Electronic Preprint for Journal of Information Processing Vol.30

Regular Paper

Providing Interpretability of Document Classification by Deep Neural Network with Self-attention

Atsuki Tamekuri[1] Kosuke Nakamura[1] Yoshihaya Takahashi[1] Saneyasu Yamaguchi[1,a)]

Received: September 10, 2021, Accepted: January 7, 2022

Abstract: Deep learning has been widely used in natural language processing (NLP) such as document classification. For example, self-attention has achieved significant improvement in NLP. However, it has been pointed out that although deep learning accurately classifies documents, it is difficult for users to interpret the basis of the decision. In this paper, we focus on the task of classifying open-data news documents by their theme with a deep neural network with self-attention. We then propose methods for providing the interpretability for these classifications. First, we classify news documents by LSTM with a self-attention mechanism and then show that the network can classify documents highly accurately. Second, we propose five methods for providing the basis of the decision by focusing on various values, e.g., attention, the gradient between input and output values of a neural network, and classification results of a document with one word. Finally, we evaluate the performance of these methods in four evaluating ways and show that these methods can present interpretability suitably. In particular, the methods based on documents with one word can provide interpretability, which is extracting the words that have a strong influence on the classification results.

Keywords: deep learning, new documents classification, self-attention, smooth-grad, LSTM

1. Introduction

Deep learning, has become widely used for a variety of purposes such as natural language processing. In particular, methods based on transformer and attention [1] or BERT (Bidirectional Encoder Representations from Transformers) [2], are increasing their importance. Deep learning uses multi-layered neural networks such as Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) based on RNN. This achieved a significant advance in many fields, especially in natural language processing and image recognition.

These achieved remarkable improvement, however, it has been stated in several papers that a system based on deep learning accurately performed inference but it was a black-box [3], [4]. Namely, its output was not interpretable and the basis of its decision was not presented. As a result, they stated that decisions of deep learning cannot be trusted. There are many situations wherein a decision must be interpreted and explained. For example, a judge has to explain its judgment, a politician has to explain for its election voters, and a manager has to explain its strategies for its shareholders. We then expect that providing interpretability for decisions done by a deep learning system is important.

In this paper, we focus on a task of two-class classification of news documents with specified themes and classify them with LSTM with self-attention. We then discuss methods for providing interpretability for the decision by a deep neural network with self-attention. We proposed five methods for providing interpretability and evaluate them in several aspects. In the case of image classification, identifying pixels that strongly influenced the classification result is one of the explaining ways [5], [6]. Based on these existing works, we aim to identify the words that strongly influence the classification results as providing interpretability and the basis of decisions.

The rest of this paper is organized as follows. Section 2 refers to related work of providing interpretability on a decision of machine learning such as deep learning. Section 3 explains our experiments for news documents classification by LSTM with self-attention. Section 4 proposes five methods for providing interpretability. Section 5 evaluates the proposed methods in several ways. Section 6 discusses suitable evaluating methods of interpretability provision. Section 7 concludes this work.

2. Related Work

2.1 NLP with Self-attention

Self-attention [7], [8] is a neural network model that consists of a Bidirectional LSTM combined with Attention [1]. The Bidirectional LSTM [3] performs computations in two opposite directions, forwards and backwards, to a single output. The output layer can obtain information from both directions simultaneously. In the case of natural language processing (NLP), the output layer can be affected by both past and future states. Context considering forward and backward is taken into account.

2.2 Providing Interpretability on Decision of Machine Learnings

Ribeiro et al. argue that the interpretability of decisions in machine learning is important [4]. They pointed out that most ma-

[1] Kogakuin University, Shinjuku, Tokyo 160–0023, Japan
[a)] sane@cc.kogakuin.ac.jp

chine learning models were black boxes, and argued that understanding the reasons behind predictions was important in assessing trust. They showed a learning model that decided whether an animal in a photo was a husky or a wolf with a basis whether its background was snow or not, and then argued that this was a "bad model."

Several papers on methods for extracting the basics of decision and providing interpretability have been published. As methods for providing interpretability of decisions by deep neural networks, several methods by showing a saliency map. Simonyan et al. proposed Vanilla Gradients [9]. This method generated an image, which maximized the class score [10], thus visualizing the notion of the class, captured by a deep convolutional network. This then created a class saliency map, which was specific to a given image and class. Smilkov et al. focused on explanation by identifying pixels that strongly influence the decision [5]. They then proposed SmoothGrad to create a sensitivity map based on the gradient of the class score function concerning the input image. SmoothGrad added Gaussian noise to the input values of a CNN. It then calculated the gradient value, which was the change in the output value in relation to the change in the input value, for each input dimension and extracted the pixels with the large gradient value as the basis for a decision. In the work of Refs. [5], [9], they applied their methods only for image recognition, and no discussion on an application on NLP was presented. Samek et al. focused on a heatmap, which quantified the importance of individual pixels with reference to the classification decision and visualized importance in pixel/input space, and tackled a problem of quantifying the quality of a heatmap [11]. They assumed that flipping the most salient pixels first should lead to high performance decay, and then proposed a region perturbation strategy based on this assumption.

Selvaraju et al. proposed Grad-CAM [12] that was a method to show the basis of CNN decisions. This method visualized the pixels that contributed to the classification by using the gradient value between the input value of the CNN and the output value of the final layer of the convolutional layers. Grad-CAM was applied to also NLP and published in some sites [13], [14]. However, a variety of evaluations and discussions on evaluating ways in NLP were not presented. Li et al. plotted neural unit values to visualize compositionality of negation, intensification, and concessive clauses in NLP [15]. They then visualized a unit's salience. They measured how much each input neural unit contributed to the final decision by approximating by first derivatives with the first-order Taylor expansion. DeYoung et al. proposed methods to evaluate rationales in NLP [16]. They aimed to capture the extent to which rationales provided by a model in fact informed its predictions. They calculated the difference between the predicted probabilities from the original model and the model that the rationales were stripped. They assumed that a large difference implied that the rationales were indeed influential in the prediction. Serrano counted how many important words needed to be erased before a prediction flipped [17]. Arras et al. also proposed to measure the impact of perturbing or erasing words identified as important on model output [18]. These works are contributing, especially for evaluating extracted words. Our evaluating methods, which are the methods 1-1 and 1-2 in Section 5, are based on these previous works. However, unlike this paper, these works presented neither a method for evaluating the direction in a classification of a word nor an evaluation based on an automatically generated table that is assumed correct. Also in the above-mentioned work of Ref. [4], the authors proposed LIME for making a decision of classification interpretable. In the work of Ref. [6], a simple method to add interpretability to support vector machine (SVM) decisions by focusing on the absolute value of the SVM weight vector was proposed.

In our previous work [19], [20], [21], we focused on the interpretability of decisions of machine learning such as deep learning, and proposed methods for presenting the basis of decisions. In the work of Ref. [19], we classified review documents as high rating or low rating ones using SVM and DNN and then proposed methods for providing interpretability. The methods for SVM and DNN used the absolute value of each dimension of the weight vector of SVM and applied SmoothGrad to NLP, respectively. We revealed that machine learning sometimes performed classification that was based on a subjectively unsuitable basis even in the case of high classification accuracy and pointed out that the model was nearly a bad model. In the work of Ref. [20], we classified two types of news documents based on self-attention and proposed a method that extended SmoothGrad to natural language processing and a method that used attention values for providing interpretability. In the work of Ref. [21], we proposed methods based on NLG, Attention, and WD, and evaluated the methods in several ways. NLG and WD are explained in Section 4. This paper is based on these previous papers, especially the work of Ref. [21].

Many works on XAI (explainable artificial intelligence) have been published [22], [23], [24]. Many researchers pointed out the black-box nature of AI. They argued that an important issue of the use of AI-based systems was its often lack of transparency and then raised a discussion on XAI. In addition, systems for XAI also were published [25]. This paper is standing also on these existing works. These many works mostly focus on image recognition. These did not present enough discussion on explanation on natural language classification with various deep neural networks such as networks supported by self-attention or transformer.

3. New Documents Classification by LSTM with Self-attention

In this paper, we classify news documents with deep learning, and discuss methods for presenting a basis for decision using this classification. In this section, we explain this news document classification.

3.1 Target Documents and Method of Classification

We selected two types of articles from the nine themes of the livedoor news corpus as classification targets. We then classified whether a randomly selected document belonged to the first type or the second type of articles by LSTM with self-attention.

Figure 1 illustrates the overview of the used neural network and its input and output data. **Figure 2** describes its pseudo code. We used MeCab 0.996 for morphological analysis and NEologd

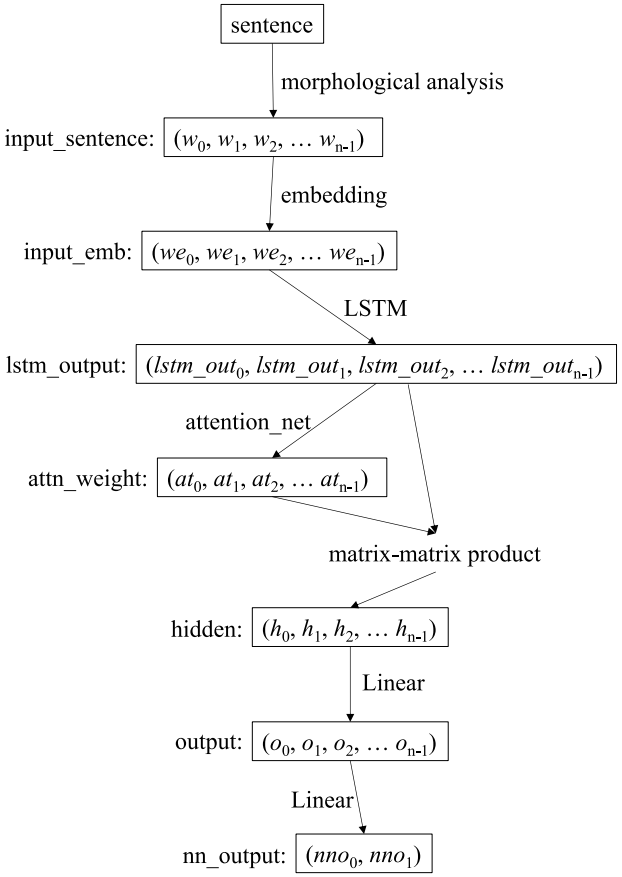


Fig. 1 Input, output, and neural network.

```
input_emb = Embedding(input_sentence)
lstm_output = bilstm(input_emb)
attn_weight = attention_net(lstm_output)
hidden = bmm(attn_weight, lstm_output)
output = Linear( hidden.view() )
nn_output = Linear( output )
```

Fig. 2 Pseudo code.

for the MeCab dictionary. Distributed representations of words are created by fastText with the pre-trained model by the Japanese Wikipedia corpus. All the words, including symbols, are not excluded. The setups of the neural network with self-attention were as follows. The number of the dimensions of the output of Bidirectional LSTM was 512. The number of dimensions of each word in distributed representation was 300. The optimization function was Adam. The learning rate was 0.001. The batch size was 128. The used loss function was CrossEntropy. PyTorch 1.3.1 was used as a deep learning library. 80% and 20% of all the documents were used for training and testing, respectively. The training was executed until the loss, which is the output value of the loss function, saturated.

The livedoor news corpus includes nine topics, which are Kaden Channel, smax, topic news, Sports Watch, IT life hack, MOVIE ENTER, dokujo, HOMME, and Peachy. The numbers of documents, numbers of words (morphemes), and numbers of kinds of words of topics are described in **Table 1**.

Table 1 Statistics of the livedoor news corpus.

	num. of documents	num. of words	num. of kinds of words
Dokujo Tsushin	870	740,828	31,660
IT Life Hack	870	571,118	24,139
Kaden Channel	864	355,769	21,523
Livedoor Homme	511	417,521	26,129
Movie Enter	870	614,786	30,986
Peachy	842	566,473	33,209
smax	870	682,105	21,129
Sports Watch	900	330,118	20,811
Topic News	770	289,514	22,276
all	7,367	4,568,232	102,286

3.2 Accuracy of Classification

First, we selected Kaden Channel and smax as classification targets. Kaden Channel and smax are sets of news documents about consumer electronics and mobile gadgets, respectively. The reason why we first selected these two themes from the nine themes is that the accuracy of classification in these two themes was the highest. The accuracy of the two-class classification was 99.1%. The numbers of documents for testing were 174 and 173 for Kaden Channel and smax, respectively. One document of Kaden Channel was incorrectly classified as a smax document. Two smax documents were classified as Kaden Channel documents incorrectly.

We think deep learning classified documents highly accurately, but the basis of the classification for interpretation was not presented.

There are nine themes in the livedoor news corpus, and there are ${}_9C_2 = 36$ combinations of two themes. The accuracy of the combination of Kaden Channel and smax was the highest among all combinations. In this study, we focus on the fact that deep learning can correctly infer but its decision is not interpretable. Therefore, we first discuss interpretability with the classification with the highest accuracy. We second discuss the other seven themes. We chose the sets of combinations of two themes that have the highest total accuracies from all the other seven sets of the combinations. The sets are (Dokujo Tsushin, Topic News), (Peachy, Sports Watch), (IT Life Hack, Movie Enter), and (Livedoor Homme, Kaden Channel). Because the number of the themes was an odd number, only one theme, which was Kaden Channel, was used twice. Extraction of the basis of classifications with low accuracies is expected to contribute to identifying the causes of incorrect classification and discussing ways to improve accuracy. This extraction also is expected to be useful.

4. Providing Interpretability

In this section, we propose five methods for providing the basis for classification decisions: the norm NLG method, the Attention method, the WD method, the NLG*WD method, and the Att*WD method.

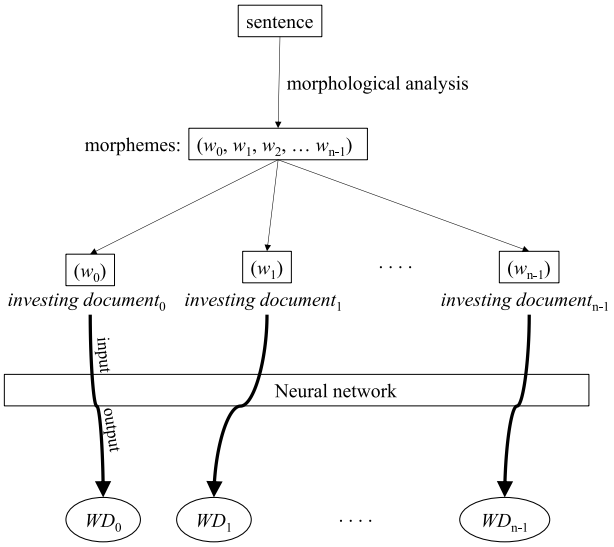


Fig. 3 WD and investing document.

4.1 Norm NLG Method

The norm NLG method, which stands for the natural language gradient method, is a naive application of SmoothGrad [5] to natural language processing by LSTM with self-attention. This method adds Gaussian noise to input words in a distributed representation and calculates gradient, which is the change in the output value relative to the size of the added noise. The gradient is obtained for each dimension of each input word. Namely, a vector of 300 dimensions is obtained for each input word. In this paper, we call the norm of this vector of gradients for each word norm NLG. This method considers that words with a large norm NLG are important words for classification, i.e., the basis for classification. This method's score is the norm of this vector.

This method takes magnitude into account but does not take direction into account. A word with a large norm NLG is a word that has a large impact on classification. However, this method does not take into account whether the word has a large influence on the true decision or false decision.

4.2 Attention Method

The Attention method checks the attention value given to each input word in documents classification using LSTM with self-attention and considers words with large attention values important basis for classification. This method's score is the attention value.

4.3 WD Method

The WD method, which stands for word direction method, creates an investigating document for each word in the input document. Each investigating document contains only a single word in the input document. This method inputs every investigating document to the model for classification and uses the output value of the neural network as the value for the basis for the classification of the word. This method's score is the output value of the neural network.

Figure 3 illustrates the WD method. First, a sentence (document) is split into multiple morphemes by morphological analysis. Second, every morpheme creates an investing document. For example, if a document contains morphemes w_0, w_1, and w_2, three investing documents are created. The first investing document is a document that contains only one word (morpheme) w_0. The second investing document is composed of only w_1. The third one is composed of only w_2. Third, every investing document is input into the neural network. An "investing document" in Fig. 3 corresponds to "sentence" in Fig. 1. Forth, the output value from the neural network is obtained. The "nn_output: ($nno_0, nno_1, nno_2, \ldots nno_{n-1}$)" in Fig. 1 corresponds to "WD_x" in Fig. 3. In the case of an investing document, the number of dimensions of the output vector is one, i.e., the output is like "nn_output: (nno_0)." This value (nno_0) is the score of the WD method.

This method takes both magnitude and direction into account. However, this method does not take the context of each word into account.

4.4 NLG*WD Method

The NLG*WD method uses the product of the norm NLG value and the sign of the WD value of each word as the basis for the classification. The norm NLG value is a positive real number, and the sign of the WD value is +1 or −1. The norm NLG and the sing of WD represent the magnitude and the direction as a basis for classification, respectively. The norm NLG value can be determined by the way in Section 4.1. The WD value is determined by the way in Section 4.3. Its sign is easily determined by checking if the WD value is larger than 0. This method's score is the product of the norm NLG value and the sign of the WD.

4.5 Att*WD Method

The Att*WD method uses the product of the Attention value, which is a positive real number, and the sign of the WD value, which is +1 or −1, as the basis of classification for each word, where the Attention value and the sign of the WD value represent the magnitude and direction of the basis, respectively. The Attention value can be determined in the way in Section 4.2. The WD value is determined by the way in Section 4.3. Its sign is easily determined by checking if the WD value is larger than 0. This method's score is the product of the Attention value and the sign of the WD.

5. Evaluation

5.1 Experimental Setup

We evaluated the five proposed methods using four different evaluation methods (methods 1-1, 1-2, 2-1, and 2-2). Appropriate evaluating methods are discussed in Section 6. Five documents were randomly selected from each of themes for classification and, in all evaluations, these documents were classified by LSTM with self-attention. The sets of themes for classification are (smax, Kaden Channel), (Dokujo Tsushin, Topic News), (Peachy, Sports Watch), (IT Life Hack, Movie Enter), and (Livedoor Homme, Kaden Channel), as described in Section 3.2. The five proposed methods provided interpretability to every classification. As we described in Section 1, providing interpretability means identifying the words that strongly influence the classification results. In all of the above classifications, deep learning

performed the classification correctly.

In the evaluation method 1-1, the words that a method specified as the basis are deleted from the document in order of their scores specified by the method. Then, the documents whose words were deleted were classified using the model. Words were deleted one by one until the model incorrectly classified. We evaluated the performance of each method based on the number of deleted words at incorrect classification.

If the model is classified incorrectly after the deletion of only a small number of words, we can assume that those words had a strong influence on the classification results and the method could extract a better basis. However, repeated word deletion results in a document that differs significantly from the original document. Evaluation with such a document may not be suitable.

In the evaluation method 1-2, to avoid the evaluation with documents that differ significantly from the original documents in the evaluation method 1-1, we deleted at most 10 words that have the highest scores of each method from the original document. We evaluated every method based on the size of the decrease in the output value of the neural network to the deletion of the words.

In the evaluation methods 2-1 and 2-2, a ranking table of words is created based on the scores given by each method. Naturally, the method that gives a higher score to words with an important basis is the suitable method. We compare this ranking table of each method with the ranking table assumed correct. We evaluated the proposed methods with the assumption that the closer the two tables are, the better the method is.

The ranking table assumed correct is created as follows. First, we delete a word from a document. Second, we input the document, a word of which was deleted, into the neural network and obtain the output value, which indicates the classification result. Third, we compare the output values before and after the deletion of a word and calculate the difference. In this paper, we refer to this difference as "the evaluation change by a deletion of a word." If the output value changes for incorrect classification, the evaluation change is positive. If it changes for correct classification more, the change is negative. Fourth, we calculate this evaluation change for every word in a document and create a ranking table of words in order by the evaluation change. This is the table assumed correct.

In the evaluation method 2-1, the correlation coefficient between the ranking table presented by each method and the ranking table assumed correct is calculated. Every word has two ranking orders in two tables. The correlation coefficient between these two ranking orders is calculated. We assume that the higher the correlation coefficient is, the better the method is.

In the evaluation method 2-2, we calculate the nDCG [6] of the top 30 words with the highest evaluation change, which are the top 30 words in the ranking table assumed correct, in the ranking table of each method. We assume that the higher the nDCG is, the better the method is.

5.2 Experimental Result

Figures 4, **5**, **6**, **7**, **8**, and **9** show the results of the evaluation method 1-1. The vertical axis shows the average percentage of the number of deleted words before incorrect classification in

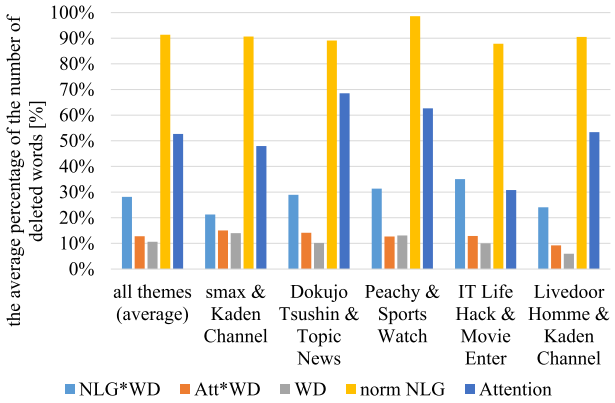


Fig. 4 Number of deleted words before incorrect classification (all themes).

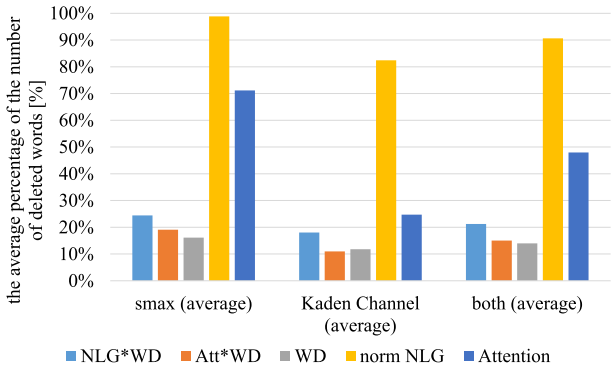


Fig. 5 Number of deleted words before incorrect classification (smax, Kaden Channel).

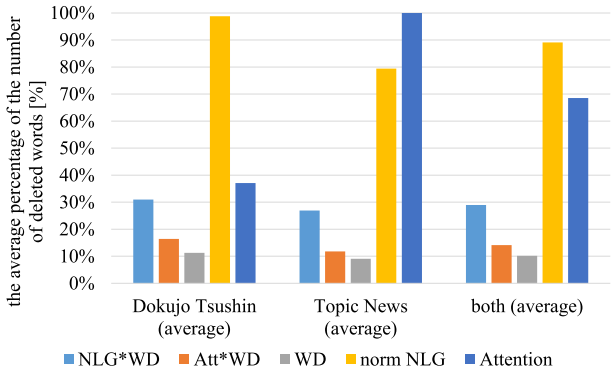


Fig. 6 Number of deleted words before incorrect classification (Dokujo Tsushin, Topic News).

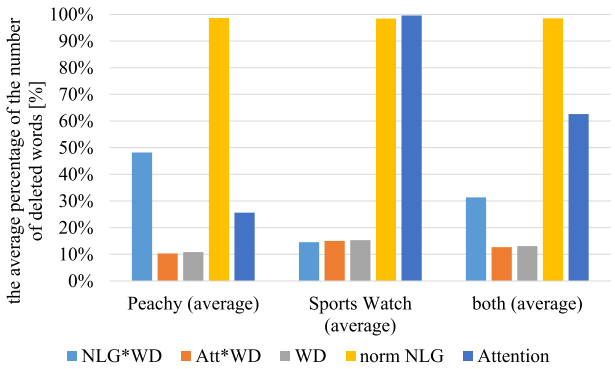


Fig. 7 Number of deleted words before incorrect classification (Peachy, Sports Watch).

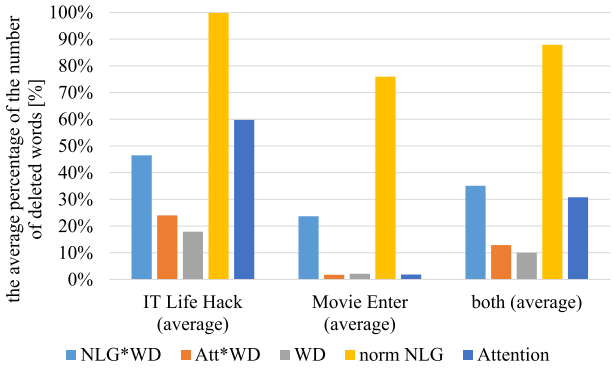


Fig. 8 Number of deleted words before incorrect classification (IT Life Hack, Movie Enter).

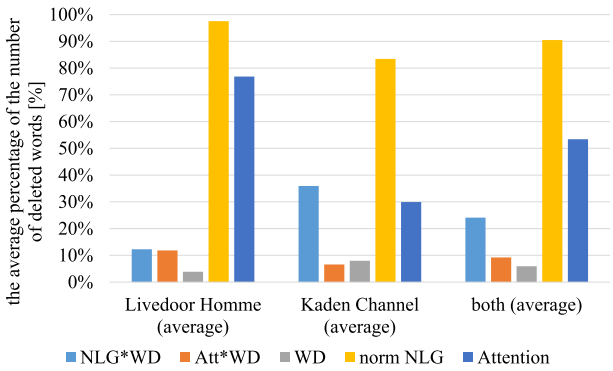


Fig. 9 Number of deleted words before incorrect classification (Livedoor Homme, Kaden Channel).

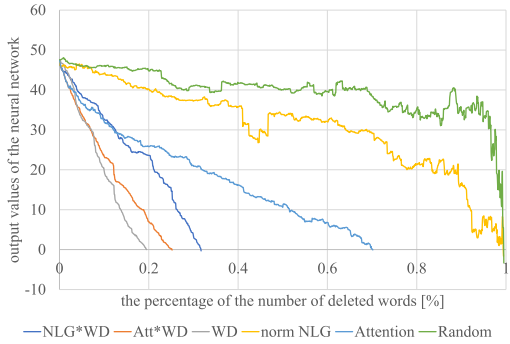


Fig. 10 Transitions of the neural network output (smax 6736017, smax and Kanden Channel).

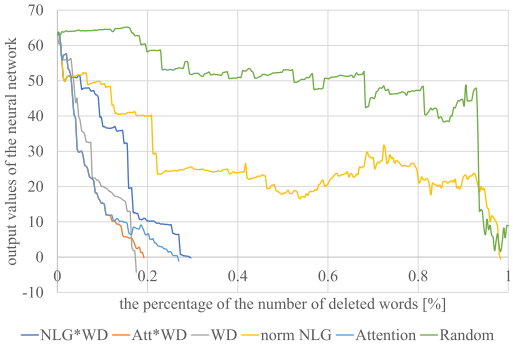


Fig. 11 Transitions of the neural network output (Kaden Channel 6237932, smax and Kaden Channel).

the five documents in each theme and the ten documents of both themes. A percentage indicates the number of deleted words divided by the total number of words in the document. Figure 4 depicts the averages of all the themes and each theme. The averages of each theme are the same with the "both (average)" values in Fig. 5 to Fig. 9. Figure 4 indicates that the WD method achieved the best performance, i.e., the least deletion rate, and the Att*WD and NLG*WD methods followed in the average in all themes. Compared with these three methods, the performance of the norm NLG and Attention methods was significantly inferior. Almost the same trends were observed in each classification in Fig. 4. Namely, the WD method achieved the best or almost best performances in all the cases, which are the best performances in four cases and performance very close to the best one in one case. Attn*WD also achieved the second-best in four cases, and NLG*WD achieved the third-best in four of the five cases.

Figures 5, 6, 7, 8, and 9 show the results of each theme of each classification. Similar results were obtained also for each theme in each classification. For example, in the case of the smax and Kaden Channel themes in Fig. 1, the WD method had the lowest deletion ratio, and the NLG*WD and Att*WD methods had similar ratios in the smax theme. In the Kaden Channel theme, these three methods performed remarkably better than the norm NLG and Attention methods. The norm NLG method had the highest percentage in 14 cases of the 16 cases in all the experiments – three cases in Figs. 5, 6, 7, 8, and 9 and one case in Fig. 4 – and did not incorrectly classify unless around 80% or more of the words were deleted. From these results, we can conclude that the NLG*WD, Att*WD, and WD methods are the most suitable methods for providing interpretability in the aspect of deleting words until they incorrectly classify.

Figure 10 and **11** show the transitions of the output values of the neural network while deleting words until the incorrect classification of a document of each theme of the first classification, which is the classification between smax and Kaden Channel as described in Section 3.1. The vertical axis of the figure is the output value of the neural network, and the horizontal axis is the percentage of deleted words. For comparison, we also show the transition of the output value while deleting words in random order. From the figures, we can see that the NLG*WD, Att*WD, and WD methods, which showed good performances in Fig. 5, showed rapid decreases, and the difference between them was not quite large. The results show also that the output value rarely decreases with deleting random words. This implies that the proposed methods, including worse ones, suitably chose words for interpretability. Focusing on the results with a small number of deleted words, especially in Fig. 10, the NLG*WD achieved pretty good performance. The transitions of the output values of the second classifications are shown in **Figs. 12**, **13**, **14**, **15**, **16**, **17**, **18**, and **19**. We can find similar trends also in these figures. Namely, the WD method worked best and the Att*WD and NGL*Wd methods followed in many cases.

Figures 20, **21**, **22**, **23**, **24**, and **25** show the results of the evaluation method 1-2. In Figs. 21, 22, 23, 24, and 25, the horizontal and vertical axes show the number of words deleted and the average of the evaluation results of 10 documents, respectively. Figure 20 shows the averages of these results. Figure 20 indicates that Att*WD and NLG*WD worked best, and WD did slightly less in the cases the number of deleted words was very small.

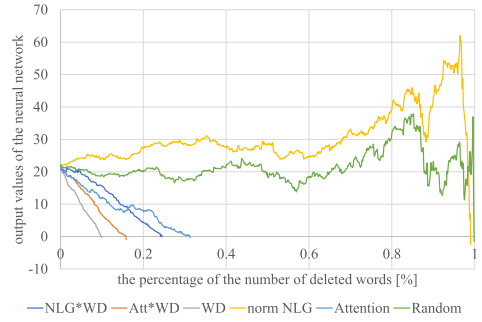


Fig. 12 Transitions of the neural network output (Dokujo Tsushin 5803851, Dokujo Tsushin and Topic News).

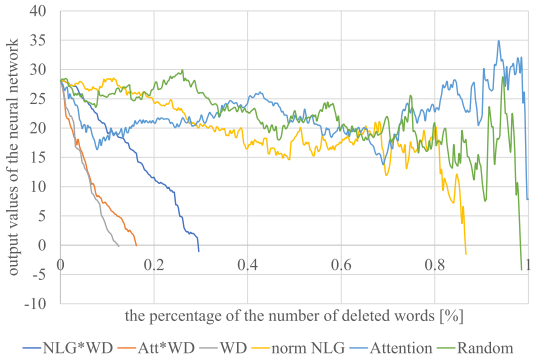


Fig. 13 Transitions of the neural network output (Topic News 6886968, Dokujo Tsushin and Topic News).

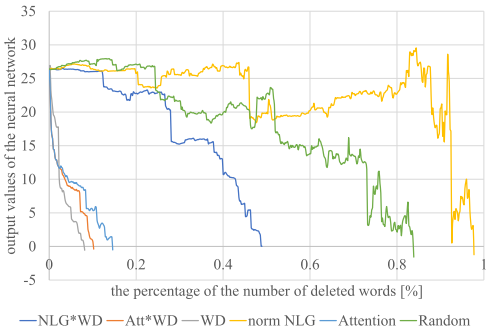


Fig. 14 Transitions of the neural network output (Peachy 6907491, Peachy and Sports Watch).

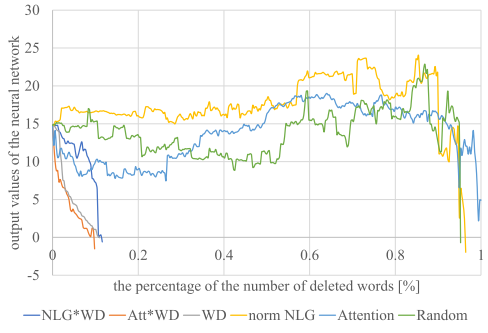


Fig. 15 Transitions of the neural network output (Sports Watch 6904496, Peachy and Sports Watch).

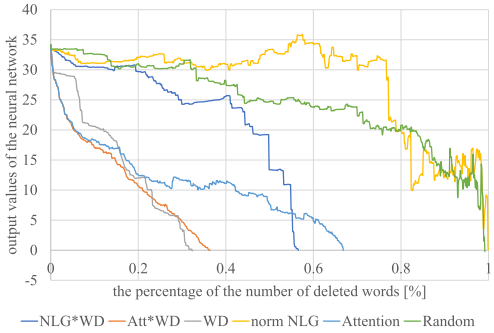


Fig. 16 Transitions of the neural network output (IT Life Hack 6631652, IT Life Hack and Movie Enter).

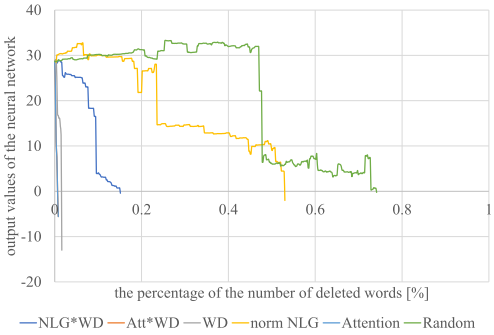


Fig. 17 Transitions of the neural network output (Movie Enter 6499721, IT Life Hack and Movie Enter).

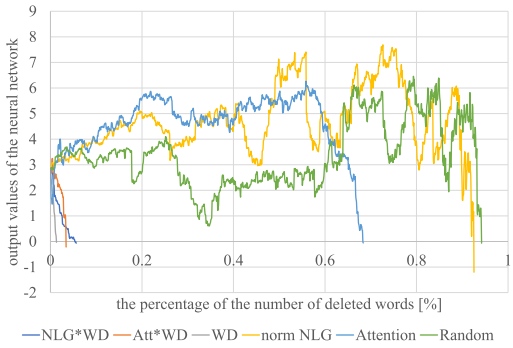


Fig. 18 Transitions of the neural network output (Livedoor Homme 6690686, Livedoor Homme and Kaden Channel).

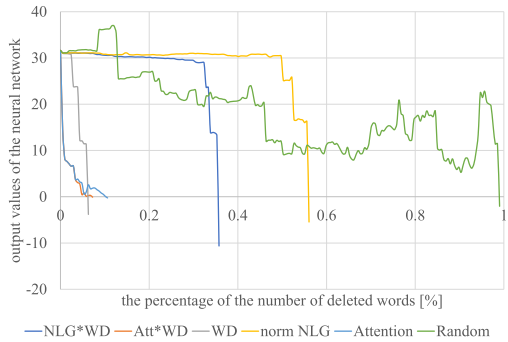


Fig. 19 Transitions of the neural network output (Kaden Channel 6908995, Livedoor Homme and Kaden Channel).

NLG and Attention did not work effectively also in this evaluation. In other words, WD and methods using WD obtained better performance similar to the evaluation method 1-1.

The WD method that performed best in the evaluation method 1-1, in which a large number of words were deleted, did not perform best in the evaluation method 1-2, in which a small number of words were deleted, and was the third-best among the five methods in Fig. 20. Focusing on the results in each classification, we can see similar trends in many cases. Att*WD performed best or nearly best in all the cases. NGL*WD and WD followed Att*WD in many cases. NLG and Attention did worse in many cases. **Figures 26**, **27**, **28**, **29**, **30**, **31**, **32**, **33**, **34** to **35** show the

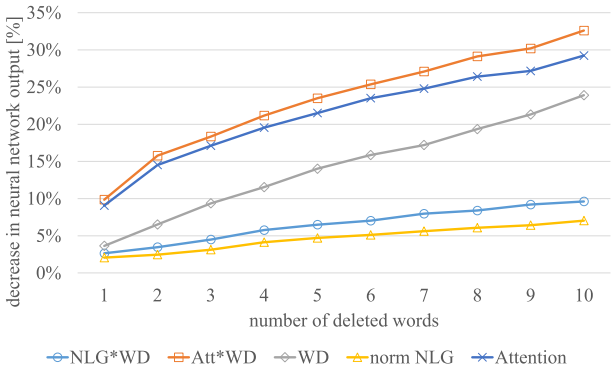


Fig. 20 Decrease in the output value with deletion of a small number of words (average of all the classifications).

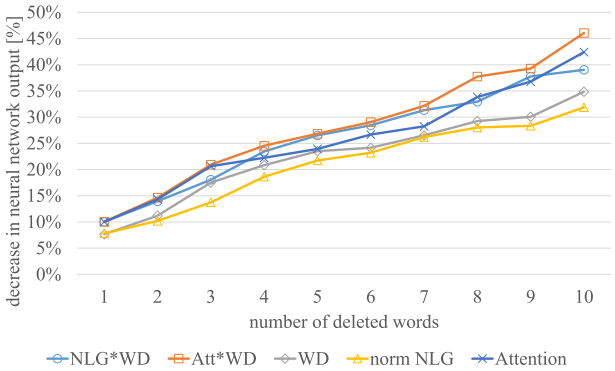


Fig. 21 Decrease in the output value with deletion of a small number of words (Kaden Channel, smax).

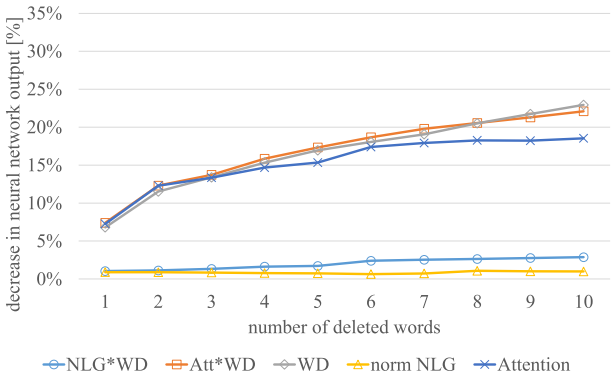


Fig. 22 Decrease in the output value with deletion of a small number of words (Dokujo Tsushin, Topic News).

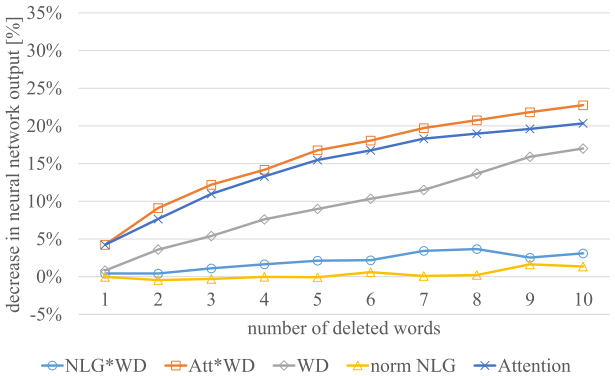


Fig. 23 Decrease in the output value with deletion of a small number of words (Peachy, Sports Watch).

result of one document in each theme of the classifications. These show a similar trend. The NLG*WD and Att*WD methods, especially the Att*WD method, performed well in most cases. The

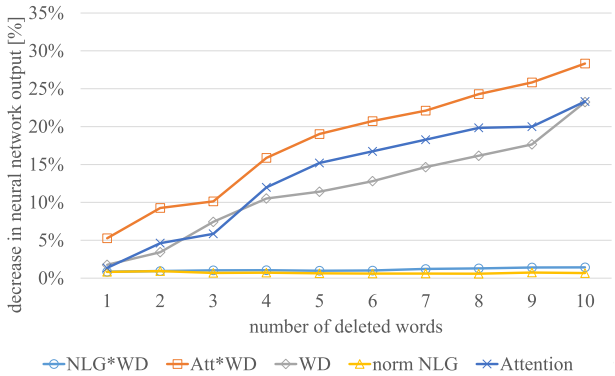


Fig. 24 Decrease in the output value with deletion of a small number of words (IT Life Hack, Movie Enter).

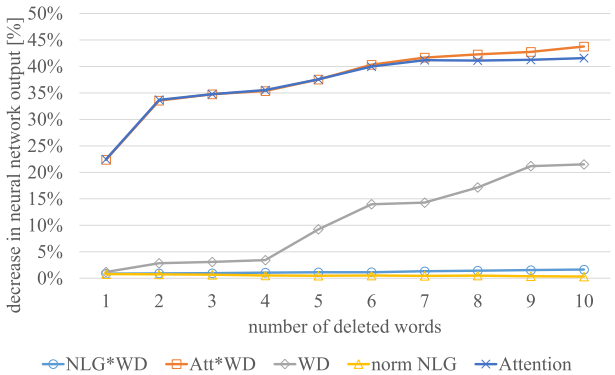


Fig. 25 Decrease in the output value with deletion of a small number of words (Livedoor Homme, Kaden Channel).

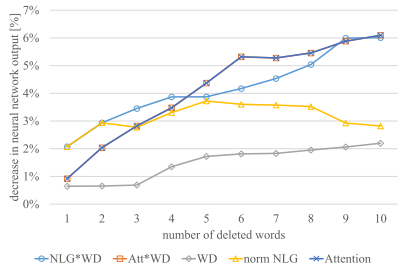


Fig. 26 Decrease in the output value with deletion of a small number of words (smax 6736017).

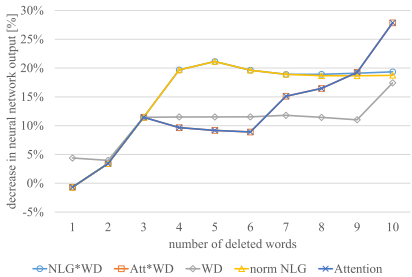


Fig. 27 Decrease in the output value with deletion of a small number of words (Kaden Channel 6237932).

NLG and Attention methods did not perform suitably. All the methods including NLG and Attention methods worked effectively in the case of the classification between Kaden Channel and smax, which was the classification with the highest accuracy.

As described in Section 5.2, it may not be appropriate to use a document in which many words are deleted from the original one. Based on this assumption, we expect that the WD method, which showed the highest performance in the performance evaluation 1-

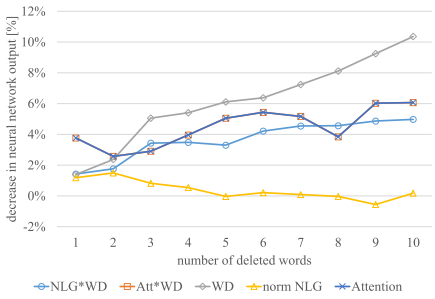


Fig. 28 Decrease in the output value with deletion of a small number of words (Dokujo Tsushin 5803851).

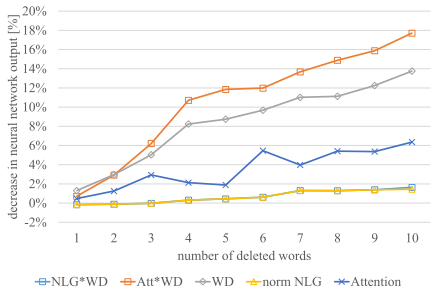


Fig. 29 Decrease in the output value with deletion of a small number of words (Topic News 6886968).

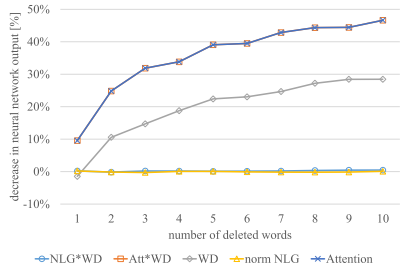


Fig. 30 Decrease in the output value with deletion of a small number of words (Peachy 6907491).

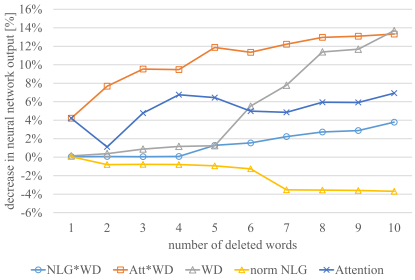


Fig. 31 Decrease in the output value with deletion of a small number of words (Sports Watch 6904496).

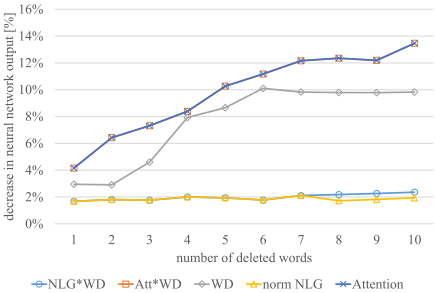


Fig. 32 Decrease in the output value with deletion of a small number of words (IT Life Hack 6631652).

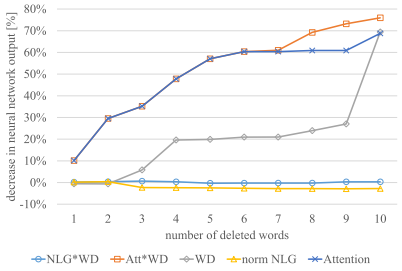


Fig. 33 Decrease in the output value with deletion of a small number of words (Movie Enter 6499721).

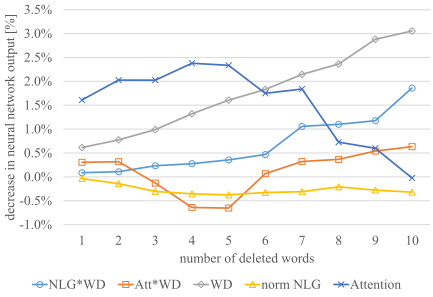


Fig. 34 Decrease in the output value with deletion of a small number of words (Livedoor Homme 6690686).

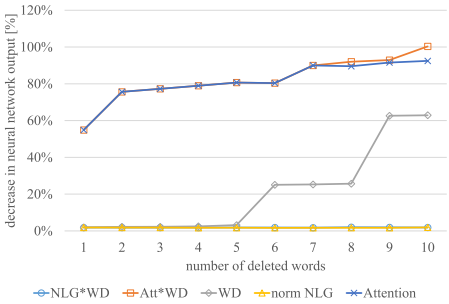


Fig. 35 Decrease in the output value with deletion of a small number of words (Kaden Channel 6908995).

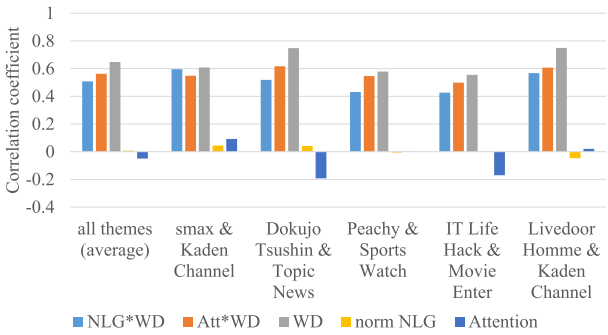


Fig. 36 Correlation coefficient between table assumed correct and that created by each method (average and all).

1 but non-best performance in the evaluation 1-2, is not an appropriate method. We can conclude that the NLG*WD and Att*WD methods, which provided good performances in both evaluations 1-1 and 1-2, are suitable methods for providing interpretability.

The experimental results of the evaluation method 2-1 are shown in **Figures 36**, **37**, **38**, **39**, **40**, and **41**. The "all theme (average)" in Fig. 36 is the average of the values in Figs. 37, 38, 39, 40, and 41. The other values in Fig. 36 are the same as the values of "both (average)" in Figs. 37, 38, 39, 40, and 41. In the document classification of this paper, the neural network out-

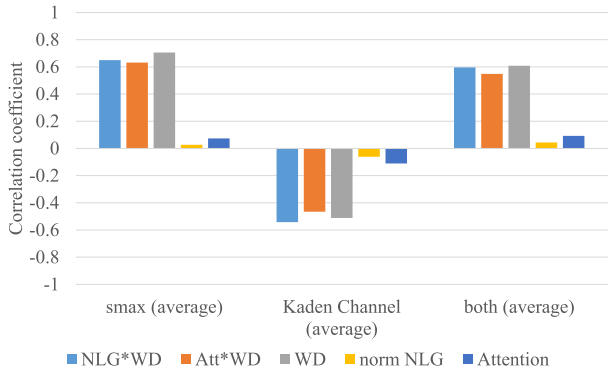


Fig. 37 Correlation coefficient between table assumed correct and that created by each method (Kaden Channel, smax).

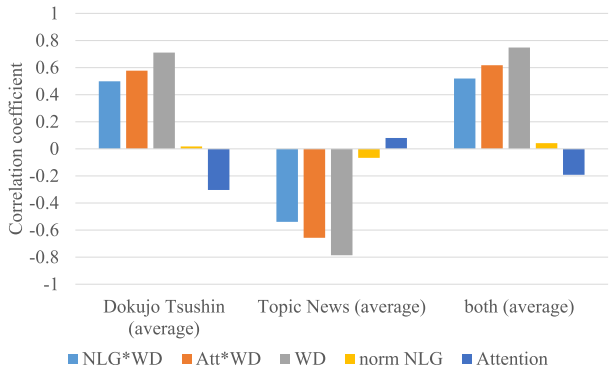


Fig. 38 Correlation coefficient between table assumed correct and that created by each method (Dokujo Tsushin, Topic News).

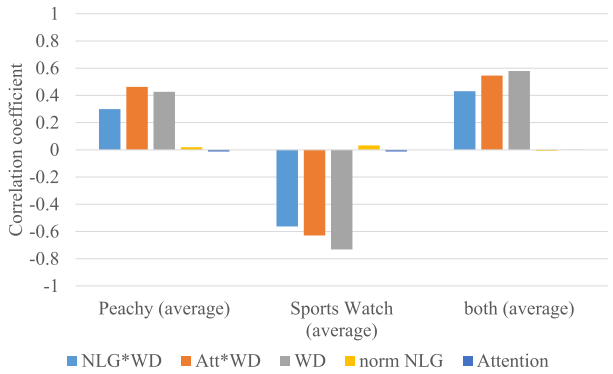


Fig. 39 Correlation coefficient between table assumed correct and that created by each method (Peachy, Sports Watch).

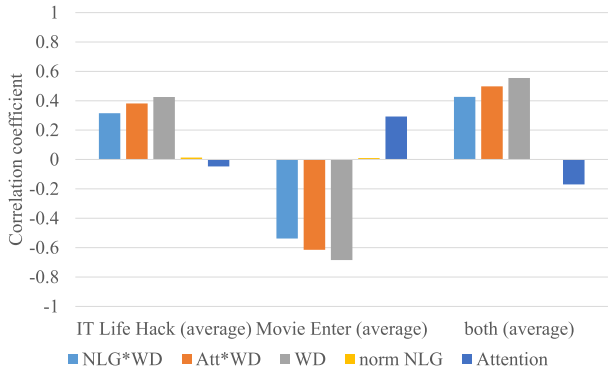


Fig. 40 Correlation coefficient between table assumed correct and that created by each method (IT Life Hack, Movie Enter).

puts a positive value if the network classifies the document as that of the first theme. It outputs a negative value if it does as one of the second theme. Therefore, the larger the correlation

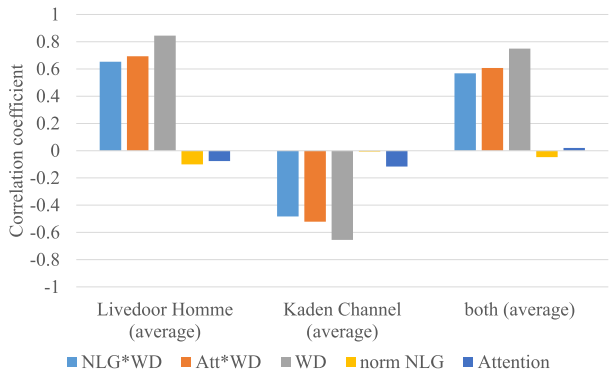


Fig. 41 Correlation coefficient between table assumed correct and that created by each method (Livedoor Homme, Kaden Channel).

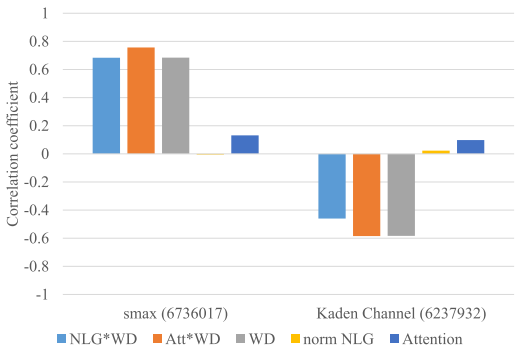


Fig. 42 Correlation coefficient between table assumed correct and that created by each method (smax 6736017, Kaden Channel 6237932).

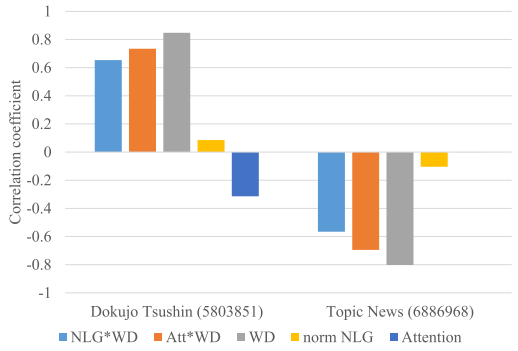


Fig. 43 Correlation coefficient between table assumed correct and that created by each method (Dokujo Tsushin 5803851, Topic News 6886968).

coefficient for the first theme, the better the presentation of interpretability. Similarly, the smaller the correlation coefficient for the second theme is, the better it is. In the figures, the average of the correlation coefficient of the first theme and that of the second theme multiplied by -1 is shown as "both (average)". In Fig. 37, we can see that the correlation coefficients of the three methods NLG*WD, Att*WD, and WD are high (good) in the average and all five cases. The performances of these three methods are almost equal. WD method slightly outperformed Att*WD, and Att*WD slightly outperformed NLG*WD. The Attention method achieved a good performance in the evaluation 2-2, its performance in this evaluation was remarkably low. The norm NGL did not work well similar to the evaluations 1-1 and 1-2. **Figures 42**, **43**, **44**, **45**, and **46** show the results of one document in each theme. The results showed a similar trend, i.e., the NLG*WD, Att*WD, and WD methods performed better.

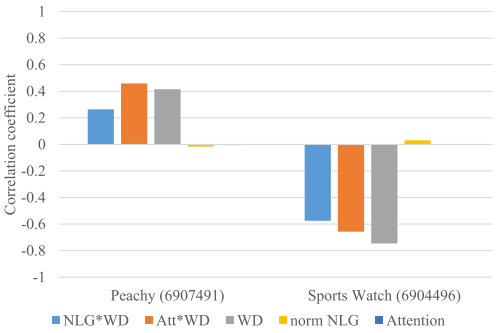


Fig. 44 Correlation coefficient between table assumed correct and that created by each method (Peachy 6907491, Sports Watch 6904496).

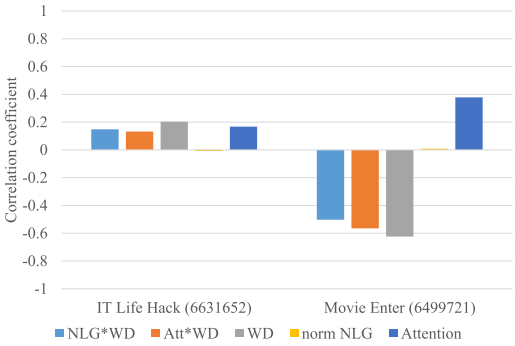


Fig. 45 Correlation coefficient between table assumed correct and that created by each method (IT Life Hack 6631652, Movie Enter 6499721).

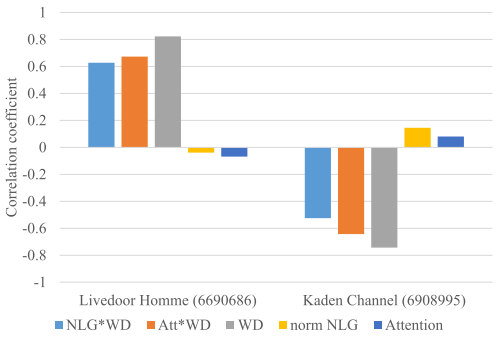


Fig. 46 Correlation coefficient between table assumed correct and that created by each method (Livedoor Homme 6690686, Kaden Channel 6908995).

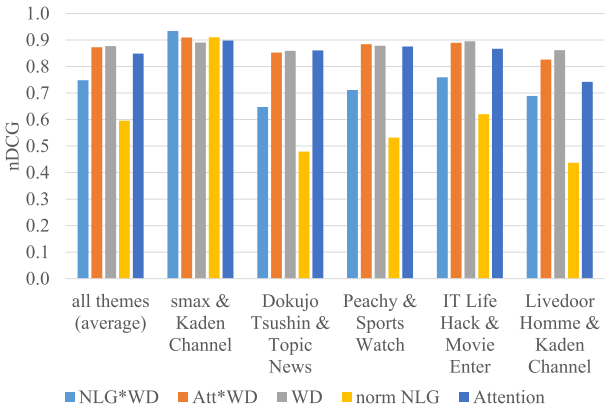


Fig. 47 nDGC in ranking table by each method (average and all).

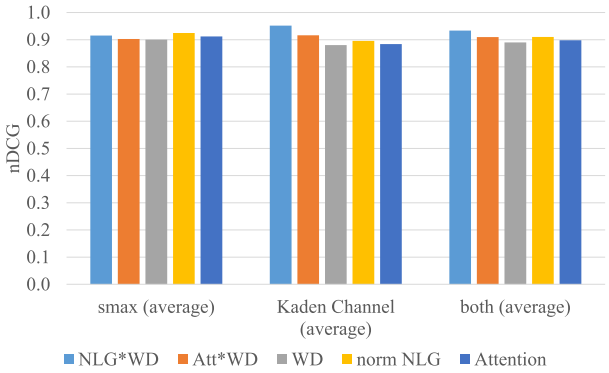


Fig. 48 nDGC in ranking table by each method (Kaden Channel, smax).

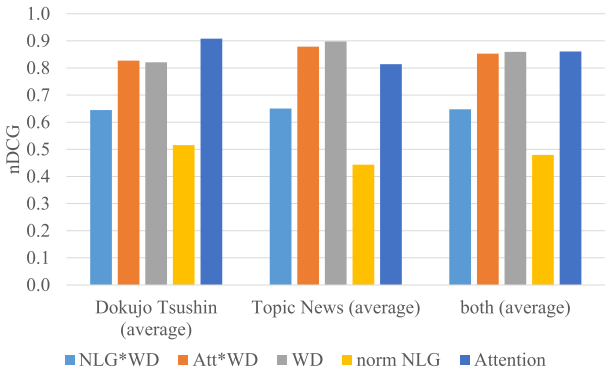


Fig. 49 nDGC in ranking table by each method (Dokujo Tsushin, Topic News).

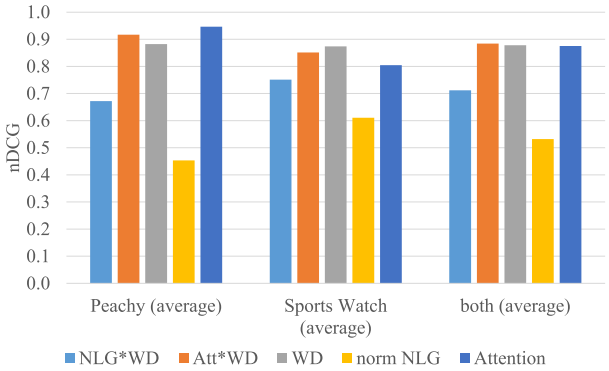


Fig. 50 nDGC in ranking table by each method (Peachy, Sports Watch).

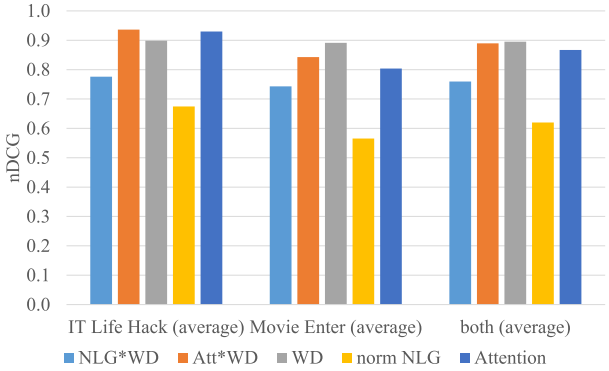


Fig. 51 nDGC in ranking table by each method (IT Life Hack, Movie Enter).

The experimental results of the evaluation method 2-2 are shown in **Figures 47**, **48**, **49**, **50**, **51**, and **52**. The "all themes (average)" in Fig. 47 is the average of all "both (average)" in Figs. 48, 49, 50, 51, and 52. The other values in Fig. 47 are the same as the values in Figs. 48, 49, 50, 51, and 52. The vertical axis shows the average of the nDCG values. This value is large if the top

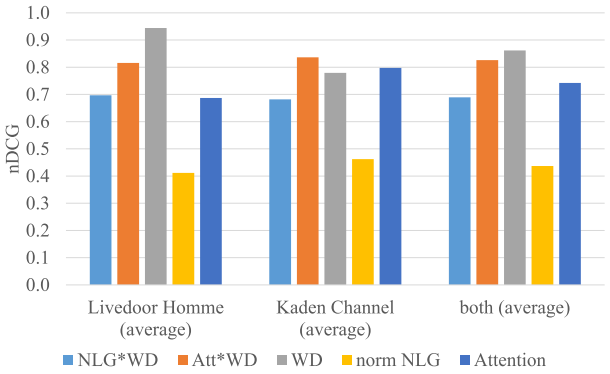


Fig. 52 nDGC in ranking table by each method (Livedoor Homme, Kaden Channel).

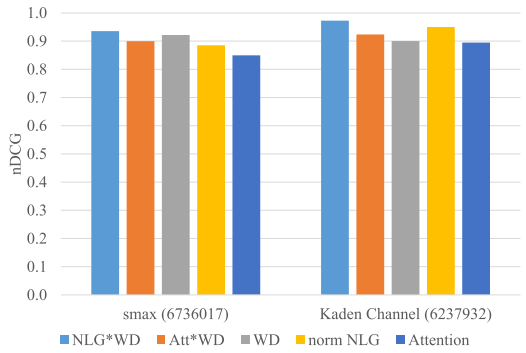


Fig. 53 nDGC in ranking table by each method (smax 6736017, Kaden Channel 6237932).

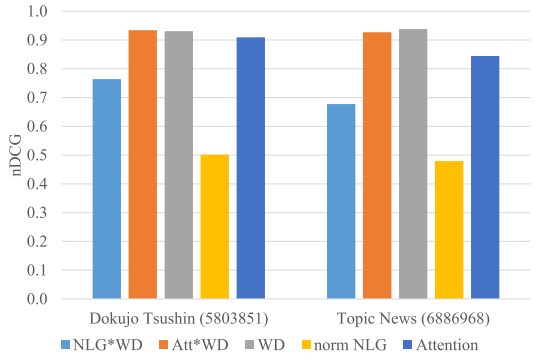


Fig. 54 nDGC in ranking table by each method (Dokujo Tsushin 5803851, Topic News 6886968).

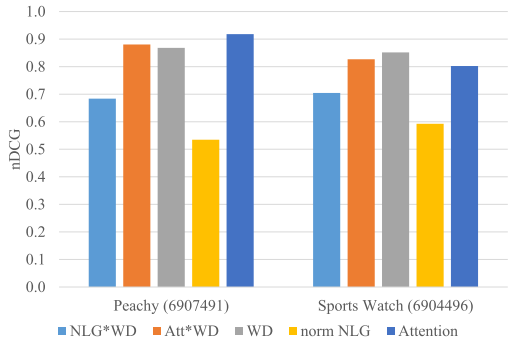


Fig. 55 nDGC in ranking table by each method (Peachy 6907491, Sports Watch 6904496).

30 words in the ranking table assumed correct are ranked highly in the ranking table of each method. In particular, the value is large if the top or near-top words in the ranking table assumed correct are ranked highly in the tables. Figure 47 shows that the

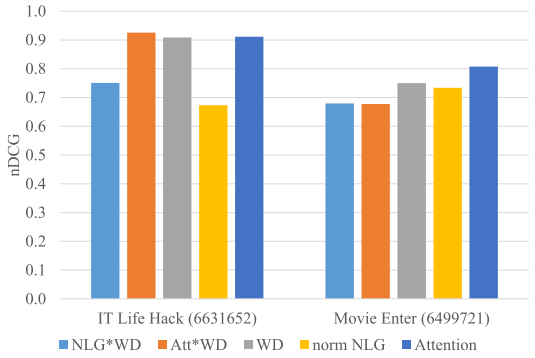


Fig. 56 nDGC in ranking table by each method (IT Life Hack 6631652, Movie Enter 6499721).

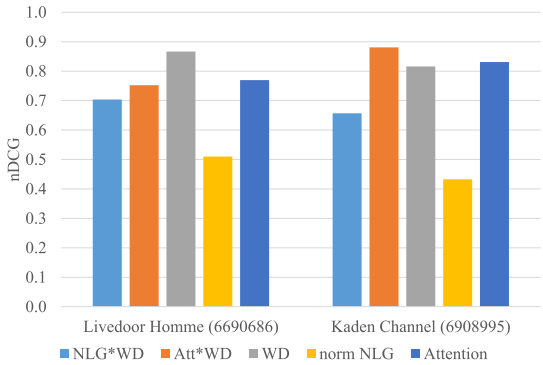


Fig. 57 nDGC in ranking table by each method (Livedoor Homme 6690686, Kaden Channel 6908995).

WD and Att*WD methods have the best and second-best performances in the average of all articles, followed by the Attention method. This trend is observed also in many of the classifications in Fig. 47. In the classification between smax and Kaden Channel, all the methods achieved good performances, which is around 0.9. In the other cases, the performances of norm NLG were remarkably less than the others. Focusing on each theme in Figures 48, 49, 50, 51, and 52, we can see that the Att*WD, WD, and Attention methods were the best or nearly best in most cases. Namely, similar trends can be observed in many of themes. **Figures 53**, **54**, **55**, **56**, and **57** show the results of one document of each theme, and their results also imply that the Att*WD and WD performed most suitably and the Attention did slightly less. From these results, we can conclude that the Att*WD and WD methods provided the interpretability, i.e., the basis of classification, most suitably in an aspect of the evaluation 2-2.

The proposed methods were evaluated in the four evaluation methods. The WD, Att*WD, and NLG*WD methods had the top-3 performances for all the evaluation methods. Comparing these three methods, the Att*WD method achieved the top or nearly top performances in all the evaluations. The NLG*WD and WD methods performed a little less in some evaluations. From these, we can conclude that the NLG*WD, Attn*WD, and WD methods can extract interpretability suitably and the Att*WD method can provide interpretability most suitably.

6. Discussion

First, we discuss suitable ways to evaluate methods for providing interpretability or the basis of a decision. This is a very difficult discussion. Jay stated that how best to measure rationale

faithfulness was an open question [16]. In this paper, we defined four types of indices and evaluated the proposed methods with them. Each evaluating index has a clear definition for calculating its quantitative values. Therefore, a method that extracts the word whose value is the highest as a basis, with calculating the index value of every word, can have the highest performance. However, it is not certain that the result assumed correct (e.g., the ranking list assumed correct answer) is the correct answer as a basis of a decision. Thus, we evaluated the methods from several aspects and then concluded. We think our evaluations in several ways contribute somehow but are not completely correct.

We think the evaluation methods 1-1 and 1-2 are suitable in an aspect. That is, previous works [16], [17], [18] were based on word deletion and their merits have been recognized. On the other hand, these have limitations due to the multi-layer nonlinear structure of neural networks. For example, deletion of all the words in a set of words has an effect on the output value and deletion of a part of the set does not. In such a case, evaluating the impact of each word individually may be inaccurate. We think that we may improve our evaluating method by measuring the impact of each set of words. However, the number of combinations of words can be significantly large and this cannot be easily achieved. Further discussion on ways to evaluate is future work.

Second, we discuss the characteristics and performance of each method. We can easily predict that Attention would have a strong correlation with the magnitude of the effect on the classification results. However, since it does not have a sign, positive or negative, we expected that Attention should be supported by a method for providing a sign, and then the Att*WD method outperformed the Attention method. Although the WD method has both the magnitude and the sign, we expected that its magnitude may not be highly accurate because the document including one word did not take context into account. Practically, its accuracy is not necessarily high as shown in the evaluation method 1-2. However, we expected more accurate inference on an easier two-class inference of whether positive or negative. We then expected that NLG*WD and other methods supported by the WD would have performed better. Focusing on NLG and Attention supported by WD, we can see that both of them performed well from the comparison with the random method in evaluation method 1-2. Comparing these two methods, we can see that NLG performed slightly better. We think that this is mainly because the gradient, based on SmoothGrad, is an index that directly investigates the influence on the final effect, while Attention is one of the causes.

Third, we discuss the effect of word deletion for LSTM-based models. We deleted some words from sentences and input these incomplete sentences into an LSTM-based neural network. We can expect that the LSTM-based model can analyze these sentences nearly as expected, at least in an aspect of classification, even though these are incomplete. Figure 10 to Fig. 19 show that the output values change continuously as the number of deleted words increases. This implies that a value near the original value can be obtained even if a few words are deleted. However, we think that another way to evaluate methods without creating incomplete sentences also should be discussed, and using mask tokens with BERT is one of the most promising evaluating ways.

Finally, we discuss the reason why the order of methods differs depending on the evaluation methods. The evaluation methods 2-1 and 2-2 presented the same order. The evaluation method 1-1 also presented very similar results to the evaluation methods 2-1 and 2-2. The evaluation method 1-2 provided a slightly different order. the NLG*WD and Att*WD methods outperformed the WD method in the evaluation method 1-2 while the WD method outperformed in the other evaluation methods. In the case of Fig. 25, the performance of the WD method was low. This implies that the performance of the WD method depends on a target sentence. We expect that the WD method does not work well in some cases such as a case its context, which may be monitored by Attention, has a strong impact on its classification because the WD method does not take a context into account. However, the difference between these three methods was not large and we can expect these methods can achieve sufficient performance.

7. Conclusion

In this paper, we focused on the provision of interpretability, which is extraction words that have a strong influence on the decision, in natural language processing by deep learning and proposed five methods for providing. We then evaluated these methods with a news documents classification task by LSTM with self-attention using four different evaluation methods. The evaluation results showed that the method using the NLG and WD could provide interpretability most suitably in most cases. The NLG method was based on SmoothGrad, which is a method for giving interpretability on image recognition and extended to NLP. The WD method evaluated each word by creating a document that contains only one word and inputting the document to the neural network.

For future work, we plan to evaluate the method using more documents, discuss the appropriate evaluation method of provision of interpretation, and discuss evaluation based on using mask tokens with BERT instead of deleting words.

Acknowledgments This work was supported by JSPS KAKENHI Grant Numbers 18K11277, 21K11854, 21K11874. This work was supported by JST CREST Grant Number JPMJCR1503, Japan.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I.: Attention is all you need, *Advances in Neural Information Processing Systems*, pp.6000–6010 (2017).

[2] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

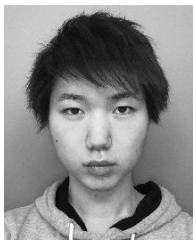
[3] Montavon, G., Samek, W. and Müller, K.-R.: Methods for Interpreting and Understanding Deep Neural Networks, *Digital Signal Processing*, Vol.73, pp.1–15 (Feb. 2018).

[4] Ribeiro, M.T., Singh, S. and Guestrin, C.: Why Should I Trust You?: Explaining the Predictions of Any Classifier, *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD '16*), pp.1135–1144, ACM, DOI: 10.1145/2939672.2939778 (2016).

[5] Smilkov, D., Thorat, N., Kim, B., Viégas, F. and Wattenberg, M.: SmoothGrad: Removing noise by adding noise, *Workshop on Visualization for Deep Learning in ICML* (2017).

[6] Järvelin, K. and Kekäläinen, J.: IR evaluation methods for retrieving highly relevant documents, *Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Re-*

trieval (*SIGIR '00*), pp.41–48, Association for Computing Machinery, DOI: 10.1145/345508.34554 (2000).
[7] Lin, Z., Feng, M., dos Santos, C.N., Yu, M., Xiang, B., Zhou, B. and Bengio, Y.: A Structured Self-attentive Sentence Embedding, *The International Conference on Learning Representations* (*ICLR '17*) (2017).
[8] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *The International Conference on Learning Representations* (*ICLR '14*) (2014).
[9] Simonyan, K., Vedaldi, A. and Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, *Workshop on ICLR* (2014).
[10] Erhan, D., Bengio, Y., Courville, A. and Vincent, P.: Visualizing higher-layer features of a deep network, Technical Report 1341, University of Montreal (2009).
[11] Samek, W., Binder, A., Montavon, G., Lapuschkin, S. and Müller, K.: Evaluating the Visualization of What a Deep Neural Network Has Learned, *IEEE Trans. Neural Networks and Learning Systems*, Vol.28, No.11, pp.2660–2673, DOI: 10.1109/TNNLS.2016.2599820 (2017).
[12] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization, *2017 IEEE International Conference on Computer Vision* (*ICCV*), pp.618–626, DOI: 10.1109/ICCV.2017.74 (2017).
[13] Jeon, H.: Let Sentiment Classification Model speak for itself using Grad CAM, available from ⟨https://medium.com/apache-mxnet/let-sentiment-classification-model-speak-for-itself-using-grad-cam-88292b8e4186⟩ (accessed 2021-09-20).
[14] Visualization of Basis of Decision of NLP Model using Grad-CAM, (in Japanese), available from ⟨https://ymym3412.hatenablog.com/entry/2019/03/19/022240⟩ (accessed 2021-09-20).
[15] Li, J., Chen, X., Hovy, E. and Jurafsky, D.: Visualizing and Understanding Neural Models in NLP, *Proc. 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.681—691, Association for Computational Linguistics, DOI: 10.18653/v1/N16-1082 (2016).
[16] DeYoung, J., Jain, S., Rajani, N.F., Lehman, E., Xiong, C., Socher, R. and Wallace, B.C.: ERASER: A Benchmark to Evaluate Rationalized NLP Models, *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, pp.4443–4458, Association for Computational Linguistics, DOI: 10.18653/v1/2020.acl-main.408 (2020).
[17] Serrano, S. and Smith, N.A.: Is Attention Interpretable? *Proc. 57th Annual Meeting of the Association for Computational Linguistics*, pp.2931–2951, Association for Computational Linguistics, DOI: 10.18653/v1/P19-1282 (2019).
[18] Arras, L., Horn, F., Montavon, G., Müller, K.-R. and Samek, W.: What is relevant in a text document?: An interpretable machine learning approach, *PloS One*, DOI: 10.1371/journal.pone.0181142 (2017).
[19] Nakamura, K. and Yamaguchi, S.: A Study on Providing Interpretability on Classification f Subjective Documents by Machine Learning, *WebDB Forum 2019*, 1C-1 (2019). (in Japanese)
[20] Nakamura, K. and Yamaguchi, S.: A Study on Provision of Interpretability of Document Classification Results based on Deep Learning with Attention, *The 83th National Convention of IPSJ*, 6L-08 (2021). (in Japanese)
[21] Tamekuri, A., Nakamura, K., Takahashi, Y. and Yamaguchi, S.: A Study on Presenting Decision Rationale for Topic Classification of Documents by Deep Learning, *IPSJ SIG Technical Reports*, Vol.2021-NL-249, No.1, pp.1–7 (2021).
[22] Feiyu, X., Hans, U., Yangzhou, D., Wei, F., Dongyan, Z. and Jun, Z.: Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges, *Natural Language Processing and Chinese Computing*, pp.563–574, Springer International Publishing, DOI: 10.1007/978-3-030-32236-6_51 (2019).
[23] Adadi, A. and Berrada, M.: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), *IEEE Access*, Vol.6, pp.52138–52160, DOI: 10.1109/ACCESS.2018.2870052 (2018).
[24] Miller, T., Howe, P. and Sonenberg, L.: Explainable AI: Beware of Inmates Running the Asylum Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences, arXiv:1712.00547 (2017).
[25] Explainable AI BETA, Tools and frameworks to understand and interpret your machine learning models. available from ⟨https://cloud.google.com/explainable-ai⟩ (accessed 2021-09-20).



Atsuki Tamekuri now stays in Kogakuin University to study information and communication technology.

Kosuke Nakamura received his M.E. degrees in Engineering Kogakuin University in 2022.

Yoshihaya Takahashi received his B.E. degrees in Engineering Kogakuin University in 2020. He now stays in Kogakuin University to study electrical engineering and electronics

Saneyasu Yamaguchi received Engineering Doctor's degree (Ph.D.) at Tokyo University in 2002. During 2002–2006, he stayed in Institute of Industrial Science, the University of Tokyo to study I/O processing. He now with Kogakuin University. Currently his researches focus on operating systems, virtualized systems, and storage system.

(Editor in Charge: *Kanako Komiya*)