Classification of Smartphone Application Reviews Using Small Corpus Based on Bidirectional LSTM Transformer

Kazuyuki Matsumoto, Seiji Tsuchiya, Takumi Kojima, Hiroya Kondo, Minoru Yoshida, and Kenji Kita

***Abstract*—This paper provides the classification of the review texts on a smartphone application posted on social media. We propose a high performance binary classification method (positive/negative) of review texts, which uses the bidirectional long short-term memory (biLSTM) self-attentional Transformer and is based on the distributed representations created by unsupervised learning of a manually labelled small review corpus, dictionary, and an unlabeled large review corpus. The proposed method obtained higher accuracy as compared to the existing methods, such as StarSpace or the Bidirectional Encoder Representations from Transformer (BERT).**

***Index Terms*—Attention mechanism, review classification, small corpus, transformer.**

I. Introduction

With the recent increase in the number of mobile terminal devices, especially smartphones, communication using social media has been increasing among the youth. In the past, e-mails, Internet bulletin boards, chats, etc. used to be main tools for communication. However, at present, social media platforms such as Twitter [1], Facebook [2], and Instagram [3] have become mainstream.

Twitter is a social media platform where we can post short messages called tweets, together with videos, images, or web links. This platform is compatible with smartphones and its users can freely post messages, making it a prime real-time property. Twitter is different from Facebook or LINE [4] in anonymity and some Twitter users use different accounts depending on their usage, such as business, private, or hobbies. On Twitter, users can read reputations of various products or services by referring to other tweets. Among such products and services, smartphone applications are considered as one of the best model systems on which users can easily post their reviews as both the applications and the Twitter platform are commonly used on a smartphone device.

The latent application users can save time in collecting information if they can see reputations of the applications from the word-of-mouth tweets regarding the corresponding smartphone application. The applications that can be used on a smartphone can be categorized as games, health, information collecting, etc. The applications are also different in terms of pricing (free or paid) and the developing agency (developed by an individual or developed by a company).

Most of the applications are registered on App Store [5] and Google Play [6], which also show reviews posted for each application. However, several of these reviews are complaints or issues with the applications and positive reviews sometimes lack detailed descriptions. Therefore, latent users might have incorrect prepossession to the application if they accept the review comments on their face value.

In this study, we aim to analyze the reviews, for the benefit of the latent application users, by targeting the word-of-mouth tweets including users' real opinions. We collected the word-of-mouth data regarding an application, annotated polarity labels on the words that can be judged as evaluation expressions, and annotated polarity labels on the word-of-mouth tweets. Based on them, we created training data and an evaluation expression dictionary. In this process, we constructed a foundation to classify application reviews that is small in size but includes all the important elements. Moreover, to classify the kind or category of the application, we trained the distributed representations from the existing large size of review corpus. This process also helped improve the robustness of unknown expressions.

In this study, by combining a Transformer with a self-attention mechanism and long short-term memory (LSTM) networks, a model with a higher performance than those of the existing methods has been developed.

Section II describes the existing studies on review classifications, and Section III provides an overview and flow of the proposed method. Section IV describes the dataset used in this study and the experimental condition. Section V mentions the results and discussion. Finally, Section VI concludes the paper.

II. Related Studies

A. Studies on Review Text Classification

This section describes previous studies on review text classification. Generally, the studies on classification of an evaluation sentence, which is written based on the writers' subjective views, such as a review sentence, are termed under sentiment analysis. The field of sentiment analysis includes studies that classify emotional polarity as positive or negative, based on word or sentence structure obtained from the document [7]-[31], studies that analyze the evaluation score, which indicates the degree of emotion, and studies that judge emotion or non-emotion.

These studies either use large training datasets or analyze

Manuscript received November 9, 2020, revised January 2, 2020.

K. Matsumoto, T. Kojima, M. Yoshida, K. Kita are with Tokushima University, Minamijosanjima-cho 2-1, 7708506, Japan (e-mail: matumoto@is.tokushima-u.ac.jp, mino@ is.tokushima-u.ac.jp, kita@ is.tokushima-u.ac.jp).

S. Tsuchiya is with Doshisha University, Kyoto, Japan (e-mail: stsuchiy@mail.doshisha.ac.jp).

H. Kondo is with Sharp Corporation, 5908522, Osakafu, Sakai-shi, Takumicho-1, Japan.

doi: 10.18178/ijmlc.2020.10.1.912

the documents based on the dictionary. However, even though the task is similar, if the domain of the target text is different, the dictionary or model corresponds to the domain, and the creation cost increases significantly.

Kobayashi *et al*. [32] constructed the dictionary [33], which includes evaluation expressions annotated with evaluation polarity, and to be public. This dictionary provides references to commonly used evaluation terms. However, for smartphone applications, which is a recently developed field, several new words or slangs are included in the review text. Hence, the general evaluation expression dictionary is not sufficient for the classification of the review texts of smartphone applications.

Asakura *et al.* [34] applied the neural attention mechanism to aspect-base sentiment analysis (ABSA). Aspect-base Sentiment Analysis estimates emotional polarity of a review text based on multiple aspects [35]-[37]. They used the dataset of SemEval2016 Task5 Subtask 1 (SE16T5S1) [38], and applied semi-supervised learning on the neural attention model by using word vectors, which were pre-trained based on Google News Corpus as initial parameters. Because this study treats short text on Twitter, it is not basically considered that the sentence which multiple views included in. However, by way of exception, even if some tweets include multiple opinions, we targeted the sentence that could be judged as having a positive/negative polarity.

B. Review Dataset

The Internet Movie Database (IMDb) [39] is now available as review data for sentiment analysis. This dataset includes 25,000 review texts about movies for training and for testing, which are classified into positive/negative depending on the review contents (positive: 12,500, negative: 12,500). The dataset is often used as a benchmark to evaluate a model for sentiment analysis. However, because IMDb is written in English, it cannot be used for classifying application reviews written in Japanese.

The Amazon Review data [40] also includes English language texts and does not include review texts in Japanese.

The UMass Amherst Linguistics Sentiment Corpora is a corpus [41] that counts word n-gram and total of the Rating of each n-gram in the review texts for the products advertised in Japanese, English, German, and Chinese websites of Amazon. This corpus, however, does not include any review text itself.

The Rakuten dataset [42], which includes reviews of Rakuten's products or accommodations, and the "Intage dataset (Minrepo)" [43] are examples of databases corresponding to a Japanese review corpus.

However, in these corpora, the review scores do not always match with the contents. All such review data whose review scores do not match with the contents can be considered as noise. In this study, because we focused on word-of-mouth reviews about applications that are posted on social media, we created an original dataset by collecting and labelling the word-of-mouth review tweets from Twitter.

C. Prediction of Review Document Quality

Recently, several studies have classified review texts by using methods based on neural networks [44]-[48]. This review classification is relatively simple, as it is basically a binary classification. However, the text data on social media platforms such as Twitter often include many hashtags, images, links to other websites, and Retweets that could be considered as noise. Therefore, using these raw data to create a classification model will reduce accuracy.

Ezaki *et al.* [49] proposed a method to assess the effectiveness of a review document by using a classifier to classify a text as review text. Their method calculates the ratio of the review sentences included in a document by using the results of the review classifier, and identifies a review text as useful if the rate is over the certain percentage.

To classify a sentence as a review sentence, the authors used bag of words as a feature and support vector machine (SVM) as a machine learning algorithm. Consequently, their proposed method, based on the percentage of the review content in a document, could obtain 10 % higher accuracy than the simple bag of words document classification. Although Ezaki *et al.* targeted reviews in documents (weblog articles), it would be difficult to apply the same method to the reviews in tweets, which is the target of our study, as there are very few sentences.

To assess the usefulness of a review sentence, Kurahashi *et al*. [50] studied the reviews presented in Amazon. To evaluate the usefulness of a review sentence, the authors combined information obtained from the sentence with other sources, such as the number of characters in the review texts, appearance frequency of each part of speech tag, the rating of the post, posting date, polarity score of the review sentence, and appearance frequency of link. To evaluate usefulness, we labeled the reviews that obtained more than a certain number of answers to the question "it served as a useful reference or not"; In addition, we labeled the reviews that obtained a high approval rate (over 0.7) as high quality and those that obtained a low approval rate (under 0.3) as low quality. An accuracy of 73 % was obtained by the proposed method in the classification experiment using SVM. This accuracy was approximately 10 % higher than that of the method using only bag of words. While analyzing the results of the experiment, it was found that the difference between the score and the average review score and the difference between the polarity score and the average polarity score was very important.

The word-of-mouth reviews on Twitter cannot post scores, and the polarity score was the same as the classification score of the review texts in our study. Therefore, to evaluate the usefulness of a sentence, "the number of characters" or "the appearance frequency of noun" was initially considered. However, on Twitter, word-of-mouth reviews are basically short texts. Hence, it is difficult to assume that the number of characters or the appearance frequency of a part of speech would represent significant features. Therefore, as training data, we used only the high quality review tweets that were judged as "useful" by other people.

Several studies have tried to predict the quality of reviews [51]-[54]. Liu *et al.* [53] detected spam reviews. It is difficult to extract spam reviews by comparing with the other spam texts, as the content quality of spam reviews is not always low. The better a sentence is written, the easier it is to fail to detect it as a spam. In our study, the negative effects of spam were avoided by manually removing the spam reviews in advance.

III. METHODS

A. Overview of the Proposed Method

Our proposed classification algorithm with high accuracy and complementarity is based on small data and large data. Firstly, we used an evaluation expression dictionary created manually as small data.

For the evaluation dictionary, we used the Japanese appraisal evaluation expression dictionary [55] and the Japanese sentiment polarity dictionary [33]. These dictionaries have collected evaluation expressions that often appear in Japanese texts and have classified them in terms of positive/negative polarity. These dictionaries have high quality and wide utility.

Meanwhile, our study targets smartphone application review texts posted on Twitter, which may require considering distinctive expressions that often appear in such review texts.

For review texts regarding a specific application, we manually annotated a polarity label (positive/negative) to individual words and tweets, which helped create high quality data.

To extract features from a review text, we used distributed word representation by CBoW and skip-gram models. These distributed word representation models were trained in advance based on a large scale Wikipedia article corpus. Because these models are not specialized in review text, these might include some redundant knowledge as well.

Therefore, we used a large sized review text corpus in addition to the existing pre-trained distributed word representation models. We constructed a more suitable distributed word representation model by learning distributed word representations with this corpus. The flow of classification model creation is shown in Fig. 1.

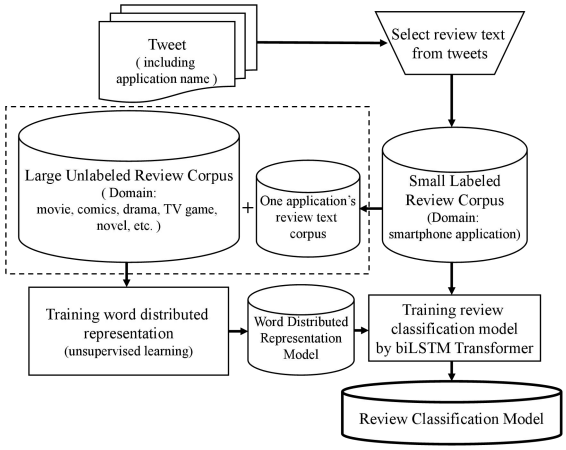


Fig. 1. Creation flow of the review classification model.

B. Transformer Classifier

This subsection describes the review text classification by neural networks using Transformer and attention mechanism. Transformer is one of the encoder-decoder models for neural machine translation (NMT) [56]. Other NMT models include seq2seq (sequence to sequence) [57] as an NMT model based on recurrent neural networks (RNN).

By self-attention, Transformer calculates feature similarity among the words in the input sentence, and encodes position information of each word. By doing so, it can consider a more global relationship than models such as convolutional neural networks (CNN) [58] or RNN. Besides, this model enables parallel computation, enabling it to operate at a higher speed than RNN.

The self-attention in Encoder-Decoder can be expressed as Eq. (1). Q indicates query, K indicates key, and V indicates value. The similarity (Attention) between query and key is calculated by the inner product of Q and K. The inner product between the normalized attention weight by Softmax and V indicates the value corresponding to the key as a weighted sum.

$$\text{Attention}(Q,K,V) = \text{Soft}\max(Q \bullet K^T) \bullet V \tag{1}$$

This mechanism helps pick up the word string that includes similar distributed word expressions by inputting a sentence as a string of distributed word expressions. In other words, a string of distributed word expressions can be decoded into similar sentences.

Eq. (2) and (3) show the positional encoding of Transformer. "pos_w" indicates word position and "i" indicates the index of the vectorized word. "d" indicates the number of word embedding dimensions. These equations calculate positional information tensor that uniquely decides the position of the word and the dimension of the distributed word expression.

$$PE(pos_w, 2i) = \sin(pos_w / 1000^{2i/d}) \tag{2}$$

$$PE(pos_w, 2i+1) = \cos(pos_w / 1000^{2i/d}) \tag{3}$$

In this study, we use the structure of Transformer Encoder, and add a Softmax layer that classifies the output layer as positive/negative. Fig. 2 shows the structure of Transformer networks used in this study.

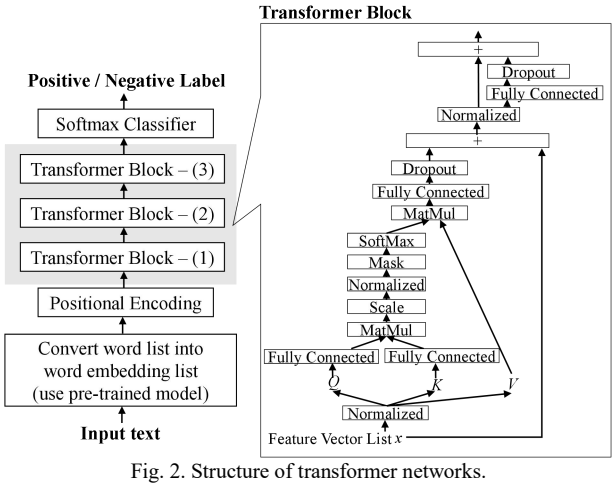


Fig. 2. Structure of transformer networks.

We used three Transformer blocks in the preliminary experiment as it has been shown that three Transformer blocks perform better than the one or two blocks.

C. LSTM Transformer

By using self-attention mechanism and Transformer, review classification model can be suited to review text classification. However, in many cases, if the number of the training data is small, the model cannot work accurately. When the data are insufficient, it tends to be affected by the

pre-training accuracy. In addition, it is necessary to consider the order of words in the review texts.

For example, it is often observed in the review texts that positive opinions are said in the anterior half, and negative opinions are said in the latter half. The correct order of opinions can be identified if LSTM [59] or bidirectional LSTM [60] is used. In this study, we tried to improve the review classification accuracy by combining LSTM-RNN or bidirectional LSTM-RNN with Transformer. The network structures of LSTM Transformer and bidirectional LSTM Transformer are presented in Fig. 3 and Fig. 4. Both of LSTM Block and biLSTM Block include one LSTM/biLSTM layer.

We used two Transformer blocks in the preliminary experiment as it has been shown that two Transformer blocks perform better than one or three blocks.

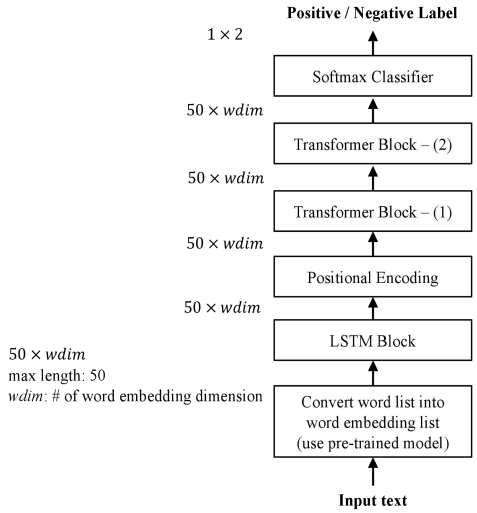


Fig. 3. LSTM-RNN transformer.

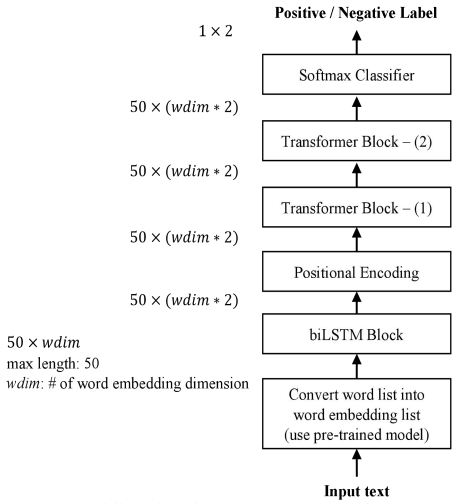


Fig. 4. Bidirectional LSTM-RNN transformer.

D. Convolutional Neural Networks

In this study, we used CNN as a comparison method. Because CNN can consider context, it is expected to demonstrate a higher accuracy than the method using bag of words. We used two-dimensional convolution in the CNN. The structure has 3 convolution layers and 3 max pooling layers. Fig. 5 shows the structure of the CNN.

E. Word Emotion Polarity Encoder Networks

We propose a method to extract word feature based on the model that classifies a text into word emotion polarities.

Based on the existing evaluation polarity dictionary and the dictionary manually made using data from the reviews of certain applications, we trained a positive/negative classifier with deep neural networks.

The distributed word representation vector can be converted into the feature quantity that considers emotion polarity (positive/negative) by using the output (64 dimensional vector) of the fully connected layer directly before the output layer as a feature.

This feature vector was referred to as PN. We fine-tuned this vector based on LSTM and biLSTM with a small review corpus, and constructed the review classification models. Fig. 6 shows the network structure when the PN vector encoder was trained.

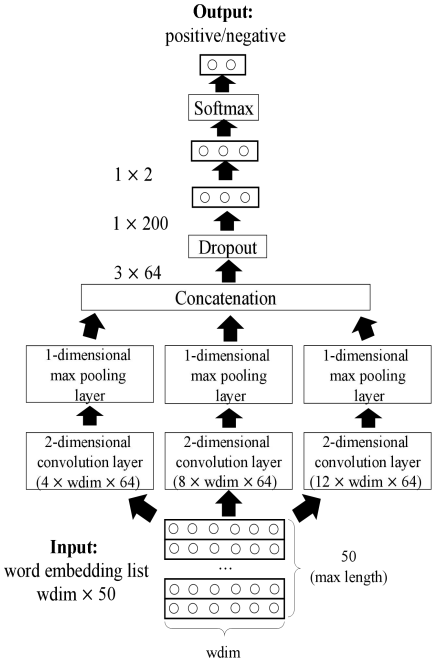


Fig. 5. Structure of CNN.

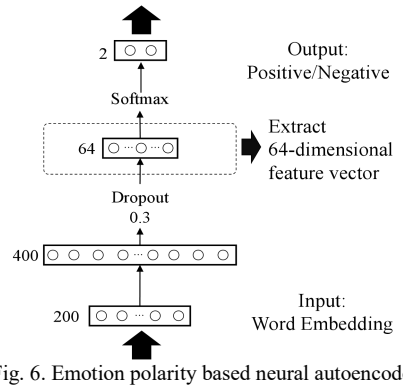


Fig. 6. Emotion polarity based neural autoencoder.

IV. Data

This section describes the application review data, the pre-training review corpus, the evaluation expression dictionary, and noise removal.

A. Word-of-Mouth Tweets Collection

Because the target sentences are not the only sentences including evaluation expression, classification of positive or negative text must be conducted even if the sentence does not

include the evaluation expressions.

Table I shows the breakdown of the target review corpus. S_p and S_n show the number of tweets and W_p and W_n show the total number of words in the positive/negative tweets. The number of the positive tweets was 228, while the number of the negative tweets was 172. In total, 400 tweets were carefully selected from a set of tweets obtained from the Twitter API [61] using the application name as query.

In this study, we manually evaluated the usefulness of the review tweets. The evaluation criteria have been presented below.

1. The reasons of the evaluation of the application or the problems with the application are concretely and clearly mentioned
2. The final evaluation can be clearly judged as positive or negative

If both of the above requirements were met, the review text was judged as a useful review tweet. The examples of review tweets and noise tweets are shown in Table II.

Among the collected review tweets, the maximum number of tweets was obtained for application ID:13 "Hole.io". Therefore, we created an evaluation expression dictionary based on the review texts of this application. Table III shows a few words from the dictionary.

B. Word Embedding

The large scale of review corpus data used to pre-train distributed representations consisted of application ID:13's review texts in addition to the approximately 196,000 review sentences posted under "work databases (*sakuhin* database)" [62].

TABLE I: STATISTICS OF REVIEW TWEETS

ID	Cate.	App. title	S_p	W_p	S_n	W_n
0	game	Pokemon GO	10	319	10	219
1	edu.[1]	Photomath	10	234	2	36
2	game	Dragalia Lost	10	424	10	174
3	game	Surreal Aquarium	9	239	2	32
4	game	Aooni online	9	210	10	306
5	game	Clash Royale	10	270	10	279
6	game	Happy glass	10	271	11	224
7	game	Puzzle & Dragons	8	235	10	273
8	usef.[2]	ZEPETO	10	383	7	108
9	game	Identity V	10	311	10	267
10	usef.[2]	Simeji	8	171	3	58
11	usef.[2]	Customcast	10	230	10	199
12	edu.[1]	Studyplus	1	29	1	54
13	game	Hole.io	46	1683	15	543
14	game	Dragon egg	5	117	10	258
15	game	Deemo	9	331	2	25
16	news	SmartNews	10	301	10	276
17	health	Sleep Meister	6	285	0	0
18	game	Jumpti heroes	10	329	10	297
19	game	Vendetta	7	272	9	302
20	game	GOETIAX	10	337	10	372
21	edu.[1]	Duolingo	10	360	10	447

[1] education application
[2] useful application

We tokenized the corpus with Japanese morphological analyzer MeCab [63] and used the same as a training data. To correctly tokenize the words in the evaluation expression dictionary, which was created based on ID:13's review texts, pre-processing was conducted by jointing the character strings.

TABLE II: EXAMPLE OF REVIEW/NOISE TWEETS

Review /Noise	ID	Cate.	Tweet	Label
Review	2	game	It is a bug that really disappears the treasure. When will it be fixed? "Dragalia Lost" has left too many bugs.	negative
Noise	13	game	I like Hole.io. I play it for killing time.	/
Review	16	news	Recently, "smart news" is hard to read with only ads.	negative
Noise	21	edu.[1]	Duolingo level is now MAX.	/

TABLE III: EXAMPLE OF WORD POLARITY DICTIONARY

Polarity	Example
Positive	ambidexterity, No scary thing, Fair, comfortable, thanks, tailor-made role, fluidity, reasonable price, carnival atmosphere, etc.
Negative	intemperance, firing, starving, stiff shoulder, heart attack, injustice, weak, etc.

The distributed word representation model (word embedding model) was trained based on the CBoW model by using the word2vec [64] module in the gensim [65] Python library. As a training parameter, we set the window size as 5 and the dimension as 200. Other parameters were set as default values. The resulting distributed word representation vector model was named RC.

Table IV presents an overview of the review corpus that was used to train the word embedding model.

TABLE IV: REVIEW CORPUS

Category	# of sentences	# of words
Game	33319	709573
Novel	9503	232429
Movie (Foreign)	15329	337803
Movie (Japanese)	9455	219976
Drama	9412	219508
Comic	46724	948125
Anime	67882	1402000
SFX	4916	113917

To perform a comparison, we also used the Wikipedia Entity Vector (200 dimensions) as "WE," a pre-trained word embedded by fastText [66] (300 dimensions) [67] as "FT," and the word learned through supervised classification embedded with the annotation of emotion polarity labels by StarSpace [68] as "SS."

C. Sentence Embedding

We converted the review tweet sentences into distributed sentence expressions by using the pre-trained model based on Bidirectional Encoder Representations from Transformer (BERT) [69].

We use the pre-trained BERT model [70] based on the Japanese Wikipedia article corpus. This model is distributed by the Kyoto University Kurohashi Laboratory. The 768 dimensional distributed word expressions thus obtained were used to train the review classifier by using machine learning algorithms such as SVM, Adaboosting, random forests, and Light GBM [71]. The hyper parameters of SVM were optimized by using the Grid Search algorithm.

D. Noise Filtering and Stop Word Removal

Generally, several noise strings specific to Twitter can be included in the word-of-mouth review texts. In this study, we defined the stop words and the character string patterns that

can be removed by regular expressions, which could be noise, by referring to the review texts of app ID:13. The examples of the stop words and the regular expressions for noise removal are shown in Table V.

V. Experiment and Discussion

A. Experimental Setup

This section describes the experimental setup. In the experiment, except for the application (ID:13), which was used for the creation of the evaluation expression dictionary and the pre-training of word embedding, the remaining 21 applications were used as the evaluation targets. The accuracy, recall, precision, and the F1-score were calculated by a cross validation test, which splits data by application. Equations (4), (5), (6), and (7) show the calculation of accuracy, recall, precision, and F1-Score.

TP indicates a true positive, i.e., the frequency of the true label that matched with the predicted label. *TN* indicates a true negative, i.e., the frequency of the false label that did not match with the predicted label. *FP* indicates a false positive, i.e., the frequency of the predicted label that matched with the false label. *FN* indicates a false negative, i.e., the frequency of the false label that did not match with the predicted label.

TABLE V: Example of Noise

Kind of Noise		Example
Regular expressions	link address	https?://[¥w/:%#$&¥?¥(¥)~¥.=¥+¥-]+
	digit sequence	[0-9０-９]+
Stop words	Name of review target application	Happy glass, PuzzDra, Pokemon, etc.
	Game character's name	Pikachu, *Yahiko* , etc.
	Particle of Hiragana 1-2 characters	*no, to, ta, node, da*, etc.
	Symbols	！、…, etc.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \times 100 \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100 \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100 \quad (6)$$

$$\text{F1} - \text{score} = \frac{\text{Precision} \times \text{Recall} \times 2}{\text{Precision} + \text{Recall}} \quad (7)$$

To avoid overfitting, we tuned the training number of trials based on the validation loss. Consequently, the optimized epoch numbers for each method were in the range of 5 to 30. Therefore, we compared the experimental result when the number of epochs was between 5 and 30.

We used PyTorch [72] as the deep learning framework. We used Ubuntu 18.04 LTS as the Operating System, and Geforce GTX980 as the Graphic Processing Unit. The SVM or Adaboosting algorithms were run using the machine learning algorithms in the scikit-learn [73] Python library. To train the bag of words feature as baseline method, we used the LBFGS logistic regression classifier of Clasiass [74].

B. Results

Tables V and VI show the accuracy and F1-Score of before noise removal and after noise removal scenarios, respectively, in descending order according to accuracy. "Vector" indicates the kind of vector used as feature in the experiment. The abbreviations are explained as follows.

- RC: unlabeled review corpus based 200-dimensional embedding which was pre-trained by word2vec
- FT: unlabeled Wikipedia article based 300-dimentional embedding which was pre-trained by fastText
- WE: Wikipedia Entity Vector (300-dimentional embedding) [75]
- BERT: 768-dimentional sentence embedding which was pre-trained by BERT using Wikipedia article corpus
- BoW: word appearance frequency based bag of words vector made using labelled training data
- SS: word embedding trained by StarSpace using labelled training data
- PN: 64-dimentional word embedding trained by FFNN using pre-trained word embedding by word2vec

TABLE V: Accuracy for Each Algorithm (Before Noise Removal)

Algorithm	Vector (epochs)	Accuracy	$F1_p$	$F1_n$
biLSTM Transformer	**RC**(20)	81.4	82.5	80.1
biLSTM Transformer	**RC**(25)	80.5	82.1	78.7
SVM	BERT	79.9	77.3	69.2
biLSTM Transformer	**RC**(30)	79.6	80.3	78.8
LSTM Transformer	FT(25)	78.4	81.1	74.7
LSTM Transformer	FT(30)	78.2	80.9	74.5
biLSTM Transformer	FT(25)	78.2	80.0	76.0
Transformer	PN(30)	77.5	78.9	75.9
StarSpace(2-gram, dim200)	SS	77.0	79.7	73.4
LSTM Transformer	FT(20)	76.1	79.9	70.5
lbfgs.logistic	BoW	76.1	51.6	48.2
Transformer	FT(20)	75.7	78.8	71.7
biLSTM Transformer	FT(30)	75.5	79.4	69.8
StarSpace(3-gram, dim100)	SS	75.5	79.1	70.5
LSTM Transformer	**RC**(30)	75.4	77.4	73.1
LSTM Transformer	WE(25)	75.1	76.1	74.1
LSTM Transformer	PN(10)	75.1	77.9	71.6
StarSpace(3-gram, dim200)	SS	74.8	78.0	70.4
Transformer	FT(25)	74.6	90.7	70.5
LSTM Transformer	WE(30)	74.3	80.8	73.9

These results show that noise removal significantly improved the accuracy levels. With respect to word embedding, better results are obtained in order of RC > FT > WE. BiLSTM Transformer could obtain high accuracy and the most stability among all the training algorithms.

Meanwhile, if the feature vector is extracted by using BERT, the number of words in the review texts decreased while applying noise removal. In addition, because valid feature vectors cannot be extracted, the accuracy decreased to 73.5 %.

Fig. 7 shows the comparison of accuracy and F1-score between the methods without noise removal. It was found that the F1-score (Positive and Negative) and accuracy were low for the baseline method of lbfgs.logistic that uses the bag of words feature.

It is thought that the frequencies of specific words affect the accuracy. In the small review corpus, proper nouns related to the application as a review target tend to be repeatedly used. Though noise removal can improve this partially, if a word has the same notation as the general noun,

it becomes difficult to distinguish between noise and data.

Meanwhile, because the method using CNN did not obtain high accuracy, CNN could identify the peripheral word's relationship. However, it could not accurately classify the text with which the review polarity changes between the first half and the latter half.

TABLE VI: ACCURACY FOR EACH ALGORITHM (AFTER NOISE REMOVAL)

Algorithm	Vector (epochs)	Accuracy	$F1_p$	$F1_n$
biLSTM Transformer	**RC**(30)	84.0	85.0	82.9
LSTM Transformer	FT(30)	83.4	85.5	80.7
biLSTM Transformer	FT(30)	83.2	85.2	80.5
biLSTM Transformer	WE(25)	82.9	84.7	80.5
biLSTM Transformer	WE(30)	82.2	84.2	79.7
biLSTM Transformer	**RC**(25)	80.8	81.7	79.9
biLSTM Transformer	**RC**(15)	80.2	81.2	79.1
LSTM Transformer	FT(25)	79.9	82.5	76.6
biLSTM Transformer	PN(25)	79.4	81.9	76.0
biLSTM Transformer	FT(25)	79.1	82.3	74.4
LSTM Transformer	**RC**(25)	78.4	80.8	75.3
lbfgs.logistic	BoW	77.9	52.5	47.3
biLSTM Transformer	**RC**(20)	77.5	79.5	75.2
LSTM Transformer	WE(30)	77.5	80.3	73.8
LSTM Transformer	**RC**(20)	77.3	80.4	73.0
LSTM Transformer	FT(15)	77.2	80.6	72.4
LSTM Transformer	FT(20)	76.7	79.7	72.7
biLSTM Transformer	PN(30)	76.4	79.1	73.0
Transformer	FT(30)	76.3	80.4	70.1
LSTM Transformer	**RC**(30)	75.8	78.3	72.7

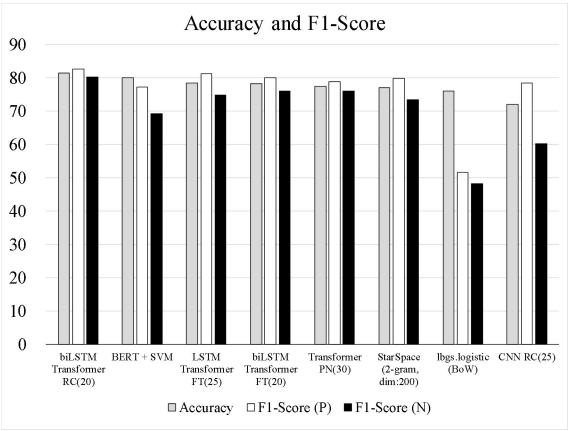


Fig. 7. Comparison of Accuracy and F1-Score.

The classifier using pre-trained word embedding based on SS could obtain a balanced classification. However, because this method cannot treat unknown expressions, which do not appear in the training review corpus, it obtained a lower accuracy than the method using word embedding based on large review corpus (RC).

Meanwhile, because a lower accuracy (68.9%) was obtained by the method based on Transformer classifier using word embedding RC, it could be due to the poor performance of the Transformer.

Fig. 8 shows the accuracy for each application with respect to the four methods; biLSTM Transformer (RC30), StarSpace (2-gram, dim:200), SVM+BERT, and lbfgs.logistic + BoW.

From this graph, we can see that there is a small variation in accuracy between the various types of applications.

Among the four methods, BERT+SVM could achieve micro accuracy (average accuracy), which was the maximum accuracy, and biLSTM Transformer (RC30) obtained 76.2 % accuracy, which was the lowest macro accuracy. Therefore, it was found that high accuracy can be achieved irrespective of the application type by using the BERT feature.

The biLSTM Transformer (RC30) achieved extremely low accuracy (ID:17, 16.7%). The number of review texts for this application is only six. A possible reason for this low accuracy could be that this is the only application that belongs to the category "health." Thus, this application is different from the applications that belong to the major category of "game," and the review content of this application's review texts was very different from the other review texts.

Overall, it was better in the unlabeled review corpus-based pre-trained word embedding as compared to that based on the large size of the Wikipedia corpus. It is expected that the accuracy improved by increasing the labelled corpus.

Because the accuracy achieved by StarSpace, which did not use pre-training, was 77 %, it was considered that the performance could be improved by integrating the pre-training word embedding used in our proposed method.

C. Error Analysis

We analyzed the errors by visualizing attention about the review texts, which were misclassified by the biLSTM Transformer + RC.

Table VII presents visualization of attentions at each step for each review text. It means that the darker the background color of a word, the higher is the weight of attention. In this study, the low frequency words were removed during pre-processing of review texts. Therefore, in this table, some words from the original review text do not appear.

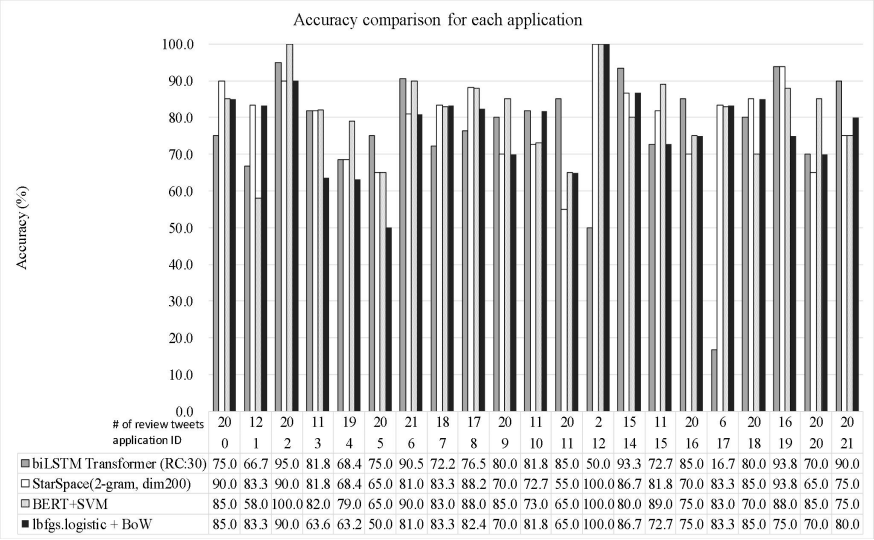
In Example-1, in the first step, higher weights were assigned to "Useful," "*te*," and "*sou*." In the second step, higher weight was assigned to "*sugi*." As "*-sugiru*" in Japanese means excess, it was considered appropriate for feature expression of review texts. However, as the step-2 weight of "useful (*Benri*)" was lower than the step-1 weight, the label was predicted as "negative." It is considered as factor of misjudge that the smaller number of words which can be key to judge.

In Example-2, the negative words such as "*Kusoge*-" (crappy game) and "*kuso*" (damn) were assigned low weights in step-1. However, these words were assigned high weights in step-2. The other non-distinctive words were also assigned high weights. The final evaluation was "positive," which is a misclassification. For the Transformer+FT method, this example predicted and assigned weights more accurately as compared to other methods. Because attention weights were accumulated by the biLSTM method, the assigning of weights was not well-modulated.

D. Discussions

The experiments showed that the review classifier can classify with high accuracy even if the training corpus is small, provided the high quality feature can be obtained beforehand.

The maximum accuracy of 84.0% was obtained by the Bidirectional LSTM with Transformer based on word embedding obtained through unsupervised learning from the large review corpus as feature and with noise removal. This accuracy is approximately 6 % higher than the baseline method, which uses the simple word frequency based bag of words feature.



# of review tweets	20	12	20	11	19	20	21	18	17	20	11	20	2	15	11	20	6	20	16	20	20
application ID	0	1	2	3	4	5	6	7	8	9	10	11	12	14	15	16	17	18	19	20	21
biLSTM Transformer (RC:30)	75.0	66.7	95.0	81.8	68.4	75.0	90.5	72.2	76.5	80.0	81.8	85.0	50.0	93.3	72.7	85.0	16.7	80.0	93.8	70.0	90.0
StarSpace(2-gram, dim200)	90.0	83.3	90.0	81.8	68.4	65.0	81.0	83.3	88.2	70.0	72.7	55.0	100.0	86.7	81.8	70.0	83.3	85.0	93.8	65.0	75.0
BERT+SVM	85.0	58.0	100.0	82.0	79.0	65.0	90.0	83.0	88.0	85.0	73.0	65.0	100.0	80.0	89.0	75.0	83.0	70.0	88.0	85.0	75.0
lbfgs.logistic + BoW	85.0	83.3	90.0	63.6	63.2	50.0	81.0	83.3	82.4	70.0	81.8	65.0	100.0	86.7	72.7	75.0	83.3	85.0	75.0	70.0	80.0

Fig. 8. Comparison for each application (4 methods).

TABLE VII: FAILURE EXAMPLE OF ATTENTION

		(*Benri* / *sugi* / *te* / *sou*)	Correct	Output
Example-1	1st Attention	便利 すぎ て そう	Positive	Negative
	2nd Attention	便利 すぎ て そう		
		(*Hontouni* / *kusoge-* / *da* / *yo* / *na* / . / *gacha* / *no* / *mo* / *kuso* / *ishi* / *mo* / *nai* / *shi* / *nani* / *ga* / *na* / *nn* / *da* / *yo* / *kono* / *game* / . / *event* / *mo* / - / . / *yori* / *daro* / *w* / *zettai* / *kono* / *game* / *tsukut* / *ta* / *ha* / *da* / *yo* / *na* / .	Correct	Output
Example-2	1st Attention	本当に クソゲー だ よ な 。 ガチャ の も クソ 石 も ない し 何 が な ん だ よ この ゲーム 。 イベント も ー 。 より だろ w 絶対 こ の ゲーム 作っ た は だ よ な 。	Negative	Positive
	2nd Attention	本当に クソゲー だ よ な 。 ガチャ の も クソ 石 も ない し 何 が な ん だ よ この ゲーム 。 イベント も ー 。 より だろ w 絶対 こ の ゲーム 作っ た は だ よ な 。		

The methods that classify using SVM, based on the feature that is obtained by using high versatile feature extraction method such as BERT, could achieve relatively high accuracy (79.9%). Therefore, the quality and quantity of the pre-trained corpus and the applied training algorithm were found to be important in developing a classifier based on the small labelled corpus.

As the result of error analysis, a lot of example which could not focus on characteristic words well by attention. For example, there were cases where the self-attention mechanism could not use the similarity between words and the unknown words that could not be pre-trained.

The accuracy can be improved by preferentially focusing on evaluation expressions by using the manually constructed evaluation expression dictionary. However, low accuracy was achieved while using the polarity score of the PN by the word polarity classifier based on the evaluation expression dictionary. The method adds the polarity score vector to the words that do not have sentiment polarity in reality, which potentially reduces the accuracy.

A similar accuracy was achieved using StarSpace algorithm where the classifier was based on Transformer using word embedding that was trained based on labelled training corpus. However, the accuracy was lower than that of the method using the other embedding method and the LSTM Transformer. Hence, we believe that the size of the pre-training data affects the accuracy.

VI. CONCLUSIONS

This study aims to create a high precision review classification model based on the small labelled review texts. In addition to the manually created small review corpus and dictionary, we propose a method that trains the classifier model by combining the large sized unsupervised review text corpus.

The proposed method, that uses word embedding dedicated to review texts as a feature based on self-attention Transformer and bidirectional LSTM networks, could achieve a higher accuracy than the algorithms such as the recently developed StarSpace or BERT, which use embedding based classification.

An improvement in accuracy could be observed by removing noise based on the stop word list, which was made manually.

In future, we would like to develop a method that would automatically detect noise words from review text and prepare a more sophisticated and flexible review classifier by adding the noise removal process as a pre-processing step.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP18K11549.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Kazuyuki Matsumoto, Seiji Tsuchiya, Minoru Yoshida and Kenji Kita concuted the research; Takumi Kojima and Hiroya Kondo analyzed the data and surveyed the related

research field. Kazuyuki Matsumoto, Seiji Tsuchiya and Minoru Yoshida wrote the paper; all authors had approved the final version.

REFERENCES

[1] Twitter. [Online]. Available: https://twitter.com/
[2] Facebook. [Online]. Available: https://www.facebook.com/
[3] Instagram. [Online]. Available: https://www.instagram.com/
[4] LINE. [Online]. Available: https://line.me/ja/
[5] App Store. [Online]. Available: https://www.apple.com/jp/ios/app-store/
[6] Google Play. [Online]. Available: https://play.google.com/
[7] B. Heerschop, F. Goossen, and A. Hogenboom, "Polarity analysis of texts using discourse structure," in *Proc. the 20th ACM Conference on Information and Knowledge Management*, Glasgow, United Kingdom, October 24-28, 2011.
[8] M. Taboada and J. Grieve, "Analyzing appraisal automatically," in *Proc. AAAI Spring Symposium on Exploring Attitude and Affect in Text*, 2004, pp. 158-161.
[9] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proc. the 40th Annual Meeting on Association for Computational Linguistics*, 2002, pp. 417-424.
[10] H. Takamura, T. Inui, and M. Okumura, "Extracting semantic orientations of words using spin model," in *Proc. the 43rd Annual Meeting on Association for Computational Linguistics*, 2005, pp. 133-140.
[11] D. Tang, F. Wei, B. Qin, L. Dong, T. Liu, and M. Zhou, "A joint segmentation and classification framework for sentiment analysis," in *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Oct. 2014, pp. 477–487.
[12] Z. Zhang and M. P. Singh, "ReNew: A semi-supervised framework for generating domain-specific lexicons and sentiment analysis," in *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014)*, 2014, pp. 542-551.
[13] B. Pang, L. Lee, and S. Vaihyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proc. the EMNLP*, Philadelphia, Pennsylvania, USA, 2002, pp. 79–86.
[14] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1-135, 2008.
[15] K. Tsutsumi, K. Shimada, and T. Endo, "Movie review classification based on multiple classifier," in *Proc. the 21st Pacific Asia Conference on Language, Information and Computation (PACLIC21)*, 2007, pp. 481-488.
[16] I. Titov and R. McDonald, "A joint model of text and aspect ratings for sentiment summarization," in *Proc. ACL-08: HLT*, Ohio, USA, Jun. 2008, pp. 308–316.
[17] X. Lei, X. Qian and G. Zhao, "Rating prediction based on social sentiment from textual reviews," *IEEE Trans. on Multimedia*, vol. 18, iss. 9, pp. 1910-1921, Sept. 2016.
[18] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 5, 2015.
[19] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social web," *Journal of the American Society for Information Science and Technology*, vol. 63, iss. 1, Oct. 2011.
[20] S. Mukherjee and P. Bhattacharyya, "Feature specific sentiment analysis for product reviews," in *Proc. International Conference on Intelligent Text Processing and Computational Linguistics*, 2012, vol. 7181, pp.475-487.
[21] L. Zhang, K. Hua, H. Wang, G. Qian, and L. Zhang, "Sentiment analysis on reviews of mobile users," *Procedia Computer Science*, vol. 34, pp. 458-465, 2014.
[22] S. Rill, D. Reinel, J. Scheidt, and R. V. Zicari, "PoliTwi: Early detection of emerging political topics on twitter and the impact on concept-level sentiment analysis," *Knowledge-Based Systems*, vol. 69, pp. 24-33, 2014.
[23] A. Ghose and P. G. Ipeirotis, "Designing novel review ranking systems: predicting the usefulness and impact of reviews," in *Proc. the Ninth International Conference on Electronic Commerce*, Minneapolis, MN, USA, 2007.
[24] Z. Zhang and B. Varadarajan, "Utility scoring of product reviews," in *Proc. the 15th ACM International Conference on Information and Knowledge Management*, Arlington, Virginia, USA, 2006.
[25] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 1555–1565.
[26] S. Batra and D. Rao, *Entity Based Sentiment Analysis on Twitter*, Stanford University, 2010.
[27] X. Lei, X. Qian, and G. Zhao, "Rating prediction based on social sentiment from textual reviews," *IEEE Transactions on Multimedia*, vol. 18, iss. 9, pp. 1910-1921, 2016.
[28] S. D. Kok, L. Punt, R. V. D. Puttelaar, K. Ranta, K. Schouten, and F. Frasincar, "Review-level aspect-based sentiment analysis using an ontology," in *Proc. the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 315-322.
[29] S. Gojali and M. L. Khodra, "Aspect based sentiment analysis for review rating prediction," in *Proc. 2016 International Conference on Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, 2017.
[30] R. Wadbude, V. Gupta, D. Mekala, and H. Karnick, "User bias removal in review score prediction," in *Proc. the ACM India Joint International Conference on Data Science and Management of Data*, 2018, pp. 175-179.
[31] P. Basile, V. Basile, M. Nissim, and N. Novielli, "Deep Tweets: From entity linking to sentiment analysis," in *Proc. the Italian Computational Linguistics Conference* (CLiC-it 2015), 2015.
[32] N. Kobayashi, K. Inui, Y. Matsumoto, K. Tateishi, and S. Fukushima, "Collecting evaluative expressions for opinion extraction," *Journal of Natural Language Processing*, vol. 12, no. 3, pp. 203-222, 2005.
[33] Japanese Sentiment Polarity Dictionary. [Online]. Available: http://www.cl.ecei.tohoku.ac.jp/index.php?Open%20Resources%2FJapanese%20Sentiment%20Polarity%20Dictionary
[34] R. Asakura, H. Niitsuma, and M. Ohta, "Effective Usage of Neural Attention on Aspect-Based Sentiment Analysis," presented at DEIM Forum 2018 G4-4.
[35] H. Niitsuma, D. Kubota, and M. Ohta, "Japanese Sentiment Analysis Using Simple Alignment Sentence classification," in *Proc. the 10th International Conference on Management of Digital EcoSystems (MEDES'18)*, 2018, pp. 126-131.
[36] H. Niitsuma, R. Asakura, and M. Ohta, "Simple alignment sentence classification for aspect-based sentiment analysis," in *Proc. 19th International Conference on Computational Linguistics and Intelligent Text Processing*, 2018.
[37] C. Sun, L. Huang, and X. Qiu, "Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence," in *Proc. NAACL-HLT 2019*, Jun. 2019, pp.3 80–385.
[38] SemEval-2016 Task 5: Aspect-Based Sentiment Analysis. [Online]. Available: http://alt.qcri.org/semeval2016/task5/
[39] IMDb (Internet Movie Database). [Online]. Available: https://www.imdb.com/
[40] Amazon Review Data. [Online]. Available: http://jmcauley.ucsd.edu/data/amazon/
[41] UMass Amherst Linguistics Sentiment Corpora. [Online]. Available: https://semanticsarchive.net/Archive/jQ0ZGZiM/readme.html
[42] Rakuten dataset. [Online]. Available: https://www.nii.ac.jp/dsc/idr/rakuten/
[43] Intage dataset (Minrepo). [Online]. Available: https://www.nii.ac.jp/dsc/idr/intage/
[44] A. Severyn, A. Moschitti, "Twitter Sentiment Analysis with Deep Convolutional Neural Networks," in *Proc. the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 959-962.
[45] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Oct. 2014, pp. 1746–1751.
[46] W. Li, P. Liu, Q. Zhang, and W. Liu, "An improved approach for text sentiment classification based on a deep neural network via a sentiment attention mechanism," *Future Interest*, 2019, 16, 2019.
[47] D. Tang, B. Qin, T. Liu, and Y. Yang, "User modeling with neural network for review rating prediction," in *Proc. the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, 2015, pp. 1340-1346.
[48] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. NAACL-HLT 2016*, San Diego, California, June 12-17, 2016, pp. 1480-1489.
[49] H. Ezaki, M. Kawaba, and T. Hirano, "Review decision about the documents by the ratio of a review sentence by classifier using a sentence review," IPSJ SIG Technical Report, vol. 2012-IFAT-105, no. 2, pp. 1-5, 2012.
[50] H. Kurahashi, and M. Anono, "The judgement experiment of the usefulness among amazon's review," in *Proc. Forum on Information Technology 2013*, 2013, pp. 101-102.
[51] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proc. the 2018 World Wide Web Conference (WWW'18)*, 2018, pp. 1583-1592.

[52] L. Martin and P. Pu, "Prediction of helpful reviews using emotions extraction," in *Proc. the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, pp. 1551-1557.
[53] J. Liu, Y. Cao, C.-Y. Lin, Y. Huang, and M. Zhou, "Low-quality product review detection in opinion summarization," in *Proc. (EMNLP-CoNLL)*, 2007, pp. 334-342.
[54] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Automatically assessing review helpfulness," in *Proc. the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006.
[55] M. Sano, "Reconstructing English system of attitude for the application to Japanese: An exploration for the construction of a Japanese dictionary of appraisal," in *Proc. 38th International Systemic Functional Congress,* 2011.
[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uzkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conference on Neural Information Processing Systems* (NIPS2017), 2017.
[57] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Proceedings of NIPS*, 2014
[58] Y. LeCunn, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436-444, 2015.
[59] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp.1735-1780, 1997.
[60] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
[61] Twitter API. [Online]. Available: https://developer.twitter.com/
[62] Sakuhin Database. [Online]. Available: https://sakuhindb.com/
[63] MeCab: Yet Another Part-of-Speech and Morphological Analyzer. [Online]. Available: https://taku910.github.io/mecab/
[64] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in Vector Space. [Online]. Available: https://arxiv.org/abs/1301.3781
[65] Genism. [Online]. Available: https://radimrehurek.com/gensim/
[66] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 2, 2016, pp. 427–431.
[67] fastText pretrained vector. [Online]. Available: https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md
[68] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, "StarSpace: Embed all the things!" in *Proc. the Thirty-Second AAAI Conference on Artificial Intelligence* (AAAI-18), 2018, pp. 5569-5577.
[69] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv:1810.04805, 2018.
[70] BERT Japanese Pretrained Model. [Online]. Available: http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT%E6%97%A5%E6%9C%AC%E8%AA%9EPretrained%E3%83%A2%E3%83%87%E3%83%AB
[71] Light GBM. [Online]. Available: https://github.com/microsoft/LightGBM
[72] PyTorch. [Online]. Available: https://pytorch.org/
[73] scikit-learn. [Online]. Available: https://scikit-learn.org/stable/
[74] Classias: A collection of machine-learning algorithms for classification. [Online]. Available: http://www.chokkan.org/software/classias/index.html.ja
[75] Japanese Wikipedia Entity Vector. [Online]. Available: http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/

Kazuyuki Matsumoto received the PhD degree in 2008 from Tokushima University. He is currently an assistant professor of Tokushima University. His research interests include affective computing, emotion recognition, artificial intelligence and natural language processing. He is a member of IPSJ, ANLP, IEICE, JSAI, and IEEJ.

Seiji Tsuchiya received the bachelor of engineering from the Faculty of Engineering, Doshisha University, Japan in 2000. He received the master of engineering from Graduate School of Engineering, Doshisha University in 2002. He entered Sanyo Electric Co., Ltd in 2002. He received his Ph.D. from Graduate School of Engineering, Doshisha University in 2007. He was an assistant professor in Bulletin of Institute of Technology and Science the University of Tokushima, Japan. He was an assistant professor in the Faculty of Science and Engineering, Doshisha University in 2009, an associate professor in 2011, and a professor in 2017 there. His research interests include knowledge processing, concept processing, and semantic interpretation.

Takumi Kojima received the B.S. degree from the Faculty of Engineering, Tokushima University in 2018. He is currently a mater degree student of Graduate School of Tokushima University. His research interest is sleeping time analysis from social media by using natural language processing method.

Hiroya Kondo received the B.S. degree from Faculty of Engineering, Tokushima University in 2019. He works in Sharp Corporation. His research interest is opinion mining.

Minoru Yoshida is a lecturer at the Department of Information Science and Intelligent Systems, University of Tokushima. After receiving his BSc, MSc, and PhD degrees from the University of Tokyo in 1998, 2000, and 2003, respectively, he worked as an assistant professor at the Information Technology Center, University of Tokyo. His current research interests include web document analysis and text mining for the documents on the WWW.

Kenji Kita received the B.S. degree in mathematics and the PhD degree in electrical engineering, both from Waseda University, Tokyo, Japan, in 1981 and 1992, respectively. From 1983 to 1987, he worked for the Oki Electric Industry Co. Ltd., Tokyo, Japan. From 1987 to 1992, he was a researcher at ATR Interpreting Telephony Research Laboratories, Kyoto, Japan. Since 1992, he has been with Tokushima University, Tokushima, Japan, where he is currently a professor at Faculty of Engineering. His current research interests include multimedia information retrieval, natural language processing, and speech recognition.