Emotion Recognition of Emoticons Based on Character Embedding

Kazuyuki Matsumoto*, Akira Fujisawa, Minoru Yoshida, Kenji Kita
Graduate School of Technology, Industrial and Social Sciences, Tokushima University, Tokushima University, 2-1 Minamijyousanjima-cho, Tokushima 770-8506, Japan.

* Corresponding author. Tel.: 770-8506; email: matumoto@is.tokushima-u.ac.jp
Manuscript submitted January 6, 2017; accepted May 13, 2017.
doi: 10.17706/jsw.12.11.849-857

Abstract: This paper proposes a method for estimating the emotions expressed by emoticons based on a distributed representation of the character meanings of the emoticon. Existing studies on emoticons have focused on extracting the emoticons from texts and estimating the associated emotions by separating them into their constituent parts and using the combination of parts as the feature. Applying a recently developed technique for word embedding, we propose a versatile approach to emotion estimation from emoticons by training the meanings of the characters constituting the emoticons and using them as the feature unit of the emoticon. A cross-validation test was conducted for the proposed model based on deep convolutional neural networks using distributed representations of the characters as the feature. Results showed that our proposed method estimates the emotion of unknown emoticons with a higher F1-score than the baseline method based on character n-grams.

Key words: Emoticon, emotion recognition, character embedding, convolutional neural networks.

1. Introduction

With the development of communication by text, more and more people, regardless of age or gender, are using Social Networking Services for their daily interactions. One of the advantages of text-based communication via the Internet is that it enables users to talk with anyone, anywhere and to respond almost instantaneously. However, as it is sometimes difficult to convey emotion using text only, opportunities to add non-verbal information such as pictures to a text are available. Emoticons are one type of such non-verbal information. Over time, these conveyors of emotion have proliferated; they are now widely used on electronic bulletin boards, chat systems, and in e-mails. In Japan, especially, the number and diversity of emoticons continue to increase at a rapid pace. There have been numerous studies dealing with emoticons as they have become increasingly important.

It is obvious that assessing emoticons by character unit is not a very effective way to estimate the emotion conveyed by the emoticon. To be successful, it is necessary to know the positions of the characters and their co-occurring characters. Moreover, emoticons look different and give different impressions depending on their font types or contexts. To capture contextual similarities between the characters and the semantic/usage features of the characters, as well as to consider the co-occurrence relationship of emoticons, requires a corpus that includes a substantial variety of emoticons. However, because the types of characters are limited, by modeling what emotion is represented by various character combinations, it is possible to achieve an effective estimation of the associated emotion even with a relatively small supervised data set.

Our study trained character embedding expressions from emoticons by applying a recently developed word embedding method and then used these expressions as the feature for emotion estimation. We estimated the associated emotion by training the character features of the emoticon with deep convolutional neural networks.

2. Related Works

There have been numerous related studies on estimating emotion from texts; however, few studies have exclusively targeted emoticons. One of the reasons for this is that emoticons tend to be treated as signs, like pictographs, and considered not suitable subjects for linguistic semantic analysis. Some studies on emotion estimation from texts have registered emoticons into the emotional expression dictionaries [1], [2].

Yamada *et al*. [3] proposed a facial expression estimation method based on character n-grams. The approach constructs a facial expression estimation model by calculating a character n-gram appearance probability for each facial expression displayed by an emoticon. Results of an evaluation experiment showed that the proposed method had a higher facial expression estimation accuracy than an approach that used a single character appearance feature.

Kazama *et al*. [4] proposed a method that extracts unknown emoticons by using an emoticon extraction algorithm and judging surface/semantic similarities using word2vec. The method was used to estimate emotions from Twitter texts. However, because their method does not consider the similarity between the constituent parts of the emoticons, it cannot be said that their method is robust to all-new emoticons that have never appeared in any corpus. Recently, IBM’s Watson [5] has been used to conduct reputation analysis by considering emoticons. However, there are practical problems with this as the varieties of the target emoticons are quite small in number.

The CAO system proposed by Ptasynski *et al*. [6] is a system for emotion analysis of emoticons using a database containing an emoticon register. In the database, emoticons are annotated with semantic information. It is important to note, however, that single emoticons do not show facial expressions and that the sense of an emoticon changes depending on peripheral words or additional parts such as hands or objects. Expanding the database will not completely eliminate unknown expressions or semantic information.

Dividing emoticons into character units would provide a way to analyze components of the face expressed by each character. The characters could be used to represent various organs in the emoticons. By analyzing a corpus with annotated names of the organs attached to each character in the emoticon, it would be possible to calculate a likelihood that indicates which organs are expressed by which characters. However, annotating each part in a huge emoticon database would require a significant cost. As a result, such annotated emoticon data have not been published, which has discouraged the advancement of associated studies.

The lack of published resources and comprehensive database of emoticons makes it difficult to conduct a large experiment. Our study aims to construct an emoticon emotion estimation (EEE) model by training a large amount of unsupervised emoticon data, capturing the feature of emoticons by character unit, and using a small emotion-labeled corpus as training data.

3. Proposed Method

3.1. Emoticon Character Embedding

Our proposed method uses Word2vec [7] to acquire character embedding (character distributed semantic representation) of emoticons. Word2vec is a method or tool to calculate word distributed representations. It generates a fixed length, real-valued vector for words that appear in the corpus by training neural networks using the feature of skip-gram or continuous Bag-of-Words (CBOW) extracted from the corpus tokenized with word units.

The generated real-valued vector is called a word distributed semantic representation or word embedding vector. The semantic/contextual similarities or relevance between words can be calculated by using this vector as a feature. There are other methods to calculate word distributed semantic representations [8, 9]. However, Word2vec was chosen here because the method is the most widely used tool at present.

In the proposed approach, emoticons are automatically extracted from Twitter texts and are split into character units. The target language is Japanese; the character encoding type is utf8. We decided to use the regular expression rule for automatic extraction and used \([^\]]{4,45}\) as the regular expression pattern for emoticon extraction, since most of the emoticons include (“, “) as a facial contour and at least one character is included in the inner contour. This means we do not treat emoticons that do not include the close bracket, “)”. We set the maximum number of characters inside the brackets at 10 to avoid miss-extraction.

3.2. Deep Convolutional Neural Networks

The use of Deep Convolutional Neural Networks (DCNN) has a record of high performance in image recognition and has been quite effective in text classification [10].

In this study, we used a published implementation of Chainer [11] and a definition of hyper parameters as our DCNN. Fig. 1 shows the network structure of the DCNN used in this study. It consists of an input layer, a convolution layer, and max pooling layers. This structure avoids over-fitting and increases the generalization capability by inserting a dropout function between the layers. Generally, it is said that over-fitting often occurs because deep neural networks have such a complicated structure. Because this study aims to perform emotion estimation with high accuracy for unknown emoticons by using a small training data set with annotation, we regarded versatility as more important.

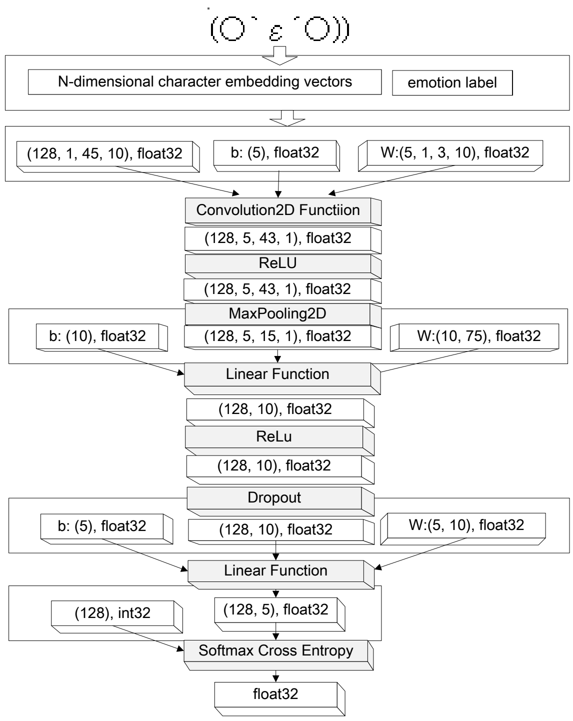

Fig. 1. Structure of deep convolutional neural networks.

We used padding since the lengths of emoticons vary depending on their type. Considering distributions of emoticon lengths, we decided on a maximum sequential length of 45. Symbol 'W' indicates kernel size. Kernel size expresses window size for the convolution process. In the convolutional process, the feature of the t^{th} character positioned from the beginning of a string is expressed by a local feature obtained from window size C. Convolutional process can be interpreted as expressing the feature of the t^{th} character by the summation of the weighed hidden state vectors around the character. Equation 1 calculates feature vector $v_t^{(1)}$ of character c_t when it is limited to 1-dimension. $W^{(1)}$indicates the parameter matrix. $h_{t-\lfloor c/2 \rfloor}^{(l-1)}$, $h_t^{(l-1)}$, $h_{t+\lfloor c/2 \rfloor}^{(l-1)}$ indicate the hidden state vector. $\lfloor x \rfloor$ indicates a floor function. For example, in the case of $x = 2.5$, $\lfloor 2.5 \rfloor = 2$. In this paper, because a character level distributed representation (character embedding) is obtained in the pre-training phase, the convolution is processed by considering N-dimension.

$$v_t^{(1)} = \boldsymbol{W}^{(1)} \begin{bmatrix} h_{t-\lfloor C/2 \rfloor}^{l-1} \\ \vdots \\ h_t^{(l-1)} \\ \vdots \\ h_{t+\lfloor C/2 \rfloor}^{(l-1)} \end{bmatrix} + \boldsymbol{b}^{(1)} \tag{1}$$

3.3. Emotion-Labeled Emoticon Corpus

We constructed an emotion-labeled emoticon dictionary based on the emoticon dictionaries [12-18] published for general utility. Emotion labels are defined based on Fischer's emotion systematic chart [19] shown in Table 1. Examples of emoticons registered into the emoticon dictionary and their emotion labels are shown in Table 2.

Table 1. Definition of Emotion Labels

Emotion label	Joy	Surprise	Anger	Sorrow	Neutral
Emotions	joy, love, etc.	surprise	anger, hate, etc.	sorrow, anxiety, etc.	neutral

Table 2. Example of Emoticons

Emotion	Example of emoticons
Joy	(゜▽＾), （＊^-^＊）, (ヽ(゜∀゜), （＊´▽`）, （^^◎), etc.
Surprise	(゜Д゜;), （○。○）, (゜ x゜), (°Д°lll), (◎゜Д゜σ), ((;;゜Д゜), etc.
Anger	((└(゜ε゜), (`･(ｪ ), (‡゜Д゜), (*`з´), ((゜∀゜`;;), ( `д'), etc.
Sorrow	(ﾉд`), (´д`*;), (T○T), (m__m), (･__:), etc.
Neutral	('∋'), (● ' I ' ●), (￣ -￣), ('･c_, ･`), (･ω･), etc.

4. Experiment

The experimental data are shown in Table 3. We target a corpus for pre-training the character embedding expressions according to the regular expression rule and automatically judge whether an expression is an emoticon. We remove the strings that are judged as non-emoticons from the pre-training corpus (emoticon filtering) and determine whether accuracy increases as a result.

At the same time, we collect the strings of emoticons from the existing emoticon dictionaries (apart from the training data), then set them as positive examples. We calculate cosine similarity between the character strings in these positive examples and those in the pre-training corpus, focusing on the appearance frequency vector of the characters. Then, the strings with similarity less than 0.9 are removed from the pre-training corpus. We use a corpus that was subject to emoticon filtering for comparison.

Table 3. Experimental Data

	Without filtering	With filtering
Pre-training corpus	5,658,013	785,569
Labeled emoticon dictionary	3,080	
Non-labeled emoticon dictionary	5,914	

We also compare the performance between the proposed model and the emotion estimation models trained by support vector machines(svm), random forest(rf), decision tree(dt), Gaussian Naive Bayes(gnb), the K-nearest neighbors algorithm(knn), logistic regression(lr), AdaBoost(adb) and quadratic discriminant analysis(qda) that used the character n-gram ($1 \leq n \leq 4$) as the feature.

In this experiment, we evaluate the proposed and comparative models using five-fold cross-validation. We adjusted the data balance in the training data according to the smallest emotion label numbers in the labeled emotion dictionary. At that time, we randomly selected the emoticons with each emotion label to be used for training data.

Fig. 2 shows the Precision-Recall curve with the highest F1-score—with and without emoticon filtering—for the pre-training corpus. Fig. 3 shows the Precision-Recall curve of the baseline method using the n-gram feature. The labels of the plot points in Fig. 3 indicate the n value of each n-gram.

The marker colors indicate type of emotion: red for joy, green for surprise, blue for anger, black for sorrow, and magenta for neutral. The combinations of parameters in Fig. 2 are the parameters of Word2vec (window, size, mincount) = (5, 300, 1) without filtering, and the parameters of Word2vec: (window, size, mincount) = (3, 300, 5) with filtering.

Fig. 2 shows that the precision and recall of the "neutral" emotion are both 0. Fig. 3 indicates that precision and recall are ill-balanced in the baseline method; as shown, recall is high and precision is low for all the machine learning algorithms.

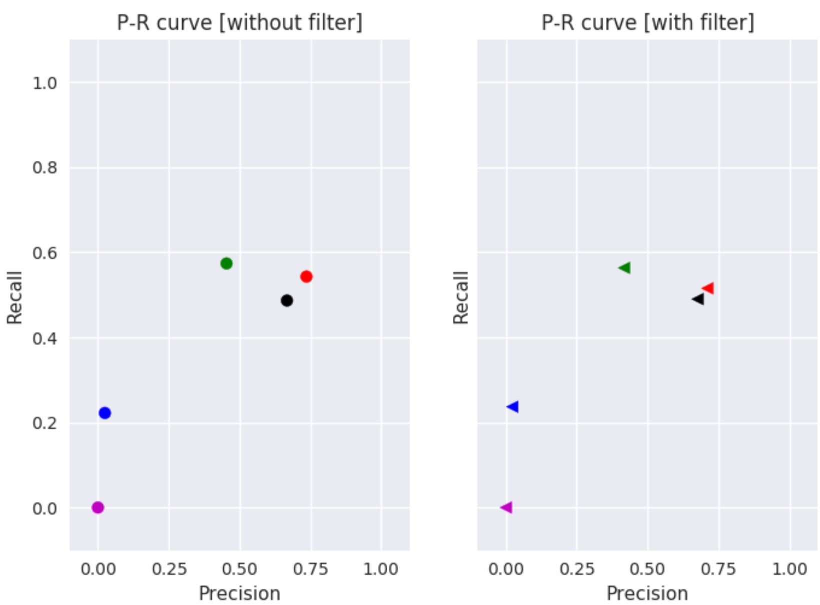

Fig. 2. Precision-Recall curve of the proposed method (with/without filtering).

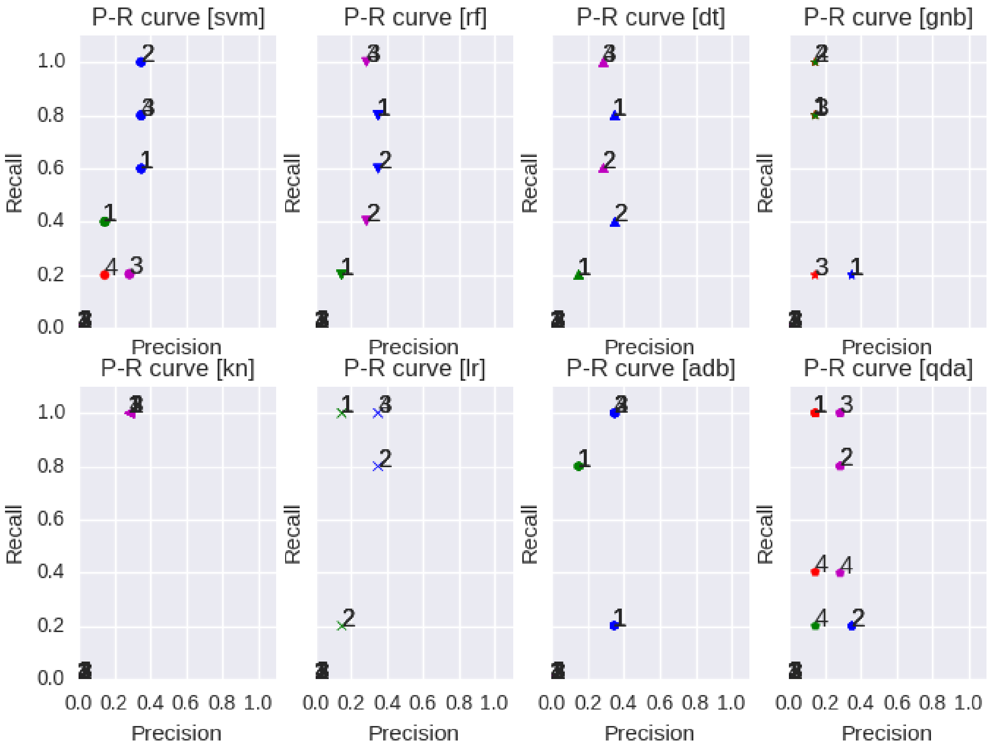

Fig. 3. Precision-Recall curve of the baseline method.

5. Discussion

It is not possible to express positional information for the characters in emoticons by using only character appearance information. One way to consider positional information is to use character n-grams. However, because the types of character n-grams increase with an increase in the value of *n*, the dimension of the feature will increase in proportion to the number of training data values, which could cause over-fitting.

In the evaluation experiment conducted here, the emotion estimation model based on character n-grams produced lower F1-scores than the DCNN method using character embedding as its feature. As seen in the Precision-Recall curves, in the evaluation results for the baseline method, there are many emotions with high recall and low precision. In addition, the various emotion estimators often output the same emotion labels. The F1-scores might be improved by adjusting the parameters for each machine learning algorithm. However, the only way to realize effective learning would be to increase the size of the training data set, as the value of *n* did not affect the F1-scores.

We examined the effectiveness of emoticon filtering for the pre-training corpus but found few differences. As the method without filtering achieved higher F1-scores than the method with filtering, we could not confirm the effectiveness of filtering.

With respect to the adjustment of the parameters of Word2vec for pre-training, it was found that the size of the dimensions of the character embedding vectors affected the F1-scores more than emoticon filtering. This indicates that in a corpus larger than a certain size, the character embedding can be trained with less influence from noise (symbols other than emoticons).

Table 4 shows a comparison of the F1-scores with and without filtering (when the 'neutral' emotion was removed) for the combinations of parameters with the highest F1-scores for the proposed method. Significantly, the F1-score for "Anger" is quite low. On the other hand, the precision for "Joy" and "Sorrow" are high. Notably, there were almost no differences in these tendencies due to filtering.

Table 4. Comparison of F1-Scores for Each Emotion Label

	Without filtering (window=5, size=300, mincount=1)			With filtering (window=3, size=300, mincount=5)		
Emotion	Precision	Recall	F1-score	Precision	Recall	F1-score
Joy	0.736	0.579	0.648	0.711	0.515	0.598
Surprise	0.454	0.581	0.510	0.417	0.563	0.479
Anger	0.025	0.256	0.045	0.022	0.237	0.041
Sorrow	0.667	0.535	0.594	0.677	0.490	0.568
Average	0.471	0.488	0.449	0.457	0.451	0.422

The versatility of emotion estimation from emoticons increased with the use of character embedding. However, it was found that the bias associated with using a small-scale annotated training data set could not be avoided, raising questions regarding the necessity of a large scale emoticon database. It goes without saying that a high quality and completely labeled database would be very helpful for the analysis of the human cognitive mechanism to deal with emoticons and the emotions they express.

In this paper, we aimed to estimate emotions from emoticons that had already been extracted from sentences. However, the sense or emotion expressed by an emoticon varies depending on the context. Moreover, because the ambiguity/polysemy of the emoticon depends on the personalities/cultural backgrounds of the users, future efforts are needed to determine how we might convert such information—information that cannot be determined uniquely—into knowledge.

We compared similarities between characters for the two conditions—with and without filtering. Table 5 shows partial results (for the character “ д ”).

Table 5. Comparison of Similarities between Characters

Target Character: д			
Without filtering		With filtering	
character	similarity	character	similarity
⊿	0.454	艸	0.614
Д	0.439	ｪ	0.604
ﾟ	0.364	Å	0.595
∀	0.361	ﾜ	0.558

The character “ д ” is often used to represent the mouth in emoticons. As can be seen in the table, in the “without filtering” condition, characters with a similar “ д ”, shape are ranked high in the list. On the other hand, in the “with filtering” condition, characters with a dissimilar shape are ranked high, with the “艸” leading the way.

One possible reason is that “艸” is often used in emoticons to show hands covering the mouth; thus, the positional relationship might have caused a high similarity value. Moreover, although vector dimension is the same for both the “with filtering” and “without filtering” conditions, parameters such as window size and mincount are different. These different parameters may have affected the similarity values. In addition, filtering the corpus reduced the corpus size, making it difficult to obtain a sufficient number of examples. As a result, a high-accuracy vector could not be generated.

6. Conclusion

In this paper, we have described a versatile emotion estimation model for emoticons by training deep neural networks and using a character embedding vector trained from the emoticon corpus as the feature. In an evaluation experiment, we produced higher F1-scores than the baseline model, which was based on a simple character n-gram feature without pre-training.

In our experiment comparing results with/without emoticon filtering on the corpus for pre-training, no noticeable improvement in performance was observed when filtering was used. In the future, we intend to improve the method for generating the character embedding vector to increase accuracy and versatility.

There are still problems in resource construction that need to be addressed. One of them involves how we might efficiently increase the size of the annotated corpus to reduce the influence of deviations in the annotated emoticon corpus. As extensive annotation would require a substantial human cost, we would like to find a semi-automatic emotion labeling approach that uses a technique for emotion estimation from sentences.

Acknowledgment

This research has been partially supported by JSPS KAKENHI Grant Number 15K16077, 15K00425, 15K00309.

References

[1] Kawakami, M. (2008). The database of 31 Japanese emoticon with their emoticons and emphases. *The Human Science Research Bulletin of Osaka Shoin Women's University*, 67 – 82.

[2] Mizuoka, Y., & Suzuki, Y. (2011). Automatic collection of emotional expressions using microblogs. *Proceedings of Forum on Information Technology*, 291-294.

[3] Yamada, T., Tsuchiya, S., Kuroiwa, S., & Ren, F. (2007). Classification of facemarks usingn-gram. *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*.

[4] Kazama, K., Mizuki, S., & Sakaki, T. (2016). Study of sentiment analysis using emoticons on twitter. *Proceedings of the 30th Annual Conference of the Japanese Society for Artificial Intelligence*.

[5] Watson. Retrieved from https://www.ibm.com/watson/jp-ja/

[6] Ptaszynski, M., Maciejewski, J., Dybala, P., Rzepka, R., & Araki, K. (2010). CAO: A fully automatic emoticon analysis system. *Proceedings of Twenty-Fourth AAAI Conference on Artificial Intelligence*.

[7] Mikolov, T., Sutskever, I., Chen, K., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Proceedings of the 27 Annual Conference on Neural Information Processing Systems.*

[8] Jeffrey, P., Rechard, S., & Christopher, D. M. (2014). GloVe: Global veectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

[9] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[10] Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (pp.1746-1751).

[11] Ichiroex. Retrieved from https://github.com/ichiroex/chainer-cnn

[12] Kaomoji-Cafe. Retrieved from http://kaomoji-cafe.jp/

[13] Kaomoji. Retrieved from http://kaomoji.ru/en/

[14] Kaomoji-Copy. Retrieved from http://kaomoji-copy.com/

[15] Facemark. Retrieved from http://www.facemark.jp/facemark.htm

[16] Tofugu. Retrieved from https://www.tofugu.com/japanese/kaomoji/

[17] Cutekaomoji. Retrieved from http://cutekaomoji.com/misc/table-flipping/
[18] Kao. Retrieved from http://www.kaomoji.com/kao/text/
[19] Fischner, K. W., Shaver, P., & Carnchan, P. (1989). A skill approach to emotional development: From basic- to subordinate-category emotions. In W. Damon (Eds.), *Child development today and tomorrow*. (pp.107-136). Jossey-Bass.

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 and IEEJ.

Akira Fujisawa received the B.S. degree and M.S. degree in 2013 and 2015 from Faculty of Engineering, Tokushima University. He is currently a PhD candidate at the Tokushima University. His research interests include multimedia information retrieval, image recognition. He is member of JSAI.

Minoru Yoshid 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.