International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:9, 2011

Indonesian News Classification using Support Vector Machine

Dewi Y. Liliana, Agung Hardianto, M. Ridok

***Abstract*—**Digital news with a variety topics is abundant on the internet. The problem is to classify news based on its appropriate category to facilitate user to find relevant news rapidly. Classifier engine is used to split any news automatically into the respective category. This research employs Support Vector Machine (SVM) to classify Indonesian news. SVM is a robust method to classify binary classes. The core processing of SVM is in the formation of an optimum separating plane to separate the different classes. For multiclass problem, a mechanism called one against one is used to combine the binary classification result. Documents were taken from the Indonesian digital news site, www.kompas.com. The experiment showed a promising result with the accuracy rate of 85%. This system is feasible to be implemented on Indonesian news classification.

***Keywords*—**classification, Indonesian news, text processing, support vector machine

I. Introduction

NEWS is information that is presented through print, broadcast, internet, or from mouth to mouth [1]. News is an important thing for the community. Every day people are looking and reading to get some information, for example the manager of a company looking for hot issues to support the policy making, but the number of articles published on the internet cause time consuming searches. Therefore, the need for automatic news classification process, namely the classification of news into specific categories is required to obtain relevant information rapidly.

News classification is one of the topics in data mining. In data mining there are two learning techniques, which is unsupervised learning and supervised learning. Clustering is an example of unsupervised learning, where a group of data is clustered by their level of similarity without any supervision, while classification is a form of supervised learning works by establishing a model which is a function of the training set.

There are several news classification methods including neural network, decision tree, single pass clustering, and Naïve Bayes classifier. The research has been done is to classify news using single pass clustering algorithm [2]. Single pass clustering calculated the similarity level using the standard cosine similarity function. This respective similarity was further evaluated to determine the feature vector that expressed similar documents based on certain threshold value. The accuracy of this algorithm was 79%. Other research was by using the Naïve Bayes Classifier [3]. This algorithm used a direct hypothesis which was formed without searching process, but simply by counting the frequency of occurrence of a word in the training data. The accuracy of this algorithm was 79% [3].

This research uses other methods in solving Indonesian news classification problems that is Support Vector Machine or often referred to as SVM. SVM is a well-known machine learning techniques to solve problems of classification. The goal of SVM is to establish an Optimum Separating Hyperplane (OSH), which is a function of linear separation between classes. The benefit of SVM is the optimization that is used is a reliable linear optimization [4].

The object of this study was taken from www.kompas.com which is one of the digital news site in Indonesian language which are sought after news seekers. Data taken from the website is news published from June until July 2010. News categorization is made into 4 categories: national, international, business and finance, and sports. It is intended to obtain appropriate training data and to simplify testing.

II. Indonesian News Classification System

The design of the system is divided into 3 phases: preprocessing, learning, and classification. Figure 1 shows the design of the system.

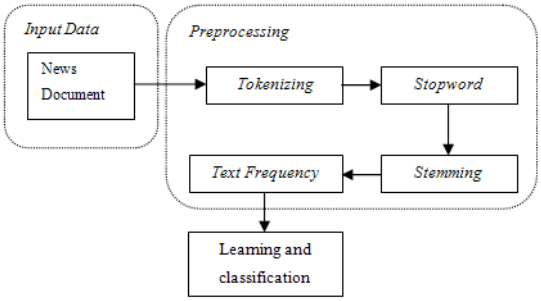

Fig. 1 Design of Indonesian news classification system

The first phase is the preprocessing input data in the form of categorized news including tokenizing, stopword filtering, stemming, and word frequency count. The word that appears at least 3 times on word frequency is considered; the weight of each word is calculated and normalized. All words and its normalized weight combined with a set of words that have been stored in the knowledge to form new word knowledge.

Dewi Y. Liliana is with the Department of Computer Science, Faculty of Mathematics and Natural Sciences, University of Brawijaya, Malang, East Java, Indonesia (email: dewi.liliana@ub.ac.id).

Agung Hardianto is the undergraduate student at Department of Computer Science, Faculty of Mathematics and Natural Sciences, University of Brawijaya, Malang, East Java, Indonesia (e-mail:pea_jack@rocketmail.com).

M.Ridok is with the Department of Computer Science, Faculty of Mathematics and Natural Sciences, University of Brawijaya, Malang, East Java, Indonesia

The second phase is the learning (training phase) based on knowledge that has been formed. News classification process (testing) of an unknown category is proceeded. Further predictions produced a single file that stores the value of the results for testing of classified documents. Positive or negative label of existing results on the test document is used to determine the category of each document testing. If positive then the document is included in the category that had been established previously and vice versa.

III. Text Preprocessing

In the text preprocessing, the steps are tokenizing, stopword filtering, stemming, word frequency counting, computation of TF-IDF features, and normalization.

A. Tokenizing

Tokenizing includes several processes, first is case folding or changing all the letters in the text to lowercase. The next process is parsing. The parsing is simple: break a text into a collection of words without regard to the relationship between words and the role or position in the sentence, the character received in the formation of words is alphabetical only. The repeated word in Indonesian rule will be parsed into two words.

B. Stopword Filtering

After completing the tokenizing the next step is to check into a stopword word list. If the word is a stopword the word is discarded. If not then the word will go through the stemming.

C. Stemming

Stemming is a process to transform word into its basic morphological unit. Stemming used here is Talla stemming [5].

D. Word Frequency Counting

Words that have been stemmed are saved as the training data, each of words in the training data is converted to a format understood by the SVM to be used as input to the SVM learning process. The process is looking for word frequency of at least 3 on each document.

E. TF-IDF Features

Each document is represented as a vector with elements of the term that is recognized from the extraction stage of the document. The vector consists of the weight of each term is calculated based on the TF-IDF method. TF-IDF is a method of weighting which is integration between the term frequency (TF) and inverse document frequency (IDF) [6]. In the TF-IDF weighting the first step is to find the number of words that we want to know its weight or frequency term in each document after it is multiplied by inverse document frequency. The formula to find the weight of words with TF-IDF is:

$$w_{ij} = tf_{ij}.idf$$

$$idf = \log \frac{N}{df_j} \quad (1)$$

Where w_{ij} is weight of word i at the document of j, N the number of documents, and the term frequency tf_{ij} is the sum of the presence of word i in document j, df_j *(document frequency)* is the number of document j which contains word i.

F. Normalization

For the determination of term weights also take into account the document length normalization. The process of normalization will make any weight of the document vector has a value of (0-1). Normalization performed with cosine normalization formula as in equation (2):

$$W_{kj} \equiv \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{r} (tfidf(t_s, d_j))^2}} \quad (2)$$

Where W_{kj} a normalized weight of word k in document j, $tfidf(t_k, d_j)$ is TF-IDF of word k in document j, r is the number of words in document j.

IV. Learning and Classification

The document that has been passed preprocessing will undergo training and testing using SVM. SVM will perform learning and classification. SVM learn functions find a number of support vector from training data by calculating the value of N (number of training data) to obtain the best separator plane using a QP (Quadratic Programming). While the SVM decision functions classify the testing data into it classes. Classification based on one-against-one voting mechanism [7] where each testing data will be voted upon objective function value. The highest votes will be the result of classification of the class.

V. Support Vector Machine

Support Vector Machine (SVM) is a learning systems that uses a hypothesis space of linear functions in a high dimensional feature space, and trained with a learning algorithm based on optimization theory by implementing the inductive bias derived from statistical learning theory [8]. The main purpose of this method is to build OSH (Optimum Separating Hyperplane), which makes an optimum separation function (linear function), that can be used for classification. Suppose $\{x_1,...,x_n\}$ is a data set and $y_i \in \{+1,-1\}$ is a class label of the data x_1.

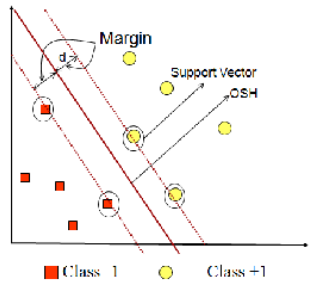

Fig. 2 Support vector and optimum separating hyperplane

In Figure 2 can be seen the separator plane that separate all the objects according to the class. The data residing in border areas is called support vector (rounded objects). Two classes can be separated by a parallel pair of delimiter plane. The first plane limits the first class while the second plane limits the second class, so that equation (3) is obtained as:

$$x_i.w + b \geq +1, y_i = +1 \quad (3)$$

$$x_i.w + b \geq +1, y_i = -1$$

w is the normal plane or the so-called support vector weight and *b* is the relative position of the plane to the center coordinates or the so-called bias. Because the data is nonlinear then it will be transformed into dimensional feature space by using a mapping function (transformation) $x_k \rightarrow \phi(x_k)$ with a kernel function K, the kernel used in this calculation is the RBF kernel as shown in equation (4):

$$k(x_i, x_j) = \exp\left(-\frac{1}{s^2}\left\|x_i - x_j\right\|^2\right) \quad (4)$$

Where x_i and x_j are feature vectors and δ is the product of parameters *C* and *gamma SVM* which are set by users. Weight vector equation is becoming as in equation (5):

$$W = \sum_{i=1}^{N} \alpha_i y_i k(x_i) \quad (5)$$

Weight vector is usually has a big value but with finite value of α. We use Lagrange multiplier to obtain α, therefore we can get *w*. For bias we use the equation (6):

$$b = \frac{1}{\#SV} \sum_{x_i \in SV} \left(\frac{1}{y_i} - \sum_{x_j \in SV} \alpha_j y_j k(x_j, x_i) \right) \quad (6)$$

Where *SV* is the number of support vectors. The best separator hyperplane formula is a quadratic programming problem so that the maximum value of α can be found. After the quadratic programming problem solution is found, then the class of some test data *x* can be determined based on the value of decision function shown in equation (7):

$$D(z) = sign\left(\sum_{i=1}^{N} \alpha_i y_i k(x_i, z) + b\right) \quad (7)$$

Where *D (z)* is the decision function for each class and *N* is the number of support vectors.

VI. Experimental Result and Discussion

Test data used is taken from www.kompas.com that has been converted into flat file format (txt). The data consists of 180 news documents which will be divided into 2 parts. 70% for training data and 30% for testing data with parameter settings as follows: *C* = 60, 70, 80… 150; b. *SVM Gamm*a = 1, 1.5, 2. This parameter setting has great impact in SVM results because it determines the kernel value and error of SVM.

Tests were carried out for three block testing with different parameters setting. The test result with parameters *C* = 60-150, and *SVM gamma* = 1 can be seen in Table 1, the test result with parameters *C* = 60-150, and *SVM gamma* = 1.5 can be seen in Table 2, and the test result with parameters *C* = 60-150, and *SVM gamma* = 2 can be seen in Table 3 respectively.

TABLE I
Experiment Result for Block Test 1

Experiment	*Gamma SVM*	*C*	Accuracy (%)
1	1	60	81.67
2	1	70	81.67
3	1	80	81.67
4	1	90	83.33
5	1	100	90.00
6	1	110	91.67
7	1	120	83.33
8	1	130	88.33
9	1	140	83.33
10	1	150	85.00

* *C*=60-150, *gamma SVM*=1

TABLE II
Experiment Result for Block Test 2

Experiment	*Gamma SVM*	*C*	Accuracy (%)
1	1.5	60	71.67
2	1.5	70	60.00
3	1.5	80	63.33
4	1.5	90	78.33
5	1.5	100	71.67
6	1.5	110	73.33
7	1.5	120	78.33
8	1.5	130	81.67
9	1.5	140	81.67
10	1.5	150	81.67

* *C*=60-150, *gamma SVM*=1.5

TABLE III
Experiment Result for Block Test 3

Experiment	*Gamma SVM*	*C*	Accuracy (%)
1	1.5	60	71.67
2	1.5	70	60.00
3	1.5	80	63.33
4	1.5	90	78.33
5	1.5	100	71.67
6	1.5	110	73.33
7	1.5	120	78.33
8	1.5	130	81.67
9	1.5	140	81.67
10	1.5	150	81.67

* *C*=60-150, *gamma SVM*=2

Results of testing performed on 3 blocks with different parameters *C* and *gamma SVM* value can be seen on table 1, 2, 3. In the first test the results can be seen in table 1, yielded the average accuracy rate of 85.00%. The second test results can be seen in Table 2 with an average accuracy rate of 74 167%, while the third test results can be seen in Table 3 with an average accuracy rate of 60.55%. Based on overall test results with the parameters *C* = 60-150, and *gamma SVM* = 1.0 - 2.0, it obtained the highest average accuracy rate of 85.00%. From table 1, 2, 3 can be summarized and presented in a graphical form as in figure 3:

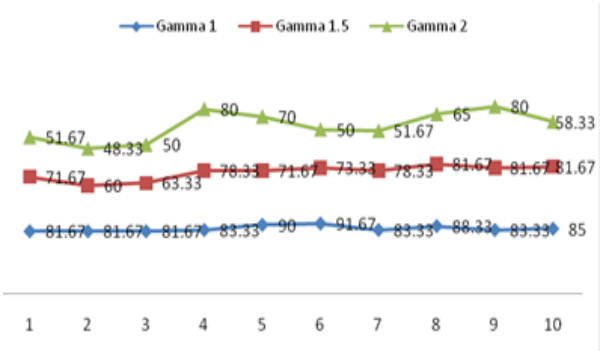

Fig. 3 Graphical view of systems accuracy rate

In the test results we got the values that most influence on accuracy. Any increasing on *gamma SVM* value led to the accuracy decreasing of about 3% to 11%. Increasing the value of the parameter *C* led the accuracy increasing of about 2% to 8%, but at a certain value after reached the top, the accuracy rate will drop significantly. Based on a few things about the effect of parameters C and gamma SVM, the best accuracy occurs when the *gamma SVM* values equal to 1 and the value of *C* equal to 110.

VII. Conclusion

News classification with SVM produced better average accuracy rate of 85%. The values of parameter *C* and *gamma SVM* had a strong effect on the accuracy achieved. With the rising value of the *C* the accuracy rate increased, while the smaller the value of *gamma SVM*, the more it increased the accuracy rate. The best parameter values that yielded the highest average accuracy rate is *C* = 110, and *gamma SVM* = 1. For further development may also attempt to influence other multiclass SVM mechanisms such as one-against all and DAGSVM. Furthermore news sources used can be more diverse, not only from a single source of news.

References

[1] W. S. Maulsby, "Getting in News", in *Mondry, 2008, pp.* 132-133

[2] A. Z. Arifin, and A. N. Setiono, "Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering", Institut Teknologi Sepuluh Nopember(ITS). Surabaya. http://mail.its-sby.edu/~agusza/SITIAKlasifikasiEvent.pdf.

[3] I. Saputra, "Analisa Dan Implementasi Klasifikasi Berita Berbahasa Indonesia Menggunakan Metode Naive Bayes Analysis and Implementation of Classification Indonesian News With Naive Bayes Method". Institut Teknologi Telkom. Bandung.

[4] M. Srinivas, and A. H. Sung. "Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines", in *Journal of Department of Computer Science*, MIT. USA, 2003.

[5] Y. Yang, and X. Liu, " A Re-examination of Text Categorization Methods", *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, 1999, pp. 42-49

[6] Tala, and Z. Fadillah, 2003, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia". Master of Logic Project. Institute for Logic, Language and Computation, Universiteit van Amsterdam, 2003 The Netherlands www.illc.uva.nl/Publications/ResearchReports/MoL-200302.text.pdf.

[7] J. C. Platt, "Sequential Minimal Optimization : A Fast Algorithm for Training Support Vector Machine", *Microsoft research*, 1998.

[8] N. Cristianini, and J. Shawe-Taylor, "*An Introduction to Support Vector Machines*" Cambridge, UK: Cambridge University Press, 2000.