Multi-Channel Distributed Representation for Classifying Tweets by using Convolutional Neural Networks

Shuichi Hashida, Keiichi Tamura, Tatsuhiro Sakai

Abstract—**With the increasing interest in social media, many text messages are being posted on micro-blogging sites. Tweets on Twitter often reflect the user's reaction for topics and events around people. Therefore, microblog mining is one of the most attractive research topics for real-world monitoring. However, since a message on a microblogging site, such as Twitter, is short, classifying such text messages is a challenging task. In our previous work, we proposed a method based on Naive Bayes classifier for classifying tweets under two classes: "relevant" and "irrelevant" to a monitoring topic. The classification performance has limitations because Naive Bayes is based on words' probabilities. To solve this problem, we propose a deep-learning-based method with multi-channel distributed representation. Distributed representation indicates word vectors representing latent features of words. Multi-channel distributed representation enhances the representing ability of distributed representation. In experiments, some models are used for comparing each other, convolutional neural networks (CNN), long short-term memory are used in performance evaluation. The results showed that the classification performance of deep learning outperformed that of the Naive Bayes. Moreover, CNN with multi-channel distributed representation can classify tweets better than CNN without multi-channel distributed representation.**

Index Terms—**Deep learning, Distributed representation, Text classification, CNN.**

I. Introduction

WITH the growing interest in social media, people have begun sharing their life experiences by posting messages [1]. Twitter is one of the most attractive micro-blogging sites; moreover, users on Twitter tweet various real-world topics including tourist spots, local events, natural disasters, accident and news. Tweets posted on Twitter are one of the most important information sources [2], [3]. The use of such information has received considering attention in several application domains [4], [5], [6].

In our previous work [7], we proposed a topic analysis system to observe dynamics of real-world topic. In the system, tweets related to a monitoring topic are identified by a classifier to extract topic-related tweets. Tweets are classified using the classifier whether the content of a tweet is related to the monitoring topic or not. We implemented a natural disaster observation system, which can detect areas suffering from heavy rain or snow. The quality of information on the system depends on the classification performance. As the classifier is based on Naive Bayes [8], there is room for improving the classification performance.

In this paper, we propose a new deep-learning-based method to improve the classification performance on the topic analysis system. The proposed method is based on convolutional neural networks (CNN), where the input of the network is represented as distributed representation, or word embedding. Yoon et al. [9] proposed this model, and a series of distributed representations of words in a sentence was inputted in the model. The key technique is that the size of the convolution filter is set to the number of dimensions of the distributed representation to maintain word information. A text sentence is a time series sequence of words, and multi-filters are used to capture features of sentences.

The main contributions of this study are as follows.

- To enhance the ability of distributed representation, we propose multi-channel distributed representation. A text message in a tweet is a sequence of words referred to as a sequence of distributed representations. Yoon's method converts this sequence into a square matrix. In the proposed method, the sequence of distributed representations is mapped to multiple sequences on time delay and each sequence is a channel.
- To evaluate the proposed method, we compared it with several deep-learning-based classifier methods and Yoon's method.

The rest of the paper is organized as follows. Sections II and III provide an overview of related work and our previous work. Section IV provides an explanation of the proposed method. In Section V, the experimental results are reported. Finally, Section VI concludes we conclude the paper and presents our future work.

II. Related work

This section provides an overview of the related work on identifying tweets for topic and event extraction through machine learning, sentiment identification, and text data classification by using deep learning techniques. Social media is the most rapid growth of a new type of information source. In particular, people obtain information instantaneously about trending topics and events from tweets on Twitter, which is one of the most widely used micro-blogging services [5]. A large number of tweets include only inconsequential information; the extraction of tweets involving useful information is important for the use of tweets in many different domains.

A survey and comparative study of tweet sentiment analysis was reported by Silva et al. [10]. In tweet sentiment analysis based on machine learning, an emotion of a tweet is identified through a classifier. Davidov et al. [11] identified tweet polarity by using emoticons as class labels. Their method defines the feature vector of a tweet and classifies the k-nearest neighbor algorithm. Aramaki et al. [12] proposed a novel method for detecting influenza epidemics through

S.Hashida are with Faculty of Information Sciences, Hiroshima City University, 3-4-1, Ozuka-Higashi, Asa-Minami-Ku, Hiroshima 731-3194 Japan.

K.Tamura, and T.Sakai are with Graduate School of Information Sciences, Hiroshima City University, 3-4-1, Ozuka-Higashi, Asa-Minami-Ku, Hiroshima 731-3194 Japan, corresponding e-mail: (ktamura@hiroshima-cu.ac.jp).

tweets; their method utilized classifiers, such as support vector machine (SVM) and Naive Bayes to extract tweets that included topics about influenza. In our previous work [7], tweet classifier utilizing the Naive Bayes technique was used to classify tweets.

Classifiers embedding neural network techniques have been studied. These kinds of classifiers have gained attention again owing to the recent development of the deep learning technique. Yoon [9] proposed a deep-learning-based sentence classifier utilizing CNN. Yoon utilized deep learning with distributed representation to classify sentences. Severyn et al. [13] proposed a deep-learning-based tweet classifier using Yoon's model. There are several methods for expressing features of words in a sentence, including one-hot vector and distributed representation. In Yoon's model, distributed representation is used as the word vector and a sequence of word vectors are input to the network.

There are many methods for expressing features of words in a sentence through neural language models [14], [15]. In neural language processing, the deep model learns the word vector to decrease the dimension of word expression. On the contrary, in [16], microblog texts were mapped to low dimensional vector space through deep learning. In this study, we focused on feature expression methods for words in a sentence. To incorporate time delay information into the distributed representation of a word, the multi-channel distributed representation is proposed.

III. Topic Analysis System based on Density-based Spatiotemporal Clustering

This section explained the topic analysis system based on density-based spatiotemporal clustering [7]. Fig. 1 shows the overview of the topic analysis system with four main stages: tweet classifier, spatiotemporal clustering, photo image classifier, and web application. The process flow of the topic analysis system is as follows:

1) The topic analysis system crawls geo-tagged tweets from Twitter by using the Geo-tagged Tweet Crawler. Then, the geo-tagged tweets are stored in the Geo-tagged Tweet Database.
2) The tweet classifier, which is based on Naive Bayes, classifies the geo-tagged tweets into relevant and irrelevant geo-tagged tweets. The relevant geo-tagged tweets are related to a monitoring topic and are entered into the Spatiotemporal Clustering.
3) The spatiotemporal clustering utilizes (ϵ, τ)-density-based adaptive spatiotemporal clustering [17] to extract spatiotemporal clusters as areas related to the monitoring topic. The photo images attached to relevant geo-tagged tweets in the spatiotemporal clusters are entered into the photo image classifier.
4) The photo image classifier classifies photo images into relevant and irrelevant photo images and is based on a pre-trained deep network [18].
5) The web application visualizes the relevant geo-tagged tweets and photo images on the map.

IV. Proposed Method

A. CNN for Text Classifier

Our proposed method is inspired by Yoon's model, which is based on CNN [19]. CNN is originally proposed for image recognition but can be used for various kinds of data such as text data, time series numeric data, and multimodal data. CNN is a multiple-layer neural network that consists of convolutional layers, pooling layers, and fully connected layers as hidden layers. In a convolution layer, while sliding a filter, feature maps are extracted through element-wise multiplication. In general, a convolution layer has multiple filters and creates some feature maps. A pooling layer is a kind of down sampling layer. A fully connected layer is the same as a conventional multi-layer perceptron neural network. This layer usually represents the middle and high-level features of input data.

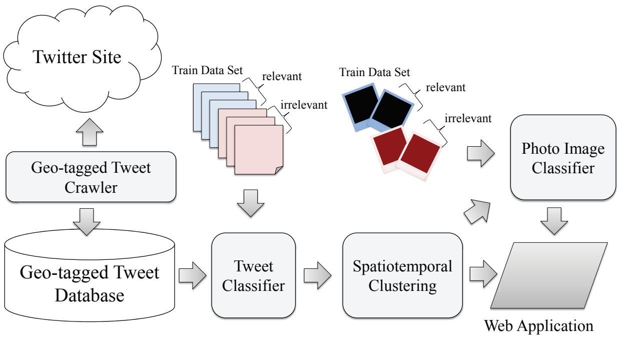

Fig. 1. The overview of the topic analysis system

The input data is usually an image data expressed by a three or two-dimensional matrix in CNN. In Yoon's model, text data is converted to a matrix, where the i-th data in a row represents the i-th word in the text data and the j-th column data represents the j-th item of distributed representation. Distributed representation is a word vector that represents the features of words in a multi-dimensional feature space (see in the next sub-section). In [9], word vectors were initialized with those obtained from an unsupervised neural language model. The authors used $word2vec$ vectors as initial vectors extracted from a learned model trained on 100 billion words from Google News [20].

In Yoon's model, the first layer is the input layer where multi-dimensional vectors are inputted. The next layer conducts convolutions over the multi-dimensional vectors by using multiple filter sizes. To maintain word information, the horizontal filter size is the same length as that as a word vector. The next layer is a max-pooling layer that max-pool the result of the convolution-layer into a one feature vector. The max-pooling layer connects to a fully-connected layer, which in turn connects to the soft-max output layer.

B. Distributed Representation

Distributed representation is a word embedding technique in natural language processing, where words are mapped to vectors of real numbers in the d-dimensional space. In natural language processing, words are usually represented as identification numbers with no meaning (e.g., “rain”→1 and “snow”→2). Distributed representation attempts to map words represented by identification numbers to d-dimensional vectors in a continuous vector space such that similar kinds of words are mapped to the same space (Fig. 2). The utilization of distributed representation enables the learning of features of text referred to as a sequence of words because similar words are represented as similar vectors.

Pre-trained and online-trained models are used to create distributed representation. In pre-trained models, distributed representation is extracted using a model trained by large-scale text data set. In online-trained models, distributed representation is extracted using embedded layers learned by user-given training data sets, while the whole neural network is learned by the user-given training data sets. In this study, the proposed method utilizes the online-trained model.

An embedding layer outputs the d-dimensional vectors of inputs represented by a sequence of identification numbers mapped to words. This layer has a transformation matrix $TF \in \mathcal{R}^{N \times d}$, where the total number of words is N and d is the dimension number of the distributed representations. The i-th row of the transformation matrix corresponds to the identification number i. $tf_{i,j}$ in TF shows the j-th dimensional value for the word mapping to identification number i. For example, let the identification number of word "rain" be 100. The word vector of "rain" is $TF_{100} = (tf_{100,1}, tf_{100,2}, \cdots, tf_{100,d})$. Fig. 2 shows an example of distributed representation, where each word is mapped to a vector in the d-dimensional space.

Fig. 3 illustrates a simple algorithm about the embedding layer. Text data in a tweet tw_i is a sequence of words $tw_i =< word_{i,1}, word_{i,2}, \cdots, word_{i,m} >$, where m is the length of word vector or distributed representation; moreover, it is a sequence of identification numbers, or a word identification array $IDS(tw_i) =< ids_{i,1}, ids_{i,2}, \cdots, ids_{i,m} >$. In the embedding layer, the word vector corresponding to each identification number is searched in TF. The embedding layer outputs a sequence of vectors $WE(TF, IDS(tw_i)) =< TF_{ids_{i,1}}, TF_{ids_{i,2}}, \cdots, TF_{ids_{i,m}} >$. The default value of $tf_{i,j}$ is set to a random value. During a training process, the value of $tf_{i,j}$ is revised iteratively through a training process.

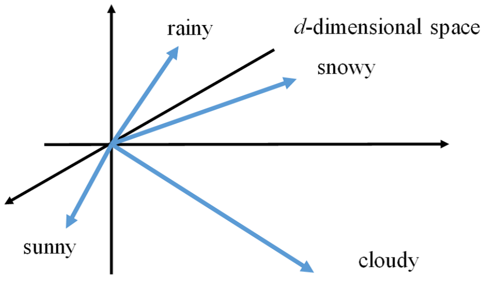

Fig. 2. Mapping to d-dimensional space

C. Multi-Channel Distributed Representation

The distributed representation of words can be extracted through an embedding layer in the proposed method. Distributed representation is a high-level representation representing the latent future of a word in a training data set; however, it cannot capture sentence structure. For example, suppose that there are two tweets : "It is rainy today" and "It will be rainy tomorrow." Even though the contexts of the words "rain" have different meaning, the word vectors of both "rain" are the same. To consider the context of a word, in this study, multi-channel distributed representation is proposed. In multi-channel distributed representation, each word has multiple word vectors, where each word vector is called a channel. In the proposed method, multiple sequences of identification numbers on time delay are generated. Moreover, the sequences of word vectors for these time delay sequences are obtained through embedded layers. Sequences of word vectors are piled up; therefore, each word has different word vectors. Even though the same word is present in various tweets, if the context of a tweet is different, the word is mapped in a multi-channeled word vector.

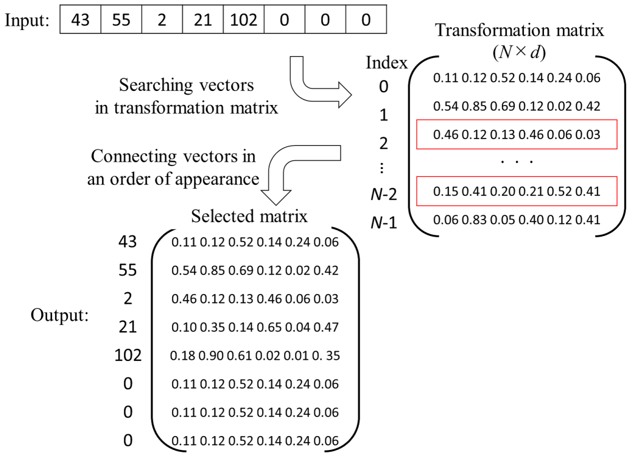

Fig. 3. Embedding Layer

Supposing that K represents the number of channels. Then, the processing of the generation of multi-channel distributed representation matrix is as follows.

1) A sequence of identification numbers is created from a tweet tw_i. Let a sequence of identification numbers be $IDS(tw_i) =< ids_{i,1}, ids_{i,2}, \cdots, ids_{i,m} >$.
2) $IDS(tw_i)$ is converted to a sequence of word vectors $WE(TF, IDS(tw_i)) =< TF_{ids_{i,1}}, TF_{ids_{i,2}}, \cdots, TF_{ids_{i,m}} >$.
3) For each channel, a sequence of word vectors $DWE(TF, IDS(tw_i)) =< TF_{ids_{i,2-K}}, TF_{ids_{i,3-K}}, \cdots, TF_{ids_{i,m,m-K+1}} >$ are extracted through the embedding layers, where if $ids_{i,j}$ and $j \leq 0$, $TF_{ids_{i,j}}$ is the zero-padding vector. Moreover, if $j > l$, where l is the length of words in tw_i, $TF_{ids_{i,j}}$ is also the zero-padding vector.
4) Each time delay sequence of word vectors is referred to as a two-dimensional matrix. A three-dimensional matrix is developed after piling up the K-extracted two-dimensional matrixes.

Fig. 4 illustrates a multi-channel distributed representation. In this example, the number of channels is three; therefore, four sequences of word vectors are extracted and a $7 \times 4 \times 3$ matrix is created. Through this process, a word being focused upon could be placed into some word vector appearing before it while including its distributed representation. Hence, this word is expressed with more detailed information.

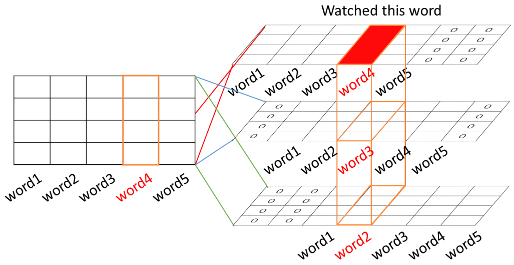

Fig. 4. Converting to multi-channel

D. Proposed Model

Fig. 5 shows the proposed model, the input of which is a integer value array numbered for each word in a string word. The dimension of distributed representation in an embedding layer is set up an integer value d. Next, this model is set in a layer that converts which convert into multi-channels from a matrix of distributed representation. This layer's outputs are given to a convolution layer and max pooling layer. Finally, this model's output is computed through a fully connected layer and activation function soft-max. Each layer is explained as follows. First, in converting to multi-channels layer, this output is overlaid for 3rd dimension direction. In the convolution layer, the filter size has height h, width d, and depth K. The filter's width was taken equal to the dimension of distributed representation to extract full information of a word.

V. Experiments

A. Experimental setups

In the experiments, we used tweets including keyword "rain" extracted from Japanese tweets on Twitter on June 4th, 2016. This data set includes 1458 tweets of "relevant" class and 1097 tweets of "irrelevant" class. The "relevant" class includes tweets related to a topic "rain." The "irrelevant" class includes tweets in this class are not related to topic "rain." In the experiments, MeCab was used as the morphological analysis. All words in tweets were used, implying that stop words were not removed. All words in the tweets used for training models were mapped to identification numbers.

The following models were evaluated in our experiments.

- Naive Bayes
  - This is our baseline and was used in our previous study.
- CNN with and without conversion converting to multi-channels layer
  - This is our proposed method. The height of filter is set to 3. In the experiments, we performed evaluation while changing the number of channels $K \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.
- CNN(Merge-3)
  - This is Yoon's method. In convolution layers, there are three sizes of filters, where heights of the filters are 3, 4, and 5.
- CNN(Merge-3) with conversion to multi-channels layer
  - This is Yoon's proposed method with multi-channel distributed representation. In the experiments, evaluation while changing the number of channels $K \in \{1, 3\}$ was experimented.
- Long Short-Term Memory[21]
  - The long short-term memory (LSTM) model is mainly utilized to classify and predict time series data and can learn long-term dependence.

In experiments, we implemented methods with Keras, which is one of the well-known deep-learning frameworks. The length of input was set to 80 words, and in the embedding layer, the dimension of distributed representation was 50. Thus, if the number of words in a tweet was more than 80 words, we only used 80 words from the head of the tweet. In addition, the number of filters in CNN layer and units in fully connected layer were set to 128. If the number of words in a tweet is less 80 words, zero-padding data was stored in the remainder. As this dataset label is binary, the number of output neural networks in this layer is 2. Furthermore, the cost function is a categorical cross-entropy, and Adadelta was used an optimizer.

B. Results

Table I shows the experimental results, presenting precision, recall, F-measure of "relevant" class, and accuracy. In addition, after learning each model on training data, each model was evaluated through cross-validation on test data, in which the number of partitions was 10. Deep models have more F-measure score than Naive Bayes model. In addition, CNN with conversion to multi-channels layer shows improvement in performance compared with CNN. However, CNN(Merge-3) with conversion to multi-channels does not show improvement on evaluations because multi-channel distributed representation can obtain information similar to that obtained which is gotten by multi convolutional filters in Merge-3 model.

Fig. 6 shows experimental results when changing the number of channels. The model with 1 channel works the same as the CNN model without multi-channel conversion. Regarding accuracy, a positive correlation is observed between the 2-channel and 6-channel models. However, this is not perfectly proportional because this multi-channel representation will be complicated when a number of channels that suit the training and test data are unavailable. In the future work, we must find the number of channels most suitable for data.

TABLE I
Evaluation models : Precision, Recall, F-measure for "relevant" class, and Accuracy

	$Precision$	$Recall$	$F-measure$	$Accuracy$
Naive Bayes	0.706	0.744	0.721	0.675
CNN	0.832	0.856	0.842	0.816
CNN (multi-channels)	0.832	0.868	0.849	0.822
CNN(Merge-3)	0.820	0.876	0.845	0.816
CNN (Merge-3, multi-channels)	0.815	0.873	0.841	0.811
LSTM	0.799	0.880	0.834	0.800

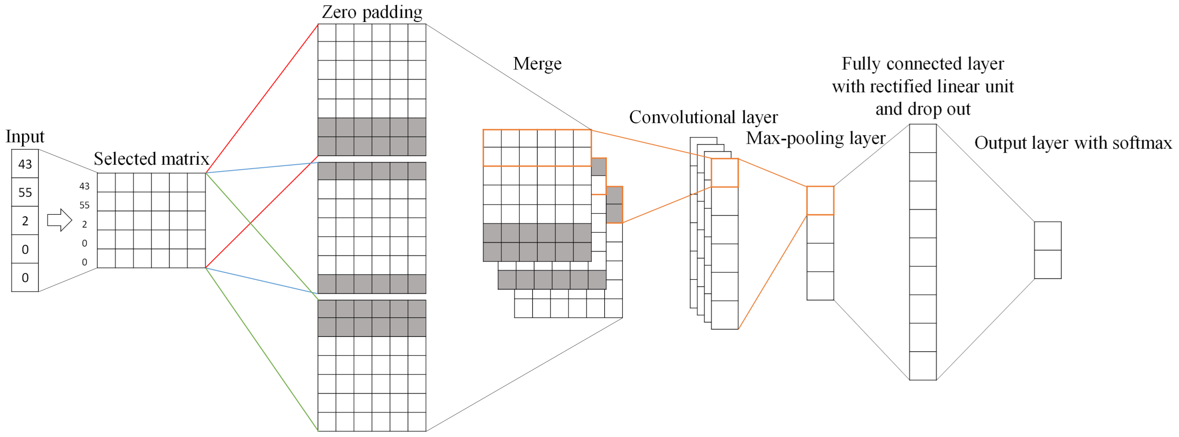

Fig. 5. Proposed model

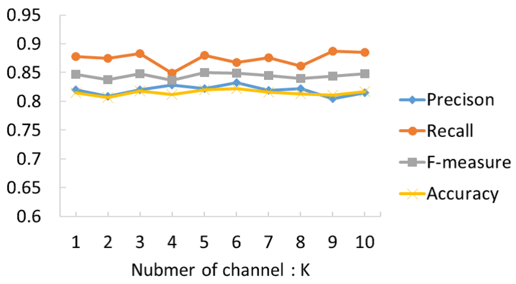

Fig. 6. Evaluation of each number of multi-channel

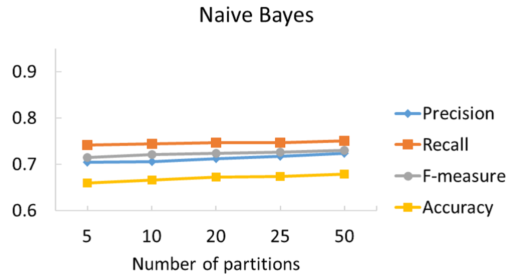

Fig. 7. Evaluation of each number of cross-validation (Naive Bayes)

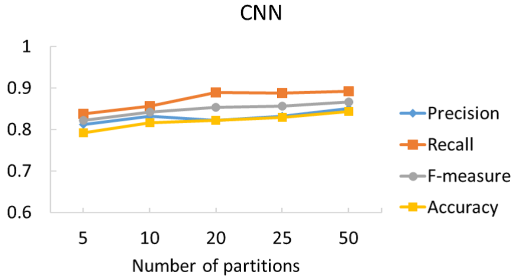

Fig. 8. Evaluation of each number of cross-validation (CNN)

Fig. 7-9 show experimental results on each number of partitions, that is, 5, 10, 20, 25, and 50 on Naive Bayes, CNN, and CNN with multi-channel, respectively.

The number of convolutional filters in CNN layer and units in the fully connected layer were set to 9 patterns, shown in Table II. In this experiment, the CNN model (with and without multi-channels) was utilized in 10-cross validations. Table III shows the best performance results. From this result, improvement of the evaluation value to multi-channel of the performance can expect when it has the number of big units in comparative way. The comparison of the model performance showed the best accuracy for the pattern (K, $Pattern$)=(3, 1) and the best F-measure for the pattern (K, $Pattern$)=(4, 8).

According to Fig. 7-9, the performance improvement of these models was proportional to the increasing number of partitions. In addition, these models learn word list in the training data but cannot learn word lists that do not appear in the training data. Considering the aforementioned factors, practice data is considered to be lacked.

VI. Conclusion

In this paper, we proposed a new deep-learning-based classifier to improve the classification performance for the tweet classifier in the topic analysis system. The main characteristic of the proposed method is the multi-channel distributed representation technique. The proposed method is based on the Yoon's model that utilizes CNN with distributed representation, which is a word embedding technique in which words are mapped to vectors in a multi-dimensional space. In Yoon's model, text data is converted to a matrix

TABLE II
Evaluations of the model parameters pattern 9 ways

Pattern n	$filters\ in\ CNN\ layer$	$units\ in\ full\ connected\ layer$
Pattern1	32	32
Pattern2	32	128
Pattern3	32	1024
Pattern4	128	32
Pattern5	128	128
Pattern6	128	1024
Pattern7	1024	32
Pattern8	1024	128
Pattern9	1024	1024

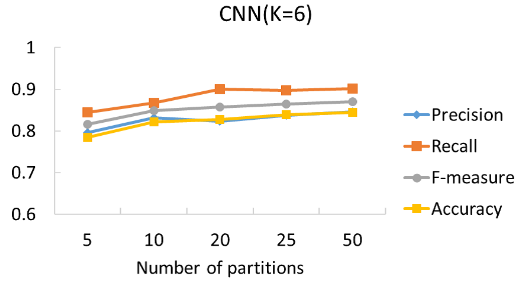

Fig. 9. Evaluation of each number of cross-validation (CNN, k=6)

TABLE III
EVALUATION MODELS(PATTERN : 9 WAYS): PRECISION, RECALL, F-MEASURE FOR "RELEVANT" CLASS, AND ACCURACY

K	Pattern n	$Precision$	$Recall$	$F-measure$	$Accuracy$
1	5	0.832	0.856	0.842	0.816
2	8	0.830	0.870	0.849	0.821
3	1	0.825	0.882	0.852	0.824
4	8	0.836	0.869	0.851	0.825
5	5	0.822	0.880	0.850	0.821
6	5	0.832	0.868	0.849	0.822
7	9	0.820	0.874	0.847	0.818
8	7	0.827	0.875	0.849	0.821
9	4	0.832	0.866	0.846	0.820
10	2	0.816	0.886	0.849	0.818

of distributed representation. To enhance the ability of distributed representation, multi-channel distributed representation combines multiple matrices of distributed representation, which are constructed based on time delay. To evaluate the proposed method, we compared the proposed method with several deep neural network models. The proposed method showed the best performance on classification of tweets. In our future work, we plan to enhance the representation of input data for further improvement of our method.

ACKNOWLEDGMENT

This work was supported by the MIC/SCOPE #162308002 and Hiroshima City University Grant for Special Academic Research.

REFERENCES

[1] A. Kavanaugh, E. A. Fox, S. Sheetz, S. Yang, L. T. Li, T. Whalen, D. Shoemaker, P. Natsev, and L. Xie, "Social media use by government: From the routine to the critical," in *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, ser. dg.o '11, 2011, pp. 121–130.

[2] D. Boyd, S. Golder, and G. Lotan, "Tweet, tweet, retweet: Conversational aspects of retweeting on twitter," in *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, ser. HICSS '10, 2010, pp. 1–10.

[3] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10, 2010, pp. 591–600.

[4] S. Asur and B. A. Huberman, "Predicting the Future with Social Media," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, ser. WI-IAT '10, 2010, pp. 492–499.

[5] A. Java, X. Song, T. Finin, and B. Tseng, "Why We Twitter: Understanding Microblogging Usage and Communities," in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, ser. WebKDD/SNA-KDD '07, 2007, pp. 56–65.

[6] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter Power: Tweets As Electronic Word of Mouth," *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 11, pp. 2169–2188, Nov. 2009.

[7] T. Sakai and K. Tamura, "Real-time analysis application for identifying bursty local areas related to emergency topics," *SpringerPlus*, vol. 4, no. 1, p. 162, Apr 2015. [Online]. Available: https://doi.org/10.1186/s40064-015-0817-x

[8] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[9] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, A. Moschitti, B. Pang, and W. Daelemans, Eds. ACL, 2014, pp. 1746–1751. [Online]. Available: http://aclweb.org/anthology/D/D14/D14-1181.pdf

[10] N. F. F. D. Silva, L. F. S. Coletta, and E. R. Hruschka, "A survey and comparative study of tweet sentiment analysis via semi-supervised learning," *ACM Comput. Surv.*, vol. 49, no. 1, pp. 15:1–15:26, Jun. 2016.

[11] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using twitter hashtags and smileys," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, ser. COLING '10, 2010, pp. 241–249.

[12] E. Aramaki, S. Maskawa, and M. Morita, "Twitter catches the flu: Detecting influenza epidemics using twitter," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11, 2011, pp. 1568–1576.

[13] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '15, 2015, pp. 959–962.

[14] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944966

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, vol. abs/1310.4546, 2013. [Online]. Available: http://arxiv.org/abs/1310.4546

[16] L. Xu, C. Jiang, Y. Ren, and H.-H. Chen, "Microblog dimensionality reduction - a deep learning approach." *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1779–1789, 2016. [Online]. Available: http://dblp.uni-trier.de/db/journals/tkde/tkde28.html

[17] T. Sakai, K. Tamura, and H. Kitakami, "Emergency situation awareness during natural disasters using density-based adaptive spatiotemporal clustering," in *Database Systems for Advanced Applications, DASFAA 2015 International Workshops, SeCoP, BDMS, and Posters, Hanoi, Vietnam, April 20-23, 2015*, vol. 9052, 2015, pp. 155–169.

[18] T. Sakai, K. Tamura, H. Kitakami, and T. Takezawa, "Photo image classification using pre-trained deep network for density-based spatiotemporal analysis system," in *Proceedings of 2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA)*, 2017, pp. 207–212.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[20] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13, 2013, pp. 3111–3119.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735