TELKOMNIKA, Vol.12, No.1, March 2014, pp. 189 ~ 200
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v12i1.1798

Two Text Classifiers in Online Discussion: Support Vector Machine vs Back-Propagation Neural Network

Erlin*[1], Rahmiati[2], Unang Rio[1]
[1] Department of Information Technology, STMIK-AMIK Riau, Pekanbaru, 28294, Riau
[2] Department of Informatics Management, STMIK-AMIK Riau, Pekanbaru, 28294, Riau
*Corresponding author, email: erlin@stmik-amik-riau.ac.id

Abstract

The purpose of this research is to compare the performance of two text classifiers; support vector machine (SVM) and back-propagation neural network (BPNN) within categorize messages from an online discussion. SVM has been recognized as one of the best algorithm for text categorization. BPNN is also a popular categorization method that can handle linear and non linear problems and can achieve good result. However, using SVM and BPNN in online discussion is rare. In this research, several SVM data are trained in multi-class categorization to classify the same set with BPNN. The effectiveness of these two text classifiers are measured and then statistically compared based on error rate, precision, recall and F-measure. The experimental result shows that for text message categorization in online discussion, the performances of SVM outperform BPNN in term of error rate and precision; and falls behind BPNN in term of recall and F-measure.

Keywords*: text categorization, support vector machine, back-propagation neural network*

1. Introduction

Text message categorization is used to assign messages into a certain number of pre-defined categories based on their content. In e-learning environment, especially in online discussion, this interest is one particular aspect of analyzing a message is posted by participant, in which messages should be filed according to content. As the volume of message in online discussion increases and categorize the message into some classes is important, hence, method and technique to automatically categorize the message is needed.

Online discussion presents major challenges to the existing text categorization technique. Online discussion messages are usually incomplete, error-prone, and poorly structured [1]. In online discussion, text categorization that used to classify the message send by student into certain category is often manual, requiring skilled specialists. However, human categorization is not effective way for number of reasons; time-consuming, labour intensive, lack of consistency in category and costly. Therefore, how to use a various computer technologies to auto-coding of message is subject of great research value.

An increasing number of researchers have attempted to produce various machines learning for automating text categorization and classification. For examples, there are Bayesian network classifier [2], decision tree [3], neural network [4], support vector machine [5], fuzzy *k*-means [6] and maximum entropy models [7]. Gabrilovich and Markovitch [8] have been proved that support vector machine (SVM) is one of the best algorithms for text categorization. Meanwhile, Yu, et al., [9] have argued that neural network (NN) also a popular categorization method that can handle linear and non linier problems for text categorization, and both of linear and non linear neural network classifier can achieve good result. Unfortunately, the use of text classifier in educational setting is rare. Most of text classifiers are used to categorize news article, emails, product reviews and web pages.

This research aims to compare the performance of two popular text classifiers in text categorization; SVM and BPNN. The research has two major goals. First, which method is better or they are equally good for categorizing the message in online discussion? Second, to describe the procedure was used for text message categorization by using SVM and BPNN.

Received October 13, 2013; Revised December 20, 2013; Accepted January 13, 2014

2. Description of The Methods

2.1. Support Vector Machine

SVM is one of relatively new method compared with other methods, but has better performance in various application fields such as image processing, handwriting and text classification. Joachims successfully applied SVM to text categorization and achieved an outstanding improvement over other method [5]. He has argued that SVM is appropriate for text categorization because SVM can handle high dimensional feature spaces and few relevant features, which are main properties of text categorization.

The simply concept of SVM can be explained as attempt to find the best hyperplane (h) which is serves as the dividing two classes in the input space. Figure 1 shows the SVM pattern of linearly separable data that has a categorical target variable with two classes. The data is a member of two classes: +1 and -1. Pattern that joined in class -1 symbolized by the green color (boxes), whereas pattern in the class +1, symbolized by the yellow color (circle).

The cases with one category are in the lower left corner and the cases with the other category are in the upper right corner; the cases are completely separated. The SVM analysis attempts to find hyperplane (i.e. a line) that separates the cases based on their target categories. The dashed lines drawn parallel to the separating line mark the distance between the dividing line and the closest vectors to the line. The distance between the dashed lines is called the margin. The vectors (points) that constrain the width of the margin are the support vectors.

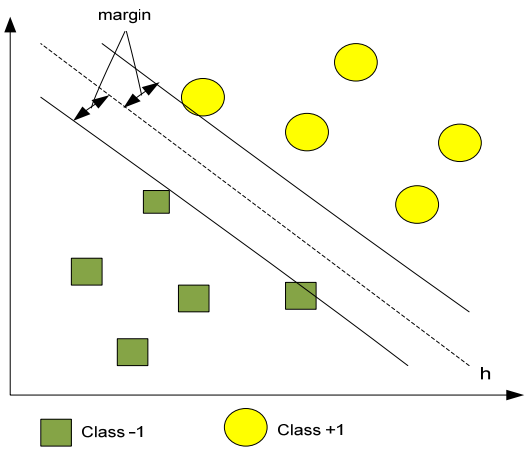

Figure 1. SVM pattern of linearly separable data

2.2. Backpropagation Neural Network

Back propagation (or feed-back) networks are called "recurrent" contains feedback connections. Back propagation networks are very powerful and can get extremely complicated [10]. Recurrent networks recalculate previous outputs back to inputs hence; output is determined both by their current input and their previous outputs. It can have signals travelling in both directions by introducing loops in the network. Weight adjustments are made to reduce error. For this reason, backpropagation neural network can be regarded very similar to short term memory in humans in that the state of the network outputs depends: upon their previous input.

Furthermore, backpropagation neural network (BPNN) is known for their ability to generalize well on a wide variety of problems. It also suitable for almost all problems if use enough hidden neurons. BPNN is a supervised type of network that is a method of training a neural network by presenting it with the correct answers during training, e.g., trained with both inputs and outputs. BPNN are used for the vast majority of working neural network applications because they tend to generalize well. It consists of a large number of simple processing units which are often referred to as neurons. The neurons are arranged in a number of layers called multi layer neural network. The three layers; input layer, hidden layer and output layer of BPNN

is shown in Figure 2. The training of a network by back-propagation involves three stages: the feed forward of the input training pattern, the calculation and back-propagation of the associated error, and the adjustment of the weight and the biases [11].

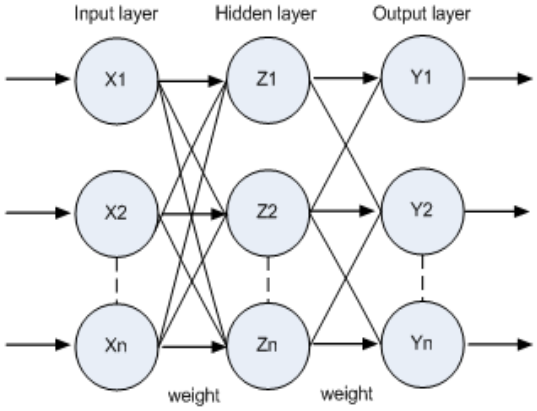

Figure 2. Three layers BPNN

3. Methodology

In this section, this research discusses how the text categorization method would be classified the message into a certain number of category. In this research, each document represents a message in online discussion. The collaborative learning skill category based on Soller's model [12] which is modified version of McManus and Aiken's Collaborative Skills Network [13] is adopted. In Soller's model each conversation act is assigned a sentence opener indicating the act's intention. Students communicated through a sentence opener interface by initiating each contribution with one of the key phrases, which conveys the appropriate dialogue intention.

The weakness of sentence opener approach is limitations in using of ideas or thinking that will be delivered in a discussion. Each student communicated by sentence opener first before posting message on the online discussion. Therefore, this research is designed to automated text categorization, hence student feel free to deliver their idea without being limited by the sentence opener that has been set previously by system. We take eight sub-skill categories; request, inform, motivate, task, maintenance, acknowledge, discuss and mediate; to be implemented and tested using SVM and BPNN.

Message of online discussion forum from one subject SCK3433-02 2008/2009: Management of Organisation Information Systems held on Moodle as learning management systems (LMS) in e-learning was examined for one discussion topic. There are 29 students completed the topic of discussion. The total numbers of messages in the online discussion (that are replies to somebody's message) are 394 messages.

The problem of text categorization may be formalized as the task to approximate an unknown classification function $\Phi : d \times c \rightarrow$ Boolean defined as:

$$\phi(d,c) = \begin{cases} true, if d \in c \\ false, otherwise \end{cases} \quad (1)$$

Where *d* is a set of document, *c* is set of categories, for any pair (*d, c*) of document and category [14]. Text categorization is defined as the process of assigning a Boolean value (true or false) to each pair of a document and its category [15]. In this research, a message that belongs to the specified category is given the Boolean value "true" otherwise, the given to the messages is "false".

In online discussion, student interaction as means of sharing knowledge and solving a problem by posting their idea or solution in the text form. All text of this forum as known as "corpus data" will be categorized into eight categories using SVM and BPNN approach. SVM

and BPNN cannot be directly interpreted the text. Text or message should first be transformed into a representation suitable for the classification algorithms to be applied.

In order to transform a message into a feature vector that suitable for text classifier, pre-processing is needed. This stage consists of identifying feature by feature extraction and feature weighting. The main goal of feature extraction is to transform a message from text format into a list of words as feature set, easier to be processed by SVM and BPNN algorithms. This step includes tokenization, stop word removal and stemming.

Tokenization is used to separate text into individual words. All upper case characters in the words are converted to lower case characters. Next, stop word removal to remove common words that are usually not useful for text categorization and ignored for later processing such as “are”, “is”, “the”, “a”, etc, based on a stop list for general English text. The remaining words then stemmed using the Porter’s algorithm [16] to normalize words derived from the same root such as “computer”, “computation”, “computing” would end up into the common form “comput”. The result is a list of words based on each message.

Furthermore, we merged the sets of word’s stems from each of the 275 training messages and removed the duplicates. As a result are 1137 terms in the vocabulary is potential as feature set. This research reduced the size of dimension by computing the document frequency (*DF*). DF is possible to reduce the dimensionality by a factor of 10 with no loss in effectiveness [17]. This seems to indicate that the terms occurring most frequently in the corpus are the most valuable for text categorization. Each message of text is classified in the same category are presumed to have similar meaning and belonging to one of the chosen category.

4. Experiment

Multi-class text categorization is designed to test the performance of SVM and BPNN. Total of 394 messages are split into two parts. First part for training data, we used of 275 messages as training set. Second part for testing data, we used of 119 messages as testing set. Regarding the unbalanced data distribution as shown in table 1, the problem is solved by assigning same proportion on the number of training data and the number of testing data in each class.

Table 1. Clustering data set

Class of Categories	# of Training	# of Testing	Total	Percentage
Acknowledge	40	15	55	13.96
Discuss	50	25	75	19.04
Inform	60	25	85	21.57
Maintenance	29	12	41	10.41
Mediate	2	1	3	0.76
Motivate	20	8	28	7.11
Request	44	20	64	16.24
Task	30	13	43	10.91
Total	275	119	394	100.00

Text classifier must be trained before it can be used for text categorization. In order to train the text classifier, a set of training messages and a specification of the pre-defined categories the messages belong to are required. We first need to transform a text into a feature vector representation. Hence the feature extraction is needed. We created a program to combine the three phases of feature extraction in C++ language based on Porter’s algorithm. The result can be seen in Figure 3 that shows a part of list of word after stemming process based on each message.

Furthermore, we merged and sorted the sets of word’s stems from each of the 275 training messages and removed the duplicates. As a result are 1137 terms in the vocabulary is potential as feature set. Figure 4 shows a part of list of potential words as a feature set from the whole message.

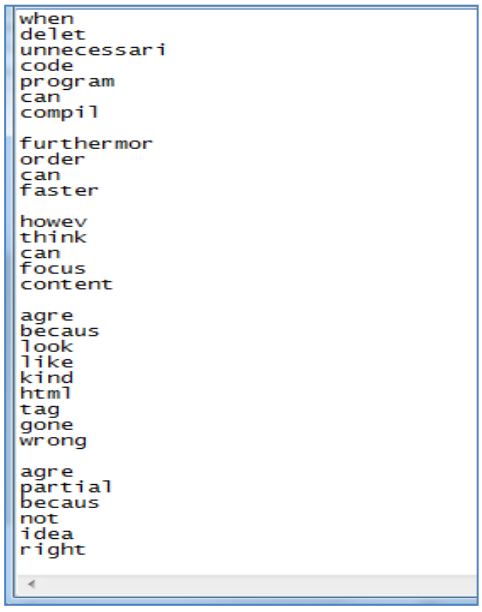

Figure 3. The part of word after stemming

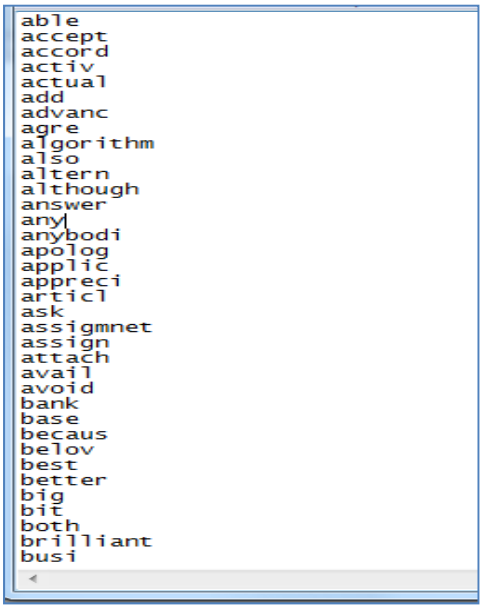

Figure 4. A part list of potential words as feature set

After feature extraction phase that select the important terms, we should do term weighting of each word. There are various term weighting approaches studied in the literature. Boolean weighting is one of the most commonly used. In boolean weighting, the weight of a term is considered to be 1 if the term appears in the message and it is considered to be 0 if the term does not appear in the message.

The size of dimension is reduced by document frequency (*DF*) method. All potential features are ranked for each category based on the term occurs in the message. The top features for each category are chosen as its feature set. We choose 370 term as text classifier's input.

Next step, text representation to transform the text into a representation suitable for text classifiers or categorization algorithms to applied. For each training and testing messages, we created the space vectors corresponding to the 275 training messages, where each space vector had a dimensionality of 370.

Training of text classifier based on supervised learning is done by using SVM^{multiclass} [18] and MATLAB Neural ToolBox [19]. The performance of SVM is very sensitive on the selection of the kernel parameters. The string *c* is a learning option to determine trade-off training error and margin, and string *t* is a kernel options to store with the vector. In SVM^{multiclass}, the value of *t* range from 0 to 4. This experiment has done using five values of *t* (0,1,2,3,4) that combines with the value of *c* range from 0.10 – 1.0. Hence, these experiments have been completed with all possible value of *t* paired with all possible value of *c*. Total six experiments was conducted. Table 2 shows the performance of SVM based on setting of learning option and kernel parameter.

From the table 2 above it can be seen that the best performance when the value of *t* is 1 and *c* is 0.80 where the accuracy reaches 79.8 with time processing of 1.06 seconds. Hence, optimal parameter setting used 0.8 as value parameter to *c* and 1 as value parameter to *t*.

In the experiment using BPNN, three layers of back-propagation neural network consist of an input layer, a hidden layer and an output layer with the sigmoid function as the activation function was used. In the input layer, the number of input node is equal to the number of feature set after dimensionality reduction. The number of hidden node depends on the number of input node and output node. In the output layer, the number of output node is equal to the number of pre-defined categories of text message categorization.

Table 2. Performance measure of SVM using kernel parameter

Experiment	t	c	correct	incorrect	accuracy	time(s)
1	0	0.10	81	38	68.07	0.00
	0	0.20	81	38	68.07	0.00
	0	0.30	80	39	67.23	0.01
	0	0.40	81	38	68.07	0.00
	0	0.50	80	39	67.23	0.00
	0	0.60	81	38	68.07	0.00
	0	0.70	80	39	67.23	0.00
	0	0.80	81	38	68.07	0.02
	0	0.90	84	35	70.59	0.00
	0	1.00	82	37	68.91	0.00
2	1	0.10	74	45	62.19	0.38
	1	0.20	77	42	64.71	0.58
	1	0.30	88	31	73.95	0.76
	1	0.40	88	31	73.95	0.75
	1	0.50	91	28	76.47	0.95
	1	0.60	91	28	76.47	0.95
	1	0.70	91	28	76.47	1.14
	1	**0.80**	**95**	**24**	**79.83**	**1.06**
	1	0.90	90	29	75.63	1.14
	1	1.00	91	28	76.47	1.23
3	2	0.10	49	70	41.18	0.14
	2	0.20	49	70	41.18	0.19
	2	0.30	49	70	41.18	0.20
	2	0.40	49	70	41.18	0.19
	2	0.50	49	70	41.18	0.19
	2	0.60	49	70	41.18	0.19
	2	0.70	49	70	41.18	0.19
	2	0.80	49	70	41.18	0.17
	2	0.90	49	70	41.18	0.19
	2	1.00	49	70	41.18	0.19
4	3	0.10	25	94	21.01	0.22
	3	0.20	26	93	21.85	0.41
	3	0.30	26	93	21.85	0.41
	3	0.40	26	93	21.85	0.42
	3	0.50	12	107	10.08	0.62
	3	0.60	12	107	10.08	0.59
	3	0.70	9	110	7.56	0.80
	3	0.80	9	110	7.56	0.84
	3	0.90	9	110	7.56	1.06
	3	1.00	20	99	16.81	0.80
5	4	0.10	25	94	21.01	0.03
	4	0.20	25	94	21.01	0.03
	4	0.30	25	94	21.01	0.03
	4	0.40	12	107	10.08	0.05
	4	0.50	8	111	6.72	0.06
	4	0.60	8	111	6.72	0.05
	4	0.70	20	99	16.81	0.08
	4	0.80	20	99	16.81	0.08
	4	0.90	8	111	6.72	0.06
	4	1.00	8	111	6.72	0.06
6	1	0.75	95	24	79.83	1.29
	1	0.76	95	24	79.83	1.39
	1	0.77	91	28	76.47	1.40
	1	0.78	91	28	76.47	1.38
	1	0.79	95	24	79.83	1.39
	1	**0.80**	**95**	**24**	**79.83**	**1.06**
	1	0.81	90	29	75.63	1.60
	1	0.82	91	28	76.47	1.60
	1	0.83	95	24	79.83	1.10
	1	0.84	91	28	76.47	1.39
	1	0.85	93	26	78.15	1.56

The network will be saved every time when the error factor reaches a new minimum average error for training data and testing data. This experiment will stop the training, if average

error below a predefined level, learning epochs exceeds a predefined number and largest error below a predefined level.

In hidden layer, the number of hidden node affects the generalization performance. Hidden nodes 10, 20, 30 and 40 were tested to find the most appropriate size of hidden node. Each of hidden nodes was tested 20 iterations. While Table 3 shows the best performance, Table 4 shows the average performance of each hidden node.

Table 3. Best performance for four hidden node

#Hidden Nodes	Best Performance			
	Accuracy	*Epoch*	*Time(s)*	*Error*
10	76.80	192	5	0.00991
20	**79.00**	**204**	**7**	**0.00977**
30	78.80	255	9	0.00998
40	78.20	291	12	0.01000

Table 4. Average performance

#Hidden Nodes	Average Performance (20 iteration)			
	Accuracy	*Epoch*	*Time(s)*	*Error*
10	66.90	413.60	10.55	0.01641
20	75.68	266.85	8.9	0.00988
30	73.83	407.55	13.15	0.01322
40	68.99	583.40	22.45	0.02743

During testing process, the best accuracy for each of four hidden nodes occurs at 19th iteration, 17th iteration, 4th iteration and 9th iteration, respectively.

As can be seen from the tables, accuracy dropped when the number of hidden nodes was less or more then 20. Accuracy was maximized when the size of the hidden layer was 20. We therefore conclude that optimal size of hidden layer should be 20.

This research then has three layers consisting of 370 input nodes, 20 hidden nodes and 8 output nodes. The best BPNN is trained through 1000 epoch with learning rate of 0.1 and momentum 0.7.

5. Experimental Result

5.1. Performance Measure

In order to evaluate a neural network task of collaborative learning skill, we define a contingency matrix representing the possible outcomes of the classification as shown in table 5.

Table 5. Contingency table for binary classification

	Category positive (C+)	Category negative (C-)
Assigned positive (A+)	true positive (*tp*)	false positive (*fp*)
Assigned negative (A-)	false negative (*fn*)	true negative (*tn*)

Several measures in the information retrieval and machine learning have been defined based on this contingency table. Recall, precision and F_1 measure shown in Eqs. 2, 3 and 4, are the evaluation measures that have been widely applied for evaluating the performance of text classifiers.

$$ErrRate = \frac{fp + fn}{tp + fp + fn + tn} \quad (2)$$

$$\mathrm{Re}call = \frac{tp}{tp + fn} \quad (3)$$

$$\Pr ecision = \frac{tp}{tp + fp} \quad (4)$$

$$F - measure = \frac{2(recall.precision)}{(recall + precision)} \quad (5)$$

In Eqs. (2), (3) and (4), *tp* (true positive) is the number of documents that are correctly categorized, *fp* (false positive) is the number of documents that are put into a wrong category, *fn* (false negative) is the number of categorized documents that actually belong to no category and *tn* (true negative) is the number of documents that are not correctly categorize. In Eq. (5), F is a balanced F-measure, which is a combined measure of precision and recall.

5.2. Comparing the Performance of SVM and BPNN

The experimental results of SVM and BPNN are illustrated in Figure 5 and Figure 6 in 8 by 8 confusion matrix. In this research, confusion matrix C, which is *n* x *n* matrix for N-class classifier is used to compute error rate, recall, precision and F-measure.

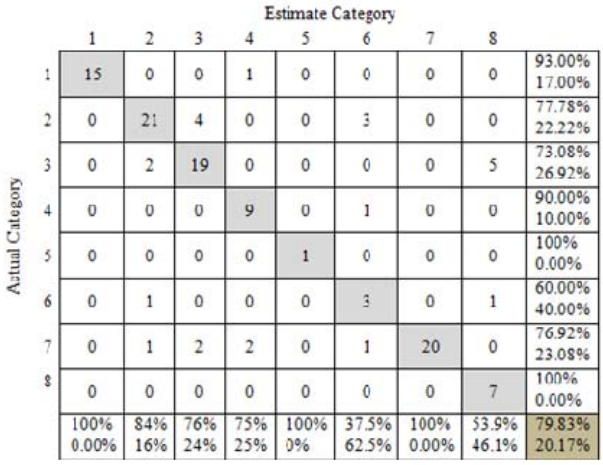

Figure 5. Confusion matrix of SVM

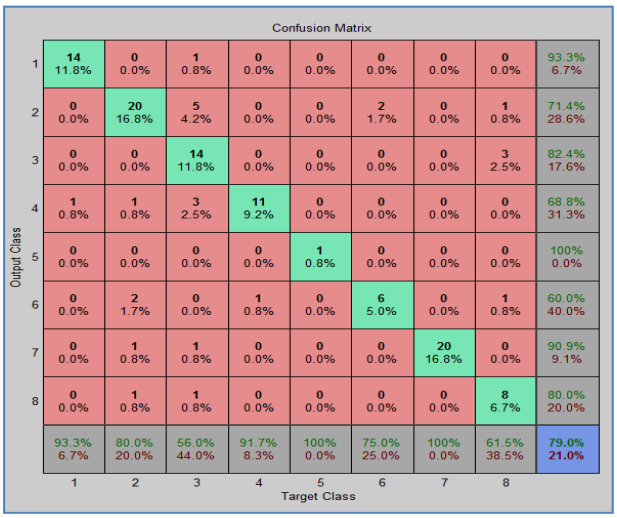

Figure 6. Confusion matrix of neural network

The confusion matrix gives the number of instances where each digit is correctly classified and also the instances where they are misclassified on using optimization dataset. The row indexes of a confusion matrix correspond to *actual values* observed and used for model testing; the column indexes correspond to *estimated values* produced by applying the model to the test data. For any pair of actual/predicted indexes, the value indicates the number of records classified in that pairing. In ideal classifier all entries will be zeroes except diagonal. The sum of the values in the matrix is equal to the number of scored records in the input data table.

There are two values (actual and estimated) for each of category. For instance in confusion matrix of SVM, the first cell (1,1) is the number of actual first category that were predicted to be first category; the value is 15. The second cell (2,2) is the number of actual second category that were predicted to be second category; the value is 21. There are 4 misclassified for second category; 2 of actual second category that were predicted to be third category, 1 of actual second category that were predicted to be sixth category and 1 of actual second category that were predicted to be seventh category.

Comparison performance of SVM and BPNN of eight categories from text of collaborative learning skill in online discussion are summarized in table 6 and table 7. Different category numbers of eight categories may cause diverse performance measures. SVM and BPNN achieved better results in smaller category classes, e.g. Acknowledge, Mediate and Request than larger ones, e.g. Motivate, Task and Inform. The lowest categorization in SVM is motivate category of 37.50%, while in BPNN is Inform of 56%. The performance of most categories is satisfactory. The micro average value of overall recall, precision and F_1 measure rate varied between 78.29% and 83.59%.

Table 6. Performance measure on SVM

Classifier / Categories	SVM			
	Error Rate (%)	Recall (%)	Precision (%)	F-Measure (%)
Acknowledge	0.00	100.00	93.75	96.77
Discuss	16.00	84.00	75.00	79.25
Inform	24.00	76.00	73.07	74.51
Maintenance	25.00	75.00	90.00	81.82
Mediate	0.00	100.00	100.00	100.00
Motivate	62.50	37.50	60.00	46.15
Request	0.00	100.00	76.92	86.95
Task	46.15	53.85	100.00	70.00
Microavg	20.17	78.29	83.59	79.43

Table 7. Performance measure on BPNN

Classifier / Categories	BPNN			
	Error Rate (%)	Recall (%)	Precision (%)	F-Measure (%)
Acknowledge	6.67	93.33	93.33	93.33
Discuss	20.00	80.00	71.43	75.47
Inform	44.00	56.00	73.68	63.63
Maintenance	8.33	91.70	68.75	78.58
Mediate	0.00	100.00	100.00	100.00
Motivate	25.00	75.00	60.0	66.67
Request	0.00	100.00	90.91	95.24
Task	38.46	61.50	80.00	69.64
Microavg	21.00	82.19	79.76	80.31

In terms of error rate, Figure 7 present that SVM outperform BPNN in category Acknowledge, Discuss and Inform with 0%, 16% and 24% respectively. On the other hand, BPNN outperform SVM in category Maintenance, Motivate and Task with 8.33%, 25% and 38.46% respectively.

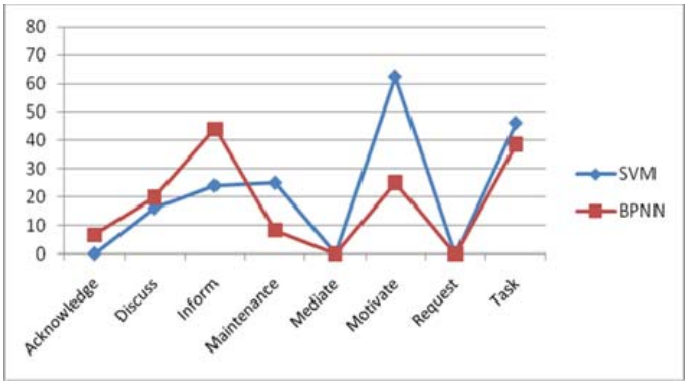

Figure 7. Error rate chart for comparison test

In terms of recall, Figure 8 present that SVM outperform BPNN in category of Acknowledge, Discuss and Inform with 100%, 84% and 76%, respectively. SVM falls behind BPNN in category of Maintenance, Motivate and Task with 75%, 37.5% and 53.85%.

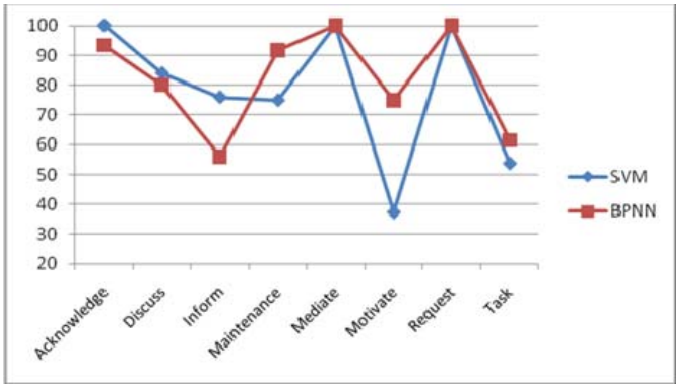

Figure 8. Recall chart for comparison test

In terms of precision, Figure 9 shows that SVM outperform BPNN in category of Discuss, Maintenance and Task with 75%, 90% and 100% respectively. SVM falls behind BPNN only in one category; Request with 90.91%.

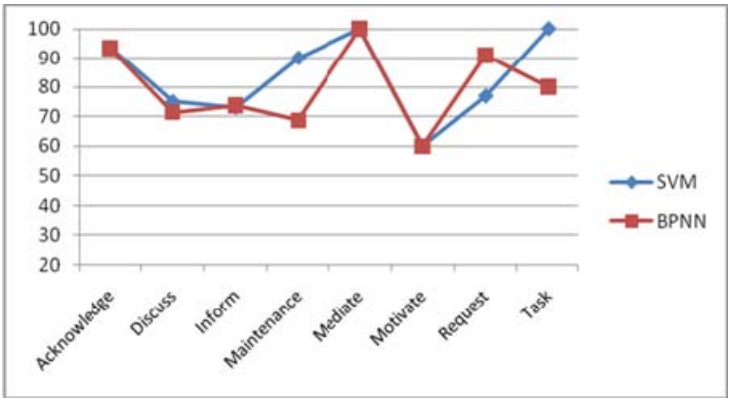

Figure 9. Precision chart for comparison test

In terms of F-Measure that is a combined of recall and precision, thus Figure 10 represent that SVM has leading in category Acknowledge, Discuss, Inform, Maintenance and Task. In the category Motivate and Request, BPNN has leading compare to SVM.

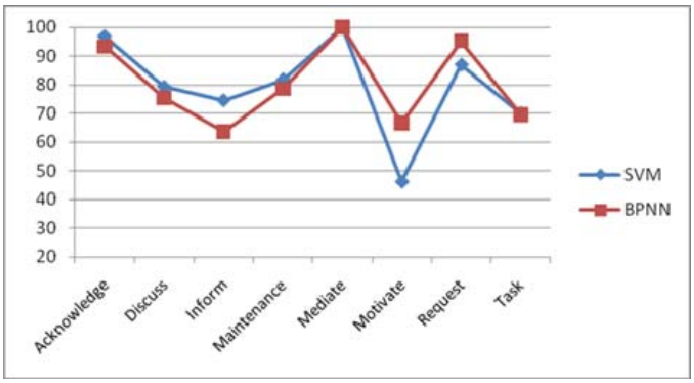

Figure 10. F-Measure char for comparison test

6. Conclusion

This research concludes the proven results that SVM and NN can achieve good classification performance of collaborative learning skill in text categorization. This makes this text classifier the best method with a theoretical justification for its use in text categorization. Such a theoretical understanding of learning methods provide the basis for selecting between text classifiers based on prior knowledge. It identifies how methods and tasks differ, so that it can also guide to development of text categorization.

This research also describes how the methods are efficiently implemented in $SVM^{multiclass}$ and Matlab Toolbox. SVM and NN suitable for training text classifiers with reasonably sized training sets. Furthermore, this research also evaluated the performance and practicability of SVM and NN for learning text classifier. It recognized that solving a learning task is not restricted to simply training the learner, but that pre-processing steps like choosing an appropriate representation are equally important.

The comparison of categorization accuracy has been presented based on error rate, precision, recall, and F-measure. The experimental result shows that one method performed better for some of categories while performing worse in others. In terms of error rate and precision, performances of SVM outperform BPNN and falls behind BPNN in term of recall and F-measure. Comparison in time taken for training and testing, the performance of BPNN is relatively long compared with SVM.

Acknowledgment

This work is supported by Ministry of National Education of the Republic of Indonesia and STMIK-AMIK Riau.

References

[1] Lui AF, Li SC, Choy SO. *An evaluation of automatic text categorization in online discussion analysis*. Seventh IEEE International Conference on Advanced Learning Technologies. 2007: 205-209.
[2] Lam W, Low KF. *Automatic document classification based on probabilistic reasoning: model and performance analysis*. IEEE International Conference on Systems, Man, and Cybernetics, 'Computational Cybernetics and Simulation. 1997: 2719-2723.
[3] Ma L, Shepherd J, Zhang Y. *Enhancing Text Clasification using Synopses Extraction.* The fourth International Conference on Web Information System Engineering. 2003: 115-124.
[4] Farkas J. *Generating Document Clusters using Thesauri and Neural Networks*. Canadian conference on Electrical and Computer Engineering. 1994; 2: 710-713.

[5] Joachims T. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. 10th European Conference on Machine Learning. 1998: 137-142.
[6] Benkhalifa M, Bensaid A, Mouradi A. *Text Categorization using the Semi-Supervised Fuzzy C-means Algorithm*. 18th International Conference of the North American Fuzzy Information Processing Society – NAFIPS. 1999: 561-568.
[7] Kazama J, Tsujii J. Maximum Entropy Models with Inequality Constrains: A Case Study on Text Categorization. *Machine Learning*. 2005; 60(1-3): 159-194.
[8] Gabrilovich E, Markovitch S. *Text Categorization with Many Redundant Features: Using Aggressive Feature Selection to Make SVMs Competitive with C4.5*. Proceedings of the 21st International Conference on Machine Learning. 2004: 321-328.
[9] Yu B, Xu Z, Li C. Latent Semantic Analysis for Text Categorization Using Neural Network. *Journal of Knowledge-Based Systems*. 2008; 900-904.
[10] Hamza AA. Back Propagation Neural Network Arabic Characters Classification Module Utilizing Microsoft Word. *Computer Science. 2*008; 4: 744-751.
[11] Hua LC, Park SC. A Novel Algorithm for Text Categorization using Improved Back-Propagation Neural Network. In: L. Wang et al (*Eds*); 2006; LNAI 4223: 452-460.
[12] Soller A. Supporting Social Interaction in An Intelligent Collaborative Learning System. *Artificial Intelligence in Education.* 2001; 12: 40-62.
[13] McManus M, Aiken R. Monitoring Computer-based Collaborative Problem Solving. *Artificial Intelligence in Education*. 1995; 6(4): 307-336.
[14] Hidalgo JMG. *Text Representation for Automatic Text Categorization*. Eleventh Conference of the European Chapter of the Association for Computational Linguisitics, EACL. 2003; http://www.conferences.hu/EACL03/
[15] Sebastiani F. Machine Learning in Automated Text Categorization. *ACM Computing Survey*. 2002; 34(1), 1-47.
[16] Porter MF. An Algorithm for Suffix Stripping. In : Morgan Kaufmann. *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc; 1997: 313-316.
[17] Yang Y, Pedersen JO. *A Comparative Study on Feature Selection in Text Categorization.* Proceedings of ICML-97. 14th International Conference on Machine Learning. 1997: 412-420.
[18] http://svmlight.joachims.org/svm_multiclass.html
[19] Neural Network Toolbox for MATLAB, http://www.mathworks.com/products/neural-net/