NOLTA, IEICE

Paper

Feature analysis of sentence vectors by an image-generation model using Sentence-BERT

Masato Izumi [1] *and Kenya Jin'no* [1]

[1] *Department of Intelligent Systems, Faculty of Knowledge Engineering Tokyo City University*
1–28–1 tamadutumi, setagaya, Tokyo 158-8557, Japan

Received October 14, 2022; Revised December 29, 20XX; Published July 1, 20XX

Abstract: In this study, using k-means and UMAP, we verified that the sentence vectors generated by Sentence-BERT as distributed representations of sentences capture the meaning of sentences well. To this end, we visualized the sentence vectors by generating images matching the meaning of the sentence from the sentence vectors generated by Sentence-BERT. The results confirm that although there were differences in the information represented by each dimension as distributional features of the sentence vector, this information overlapped substantially.

Key Words: BERT, Sentense-BERT, image generation, latent variable, expressive learning

1. Introduction

In natural language processing, Google proposed BERT[1] in 2018, and it has since attracted considerable attention as a model that can capture the context of sentences. Sentence-BERT was proposed in 2019 to generate very accurate sentence vectors that capture the meaning of the context of input sentence using a BERT model and a Siamese network[2] pre-trained on a very large dataset[3]. We verified the extent to which the sentence vectors, which comprise distributed representations of sentences generated by Sentence-BERT, capture the meaning of sentences using clustering algorithms such as k-means and UMAP[4]. The UMAP results are shown in Fig. 1. As may be observed from the figure, some clusters are indicated by color and some refer to names of cars. The "blue" cluster also includes similar colors such as "sky blue" and "dark blue". Sentences with similar meanings are assigned to the same cluster. Because Sentence-BERT can identify similar sentences with different or cryptic words that do not carry their original meaning, the model is considered to understand the meaning of sentences as represented by the generated sentence vectors. High-quality sentence vectors can be used for sentence classification, generation, proofreading, and other applications.

In this study, we focus on the nature of the sentence vector itself. Whereas BERT generates a sentence vector for each token in the input document, Sentence-BERT generates a sentence vector for each sentence. Therefore, the sentence vector generated by Sentence-BERT can be regarded as encompassing the meaning of the encoded sentence. Therefore, we consider a system that generates

  DOI: 10.1587/nolta.2.1101

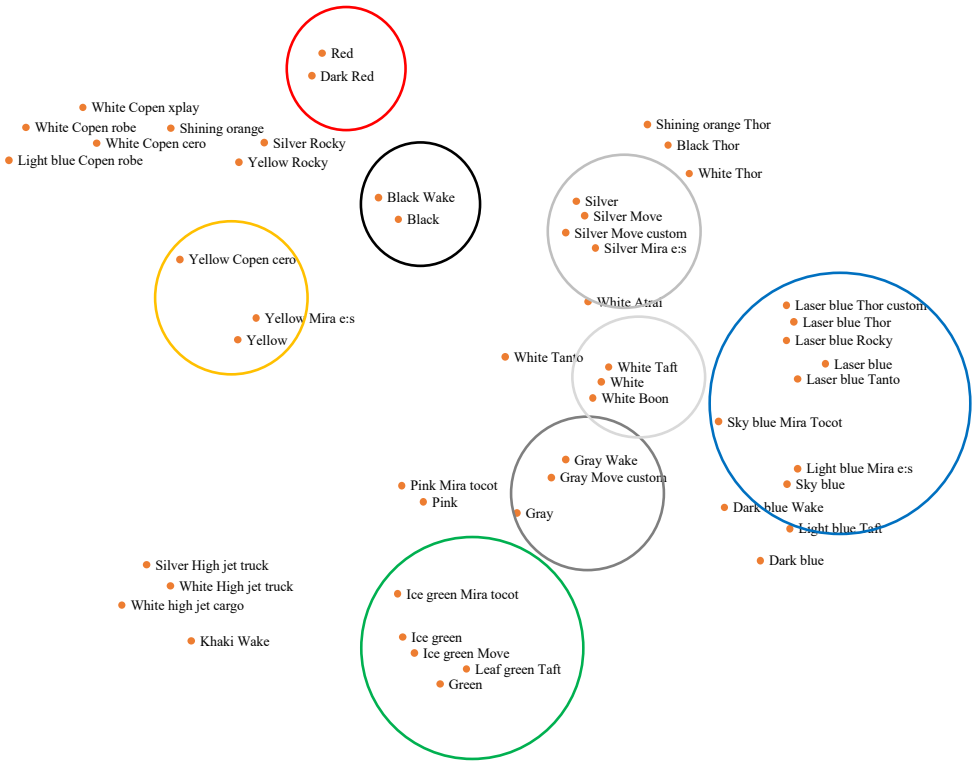


Fig. 1. UMAP

images using this sentence vector as a latent variable vector. This system learns the relationship between sentences in given training data and the images they represent via sentence vectors. We considered that a system that could learn the relationship between the meaning of a sentence and the elements of an image contained in the sentence vector should also be able to continuously visualize the latent variable of the sentence vector, which is discrete. Recently, OpenAI released "DALE·E 2" [5] and Google released "Imagen" [6] as models designed to understand the meaning of sentences and generate images accordingly. Various efforts are being made to generate high-definition images from sentences. In contrast, we sought to elucidate the sentence vector itself. Therefore, we conducted two experiments as part of this work. The first involved training an image-generation model to use sentence vectors generated by Sentence-BERT. The second experiment examined the characteristics of the sentence vectors using the constructed imaging model. Based on the results of these experiments, we examined the sentence vectors generated by Sentence-BERT.

2. BERT[1]

In natural language processing, it is fundamental to replacing the words of a sentence with a distributed representation vector is fundamental. A sentence is a sequence of word data. A sentence is a sequence of word data. BERT (Bidirectional Encoder Representations from Transformers) can predict a sentence from a sequence of input word data. It can also find distributed representations of words in both directions for the words it seeks. Conventional recurrent neural network (RNN)-based natural language processing models can only refer to words before they are sought as a variance representation vector of a word in a sentence. BERT also includes an attention mechanism, which can direct attention to tokens located at a distance to output highly accurate sentence vectors even for long input sentences.

The input sentences are first tokenized by morphological analysis and converted into a sequence of tokens. The token sequence is converted to a token vector and used as an input to BERT. BERT then converts the input token vector into a sentence vector. The pretrained BERT model converts an input sentence into a 768-dimensional sentence vector. To convert it to an appropriate sentence vector, a model must be trained on a large corpus of documents. Learning a distributed representation

from a large corpus is classified as unlabeled learning. Such unlabeled learning involves considerable computational complexity. Therefore, BERT derives common language patterns from a large corpus of documents through two types of pretraining. The pre-training process is shown in Fig. 2.

The first type of pre-training is called a masked language model (MLM). The MLM hides a part of the input token with a "[MASK] token" and guesses the hidden original token. A distributed representation of the masked word is obtained by referring to all the words in the sentence, except for the masked word. BERT provides interactive learning through the MLM.

The second pre-training task is referred to as next sentence prediction (NSP). Because MLM focuses only on relations between words, it cannot handle problems that consider relations between sentences. Therefore, the NSP inputs two sentences and learns whether they are related. The tokenized sentences are prefixed with "[CLS] token" and the "[CLS] token" is used to guess whether the sentences are consecutive.

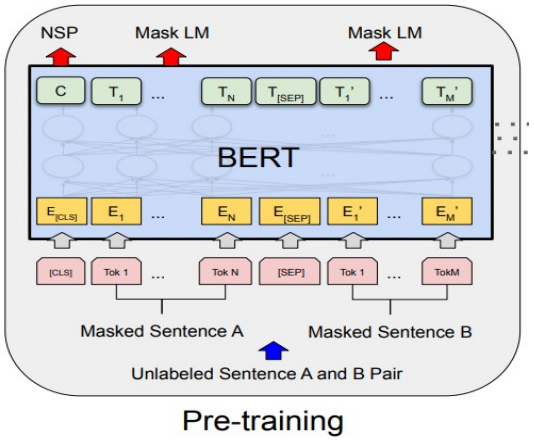


Fig. 2. BERT pre-training[1]

3. Sentence-BERT

Sentence-BERT is a natural language model that performs fine tuning on pre-trained BERT to render it more semantically correct and to transform similar sentences into similar vectors. Particularly, a sentence and an associated similar-sentence pair are selected as training data, and BERT is fine-tuned such that the sentence vectors generated from these similar sentences become similar vectors. This is achieved by fine-tuning the sentence vectors generated by BERT using a Triplet network [7] and Siamese network [2]. The Triplet network enables the fine-tuning of two sentences. These models are used to concatenate the differences between two sentence vectors and learn according to the labels of the sentences. The Siamese Network calculates the cosine similarity between two sentence vectors and learns to encode the sentence vectors of similar sentences as being similar.

Consequently, Sentence-BERT produces better sentence vectors than BERT. An evaluation of Sentence-BERT sentence embedding using SentEval toolkit [8] is shown in Fig. 3.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Avg. GloVe embeddings	77.25	78.30	91.17	87.85	80.18	83.0	72.87	81.52
Avg. fast-text embeddings	77.96	79.23	91.68	87.81	82.15	83.6	74.49	82.42
Avg. BERT embeddings	78.66	86.25	94.37	88.66	84.40	92.8	69.45	84.94
BERT CLS-vector	78.68	84.85	94.21	88.23	84.13	91.4	71.13	84.66
InferSent - GloVe	81.57	86.54	92.50	**90.38**	84.18	88.2	75.77	85.59
Universal Sentence Encoder	80.09	85.19	93.98	86.70	86.38	**93.2**	70.14	85.10
SBERT-NLI-base	83.64	89.43	94.39	89.86	88.96	89.6	**76.00**	87.41
SBERT-NLI-large	**84.88**	**90.07**	**94.52**	90.33	**90.66**	87.4	75.94	**87.69**

Fig. 3. Evaluation of SBERT sentence embedding using the SentEval toolkit

SentEval is a general toolkit for assessing the quality of sentence embedding. Sentence embeddings

were used as the features in the logistic regression classifier. The logistic regression classifier was trained on various tasks in a 10-fold cross-validation setup and the predictive accuracy was calculated for the test fold. The SBERT in Fig. 3 is Sentence-BERT. The results are shown in Fig. 3. Notably, SBERT achieved the best performance for five of the seven tasks. The average performance increased by approximately 2 percentage points compared to InferSent and the Universal Sentence Encoder. Sentence-BERT is also computationally efficient: on a GPU, it was approximately 9% faster than InferSent and 55% faster than the Universal Sentence Encoder. Sentence-BERT can be used for tasks that are computationally infeasible to model. For example, clustering 10,000 sentences with hierarchical clustering requires approximately 65 hours with BERT, because approximately 50 million sentence combinations must be computed. Using Sentence-BERT, that effort could be reduced to about 5 seconds.

3.1 Triplet Network

In learning using the Triplet network, three sentences, "Sentence A" , "Positive sentence p" and "Negative sentence n" were input and converted into vectors. For each vector, the distance between "Sentence A" and "Positive Sentence p" was learned to be closer than the distance between "Sentence A" and "Negative sentence n". In addition, the distance between "Sentence A" and "Negative Sentence n" is learned to be farther than the distance between "Sentence A" and "Positive sentence p". The relationship between sentences is clarified by adjusting the feature vectors such that the positional relationship between each of the three feature vectors is easier to understand and not too exaggerated.

3.2 Siamese Network

Siamese network models learn two sentences. If the labels of two sentences are the same, they learn to bring the two sentences closer together. However, if the labels of the two sentences are different, they learn to move them apart. Thus, similar sentences become similar vectors and vice versa.

4. Model for generating images from text

In this study, we prepared multiple pairs of images and sentences describing the images, and the images were learned to be generated from sentences describing the images. Particularly, sentences describing images were converted into sentence vectors using Sentence-BERT. Next, a CNN model was trained to use this sentence vector as input to synthesize images using inverse convolution operations. If the CNN can output an image that describes the meaning of the sentence vector, it can be called a "decoder". The CNN is trained to output an image that corresponds to the sentence vector generated from the sentence describing the image using the mean squared error (MSE) between the generated image and the teacher data image as the loss function. The proposed model is illustrated in Fig. 4.

The sentences describing the input image were separated into words using the morphological analysis tool Fugashi. These were tokenized and input into Sentence-BERT. The Sentence-BERT used in this study was based on a large Japanese corpus distributed by Inui Suzuki ' s laboratory at Tohoku University, and is pre-trained and fine-tuned by Hugging Face. The pre-training model was trained on a corpus of approximately 30 million sentences from Japanese Wikipedia as of August 31, 2020. Sentences describing the input images were converted to sentence vectors using Sentence-BERT. We trained a decoder to generate images corresponding to the sentence vectors obtained as input using the mean square error (MSE) of the output image as the loss function. The decoder is illustrated in Fig. 5.

To generate 128×128 color images from 768-dimensional sentence vectors obtained using Sentence-BERT, we designed a decoder as shown in Fig. 5. The decoder consists of three inverse convolution layers and a batch normalization layer. The input 768-dimensional vector is converted into 192 channels of 2×2 data. The first inverse convolution layer expanded this data to 16×16 on 256 channels, the second inverse convolution layer expanded it to 64×64 on 256 channels, and the third convolution layer converted it to 128×128 on three channels. The input to each inverse convolution layer was normalized using a batch normalization layer. The MSE during the training is shown in Fig. 6.

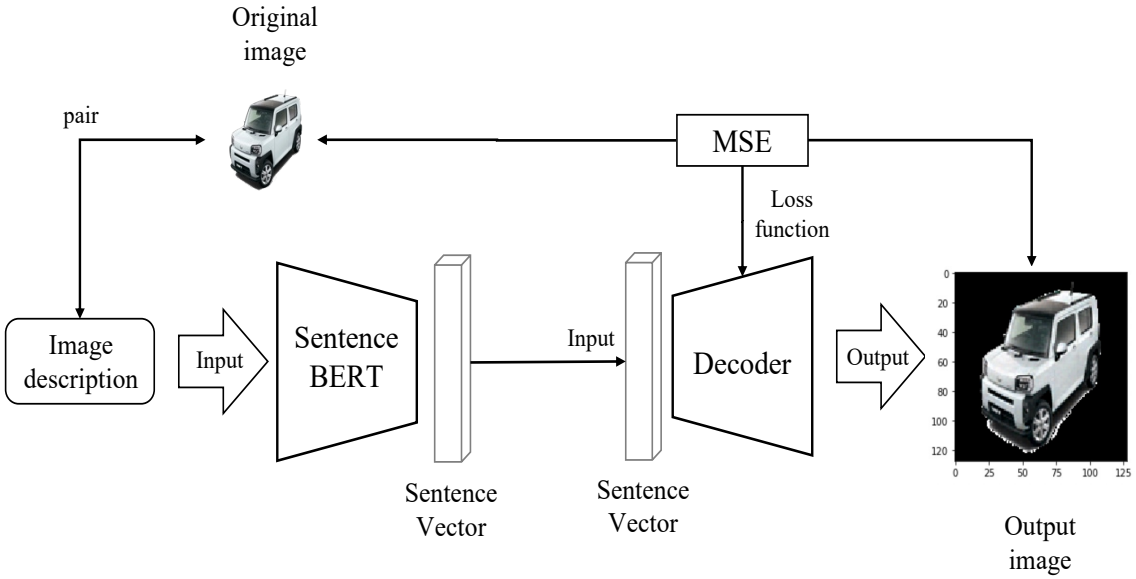


Fig. 4. Model

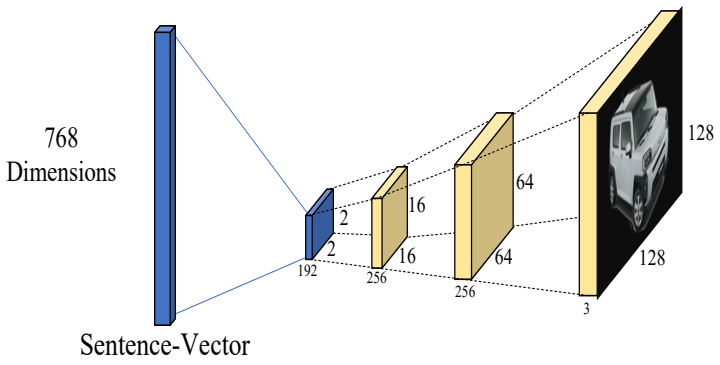


Fig. 5. Decoder

5. Experiment1

We considered images of cars with a pixel resolution of 128×128. The images had a transparent background. The images comprised 80 car models. The sentence describing the car model images was formatted as "color + conjunction + car model name". In addition, the input sentences were in Japanese because the Japanese Sentence-BERT model was used in this study. We identified five to eight different colors of cars depending on the model. Examples of images of cars are shown in Fig. 7.

There were 30 different car colors, including black, red, green, and blue. There were a total of 500 images for each combination of car type and color. Variations in car colors are shown in Fig. 8.

6. Results of Experiment1

6.1 Learning Results

First, we confirmed the feasibility of generating images using the Sentence-BERT sentence vector. Fig. 9 shows the output images when a sentence was input for a combination of "color name" and "car model name", which was real in the dataset.

As shown in Fig. 9, clear output images were obtained in this case. From these results, we confirmed that training an image generation model using the Sentence-BERT sentence vector as a latent variable for the decoder is feasible.

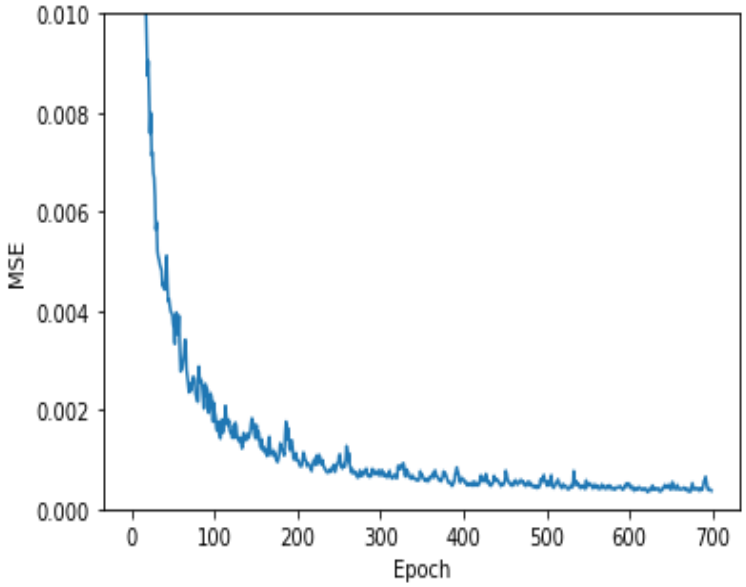


Fig. 6. MSE

Fig. 7. Dataset

black		orange		leaf green	
gray		red		green	
silver		dark red		sky blue	
Luna Silver		reddish brown		light blue	
white		violet		turquoise blue	
yellow		beige		blue	
golden yellow		khaki		dark blue	
pink berry		brown		laser blue	
pink		ice green		indigo blue	
shining orange		lime green		navy blue	

Fig. 8. Color variation

6.2 Non-existent output combinations of ”color” and car model name

The above experimental results suggest that the sentence vector output by Sentence-BERT captures the meaning of the sentences. To confirm that the sentence vector captured the meaning of the sentence, we considered the output obtained by inputting a sentence combination of ”car color” and ”car model name” that was not part of the dataset. The results are shown in Fig. 10.

The upper part of Fig. 10 shows the output results when a combination of ”car color” and ”car model name”, which was not included in the dataset, was input. The lower part of Fig. 10 shows

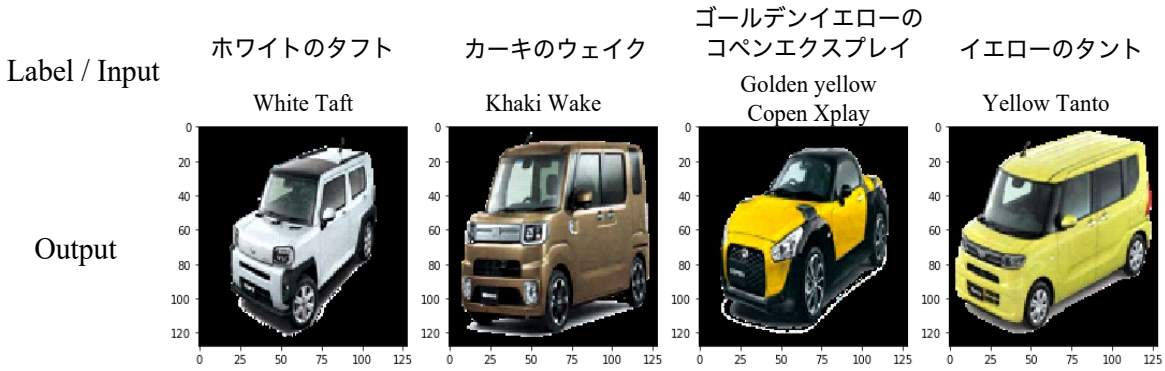


Fig. 9. Learning Results

Fig. 10. Results in output of "color" and "car model name" combinations that are not present in the dataset

the output results when a "car color" that does not exist in the dataset is input and the most similar color among the "car colors" in the dataset. A table of the "color" combinations present and absent for the four cars is shown in Fig. 11.

	Taft	Wake	Copen Xplay	Tanto
Red	×	×	○	×
Orange	○	○	×	×
Blue	×	×	○	×
Dark Blue	×	○	×	×
Silver	○	×	○	○
Luna Silver	×	○	×	×
Pink	×	×	×	×

Fig. 11. Table of availability of "color" combinations

We confirmed that when we input "red Taft" which was not in the dataset, the output took on a

strong "orange" tint. From this fact, it can be inferred that the input of a non-existent car color was strongly influenced by similar colors present in the dataset. For the car models "Wake" and "Copen Xplay", we confirmed that removing the car color "Dark" and adding the car color "Luna" changed the output. From this result, it can be inferred that the sentence vector changed slightly depending on the meaning of the sentence. When "Orange Tanto" is input, the output is colored "orange" with a strong "pink" component. This may be due to the fact that the dataset used in this experiment contained a small amount of data related to the color name "orange" and the output may have been influenced by the component related to the car name "Tanto" in the input sentence.

By generating images with the sentence vector as a latent variable, we confirmed that the sentence vector generated by Sentence-BERT may contain a component that captures the meaning of the word.

6.3 Change in color intensity

We confirmed that the sentence vector was generated when the "car color" in the input sentence changed trivially to encompass the meaning of that color. Therefore, we experimented to see what kind of image was generated from the sentence vector when the "car color" of the input was changed to a slightly different word of the same color system. The results are shown in Fig. 12.

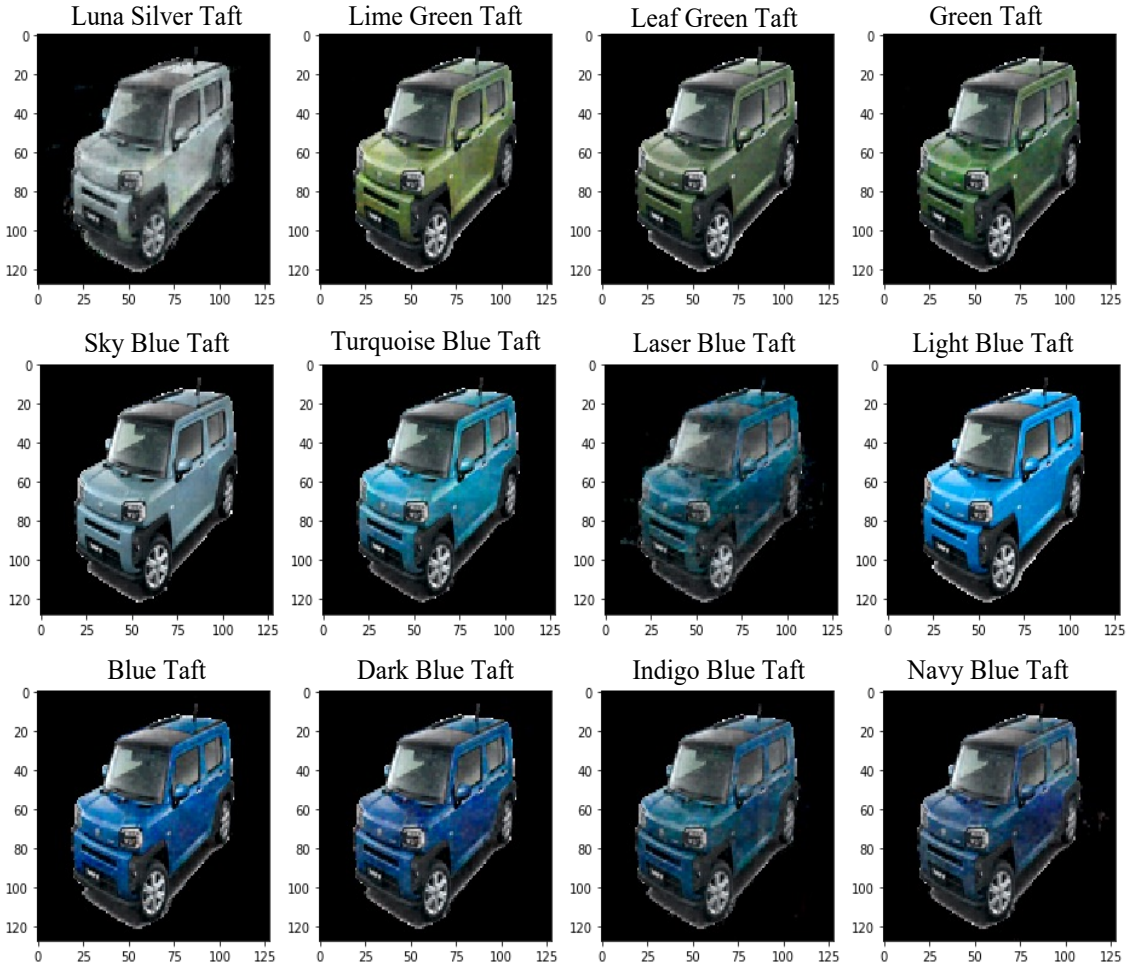


Fig. 12. Results in changes in color intensity

The car model named "Taft" in Fig. 12 comes in three colors: "light blue", "sky blue", and "leaf green". The results in Fig. 12 indicate that we can identify trivial color changes in the input text in each of the same color systems. Because a change in the output image appears with only a small change in the input sentence, it can be assumed that the colors in the same color system are similar for the 768-dimensional sentence vectors. Because the car color in the image changes simply by changing

the car color word in the sentence, we assume that the sentence vectors in Sentence-BERT generate similar vectors when the car colors are similar.

6.4 Non-existent "colors"

In the previous subsection, we confirmed that the sentence vectors generated from sentences containing similar car colors are very similar because the generated images are similar. Next, we examined the properties of the sentence vectors by observing the vectors produced when two different colors were used for the car color. We conducted an experiment to determine whether a composite input of two different colors would produce an output of a color mixed with the two colors. We construct an input sentences in the form "first color" + "second color" + "conjunction" + "car name". For example, we apply the color name "red white", which is composed of "red" and "white". The experimental results are shown in Fig. 13.

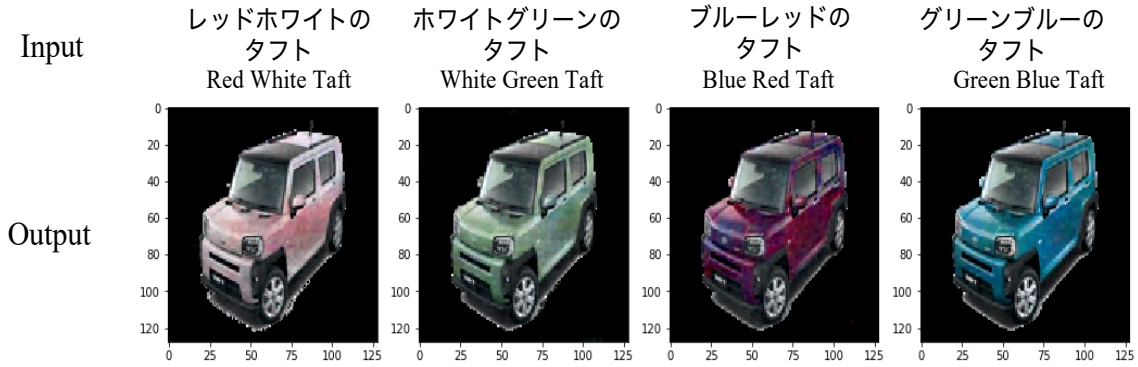


Fig. 13. Results for "color" not present in the dataset

As shown in Fig. 13, when the car color "red white" is given in the input sentence as the color, the output car image becomes "pink"-like color. We confirmed a similar result in the case of the car color "white-green". In addition, when darker colors such as blue and red are combined, the system produced a violet hue, which is considered to be a color between red and blue.

These results suggest that the relationship between clusters of similar colors is preserved in the sentence vector well. We confirmed that the sentence vectors generated by Sentence-BERT capture the meaning of the sentences and preserve the relationship between the meanings of the sentences.

7. Experiment2

In this section, we analyze the structure of the 768-dimensional sentence vector generated by Sentence-BERT based on the generated images First, to confirm if a part of the sentence vector refers to the car color, we consider the difference between the sentence vectors generated by sentences that differ only in terms of car color. We prepare four input sentences: "red Tanto", "red wake", "blue Tanto" and "blue wake". If there was a part of the sentence vector that expresses the color of the car, the sentence vector expresses the color of the car in two sentences that match the color of the car. If the names of the cars are different, different parts of the sentence vector may express the color. Therefore, we consider the matching parts of the sentence vectors "red Tanto" and "red wake". An overview of this process is shown in Fig. 14.

Two sentences of the same color and different car models were converted into sentence vectors. The two vectors are then compared for each dimension. Values that are closest to each other may represent "red". This is because there is no difference between the two sentences except for the name of the car. Conversely, values far apart are more likely to represent a car. The proximity of the values is determined by finding the absolute value of the difference between each vector, and if the difference is less than a threshold value, it is considered to be close.

Next, the dimensions representing the colors were extracted. An overview of this process is shown in Fig. 15. The process illustrated in Fig. 14 was performed twice. The second processing step

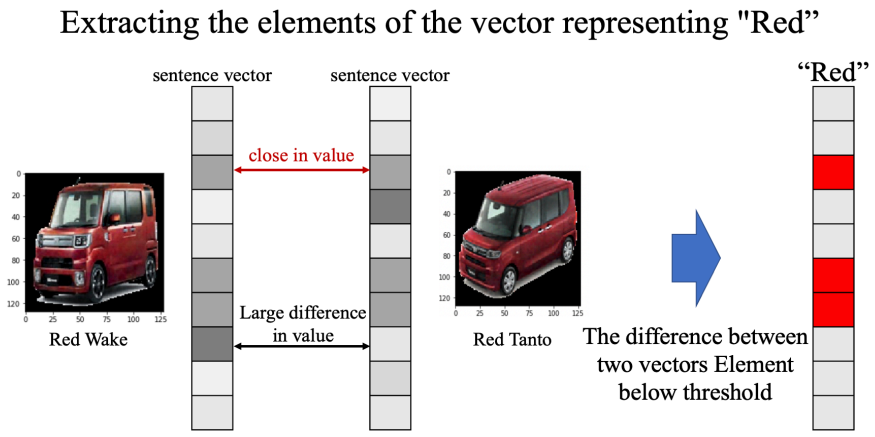


Fig. 14. Extraction of the dimension corresponding to the color "1"

was performed using a color different from the first. In this study, "red" and "blue" were used. As a result, we confirmed that there were 93 elements considered to represent "red" and 103 elements considered to represent "blue". Of these extracted dimensions, we assume that the same dimensions in "red" and "blue" represent color information. There were 22 dimensions matched by "red" and "blue". In other words, 22 of the 768 dimensions in the sentence vector could represent "color".

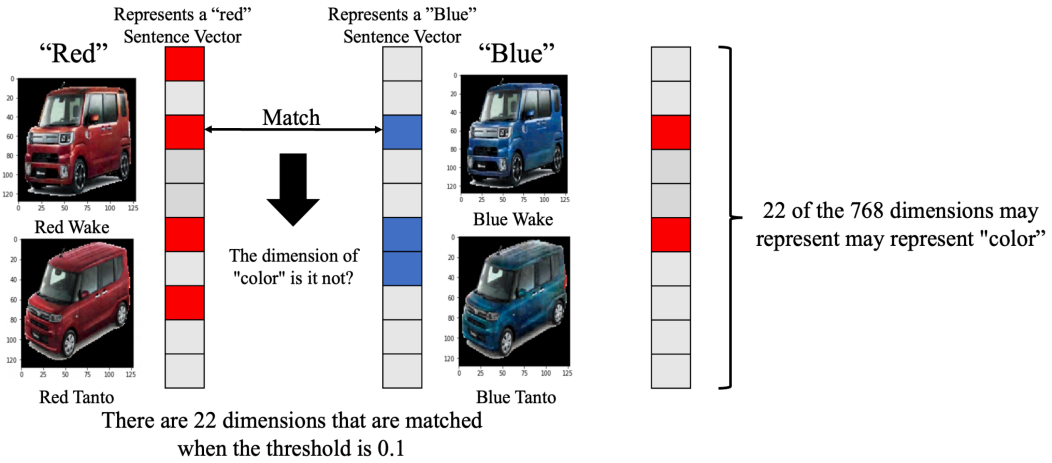


Fig. 15. Extraction of the dimension corresponding to the color "2"

Therefore, using the dimensions considered to represent the "color" obtained in Fig. 15, we conducted an experiment. A summary of the experimental results is presented in Fig. 16. The experiment was conducted to determine if we can approach the "blue wake" sentence vector by replacing the dimension considered to represent "color" in the "red wake" sentence vector with the dimension considered to represent "color" in the "blue Tanto" sentence vector.

First, the experiment was conducted with a threshold of 0.1 for the difference to be considered a match. The threshold is the difference between the corresponding dimensions of the sentence vector. Consequently, 22 dimensions were extracted as candidates for the dimension representing "color". The results show that the cosine similarity of the sentence vector to "red wake" and "red Tanto" is 0.296, and the cosine similarity of the sentence vector replaced by "blue Tanto" is 0.302, with minimal change.

In this experiment, 22 dimensions were extracted as the "color" dimension, but because the input sentence is a short sentence, "color + car name", approximately half of the 768 dimensions in the generated sentence vectors may represent "color". Therefore, the threshold for the difference between

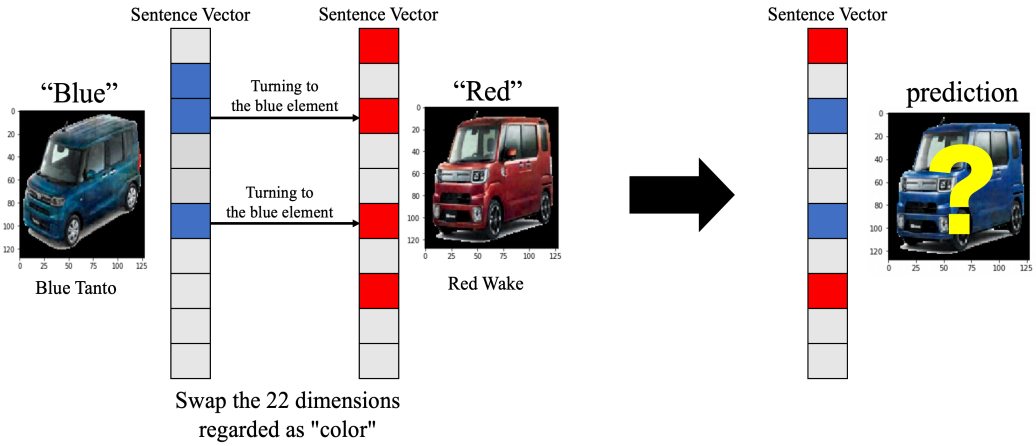


Fig. 16. Vector replacement

the two vectors to be considered a match between the two vectors was increased to 0.5.

Consequently, the number of possible dimensions for "color" was 346, and the cosine similarity of the sentence vector replaced by "blue tanto" increases to 0.482.

The threshold was further changed to 0.8. The number of dimensions judged to represent color was 532, and the cosine similarity of the sentence vectors of the sentences we replaced with "blue tanto" increased to 0.646.

Because there was not much change in the cosine similarity between sentence vectors with varying color information, we examined the changes in the images generated from these sentence vectors. The flow chart is shown in Fig. 17. The following processing is performed on a model that generates images from the learned sentences. The sentence vector output from Sentence-BERT was processed as shown in Fig. 16. Sentence vectors that had undergone color transformation are input into the decoder and converted into images.

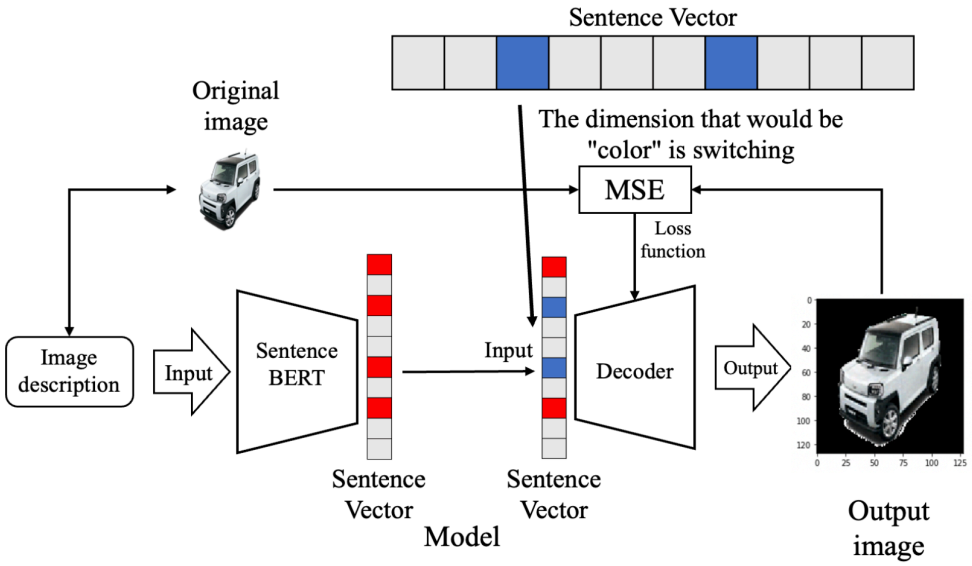
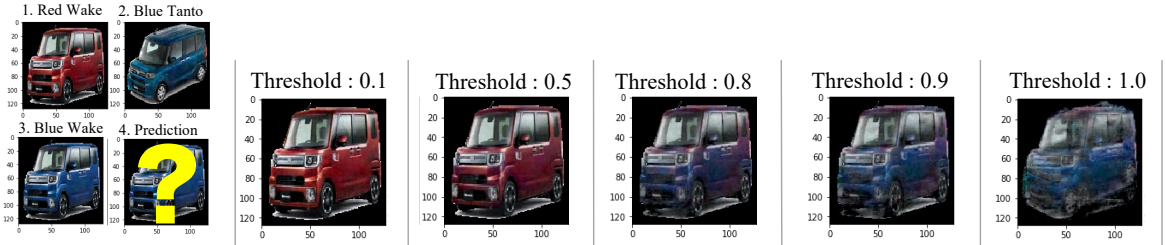


Fig. 17. Visual Verification

8. Results of Experiment2

The cosine similarity of the sentence vectors obtained by exchanging color information and the resulting images is shown in Fig. 18 when the threshold value at which an element of a sentence vector

is considered as a match is varied from 0.1 1.0.



Number of dimensions Considered "color"	22	346	532	585	633
cos similarity between 3 and 4	0.558	0.714	0.790	0.786	0.774
cos similarity between 1 and 4	0.991	0.815	0.661	0.622	0.581

Fig. 18. Results of visual verification

The results of Fig. 18 indicate that the threshold of 0.1 does not change the generated image, just as the cosine similarity between the sentence vectors does not change. We observed that as the threshold increased, the red elements in the output car image decreased, and the blue elements in the output car image gradually appeared. However, when the threshold was raised to 1.0, the shape of the car changed. This suggests that color and shape are stored in a distributed representation that cannot be separated in the sentence vector easily.

9. Conclusion

The results of the first experiment suggest that Sentence-BERT outputs a sentence vector after capturing the meaning of the sentence. We confirmed that the sentence vectors correctly captured the relationship between sentences based on our experimental results.

The second experiment confirm that although there were differences in the information represented by each dimension as distributional features of the sentence vector, this information overlapped substantially.

In the future, we plan to further improve sentence vectors by taking advantage of this property.

Acknowledgments

This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (C) Number: 20K11978. Part of this work was carried out under the Cooperative Research Project Program of the Research Institute of Electrical Communication, Tohoku University.

We would like to express our deepest gratitude to Daihatsu Motor Co., Ltd. for providing the car image datasets used in this project.

References

[1] J. Devlin, M. W. Chang, K. Lee, K.Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Proc. NAACL-HLT2019.

[2] G. Koch, R. Zemel, R. Salakhutdinov, "Siamese Neural Networks for One-shot Image Recognition", ICML deep learning workshop, vol. 2. 2015.

[3] N. Reimers, I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks", Proc. EMNLP 2019, pp. 3982–3992, 2019.

[4] M. Izumi, K. Jin'no, "Feature analysis of latent vectors of Sentence-BERT by UMAP," Proc. The 2022 IEICE NOLTA Society Conference, NLS-17, 2022.

[5] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents", arXiv:2204.06125, 2022.
[6] C. Saharia, W. Chan, S. Saxena, and et. al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv:2205.11487, 2022.
[7] E. Hoffer, N. Ailon. "Deep Metric Learning using Triplet Network", In ICLR workshop, 2015.
[8] A. Conneau, D. Kiela, "SentEval: An Evaluation Toolkit for Universal Sentence Representations", arXiv:1803.05449, 2018.
[9] acl-tohoku/bert-japanese: https://github.com/ cl-tohoku/bert-japanese,
[10] sonoisa / sentence-bert-base-ja-mean-tokens-v2 https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens-v2, 2021.