Short Message Classification using Tree Kernel and Random Forest with Feature Vectors based on Wikipedia Categories

Nobyuki Kobayashi[1], Masahiro Takeda[2], and Hiromitsu Shiina[3]

[1] Faculty of Human Sciences, Sanyo Gakuen University, koba_nob@sguc.ac.jp
[2] Graduate School of Informatics, Okayama University of Science
[3] Faculty of Informatics, Okayama University of Science, shiina@mis.ous.ac.jp

Abstract. The research to automatically classify the categories prepared for the Tweet is important from the perspective of making it easier to apply services and develop natural language processing. However, with general learning methods using labeled data for the construction of direct models, there is the concern that a massive cost will be attached to the creation of training data. Further, in the case of short message data such as tweets, the attributes obtained are limited, and the classification accuracy is greatly affected by minimal feature values. In this study, we propose a classified method by Tree kernel using Wikipedia category tree assigned to short message. In addition, we obtain the affiliation probability by Naive Bayes which learned Wikipedia category information. And add probability of affiliation due to influence from surrounding category. Finally, for each Tweet, generate a feature vector with the probability of belonging to the category as a vector.

Keywords: Twitter · Naive Bayes · Wikipedia · Tree kernel · Random Forest

1 Introduction

Currently, with Twitter[1], there are more than 500 million short messages (tweets) posted every day, and these are the source of a wide variety of information. This has resulted in a flourish of research aimed at text mining and user feature analysis etc. aimed at Twitter. In particular, research to automatically classify the categories prepared for the Tweet is important from the perspective of making it easier to apply services and develop natural language processing. As a reference example for methods using category classification, there are studies that classify categories of Tweets by topic using Naive Bayes[2, 3]. Naive Bayes works rapidly in terms of learning and identification and has high identification accuracy, so is widely used as a practical method of text classification. However, with general learning methods using labeled data for the construction of direct models, there is the concern that a massive cost will be attached to the creation of training data. Further, in the case of short message data such as tweets, the

attributes obtained are limited, and the classification accuracy is greatly affected by minimal feature values.

In this study, using Wikipedia[4] category structure, we propose two methods to categorize short message as Tweet. First method converts short messages into tree structure data. We attempt to achieve accurate text classification by calculating the similarity between the tree structure data. On the basis of generated tree structure data, we classify the short messages and measure the classification accuracy using an SVM tree kernel[5, 6].

Second method attempts to generate feature vectors of short messages using Naive Baiyes for which the Wikipedia category structure has been learned, and by learning the generated feature vectors using Random Forest[7], to extend the attributes and provide simple and accurate short message classification from a small volume of labeled data. Random forest has the merits of rapid learning and identification, and being resilient to noise, so is widely used in such fields as object recognition and character recognition. On the other hand, as it is necessary to converge learning while maintaining randomness, there are not many usage examples in natural language processing, which often uses space feature vectors. However, if the probability that the Tweet belongs to one of the respective Wikipedia categories can be expressed as a vector, it is considered that not only can identify information be extended, but close feature vectors can be generated, and sufficient training data can be obtained. Further, as a large number of weak learners based on decision trees using rich category information are generated using Random Forest, it is considered that a learning model can be constructed that is not impacted too much even if some of them extract wrong features. The affiliation rate of the categories is calculated using Naive Bayes, often used in the preceding research. Further, the values after the neighboring category probability mean value is added to the category under attention generate a feature vector as feature value. In the evaluation experiment, we compare the accuracy using SVM[8] etc. and make observations on the model.

2 Related Works

The method of learning known as distant supervision[9] is a method for automatically generating large quantities of labeled data from small quantities of labeled data. This method constructs a knowledge base using external information sources such as Wikipedia etc. and, in relation to the unlabeled corpus, automatically attaches labels to text that conforms to relationships held in the knowledge base. There has also been a flurry of research aimed at Tweets using distant supervision, and these include, for example, studies in with Tweets are classified by emotional expression, by enriching the label information [10]. There is also research related to Tweet category classification using the articles and category information of Wikipedia, and these have achieved highly accurate category classification from a small volume of labeled data[11]. As in this research, by making good use of external information sources such as Wikipedia, it is

considered possible to reduce the convergence cost of training data and secure accuracy by extending the identity information.

3 Tree Kernel Using Generating Feature Vectors for the Wikipedia Category Structure

We shall explain a method of extending the Tweet identity information using the Wikipedia category structure. In general terms, with Wikipedia, the each page (article) explained in relation to the main theme is allocated to one or more categories. Further, one category is linked to multiple highly related categories, and the category network structure is formed as a whole. By skillfully using the above characteristics, it is considered possible to extend the Tweet meaning information. In this study, as the Tweet to be classified is converted to an affiliation rate feature vector in relation to each Wikipedia category, the category affiliation rate is first calculated using Naive Bayes. With Naive Bayes, a model in which the Wikipedia categories and the text of the articles included in the categories is learned, is used as a classifier. Further, using the Wikipedia categories, a feature vector containing more appropriate feature values is generated.

3.1 Affiliation Rate Calculation Using Naive Bayes

Naive Bayes, in which the Wikipedia category information is learned, is defined as follows.

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \propto P(c)P(d|c) \ . \tag{1}$$

With Naive Bayes, when document d is provided, the posterior probability that category c can be obtained is calculated. Here, c is the Wikipedia category and d represents the articles included in the Wikipedia category. Posterior probability $P(c)$ is the ratio of the total number of documents comprised of documents in the various Wikipedia category c, and this is defined as follows.

$$P(c) = \frac{\text{Total number of documet in category } c}{\text{Total number of document in Wikipedia}} \ . \tag{2}$$

When likelihood $P(d|c)$ is applied to category c, this is the probability generated based on assuming document d as a proper noun set model. With document $d = (w_1, w_2, \ldots, w_{|n|})$, likelihood $P(d|c)$ is defined as follows.

$$P(d|c) = P(w_1, w_2, \ldots, w_k|c) = \prod_{i=1}^{k} P(w_i|c) \ . \tag{3}$$

$P(w_i|c)$ expresses the ratio of proper nouns w_i appearing in category c, and using the frequency of proper nouns w_i in category c, this is defined as follows. $N(c, w_i)$ is the total number of proper nouns appearing in category c.

$$P(w_i|c) = \frac{N(c_i, w_i)}{\sum_i N(c_i, w_i)} \quad . \tag{4}$$

The words within Tweet T are put into a vector, and with $T = (\ w_1, w_2, \ldots, w_{|n|})$, using learned Naive Bayes, the Tweet T affiliation rate for each Wikipedia category is calculated using the following definition.

$$P(c|T) = P(c)P(T|c) \quad . \tag{5}$$

Note that the Wikipedia articles learned through Naive Bayes are analyzed morphologically using MeCab[12, 13]. Here, only proper nouns are considered as features. In addition, the first MeCab dictionary could not be used to extract various proper nouns; therefore, we used T. Saito's Mecab-ipadic-neologd dictionary[14].

4 Weighted Tree Kernel

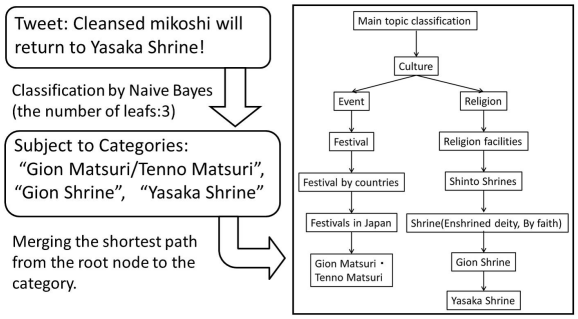


Fig. 1. Generation of tree structure data

4.1 Generating Tree Structure Data

To create the category tree structure, classification is performed using multiple categories. For example, when creating a tree structure with three leaf nodes, three categories are chosen in order of the highest posterior probability, and once the categories of the text are determined, the tree structure is created by integrating between the nodes using the shortest path for different categories.

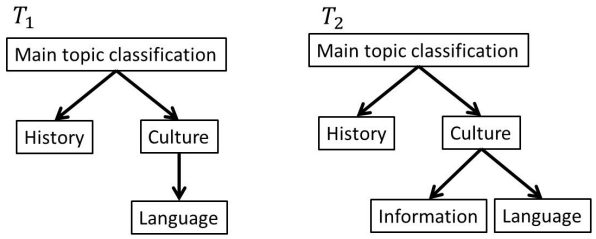
We calculate the similarity using a tree kernel for a text whose tree structure was determined using the Wikipedia categories. Then, the SVM method is employed for learning. We define the tree kernel $k(T_1, T_2)$ as follows:

$$K(T_1, T_2) = \sum_{p_1 \in Path(T_1)} \sum_{p_2 \in Path(T_2)} \gamma \cdot \text{matchpath}(p_1, p_2) \ . \tag{6}$$

$$\text{matchpath}(p_1, p_2) = \begin{cases} 1, & p_1 = p_2 \\ 0, & p_1 \neq p_2 \end{cases} \ . \tag{7}$$

Here, $Path(T)$ is a set of partial paths of a tree T, and $p_1 \in Path(T_1)$ and $p_2 \in Path(T_2)$ are partial paths that are contained in trees T_1 and T_2, respectively. Furthermore, γ is a weight parameter and matchpath returns 1 is p_1 and p_2 is the same path.

Figure 2 shows an example of a set of common partial paths for trees T_1 and T_2 that are gerated from Wikipedia category. For example, the two trees of the major categories "history", "culture" and "language" and major categories "history", "culture", "information", and "language", respectively, have six common partial paths. On the basis of the above features, this tree kernel creates all possible partial paths of tree T as feature vectors and calculates the inner product by the weight depending on the length of the partial paths. Figure 2 shows an example of a set of common particle paths consisting of six paths.



A set of common particle paths:
{Main topic classification}, {Culture}, {Language},
{Main topic classification {Culture}},
{Culture {Language}},
{Main topic classification {Culture{Language}}}

Fig. 2. Common particle paths

4.2 Weighted Tree by TF–IDF

To measure the similarity between normal trees, a partial match is important. However, even if there is a partial tree match, the similarity of partitional trees close to branches is more important.

In addition, when comparing the nodes belonging to abstract upper tier categories, such as history or culture, the nodes belonging to categories with concrete

meanings, such as “Gion festival” and “Yasaka Shrine” are considered to be more distinctive elements. Therefore, by attaching a weight reflect the importance of each node in the category tree; thus, it is possible to improve the classification accuracy using similarity measurements. With regard to such weighted trees, a text classification method proposed by Mikami et al. [15] employed weighted processing of the tree editing distance using term frequency-inverse document frequency (TF–IDF) [16]. In this study, the TF–IDF obtained using the appearance frequency between category nodes in the Wikipedia category structure is defined as follows:

$$\mathrm{TF-IDF}(v) = \mathrm{TF}(v) \cdot \mathrm{IDF}(v) \ . \tag{8}$$

$$\mathrm{TF}(v) = \frac{n_v}{\sum_{\forall k \in d, \forall d \in D} n_k} \ . \tag{9}$$

$$\mathrm{IDF}(v) = log \left\{ \frac{|D|}{|\{D : v \in d\}|} \right\} \ . \tag{10}$$

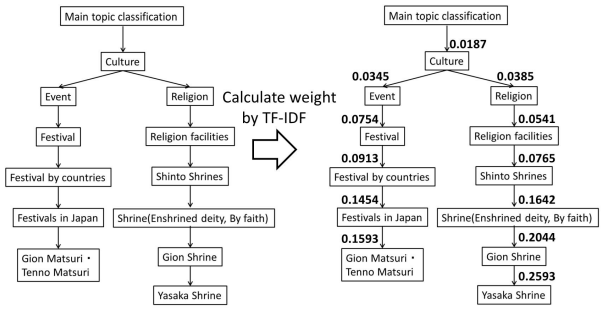


Fig. 3. Weighted Tree by TF–IDF

Here, n_v is the frequency at which node v appears in the category tree generated from the tweet text group, D is the tweet text group, and d is the unit tweet text included in the tweet text group D. TF (v) is the frequency of appearances of node n_v in each category divided by the total number of category nodes appearing in the tweet text group. The TF value increases with a higher appearance frequency of the node. In contrast, for IDF(v), the IDF decreases as more nodes appears in the tweet text, and this facts as a node filter that the IDF value for more representative nodes with low appearance frequency.

An example of a category tree in which weighted processing was performed is shown in Figure 3. As can be seen in Figure 3, category nodes with abstract

meaning have a small TF–IDF value and category nodes with more concrete meaning have a high TF–IDF value. For example, the score of nodes between "Main category" and "Culture" is 0.0187, whereas the score between "Festivals in Japan" and "Gion Matsuri · Tenno Matsuri" is 0.1593, which is considered to reflect the importance of each node.

4.3 Classification via SVM using Weighted Tree Kernel

The similarity is calculated with regard to the text, e.g., a tweet, by determining the tree structure from the Wikipedia categories using the weighted tree kernel, and this is learned using the SVM. The weighted tree kernel $K(T_1, T_2)$ is defined as follows:

$$K(T_1, T_2) = \sum_{T_{1p} \in PT(T_1)} \sum_{T_{2p} \in PT(T_2)} \sum_{v \in N(T_{1p})} \text{TF-IDF}(v) \cdot \text{matchtree}(T_{1p}, T_{2p}) \ . \tag{11}$$

$$\text{matchtree}(T_{1p}, T_{2p}) = \begin{cases} 1, & T_{1p} = T_{2p} \\ 0, & T_{1p} \neq T_{2p} \end{cases} . \tag{12}$$

Here, $PT(T)$ is the set of partial trees obtained from T, and matchtree(T_{1p}, T_{2p}) returns 0 or 1.

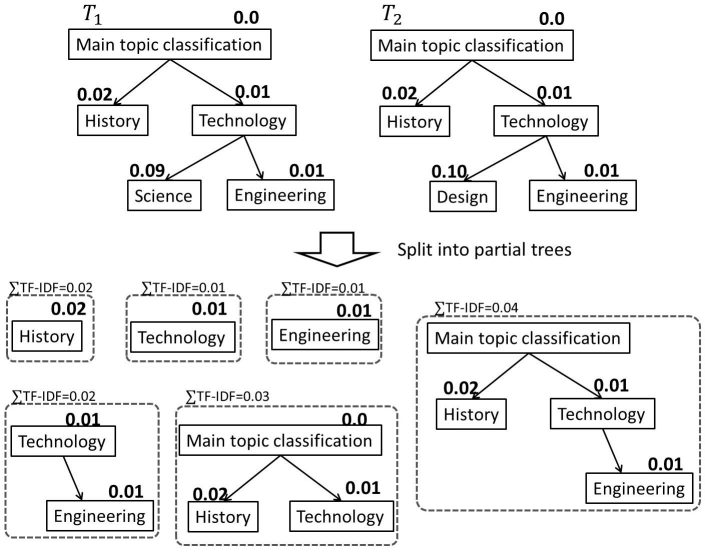


Fig. 4. Similarity of weighted trees

4.4 Calculating Similarity using Weighted Tree Kernel

We explain the process of calculating similarity using the weighted tree kernel method(Figure 4). In this study, the total weight of the nodes attached to the respective paths with regard to the common sections between T_1 and T_2 is calculated as the similarity between the trees. The weight of paths that do not have nodes between categories, such as the Main topic classification, is given the weight 0.

An example set of common partial tree obtained from trees T_1 and T_2 is shown in Figure 4, where T_1 ={Main category {History}{Technology{Science Engineering}}} and T_2 ={Main category {History}{Technology{Design Engineering}}}. Here, the similarity T_1 and T_2 is the sum of the six weighted trees, i.e., $K(T_1, T_2) = 0.02 + 0.01 + 0.01 + 0.02 + 0.03 + 0.04 = 0.13$.

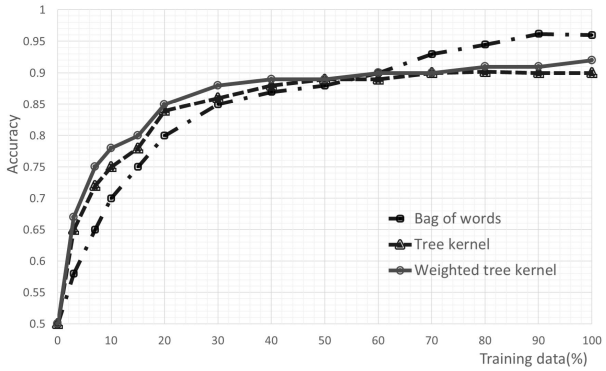


Fig. 5. Comparison of accurancy

4.5 Experimental result

Text classifications are rarely completed merely through the automatic classification of text. In this study, we evaluated the proposed method in terms of its application to a system that provides beneficial tourist information using Twitter as the target to automatically classify "Tourist information" and "Other" categories. Note that we considered the information recommendation task of using only tweets classified as "Tourist information." As a rating method, we conducted a 10-split cross-validation of 1200 tweets, manually attached with the correct labels. Note that the three leaf nodes for the tree structure data were used in this rating LIBSVM(v3.20)[17] developed by UCI [18] was used to implement the baseline SVM. Note that the SVM model used C-SVM.

For comparison, we show the results of the feature vector (Bag-of-Words), the tree kernel, and the weighted tree kernel (Figure 5). As can be seen, the feature vector accuracy is greater than that of the tree structure data on an average. However, the tree structure data demonstrates good accuracy relative to the small amount of data used for the learning. The text is expressed in terms of the tree structure category and it is possible to simply project a feature vector based on tree structure similarity; therefore, this can be categorized efficiently even with a small amount of data. In contrast, as the feature vector is weak in terms of unlearned words, accuracy will not stabilize unless a sufficient volume of learning data is used. The average accuracy of the tree structure data is not quite as good as that of the feature vector because of the noise between the frequency category in the upper tier concept of the tree structure data and the categories incorrectly classified by Naive Bayes. This is thought to have led to a decrease in the identification accuracy.

A comparison of the identification accuracy when calculating the frequency of the common section path in the tree kernel and when performing a calculation with a weight attached to the category tree node revealed that identification accuracy of the weight attached to the node was higher, because by attaching a weight to the node for the category tree, the features of the category tree in relation to the tweet are reflected. Furthermore, the accuracy was found to increase when the volume of learning data was low; thus, this model is suited for learning with small-scale data.

With the proposed method, even with only a small amount of text, it is possible to automatically generate training data that achieves highly accurate classification, which is effective in acquiring information about a large number of regions or facilities, or information about features for which it is difficult to prepare training data from words.

5 Random Forest using Extend Feature Vector by affiliation rate in Wikipedida Category

In order to generate training data, learned Tweets are collected from Tweet sets, and labels are attached for each class. Next, using Naive Bayes that has learned Wikipedia articles and categories, the posterior probability of the Tweet is calculated for each category in Wikipedia. Following this, the values obtained by adding the mean value for the probability of the subcategories in each category generates a feature vector as feature value, and this becomes training data. The generated training data is used to perform learning using Random Forest, and unlearned Tweet categories are estimated using the post-learning model.

5.1 Feature Value Calculation Using Category Links

Vectors based on the posterior probability sought using Naive Bayes are space vectors that co-occur with the words w_i that exist in certain categories c, and

there is the concern that learning cannot be converged well. Further, as the affiliation rate is only expressed as co-occurrence of words, it is insufficient for including Tweet relationships in relation to categories. Therefore, using the Wikipedia category structure, we calculate the feature value of the posterior probability mean values for the subcategories in each category (Figure 6). As the subcategory set $C(c)$ linked to category c, feature value $f(c, T)$ for category c in relation to Tweet T is defined as follows.

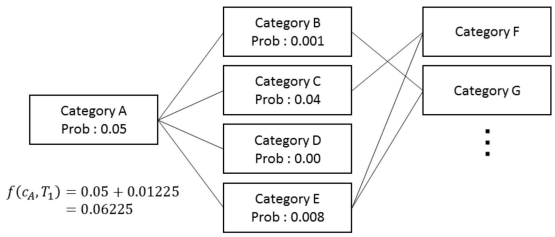


Fig. 6. A probability of affiliation due to influence from surrounding category

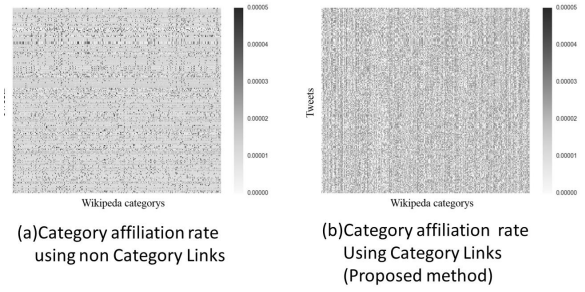


Fig. 7. Heat map of category affiliation rate

$$f(c, T) = P(c|T) + \frac{1}{|\boldsymbol{C(c)}|} \sum_{c_i \in \boldsymbol{C(c)}} P(c_i|T) \ . \tag{13}$$

As a concrete example of changes in feature value, the heat map expressing the feature value for which the Wikipedia category affiliation rate and subcategories probability mean value calculated using Naive Bayes were added, in relation to Tweet sets with the subject “amusement park” is shown in Figures 3 and

4. The affiliation rate in the heat map (figure 7(a)) is handled independently for the category, so the relationship between category links is not considered. Therefore, we can see that the concentration of certain characters in which keywords appear in the Tweet is significantly high, and the category affiliation rate for most of the other categories is made up of uniform or thin vectors. On the other hand, in the heat map (figure 7(b)), we can see that as this is the feature value after adding the probability mean values of the subcategories, a close feature vector is generated. In particular, a striped pattern appears on the vertical axis expressing the various categories of Wikipedia, and this is considered to express the features related to the Tweet categories more significantly.

5.2 Method of Classification Using Random Forest

In this study, the feature vector data generated by calculating the category affiliation rate generates multiple subsets based on bootstrap sampling, and various decision trees are created for each subset. As the predicted values of each decision tree differ, for analysis problems, the histograms for each leaf are collected, and, by obtaining the mean values, the final predicted value is obtained. The learning processing procedure is shown below.
Step 1: B type subset is generated from data sets S of the generated vector data using the bootstrap sampling method.
Step 2: One subset D is extracted from within the B type subset.
Step 3: Decision tree T_d is generated from D subsets, and the following step is repeated until decision tree T_d reaches the end node or the layer of the specified height.
(Step 3-1) With subset D, acquisition of information is evaluated, and the optimal separation point is selected.
(Step 3-2) The node is partitioned into two child nodes.
Step 4: If the learning of the B type node is incomplete, return to Step 2.
Step 5: Output of decision tree set T_B : Using the obtained decision tree set T_B, identify classes based on the following likelihood. Here, the equation (7) means returning the T_B probability mean value when unlearned Tweet T is applied to the respective decision trees. Further, the equation (8) outputs the maximum value for Tweet T attribute value.

$$P_{\text{ave}} = \frac{1}{B}\sum_{b=1}^{B} P_b(c|T) \ . \tag{14}$$

$$C_t = \underset{c_i}{\text{argmax}}(P_{\text{ave}}(c_i|T)) \ . \tag{15}$$

5.3 Experimental Results

We shall verify the classification accuracy of the proposed method. We obtained 500 cases respectively of verification data from the various categories of “IT”,

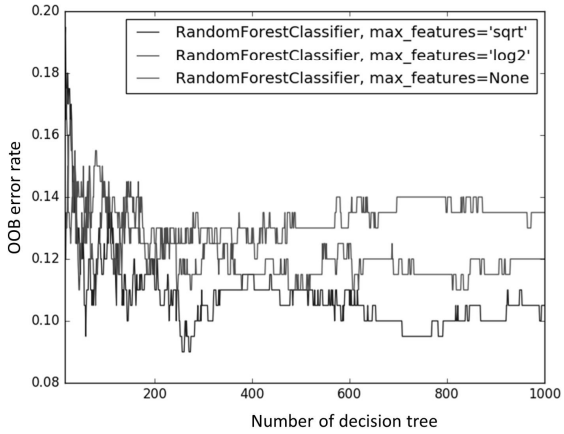


Fig. 8. Comparison of accuracy Random Forest

"home appliances", "movies", "sport", and "news", and used labeled Tweets. As a method of creating the labeled data, we created this by comprehensively collecting items related to the various categories from the hash tags applied to the Tweets, and excluding Tweets posted with unrelated content as a category, such as simple advertising and inducements to other sites.

Firstly, we verified the Forest Random out-of-bag(OOB) error rate in the proposed method. Figure 8 is the transition in the OOB error rate when generating decision trees from 15 to 1000. The three graphs each show the maximum feature value $\sqrt{B}$, $\log B$ using the respective decision trees.

In figure 8, we can see that the OOB stabilizes from about where the decision tree exceeds 300, and the learning convergences. Further, as a parameters t lower the OOB, the maximum feature value is $\sqrt{B}$, and we can see that the generated decision tree should be set to approximately 300. Next, we performed an accuracy evaluation using cross-validation. Here, in addition to the accuracy of the proposed method, Figure 9 shows the accuracy of SVM using the linear kernel, Random Forest as a feature vector for only Naive Bayes posterior probability, and the accuracy of Random Forest using a word frequency vector for comparison As the final classification accuracy from the learning curve in figure 9, Random Forest using the proposed method has higher accuracy than any of the other learning methods. In particular, when using word vectors, compared to the proposed method, the accuracy is not stable. This is because as the word vector is a direct model constructed from the converged training data, it was not possible to obtain a sufficient data quantity for learning convergence. On the other hand, from the small number of labeled data in the proposed method, a close feature vector could be learned using the Wikipedia category structure, and it was confirmed that a classifier with a high level of accuracy could be

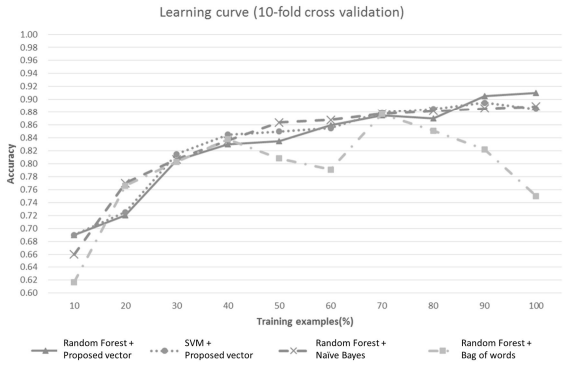


Fig. 9. Comparison of accuracy

obtained. In summary:
(1) The method using the random forest and the proposed vector has a large correlation with the learning amount.
(2) The method using the SVM and the proposed vector has the same tendency as the method using the random forest and the proposed vector. In both cases, the accuracy tends to be low when the amount of learning is low.
(3) When fixed to random forest, it is best to use naive Bayes. It is bad to use BOW(Bag-of-words). In particular, accuracy is not stable. Since the calculation amount is small using the proposed vector, it is an effective, the case of considering learning time.

6 Conclusion

In this study, we performed classification of Tweets using Tree Kernel and Random Forest over a feature vector calculated by considering the affiliation rate of Wikipedia categories up to the sub-category level. The proposed method could achieve higher accuracy than comparative experiments. Moving forward, we would like to develop its practical aspects, through the construction of a classifier that can be applied to a wider variety of short message data.

References

1. *Twitter*, [Online]. Available: https://twitter.com/
2. I. Rish, "An empirical study of the naive Bayes classifier," Proc. IJCA 2001 workshop on empirical methods in artificial intelligence, vol. 3, no. 22, pp. 41–46, 2001.
3. K. Lee, D. Palsetia, R. Nara-yanan, M.M.A. Patwary, A. Agrawal, A. Choudhary, "Twitter Trending Topic Classification," ICDMW '11, Proc. of the 2011 IEEE 11th International Conference on Data Mining Workshops, pp.251–258, 2011.

4. *Wikipedia*, [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia/
5. J. S. Taylor and N. Cristianini, *Kernel methods for pattern analysis*, New York: Cambridge Univ. Press, 2004.
6. D. Kimura, T.Kuboyama, T.Shibuya, H.Kashima, D. Kimura *et al.*, "A Subpath Kernel for Rooted Unorderd Trees," Trans. Japanese Soc. for Artificial Intell., vol. 26, no. 3, pp. 473–482, Apr. 2011. (In Japanese)
7. L. Breiman, "Random Forests," J. Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
8. V. N. Vapnik, *Statistical Learning Theory*, New York: John Wiley and Sons, 1998.
9. M. Mintz, S. Bills, R. Snow, D. Jurafsky "Distant supervision for relation extraction without labeled data," ACL2009, Vol.2, pp.1003–1011, 2009.
10. A. Go, R. Bhayani, L. Huang "Twitter sentiment classification using distant supervision," CS224N Project Report: Stanford Vol.1, pp.12–18, 2009.
11. M. Shirakawa, K. Nakayama, T. Hara, S. Nishio, "Wikipedia-Based Semantic Similarity Measurements for Noisy Short Texts Using Extended Naive Bayes," IEEE Transactions on Emerging Topics in Computing: Vol.3, pp.205–219, 2015.
12. T. Kudo, "MeCab: Yet Another Part-of-Speech and Morphological Analyzer," [Online]. Available: http://taku910.github.io/mecab/
13. T.Kudo *et al.*, "Applying Conditional Random Fields to Japanese Morphological Analysis," Proc. 2004 Conf. Empirical Methods in Natural Language Process. (EMNLP-2004), Barcelona, Spain, 2004, pp. 230–237.
14. T. Sato, *mecab–ipadic–neologd: Neologism dictionary for MeCab*, [Online]. Available: http://github.com/neologd/mecab–ipadic–neologd
15. M. Mikami *et al.*, "Calculating similarity between sentences using tree edit distance," IPSJ SIG Tech. Rep., vol. 2010-NL-196, no. 3, pp. 1–6, 2010.
16. K. S. Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," J. Documentation, vol. 28, pp. 11–21, 1972.
17. C. Chang and C. Lin, *Libsvm – a library for support vector machines*, [Online]. Available: https://www.csie.ntu.edu.tw/˜cjlin/libsvm
18. *UCI Machine Learning Repository*, [Online]. Available: https://archive.ics.uci.edu/ml/