five most similar colors of the added colors from real RGB and embedding vectors. Since the restriction is relaxed, it is expected that score will be better compared to the color-adding agreement rate. The score is expressed in the following formula.

$$Score = \frac{length(Agreements)}{length([(x,y)|x \in Colors, y \in Colors])} \text{ where;}$$

$$Agreements = [(x,y)|x \in Colors, y \in Colors; 5\_most\_similar(add(RGB(x), RGB(y))) \cap 5\_most\_similar(Vec(x), Vec(y)) \geq 2]$$

RGB(x) = real RGB vector of color x
Vec(x) = word embedding vector of color x

The purpose of this task is to observe the effect of introducing the model's ability to distinguish similar colors. If the score is improved compared to the exact match one(color-adding agreement rate), it will show that the word embedding model has the ability to distinguish similar colors. The results are presented in the following table(Table3) compared to the score of exact match one(color-adding agreement rate). About the FNN conversion, we can observe the exact same effect as the exact-match one(color-adding agreement rate) shown. Except for 3-dim to 3-dim conversion, all the performance declined. Also regarding the normalization, there is no effect on performance as observed in the exact-match task. As expected, for all of the models, the performance is improved compared to the exact-match one. We can observe more improvements in the Word2Vec model compared to the BERT-based model, and it is a very natural consequence since Word2Vec had more ability on finding out similar colors compared to the BERT-based model.

4.5 Overall evaluation from scores

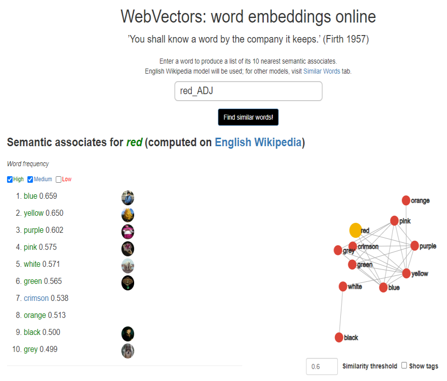
The interesting fact that can be observed from these results is that actually, Word2Vec model is better than the BERT-based model in these tasks. As a possible factor, a more consistent data set could be considered. Since the

Table 3: Nearest adding color overlapping agreements rate

Embedding vectors	color-adding agreement rate	Nearest adding color overlapping agreements rate
Word2Vec Raw 3dim	0.015	0.067
Word2Vec Raw 100dim	0.061	0.136
Word2Vec 3dim converted	0.027	0.078
Word2Vec 100dim converted	0.022	0.089
BERT Raw 768dim	0.014	0.035
BERT 768dim converted	0.003	0.004
BERT 768dim converted normalized	0.003	0.004

Word2Vec model is only trained on the color words corpus we prepared, it can concentrate on the color aspect of information compared to the BERT-based model which is trained on more general and large data sets which include many sentences without colors. We concluded that for the learning of color representation, the sentences without color words will be just noise to obtain a visual representation of the color. There is more benefit for concentrate on color words rather than obtaining information about non-color words which could have some contextual relationship with color words.
By observing the performance on each task, we can conclude that word embedding model is more suitable to obtain similar color information rather than an inter-color relationship like color adding. This conclusion can be backed up by the improvement of the score in "Nearest adding color overlapping agreements rate" which included the model's ability to consider similar colors compared to "Color-adding agreement rate" which did not include this ability. Also, relatively high performance in the task of finding similar colors compared to other tasks led to the same conclusion(Although it is too simple to just compare scores with different computations, we can observe the model did well on the second task). By thinking about this phenomenon from a more cognitive science point of view, it is reasonable to say that for word embedding models that can not have access to exact color information(like RGB), it is more difficult to capture the inter-color relationship like color adding. Assuming there is a people who can not distinguish the colors (the world will look grey-scaled). This person will be able to obtain the knowledge "red and scarlet are similar" by reading natural language sentences ["burning red apple", "burning scarlet apple"]. Furthermore, this person will be able to find out "Black is the opposite concept of white" from the sentences ["Dark black", "Bright white"](assuming this person knows that "dark" and "bright" is kind of opposite concept). However, this did not mean this person will be able to infer "Black+White=Grey". Since the result of adding different colors is not the information that can be observed or inferred from natural language context, the word embedding model that only considers semantics comes from distributed word context will not be able to obtain the knowledge about color adding.

About the word embedding mode's ability to find a similar color, we concluded that the model actually has the ability to obtain such a feature of the colors. By comparing the "Five nearest color overlapping agreements rate" of the model Word2Vec with another pre-trained BERT-based model, it is reasonable to say that the score achieved by the Word2Vec model is not coincident and proved the ability of the word embedding model to capture the similarity between the colors. This conclusion can be inducted by comparing the nearest 5 colors observed by our Word2Vec model trained on our color word sentences with the existing pre-trained Word2Vec model's output[8](Figure4). Our model predicted more similar colors while the pre-trained model presented colors that do not have any visual relationship with the given color.


WebVectors: word embeddings online

'You shall know a word by the company it keeps.' (Firth 1957)

Enter a word to produce a list of its 10 nearest semantic associates.
English Wikipedia model will be used; for other models, visit Similar Words tab.

red_ADJ

Find similar words!

Semantic associates for *red* (computed on English Wikipedia)

Word frequency

☑ High ☑ Medium ☐ Low

1. blue 0.659
2. yellow 0.650
3. purple 0.602
4. pink 0.575
5. white 0.571
6. green 0.565
7. crimson 0.538
8. orange 0.513
9. black 0.500
10. grey 0.499

0.6 Similarity threshold ☐ Show tags

Figure 4: 5 nearest colors for "red" from pre-trained word2vec model online ['blue', 'yellow', 'purple', 'pink', 'white']. Our model suggested ['plum', 'pink', 'ruby', 'red', 'russet'] for 5 nearest colors of "red"

4.6 Prediction of non-color concept

As stated in our motivation for the research, in this section, we will look at the predicted color for the non-color concept. Although it can be observed that our BERT-based model did not have high performance in capturing color information compared to the word2vector model, since the word2vector model could not apply to unseen words, we only have one option to use the BERT-based model. However, even if it is worse than the word2vector model, it is found by the above experiments, the word embedding has a partial ability to capture the features of the color if it is trained on sentences with color words, we think it is not totally meaningless to use this model to predict the colors. Actually, by observing results, we could find out more information about word embedding which can not be observed from only scores presented above. We observe not only the predicted color for given words but also the 5 nearest colors to the concept's embedding vector since we observed that the model performed better in finding similar colors. The results are shown in the following figures(Figure5,6). From fruit

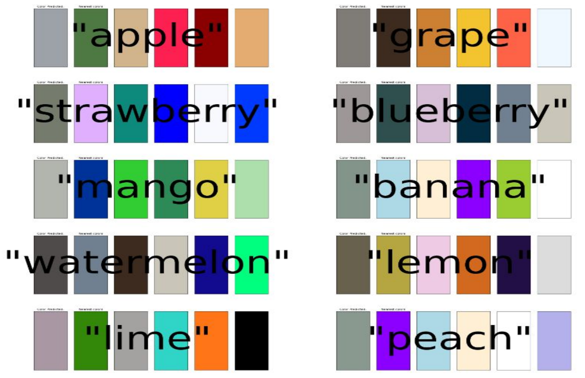


Figure 5: predicted color (most left) and 5 nearest colors (right 5) for fruit concepts

color prediction, we can not observe any relationship between the predicted color and given fruit words. Also about the 5 nearest colors, we can not observe any clear pattern that explains the fruit concept given. It is possible to analyze the result of the apple as somewhat reasonable since it presented red-like color and green as the 5 nearest colors since the apple has two kinds a green apple and a red apple. In addition, it is interesting to see the 5 nearest colors of the "grape" are all red-orange-like colors. Our intention of the word "grape" is a fruit used for wine, however, it looks like the model treats it as "grapefruit". The model also listed only cold colors for the nearest colors for "blueberry" and we assume the model is affected by the word image of "blue". For the non-color concept like digits and the alphabet, we could not find any interesting relationship between a given character and the colors presented. The same observations can be found for unseen words(like 0range, b1ue), adjectives, and nouns. It is reasonable to conclude it is difficult to predict or find out consistent nearest colors for concepts that did not have a strong relationship with visual images. Interesting results can be observed in very similar nearest colors for "blooming" and "catastrophic". We could not come up with any similarity between these two concepts. It is interesting that the presented color for

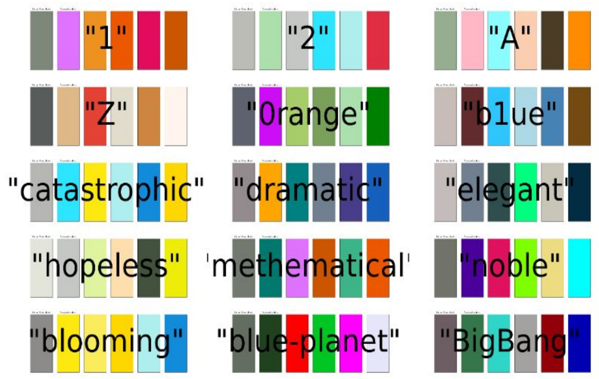


Figure 6: predicted color (most left) and 5 nearest colors (right 5) for non-color concepts

"catastrophic" which should have negative images are all light colors like yellow and sky blue. The nearest colors of these are not random and very scattered, therefore we can observe the embedding representation has captured some color features even in the BERT-based model. The predicted colors are basically grey-like colors and we suspect it is due to the low performance of RGB mapping of FNN.

5 Conclusion

By comparing the experiment results of word2vector with the BERT-based one which is pre-trained, it is reasonable to conclude it is possible for the word embedding model to capture the color features of the words if proper training data is given. This representation is able to capture the similarity between the color concept, however, it is not sufficient for expressing the physical property of colors like color addition (which means the representation did not capture the RGB-like features inducted from the physical property of the colors). This observation leads to the conclusion that the task of capturing the physical property of the color (like RGB) is much more difficult than finding out the similarity of colors since the word embedding model only learns the representation from distributed contexts. Due to our poor ability of mapping embedding vectors to RGB, it is difficult to find out any information from our color prediction for non-color concepts, however from the nearest colors presented, we can observe that the model sometimes could infer the visual feature of the words.

6 Future works

As stated above, we concluded it is possible for the word embedding model to capture the color feature of the word concept. We assume that preparing more sentences with the color words inside should contribute to more effective learning of the model and leads to a higher score in experiments (basically in the task of finding similar colors). This situation should hold also for the BERT-based model. If all the data for training is dominated by sentences with color, the BERT-based embedding could have a more flexible ability to capture the color representation. If it is properly trained, some kind of inter-color representation (like the following) should be able to be captured.

$$Vector(blue) - Vector(light\_blue) + \\ Vector(light\_green) \approx Vector(green)$$

Other than preparing more training data and a better model, it is also an important task to get proper FNN for mapping embedding vectors to RGB. As stated, it is a very difficult task to capture the physical property like RGB, but we suspect that it is possible to fit the FNN model if there is a more concise embedding representation and more training data(simply the number of color words with an RGB value). Since our purpose was to find out whether a word embedding vector could capture the color feature, we did not spend much time designing a mapping model of the FNN, however, focusing on this topic will be also interesting research. In addition, using these techniques to model human synesthesia is also one of the interesting topics.

References

[1] Joel Bock. A deep learning model of perception in color-letter synesthesia. 12 2017.

[2] SakiIibaRyuichi Doizaki and MakiSakamot. Color and fontrecommendationsbasedon mentalimagesof text. , 18(3):217–226, 2013.

[3] Hugging Face. huggingface/transformers. https://github.com/huggingface/transformers. [Online; accessed 05-December-2022].

[4] Face Media Group. A to Z of Colours. https://www.facemediagroup.co.uk/resources/a-to-z-guides/a-to-z-of-colours. [Online; accessed 05-December-2022].

[5] Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954.

[6] Hidehiko Komatsu, Ami Maeno, and Eiji Watanabe. Origin of the ease of association of color names: Comparison between humans and ai. *i-Perception*, 13(5):20416695221131832, 2022.

[7] Nordic Language Processing Laboratory. Semantic Calculator. http://vectors.nlpl.eu/explore/embeddings/en/calculator/. [Online; accessed 13-December-2022].

[8] Nordic Language Processing Laboratory. WebVectors: word embeddings online. http://vectors.nlpl.eu/explore/embeddings/en/. [Online; accessed 13-December-2022].

[9] Saki Liba, Tetsuaki Nakamura, and Maki Sakamoto. Color recommendation for text based on colors associated with words. *Journal of the Korea Industrial Information System Society*, 17, 02 2012.

[10] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[11] Takuro Miyabayashi, Saki Iiba Natsumi Hara, and Sakamoto Maki. Measuring the similarity between texts and fonts via colors. , 2010(6):1–6, 04 2011.

[12] Research Group of Sakamoto Maki. Research Group of Sakamoto Maki. http://www.sakamoto-lab.hc.uec.ac.jp/research/. [Online; accessed 13-December-2022].

[13] Nakamura Tetsuaki Saki Iiba and Sakamoto Maki. A proposal of color for text using the association between colors and words. , 2011.

[14] Spokenenglishtips. 100+ Amazing All Colors Name in English, List of Colors, Colours Name with Pictures. https://spokenenglishtips.com/colors-name-in-english/. [Online; accessed 05-December-2022].