Paper 5029-2020

Transfer Learning for Mining Digital Phenotype by SAS® Viya®

Satoki Fujita, Shionogi & Co., Ltd.; Ryo Kiguchi, Shionogi & Co., Ltd.;
Yuki Yoshida, Shionogi & Co., Ltd.; Katsunari Hirano, Shionogi & Co., Ltd.;
Yoshitake Kitanishi, Shionogi & Co., Ltd.;

ABSTRACT

With the remarkable development of various AI and machine learning methods in recent years, various advanced technologies based on collected data have been born in each industry. One of the technologies represented by them is deep learning. Image recognition, speech recognition, and natural language processing are well-known usage applications for deep learning. Computers automatically extract important features that affect the results from data without human intervention. This method achieves recognition and identification accuracy that is much higher than that of the method, and is not inferior to humans. By using this, from data collected from digital devices that are now widespread worldwide, such as posts to social media such as SNS and blogs, call logs from smartphones, and accelerometer data obtained from sensors, it may be possible to discover and quantify the human phenotypes, so-called digital phenotypes, that characterize the owner, which cannot be easily discovered by human hands. However, the implementation of deep learning requires a large amount of training data. Although barriers to obtaining big data have disappeared from several years ago, there are still many cases where it is difficult to obtain a sufficient amount of data depending on the target problem and the environment where it is located. One way to solve such a problem is transfer learning. In this paper, we implemented transfer learning in SAS® Viya etc. as a method that can be expected to be applied to digital phenotyping in the situation where the amount of data is insufficient. In particular, we examine the usefulness of transfer learning by dealing with cases where deep learning is used to determine whether or not a sticky note is attached in a document image. Furthermore, we try to find out the characteristics of the author and the story by applying transfer learning to the case that identifies the author from a sentence of the book. It imitates mining phenotype.

INTRODUCTION

With the development of computers in recent years, an environment that can collect and process a large amount of data has been set up, and deep learning methods such as Convolutional Neural Network(CNN) have achieved great success in solving various problems of artificial intelligence. Especially in the field of image analysis, the techniques have evolved remarkably, and image recognition has reached the recognition level with more accuracy than humans. In the field of natural language processing, breakthroughs are expected in the future due to the emergence of a new deep learning model called Bidirectional Encoder Representations from Transformers(BERT), which was recently announced. Along with this, the area where technology can be applied has also expanded significantly, and it has contributed to solving various issues such as inspection of products and equipment, object recognition in automatic driving and unmanned drones, realization of smart speakers using interactive systems, machine translation, etc.. One of the factors in which deep learning plays an active role is the unique characteristic of deep learning, in which machines automatically capture and extract useful features of data from learning data. Due to this property, there is no need for the person to search for features that might affect the task result, and useful features that cannot be grasped by human hands may be

introduced. In the healthcare field, for example, deep learning technology has been diverted to diagnostic and evaluation technologies using images, and support for physician diagnosis has been realized, such as detecting specific tissues that are hard to find with the physician's eyes. It is natural and interesting to pay attention to these characteristics of deep learning and to acquire from the behavioral data of an individual the characteristics of that person, that is, the phenotype that has not yet been recognized and quantified. Especially in recent years, digital devices such as personal computers, smartphones, and various wearable devices have spread worldwide, and given that abundant data can be obtained from them, we focused on the "digital phenotype" hidden in these data. By linking data collected by digital devices, such as posted texts / posted photos to SNS and blogs, call data, and action logs etc., to the user's disease information with deep learning, it may be possible to discover new causal relationships and signs of disease.

However, when trying to realize the above by deep learning, there is a big problem that a large amount of training data is required. Especially in the medical field, it is often difficult to prepare sufficient training data. For example, in clinical trials for drug development conducted by pharmaceutical companies, the number of subjects is often hundreds, and it is difficult to obtain a sufficient amount of learning data. Transfer learning is one of the means to overcome this situation. Transfer learning is a method in which a model learned in one task is diverted to another task, and learning can be performed more efficiently with less data than learning from a place without any prior information. Although research on transfer learning has existed for decades, transfer learning in the field of machine learning began to attract attention after 1995 [1]. In the 2010s, the release of datasets for large-scale object recognition, such as Image Net, and the improvement of computer processing capacity would show the overwhelming object recognition performance of deep learning methods led by CNN. Attention has been given to the question of how well a network trained on large data for a particular task can transfer to other different tasks [4]. With this trend, methods of transfer learning have been studied, and their usage has been established. Until now, such transfer learning methods have been developed and used mainly in the field of image analysis. However, with the announcement of a transfer learning model for natural language processing called BERT [5] developed by Google in 2018, it is expected that transfer learning will be actively used in the field of natural language processing in the future as well.

In the future, we intend to use deep learning to detect digital phenotypes from data overflowing on digital devices and develop them into product innovations. However, as mentioned earlier, there are many situations where there is not enough training data, so the option of transfer learning is considered essential. In this study, as a first step in introducing transfer learning, we compared efficiency and accuracy with and without transfer learning by trying the task of automatically determining whether a sticky note is included in a document image with CNN. Then, as the second example which imitates mining phenotype with transfer learning, we deal with the task of author identification from sentences.

In this study, SAS Viya and Python were used as analysis environments. SAS Viya is characterized by its openness that can be operated in various languages such as Python, R, and Java, and the simplicity of requiring no preprocessing for reading image data. Therefore, SAS Viya was a useful tool in this study. However, some methods are not yet supported, so we used Python too.

Chapter 1 gives an overview of deep learning, which is the basis of transfer learning, and then describes CNN, a special model of deep learning often used for image analysis. In Chapter 2, we explain the concept of transfer learning. In Chapter 3, we outline BERT, a transfer learning model in the field of natural language processing. Chapter 4 introduces examples of transfer learning that are applied to QC (Quality Check) work and examples that mimic phenotype mining. A summary is given in Conclusion. The images in the figures in Chapters 2 and 3 are quoted from Image-net (http://www.image-net.org/).

1. DEEP LEARNING

This chapter describes deep learning, which is the basis of transfer learning. First, the basic concept of deep learning is described, and then CNN, a representative model of deep learning, used in the verification of this paper is explained.

1.1. WHAT IS DEEP LEARNING?

Deep learning is an approach in which a computer learns from a graph with multiple hierarchies for an artificial intelligence task and finds features. Usually, the subjective actions we take for granted in everyday life, such as distinguishing between sounds and voices and recognizing what is seen in photographs. It is difficult to formally describe intuitive tasks. Therefore, we focus on learning how various features included in given data, such as age and gender, are correlated with the results. However, in terms of what features should be extracted, the performance of the conventional machine learning method greatly depends on its expression, and it was difficult to design the features manually. However, in deep learning, by leaving that part to the computer, it is possible to obtain an appropriate representation of the data and explain the causes of fluctuations in the observed data.

As an example, we describe an image of how a deep learning model extracts and identifies a cat image (see Figure 1.). When a cat image is given as data, the luminance information (three primary colors or black and white) of each pixel of the image is input in the input layer. Using those pixels as input, the first layer of the hidden layer compares the brightness of neighboring pixels and detects places (edges) that change discontinuously. Then, in the second hidden layer, we search for corners and outlines that can be recognized by combining them. The resulting corners and contours are further combined, and the third hidden layer detects parts of the object, such as the ears and nose. Finally, it has a structure that recognizes what the input image is from each part obtained in the third layer. In other words, deep learning successfully acquires complex concepts that are difficult to express by dividing them into simple concepts and nesting them.

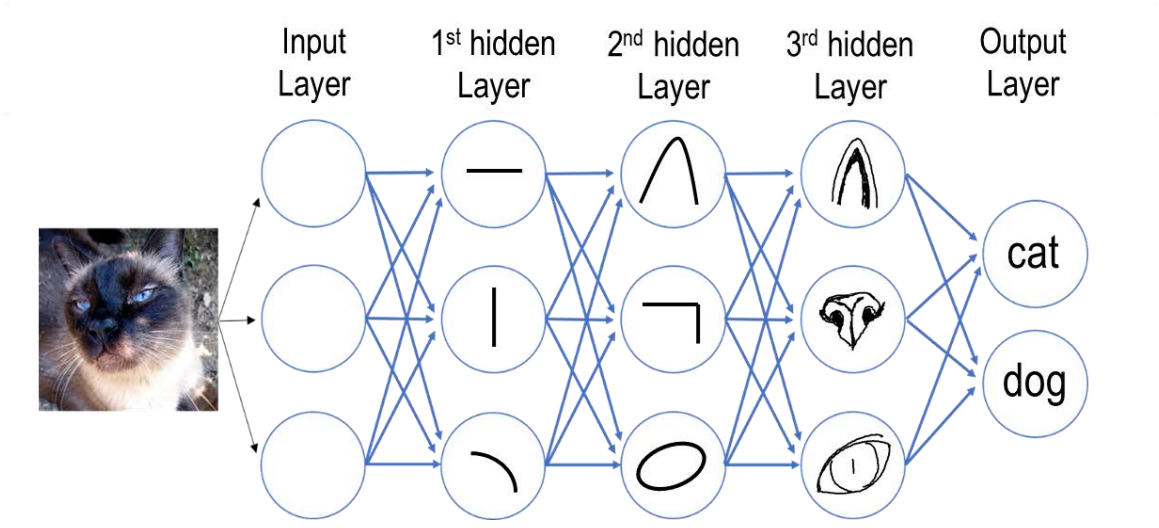

Figure 1. Image of deep learning (assuming the task of classifying cats and dogs)

The formula that represents the model for ordinary deep learning is as follows.

$$h^{(i)} = g^{(i)}\left(W^{(i)^T} h^{(i-1)} + b^{(i)}\right).$$

Here, (i) represents a layer, and in particular, $h^{(0)} = x$ represents an input layer and $h^{(n)}$ represents an output layer. $W^{(i)}$ and $b^{(i)}$ represents the parameter to be estimated, and $g^{(i)}$

is the activation function. Rectified linear unit (ReLU) defined by $g(x) = \max(0, x)$ is well known and used as standard. The above equation represents a model where the hidden layer is the $n-1$ layer and the width of the i-th layer is the dimension of $h^{(i)}$.

1.2. CONVOLUTIONAL NEURAL NETWORK

Various models of networks in deep learning have been proposed, but CNN (Convolutional Neural Network) exists as one of the most frequently used models for data with a grid topology, such as time-series data and image data. As the name implies, CNN is a special model of deep learning that performs convolution processing at a certain layer in the network. First, we will explain what convolution processing is. If a two-dimensional image I is input and K is a weight function (here called a filter), the convolution process is defined as follows.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n)$$

$(i.j)$ represents the pixel at row i and column j of the 2D image. Figure 2. shows an image of this process.

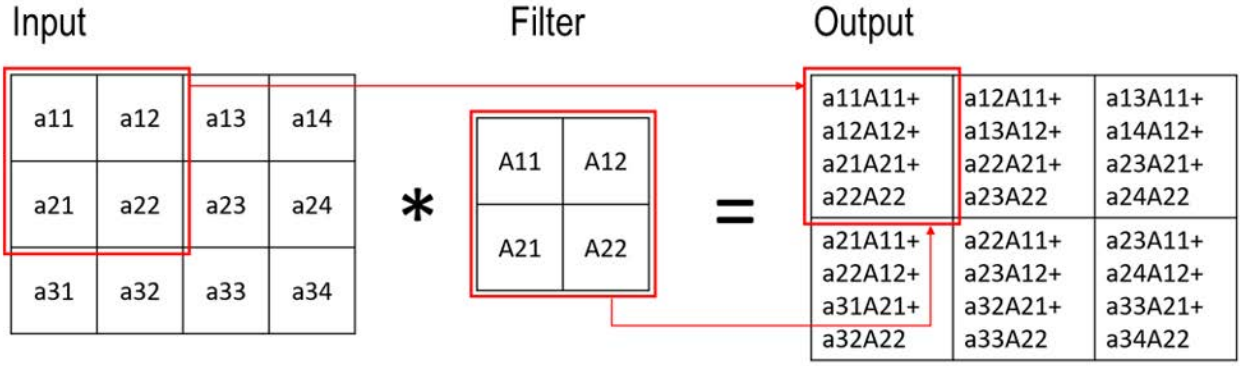

Figure 2. Example of 2D convolution

Next, the basic configuration of the CNN model will be described. A typical CNN layer consists of convolution and pooling layers, as shown in Figure 3..

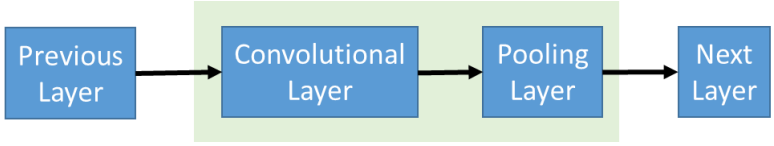

Figure 3. Typical CNN components

In the first stage, the convolution layer, the convolution process described above is performed, and the output is normalized via a nonlinear activation function such as ReLU. Then, in the second stage, pooling is performed. Pooling is an operation that replaces the output of the network with one summary statistic for each fixed area. One of the commonly used functions for pooling is max pooling, which returns the largest output in the vicinity of a rectangle (see Figure 4.). In addition, there are other pooling by means of the neighborhood of the rectangle and L^2 norm.

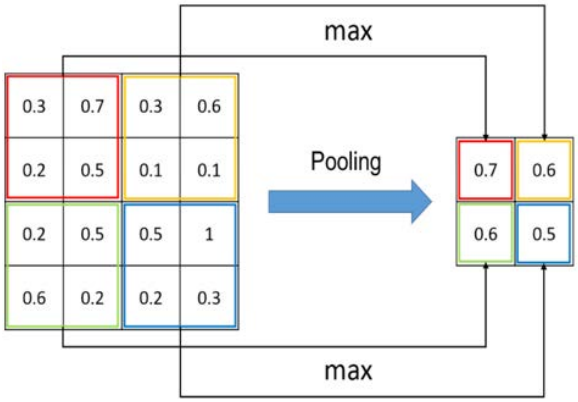

Figure 4. Pooling (max pooling)

There are two main advantages of pooling. The first is that by performing pooling, it is possible to form an expression that is almost invariant to minute movements of the input. This is especially useful when you are more concerned about whether or not a feature exists in an image than the exact location of that feature in the image. Second, the input size in the next layer can be reduced in order to replace certain regions with their summary statistics. This leads to an improvement in the computational efficiency of the network and a reduction in the amount of memory required to hold the parameters. A series of operations of convolution, normalization, and pooling are performed several times with the addition of such a pooling layer. Finally, it is connected to the fully connected layer, and the result is output on the output layer. This is the general model of a CNN network. In the output layer, if the purpose is to classify, an activation function called Softmax function, defined below, is often used.

$$\mathrm{softmax}(z)_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}, \quad i = 1, \ldots, D$$

D is the number of classes, and $z = W^T h + b$ is the output of the fully connected layer. Among $i = 1, \ldots, D$, the class that takes the value of the largest Softmax function is the class derived by CNN for the input data.

2. TRANSFER LEARNING

This chapter describes transfer learning, which is the main subject of this paper. Transfer learning is based on the idea of diverting what has been learned in one task to help improve generalization capability in another, and transfer learning performs two or more different tasks. As examples, consider the task of classifying cat and dog images (Task 1), and the task of classifying dolphin and shark images (Task 2). If there are a large number of images in Task 1, the learning in Task 1 can be used to quickly learn useful expressions from a small amount of data in Task 2. Specifically, the weight of the lower layer (here, $W^{(1)}, W^{(2)}$) in the network learned in Task 1 is diverted to Task 2, and in Task 2, the weight of the upper layer (here, $W^{*(3)}, W^{*(4)}$) only learning (see Figure 5.).

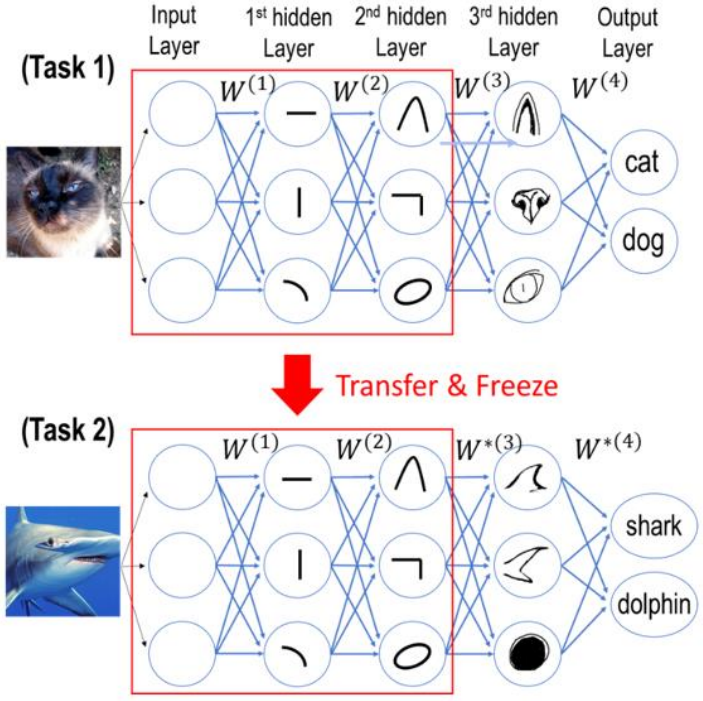

Figure 5. Image of transfer learning

Many visual categories, such as images, share underlying concepts such as edges and visible shapes, the effects of geometric changes, and potential factors such as lighting changes. Therefore, even if the goals of Task 1 and Task 2 are different,

the learning result of the lower layer of Task 1 can be diverted to Task 2. In other words, in Task 2, the upper layer depends on the task, but on the other hand, the lower layer is the feature extractor learned in Task 1. Here, the lower layer is diverted, but the upper layer may be diverted depending on the task. Speech recognition systems correspond to this. In this way, sharing parameters corresponding to expressions shared between different tasks often improves generalization capability if the sharing is appropriate. Here, the most typical method in transfer learning is assumed, in which the lower weights transferred (Transfer) are fixed (Freeze), and the lower layer of the transfer source is directly used as a feature extractor for the target task. However, in some cases, the learned weights are used as initial values without fixing the transferred weights, and re-learning (Fine tuning) is performed including the lower layers. In this case, learning is further performed from a good initial value, and faster convergence to a better value can be expected than learning all weight parameters from a place where there is no prior learning at all. Also, by re-adjusting the weight parameters including the diverted part, the model can be more suitable for the target task. Transfer learning approach provides an efficient and effective way to learn by applying the experience gained in one area to another. Therefore, if there is nothing shared between tasks or the network is too different, transfer learning may not be effective.

3. BERT

This chapter outlines BERT that realizes transfer learning in the domain of natural language processing.

3.1. TRANSFORMER

This section briefly describes Transformer, the deep learning model on which BERT is based. In natural language processing, it is necessary to consider the entire input data (sentence in this case) and its word order, unlike image analysis that handles images that are data that can be meaningful for local parts alone in the image. So far, models that process text sequentially, such as Recurrent Neural Network (RNN) and Long short-term memory (LSTM), have been developed and used. However, in these models, since words are input to the model one word at a time from a sentence, there is a drawback in that it takes several tens of times longer than other deep learning model such as CNN which can process all data collectively in one step. Of course, using CNN for language processing enables high-speed processing. But, in sentence data, there are many cases where words at distant positions are related, and CNN cannot consider this because only the information of several adjacent words is considered. To address these problems, Transformer, a model that incorporates the mechanism called Attention that can process relationships between distant words at once, has been developed.

Transformer is the model that converts each word of the input sentence data into a vector expression, adds the position information of the word in the sentence by Positional Encoding, and performs the feature conversion by repeating the following mechanism multiple times (see Figure 6.).

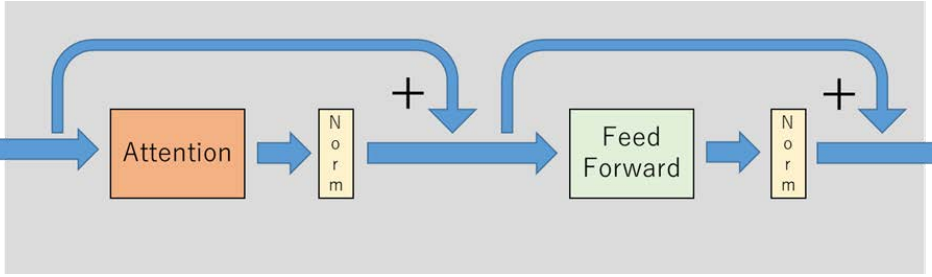

Figure 6. Core components of Transformer

Attention in the above is a mechanism for scoring each target word based on which of any words in the sentence should be noted to give the meaning of the target word. The internal operations are as follows: First, the word (Target) to be searched is converted to $\boldsymbol{Q}$ (query), and the sentence data (Source) for checking the relevance to the word is converted to $\boldsymbol{K}$ (key) and $\boldsymbol{V}$ (value) by using the fully connected layer. Then, $\mathrm{softmax}(\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}})\boldsymbol{V}$ is the output of Attention. Here, $\boldsymbol{K}$ serves as an index for each word in the sentence. In other words, firstly, the inner product of $\boldsymbol{Q}$ and $\boldsymbol{K}$ derives the weight of how relevant the search word is to each word in the sentence (attention weight), and then, in the form of comparing with the weight, the weighted sum of $\boldsymbol{V}$ corresponding one-to-one with $\boldsymbol{K}$ is obtained as the output (see Figutre7.).

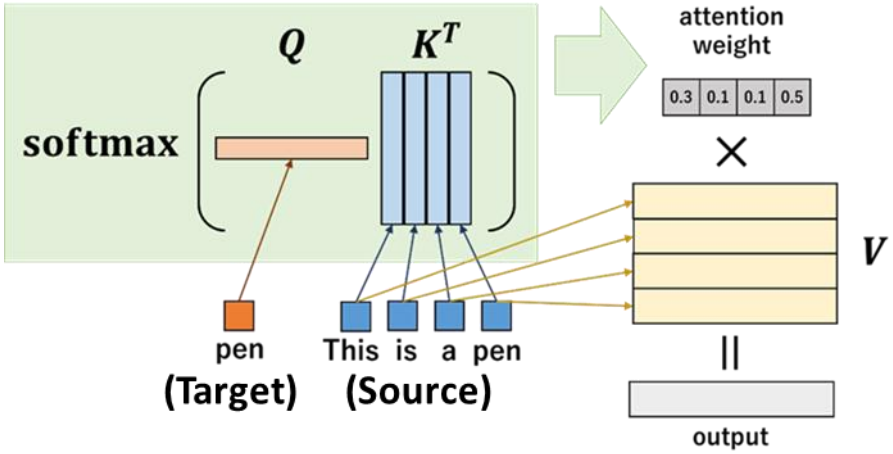

Figure 7. Operations in Attention

Here, the purpose of dividing Source into $\boldsymbol{K}$ and $\boldsymbol{V}$ is to obtain high expressive power by non-trivial conversion between $\boldsymbol{K}$ and $\boldsymbol{V}$. The output derived by Attention is normalized, added to the value before applying Attention, and input to the next FeedForward layer.

An advantage of Transformer configured as described above is that it can achieve the same performance using Attention alone without performing recursive processing like RNN. In addition, since parallel processing is possible, it is possible to learn much faster than RNNs.

3.2. BERT

BERT is a pre-training model that gains the ability to learn context in advance by learning in a large corpus and use it to transfer learning to various other tasks such as translation, summarization, document classification, and sentence generation. In the area of natural language processing, several pre-training models such as ELMo have been proposed before BERT. However, they predict the next word in a sentence as pre-training, but use only past word information without using future word information so as not to cheat at that time. The model structure was such that only unidirectional learning could be performed. So, this model makes it difficult to express the word in a sentence based on both all words before the word and all words after the word. It leads that the expressive power is somewhat limited. On the other hand, BERT has multiple Transformers, so it can extract information from the whole input word group at once. Furthermore, since Transformer is connected in a fully connected manner, the network extends from both right and left word position groups for a specific word position. Then, bidirectional learning of the context is realized by pre-training with two types of language tasks described below, which were designed to be able to learn models of such shapes.

First, describe the input format to BERT before explaining the two pre-training tasks that are the key to BERT. In BERT, as shown in Figure 8., a special token called [CLS] is inserted at the beginning of the entire input. Usually, when solving a classification problem, this

token have the features of the entire input, and is used for classifying. Also, two sentences can be handled at the same time so that a classification problem using sentence pairs as input can be handled, and the sentence is separated by a token [SEP].

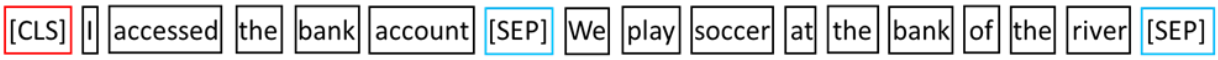

Figure 8. Input format in BERT

Next, describe the two pre-training tasks that is the core of BERT. The first pre-training task is the Masked Language Model (see Figure 9.). This is a task that randomly masks multiple words from the input text and predicts the masked part from the unmasked part. With such a task, learning is possible even with the BERT architecture that passes the token sequence to the input at once.

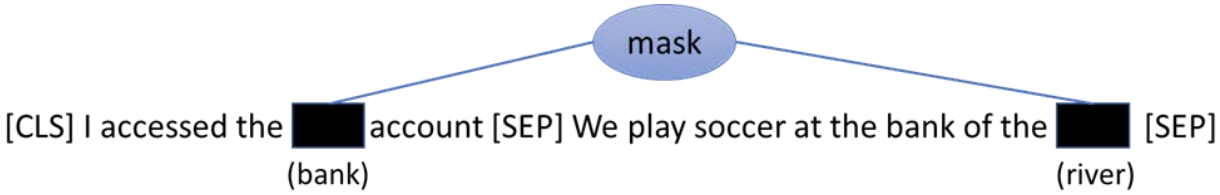

Figure 9. Masked Language Model

The second pre-training task is Next Sentence Prediction (see Figure 10). In this task, two sentence data are given, and BERT try to determine whether the two sentences are two consecutive sentences that are connected in context. In this way, learn the relationship between sentences.

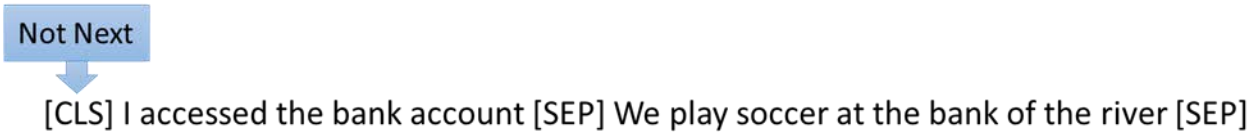

Figure 10. Next Sentence Prediction

A feature common to the above two pre-training tasks is unsupervised learning. In both tasks, there is no need to manually label the answer, and the teacher data is automatically generated by setting random numbers in the program. The ability to train the model to understand context simply by reading a large amount of sentences without teacher labels is an attractive feature where large-scale labeled data is difficult to obtain in these days.

Based on the above, summarize the features of BERT. First, Transformer and two pre-training tasks enabled simultaneous learning in both directions, enabling the creation of more context-dependent vector representations. For example, even if two words have exactly the same notation but different meanings, such as the two "bank" that appear in the example above, the context is learned by considering the surrounding words so that the Masked Language Model can be solved, and different vector representations are created accordingly. The next feature is that transfer learning as described in Chapter 2 is now possible even in the task of natural language processing. By setting the parameters learned in the two pre-training tasks described above in the BERT model and attaching them to the module that matches the natural language processing task that you want to perform (for example, one fully connected layer in the case of a classification problem), it can be diverted as an architecture that understands the meaning of the input words and sentences and reflects it on the feature vector.

4. APPLICATION EXAMPLES OF TRANSFER LEARNING

In this chapter, we explain the usefulness of transfer learning by two examples.

4.1. APPLICATION EXAMPLE 1 (IMAGE CLASSIFICATION)

4.1.1. Settings

As described in Chapter 3, if there was already a trained model by a large amount of data, transfer learning can be used to construct new model with good performance by only a small amount of data. In this section, we explain the benefits of transfer learning for a small amount data by comparing between the case with transfer learning and the case without transfer learning. In the first example, we used scanned-documents with the sticky note and without the sticky note.

In our creation processes of Certified Copy for eTMF, there are some check processes. One of the processes is the visual check whether or not the scanned-document of source document includes sticky notes. Therefore, we had a motivation to create an automatic classification tool whether the scanned-document includes sticky notes.

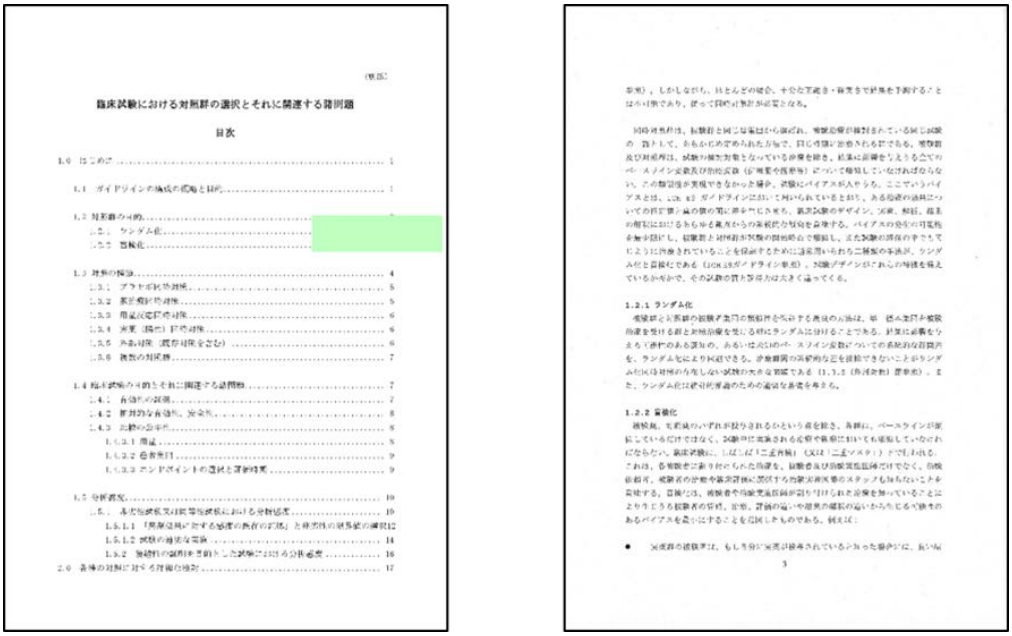

Figure 11. Image data example (left: tag, right: no_tag)

The number of prepared training image data and test image data is as follows.

	For training	For test
sticky note (tag)	400	100
no sticky note (no_tag)	400	100

Table 1. Number of image data

CNN is often used for image classification, there are VGG16, ResNet50, InceptionV3, InceptionResNetV2, MobileNet and DenseNet201 as famous CNN models. Also we got the VGG16 model trained by a large-scale image datasets called ImageNet, and implemented this image classification by this trained VGG16 model because the data prepared for learning is small.

VGG16 is a deep learning network model consisting of a total of 16 layers with 13 convolutions and 3 fully connected layers (see Figure.). Originally, it was a model trained by a large-scale image dataset called ImageNet in order to classify 1000 classes of images in ImageNet Large Scale Visual Recognition Challenge. The structure of VGG16 model is as follows.

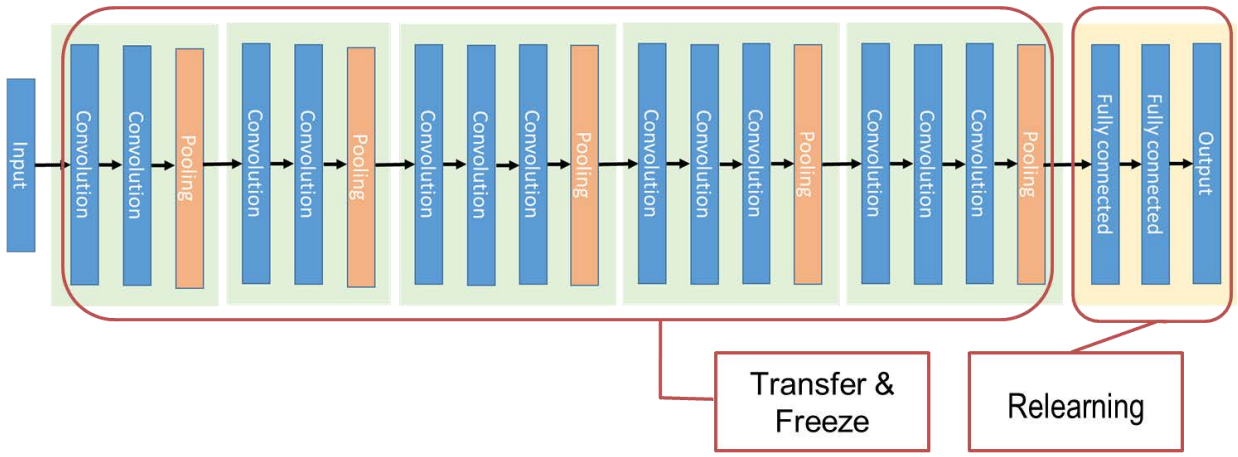

Figure 12. VGG16 model structure

The weight parameters of the convolutional layer of trained model were kept with no changes, and only the weight parameters of the last fully connected layer were trained by 800 training images. In other words, the lower layer that extracts features from an image uses what has been learned from hundreds of thousands of large-scale data, and only the higher layers involved in the task whether the scanned-document includes sticky notes were re-learned by a relatively small amount of data.

For the actual implementation, SAS Viya's Python interface (Jupyter notebook) was used. SAS Viya is thus an open platform that can be connected and operated from Python. Therefore, not only VGG16, but also various models with pre-trained models such as VGG19 and ResNet50 can be used. Furthermore, in order to handle images-file by program, pre-processing such as conversion into an array of pixels was necessary, but SAS Viya is equipped with a function dedicated to image processing, and can handle image-file directly. It is a very useful tool for performing image analysis.

To evaluate the performance of transfer learning, training and classification without transfer learning were also performed. In other words, the parameters pre-trained by large-scale data were not transferred, and all parameters are learned from the beginning using only 800 training images. The model itself used the layer configuration of VGG16.

4.1.2. Result

The results of classifying 200 test images using the trained models with and without transfer learning are shown below (see Table 2.).

	With transfer learning	Without transfer learning
Number of test images	200	200
Accuracy (%)	97.5	88.5

Table 2. Classification accuracy on test images

The misclassification rate when using transfer learning was about 2.5%, while the misclassification rate when not using transfer learning was about 11.5%, and the classification accuracy is higher when using transfer learning. You can see Figure 13. below shows the history of iterative learning.

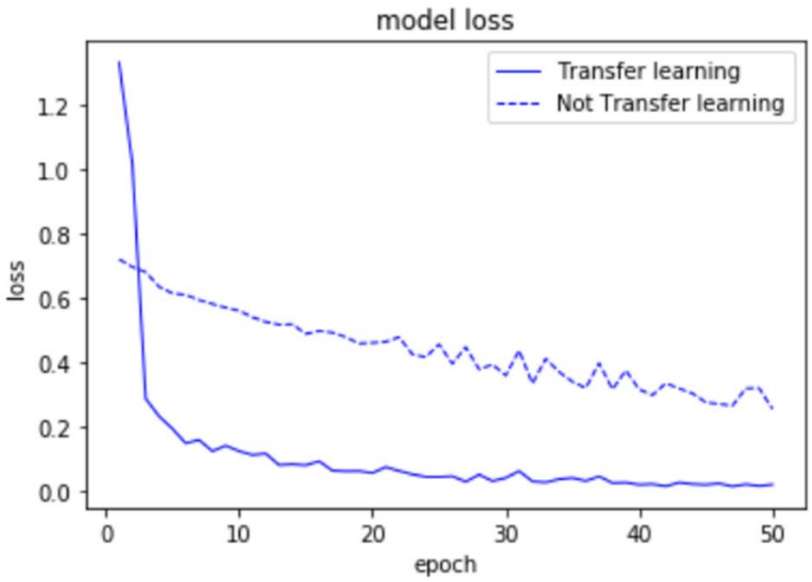

Figure 13. Training history
(solid line : with transfer learning, dotted line : without transfer learning)

In the learning curve (solid line) using transfer learning, the value of the error function (loss) approached 0 very quickly as the learning progresses (the epoch increases). On the other hand, the learning curve (dotted line) without transfer learning did not approach 0, indicated that the learning efficiency was poorer.

In addition, the effect of reducing training images was examined. In particular, the same prediction was performed with the training images reduced to 200. The training history is as follows (see Figure .). From the learning curve, the loss quickly approaches 0 when using 800 training images, and it can be seen that the learning is more efficient with 800 training images than with 200 training images. However, the correct answer rate leaded by the model trained with 200 training images was 95.5%, which did not decrease significantly.

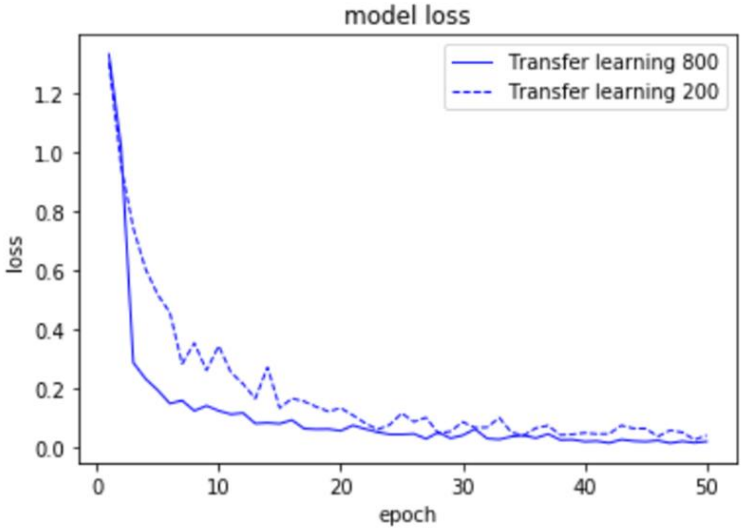

Figure 14. Training history
(solid line : using 800 training data, dotted line : using 200 training data)

From the above, it was confirmed that by using transfer learning, learning can be performed efficiently and a certain degree of accuracy can be realized even in a situation with relatively little data.

DLPy, Python package we used, had a function to visualize where the machine focused on the image by using a heat map, and the results were shown below.

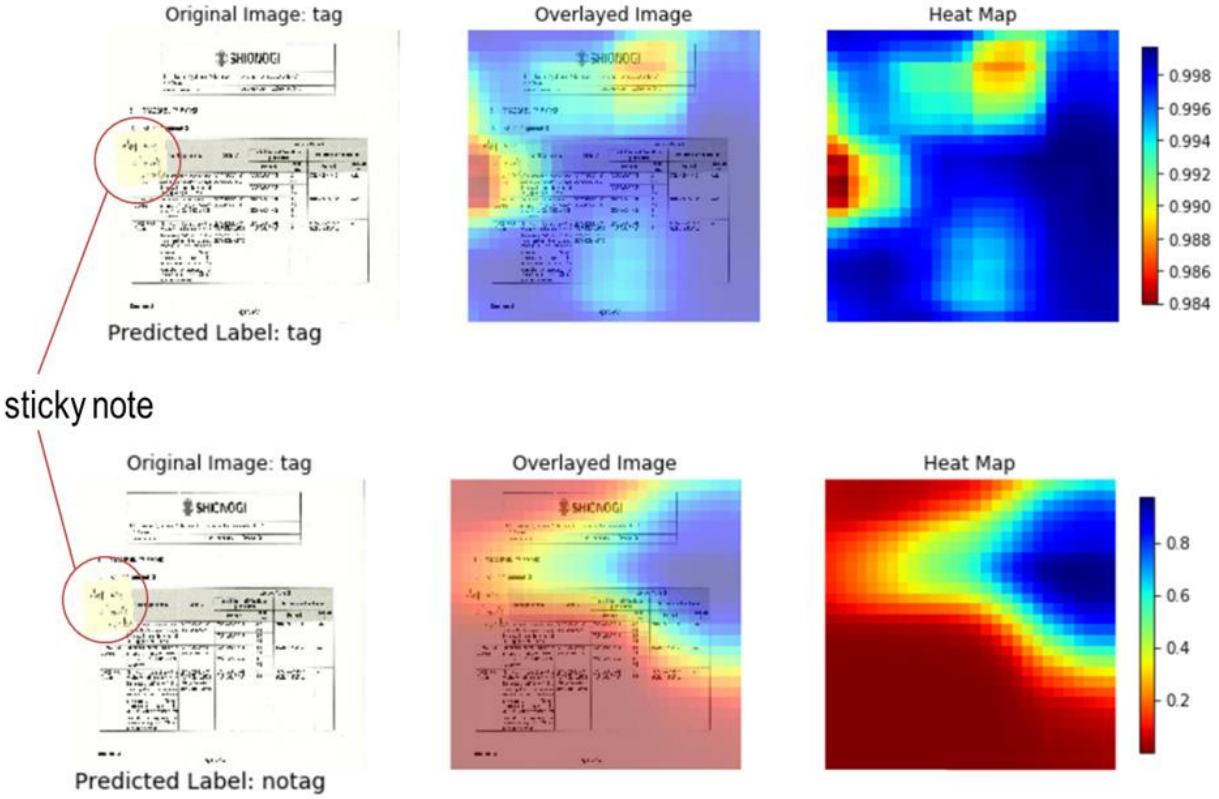

Figure 15. Example of heat map
(top: transfer learning, bottom: without transfer learning)

In Figure ., the upper figure is a heat map when transfer learning was used, and the lower figure is a heat map when transfer learning was not used. This makes it possible to visualize the information on the basis of judgment of the model, and to check the places that are affecting the discrimination by looking at the shades of blue, green, and red. From this figure, it can be seen that learning is not sufficient when transfer learning was not used, because the red area was wider.

This time, the usefulness of transfer learning was confirmed through an actual case. But, some improvements can be considered.

The first is the quantity and quality of image data. Transfer learning works with a relatively small amount of data, but of course, it is better to have a large amount of data. With more training data, in addition to the progress of learning, more layers can be re-learned in transfer learning, and the range of the model expands. In any situation, you must make every effort to obtain as much training data as possible. It is also important to train the model with a wide variety of data so that the generalization performance is improved. In this example, there are various image data patterns depending on how the original paper is scanned, the color of the sticky note, and the position where the sticky note is attached. Therefore, it is preferable that each of them is at a sufficient number. In order to prevent over-fitting and to form a model with excellent practicality, it is necessary to pay attention

not only to the structure of the model but also to the data to be learned. It is also useful to inflate an image by inverting, rotating, sliding, and changing the brightness of the learning image data when the number of training data is small. This is expected to increase the data several times and improve the accuracy of the model by several percent. However, it is important to note that such inflating must be performed after the data is divided into training data and test data. If inflating is performed before separation, data similar to the training data will be included in the test data. As a result, the accuracy of the test data becomes abnormally high even though the prediction performance is not good for completely new data, which leads to misunderstanding that it is a good model.

Another possible improvement is tuning the hyperparameters. In this case, in both cases with and without transfer learning, the number of epochs was set to 50, the batch size was set to 32, the learning rate was set to 0.01, the network was set to VGG16, and learning was performed under settings that did not perform regularization. It is necessary to verify how the result changes by changing these hyperparameters, and to set a more suitable hyperparameter.

The future task is to construct a more practical model in consideration of the above points.

4.2. APPLICATION EXAMPLE 2 (SENTENCE CLASSIFICATION)

4.2.1. Settings

In this section, we describe an application example of transfer learning in a natural language processing task using BERT described in Chapter 3. As a task close to the

I feel I am dying of weakness, and have barely strength to write, but it must be done if I die in the doing. 0
If you do not choose to understand me, forgive my impertinence. 1
When she is secure of him, there will be more leisure for falling in love as much as she chooses. 1
Am I to take it that I have anything in common with him, so that we are, as it were, to stand together; or has he to gain from me some good so stupendous that my well-being is needful to him? I must find out later on. 0

Figure 16. Input training data format

assumption of searching for digital phenotype from blogs and SNS posts, we read the sentences of a book into BERT and perform the task of predicting the author. The aim is to discover writing styles and expressions specific to the author and story. Here, binary classification of "Dracula" by Bram Stoker and "Pride and Prejudice" by Jane Austen was performed. The sentence data used for this study was downloaded from Project Gutenberg (https://www.gutenberg.org/). First, separate the original data with periods and prepare a simple preprocessed sentence as follows. As a preparation, the original data is separated by a period and a sentence with simple preprocessing is prepared as follows. After the sentence, a tab is added and a tag is added to indicate which book the sentence came from. The tag 0 here is a sentence of "Dracula" and 1 is a sentence of "Pride and Prejudice". The number of sentences in the prepared training data and test data is as follows.

	For training	For test
Dracula	4500	500
Pride and Prejudice	4500	500

Table 3. Number of sentence data used

Next, the model used is described. In this case, we used a model in which a fully connected layer for classification was attached to a BERT-Base model that was pretrained by using a corpus such as Wikipedia. That pre-trained model has been published by huggingface (https://github.com/huggingface/transformers). As can be seen from the figure below, the model structure consists of 12 Transformer blocks.

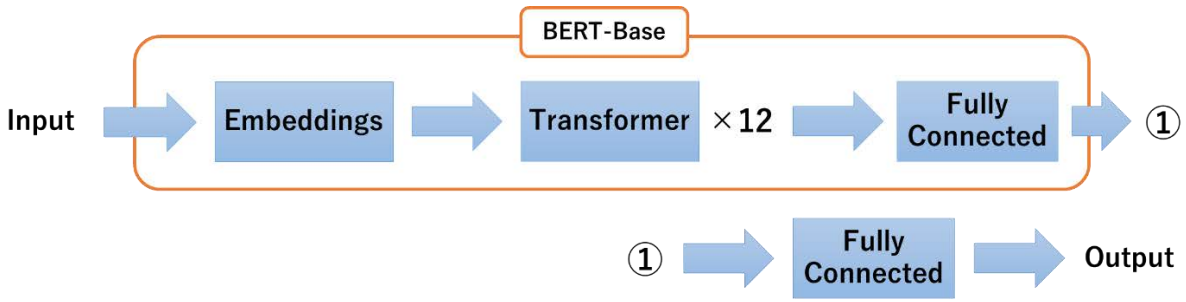

Figure 17. Overview of the BERT model structure applied this time

Fine-tuning is performed on this model. In particular, the last fully connected layer attached for the classification task and the last one Transformer of the pre-trained BERT-Base model part was retrained using the sentence data described above.

4.2.2. Result

The result obtained by inputting test data to the BERT model described in the previous section is as follows. For comparison, the result in the case where the model in which two layers of the Transformer are stacked is performed without transfer learning (BERT) is also shown.

	With transfer learning (BERT)	Without transfer learning (Transformer)
Number of test data	1000	1000
Accuracy (%)	90.1	86.4

Table 4. Classification accuracy on test data

From the above table, it can be seen that the prediction accuracy when using BERT exceeds 90%, and the classification is relatively correct. Here, the number of epochs was 3, the batch size was 32, and the learning rate was 0.00001.

Next, we visualize the rationale for the classification performed by the model. Specifically, we consider the attention weight in the last Transformer part of the BERT model as a measure of how much attention the model is paying to the word for the classification, and give a shade of red according to its weight size. In other words, it can be interpreted that the darker the red word, the more the model used for the basis of classification judgment. However, there is no theoretical guarantee that Attention weight is really an explanation of the basis of judgment, and counterexamples [7] have been given, so it should be noted that it is used as a guide.

Here is a visualization of Attention weight for some sentences in the test data. There are sentences which were predicted to be one sentence of "Dracula" (see Figure 18.). Words such as "I", "we", and "myself" have attracted attention in sentences ①②③, and well represented the characteristics of "Dracula", which is drawn in the style of letters and diaries, newspaper articles etc.. This can be seen from the fact that the word "writing" has attracted attention in ⑥. Although omitted here, we could also confirm sentences that focused on "written" and "letter". In ④ and ⑤, words such as "madman" and "horror" are attracting attention, and this is understandable given the contents of books depicting extraordinary things that doubt their own normality.

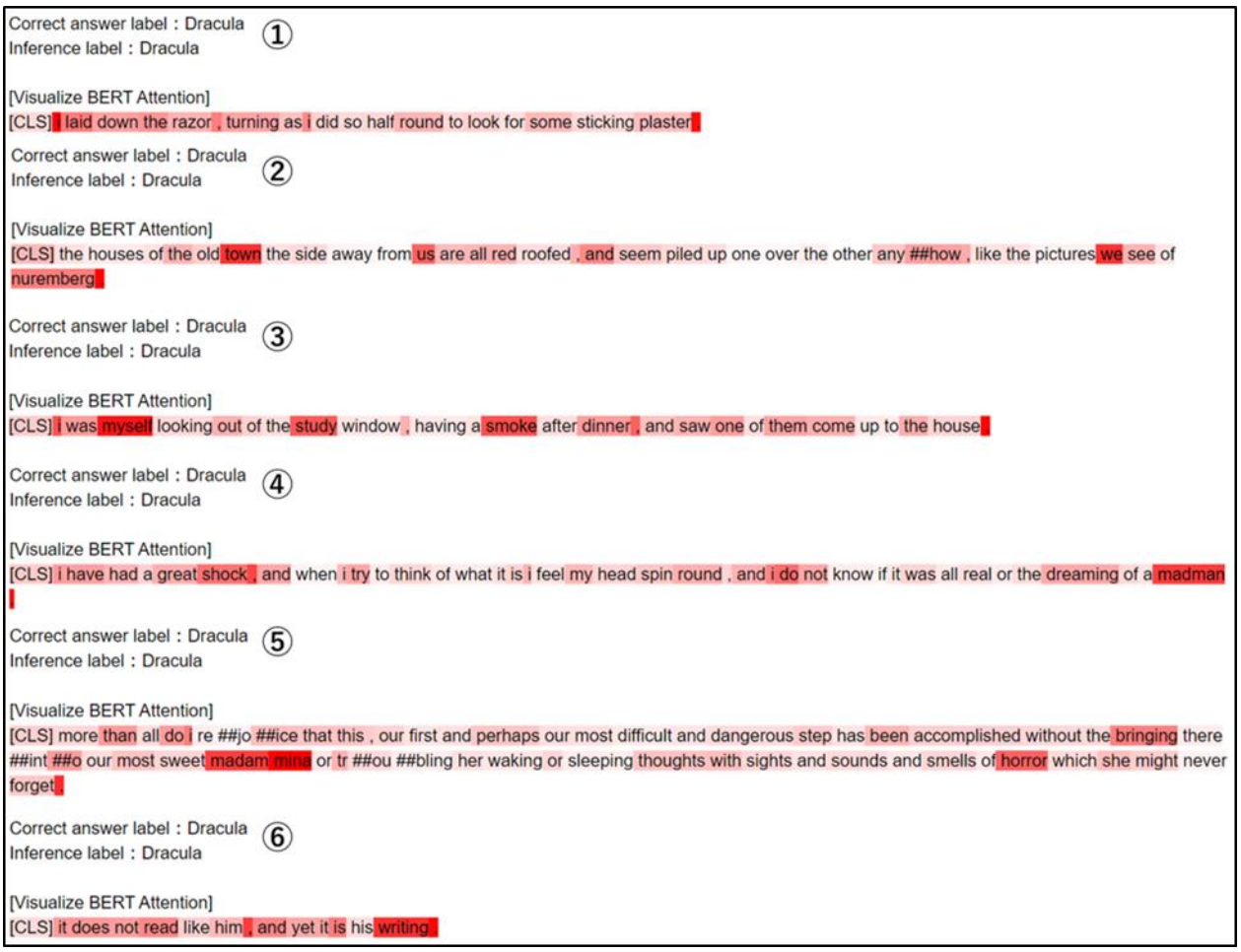

Figure 18. Visualization of Attention in Dracula

There are sentences which were predicted to be one sentence of "Pride and Prejudice" (see Figure 19.). From ⑦ and ⑧, "she" and "her" are attracting attention, and clearly show that "Pride and Prejudice" is a story centered on women. Also, in ⑧ and ⑨, attention is paid to words that represent relatives, such as "uncle" and "mother", which indicate the characteristics of stories in which stories are also developed about their own families. ⑩ and ⑪ pay attention on words such as "vanity" and "moral". "Pride and Prejudice" focuses on what others think of yourself and your family, so words that indicate a person's mental characteristics are weighted. For ⑫, the word "conjecture" is red. This word is a formal expression, and it is possible that the author prefers to use it. In other words, it may be a phenotype that represents the characteristics of the author.

Figure 19. Visualization of Attention in Pride and Prejudice

Above, the cases where the guess by BERT was correct are listed, so we also show the case where the guess was wrong below.

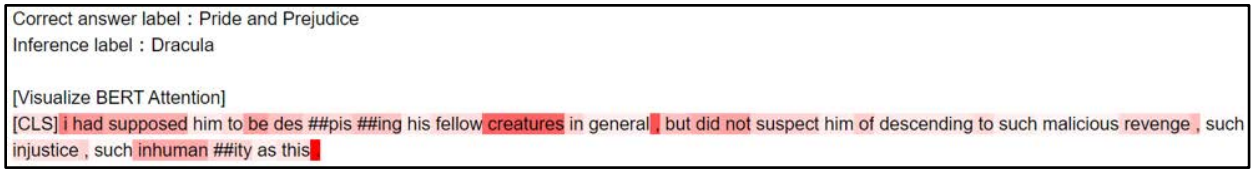

Figure 20. Example of misclassification

In this example, one sentence of "Pride and Prejudice" is incorrectly classified as one sentence of "Dracula". The sentence focuses on the word "creatures", which has different meanings in the two stories. It may have been necessary to look at the context of the sentence before and after in order to grasp the meaning of the word correctly. However, in this case, the input was only one sentence, so the meaning of the word could not be fully understood, leading to the misclassification. If the input is two consecutive sentences, the meaning of the word may be properly grasped.

In this section, by visualizing Attention, we found words that seemed to be related to the basis of BERT's judgment and interpreted them as features of stories and authors. However, it is not clear whether the words listed above really represent the features of the story, or even the phenotype of the author, because the binary classification of only two books was performed this time. In order to capture more certain features, it is necessary to use multi-class classification of various stories and to use multiple stories of the same author as

training data. In the future work, including those trials, we will try more practical tasks in order to lead to the discovery of useful phenotypes.

CONCLUSION

In this paper, we considered digital phenotype mining as a future motivation, and focused on transfer learning that can be used in the mining, and confirmed its usefulness. In particular, we applied transfer learning in image classification to verify the efficiency of transfer learning, and then used transfer learning in sentence classification as an example of imitating practical phenotype mining. As a result, it was confirmed that the use of transfer learning improved model accuracy and learning efficiency, and it could be used for digital phenotype mining in the future.

There are not many scenes in which a sufficient amount of labeled data can be prepared for training a deep learning model from the beginning. In such a situation, transfer learning shows its worth. Of course, when performing transfer learning, it is necessary to determine what part of the model shares between tasks, how to shape the model structure, and how to tune during learning. There are still many issues to consider. Roughly transferring the layer weights of the model trained by large amounts of data without thinking would not yield valid results in some situations. Under the target task and available data, we should judge whether there is a trained model that can be used transfer learning and which layers will be applied fine-tuning etc. It must be determined appropriately according to the data environment in which it can be performed.

For the implementation of transfer learning, we used DLPy, a package for the Python API, on SAS Viya. DLPy is coding similar to Keras, a package for deep learning in Python, and can easily build and train deep learning models. In addition, in the use of CNN for image analysis, since all the extraction of features is left to the machine, it is black-boxed on what basis the machine judges the image. Although there is a issue of interpretability, as we saw in Chapter 4, DLPy has a function to display where the machine focused on the image with a heat map, so that the results can have some explanatory power. SAS Viya can use this convenient package due to its openness, it can handle images themselves as image types instead of pixel arrays, so SAS Viya is a very useful platform for image analysis. However, some new models like BERT are not yet implemented, we can not easily handle them by SAS Viya, so future implementation is desired. In the field of deep learning, an enormous amount of activities and technological innovations have been carried out every day. In the future, we would like to make use of the openness of SAS Viya and perform analysis suited to the times.

In Chapter 4, we performed the author classification of books, which is close to phenotype mining, in the next step we hope to work real phenotype mining using real-world data. If we can link the patients with specific disease and their behavioral data, and find some digital phenotypes of the patients with the disease, this could lead to valuable innovation for a society where prevention is more important than treatment. In this process, the collection of labeled data becomes a barrier, especially in the medical and healthcare fields, including clinical trials associated with drug development. Therefore, expectations for transfer learning are great. We will continue to explore from various viewpoints to make the most of transfer learning for phenotype mining.

We hope this paper will serve as an opportunity for data scientists involved in drug development to take an interest in deep learning and eventually transfer learning.

REFERENCES

[1] Jialin Pan, S. and Yang, Q. 2020. "A survey on transfer learning." IEEE Transactions On Knowledge and Data Engineering 22.

[2] Goodfellow, I., Bengio, Y., and Courville, A. 2016. *Deep Learning*. MIT Press.

[3] SAS Institute Japan. 2018. "SAS Viya：ディープラーニング＆画像処理用 Python API 向けパッケージ：DLPy" Accessed June 22, 2019. https://blogs.sas.com/content/sasjapan/2018/05/21/sas-viya_dlpy1/

[4] 中山 英樹. 2015. "深層畳み込みニューラルネットワークによる画像特徴抽出と転移学習." 電子情報通信学会音声研究会 7 月研究会.

[5] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. 2018. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *arXiv preprint arXiv:1807.03819*.

[6] 小川 雄太郎. 2019. *つくりながら学ぶ！PyTorch による発展ディープラーニング*. マイナビ出版.

[7] Jain, S. and Wallace, B.C. 2019. "Attention is not explanation." *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long and Short Papers), 3543–3556.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at the following addresses:

Satoki Fujita

Shionogi & Co., Ltd.

satoki.fujita@shionogi.co.jp

Ryo Kiguchi

Shionogi & Co., Ltd.

ryo.kiguchi@shionogi.co.jp

Yuki Yoshida

Shionogi & Co., Ltd.

yuki.yoshida@shionogi.co.jp

Katsunari Hirano

Shionogi & Co., Ltd.

katsunari.hirano@shionogi.co.jp

Yoshitake Kitanishi

Shionogi & Co., Ltd.

yoshitake.kitanishi@shionogi.co.jp