Text Classification Based on SVM and Text Summarization

Vo Duy Thanh
IT Department,
Vietnam-Korea IT College,
Danang, Vietnam

Vo Trung Hung
IT Department,
The University of Danang,
Danang, Vietnam

Ho Khac Hung
IT Department,
Mekong Housing Bank,
Hue, Vietnam

Tran Quoc Huy
Danang Department of Information and Communication,
Danang, Vietnam

***Abstract* - This paper presents the results of our research on text classification which the proposed model is a combination of text summarization technique and semi-supervised learning machine based on the Support Vector Machine (SVM). We propose a solution which is combined two algorithms: searching maximal frequent wordsets and clustering algorithms, extracting the main idea of the text before classifying. The novelty of the proposed method is to summarise the text before constructing of the feature vector in order to minimize the dimension of the vector. In addition, we employ semi-supervised machine learning methods to minimize the number of labelled text used for training (generating the feature model). The experimental results show that the solution achieved a high accuracy; it is more stable and faster than that of the supervised learning or semi-supervised learning based on the support vector.**

***Keywords*: *Text classification; support vector machine (SVM); semi-supervised learning; manifold learning; text summarization*.**

I. INTRODUCTION

Text classification is a significant problem which is widely applied in various areas such as search engine, pattern recognition, data mining, etc. Most of the text classification methods which have been previously proposed are based on machine learning, probabilistic, decision tree, inductive properties, k-nearest neighbour, and recently support vector machine. The aforementioned methods typically aim to classify data into two classes (binary classification), thus, often facing challenges when the data has a large size.

In this paper, we combine the searching maximal frequent wordsets and clustering algorithms to extract the main idea of the text before classifying. By doing so, the text is summarised before constructing the feature vector in order to minimise the dimension of the vector. Additionally, we employ semi-supervised learning technique to mitigate the number of labelled text used for training (to construct the feature model).

We have evaluated the proposed model and compared it with the supervised learning method (using labelled text for training) and the semi-supervised learning method based on the support vector machine. The experimental results show that the proposed model obtained a higher accuracy and more stable than other methods.

The paper is organized as follows. Section 2 reviews the other works related to text summarization, the feature vector and construction of the feature vector, text clustering and extracting the main idea of the text based on the maximal frequent wordsets, text classifying model. Section 3 introduces our model for which it combines the search of maximal frequent wordsets and clustering algorithm to extract the main idea of the text before classifying. Section 4 presents the experiment results and evaluates the proposed model. Finally, section 5 concludes the paper and opens some future work.

II. RELATED WORKS

A. *Text summarization*

Text summarization is an important problem for data mining in general and for text classification in particular. It helps to reduce the text size but still guarantee to express the main idea of the text.

There are many models which have been recently proposed for automatic text summarization of English, Japanese, and Chinese. W. B. Cavnar (1994) [5] have represented the document based on the n-gram model instead of conventional keyword model. A. Chinatsu (1997) [8] have developed the DimSum system for text summarization using natural language processing techniques and statistical method based on the co-efficient of tf-idf. J. Carbonell (1998) [6] has summarised the text by ordering and extracting the excel sentences (representing the main idea of the text). J. Goldstein (1999) [14] has classified the text summarization based on the relevant measurements. The method combines between linguistic features and statistics. Each sentence is characterised by linguistic features and statistical measurements, J. L. Neto (2000) [21] has generated the summary of text through the relevant importance of topics. D. Radev (2000) [27] has built text summary based on the centroid of the text in order to extract the key sentence. Y. Gong (2001) [15] has proposed two simple methods for text summarization: based on statistical measurements, frequency analysis and approach latent semantic.

In terms of Vietnamese text processing, there are several well-known models which have been introduced such as N.T.M. Huyen's model (2003) [29] on how POS tagging; the model of D. Dien et al. (2001) [11] which is proposed for separating Vietnamese words; the model of H. Kiem and D. Phuc (2002) [20] for text classifying based on the most frequent phrases; the D. Phuc's model applying frequent wordsets and association rule for classifying of Vietnamese text with the concern of context.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

B. *Definitions*

1) Definition 1: Wordset is a set of words that sequentially occurs in a sentence. The frequency of a wordset is the number of sentences which contains that wordset. Let t be a wordset, $sp(t)$ be the ratio of the number of sentences that contains a wordset over the total number of sentences in the text. Let $min\_T \in [0,1]$ be the number of minimal frequent thresholds, wordset *t* is considered as frequent according to threshold min_T if $sp(t) \geq min\_T$.

2) Definition 2: Maximal frequent wordset is the wordset that is not the subset of any frequent wordsets.

There are several pieces of work studied for word separating, which have a high accuracy. However, in this paper, we apply the n-gram method to analyse word/phrase in Vietnamese documents and combine with Vietnamese dictionary to determine a meaningful word/phrase. It should be noted that Vietnamese have 81.55 % syllables which are single words; about 70.72 % compounds which are double-syllable; around 13.59 % compounds having 3-syllable, 4-syllable; and only 1.04% compound having above 5-syllable. Therefore, in this study, we employ an n-gram of size 3 (to investigate all words which have from 1 to 3 syllables).

C. *Creating the feature vector of text*

To generate the feature vector, we first utilise the algorithm of finding maximal frequent sets appeared in sentences of the text and then build the feature vectors of sentences. Particularly, for binary vector, the *k*-th element of the vector corresponding to the *j*-th sentence is equal to 1 if the *j*-th sentence contains the frequent wordset of the *k*-th element; otherwise, it is equal to 0.

Frequent wordset finding algorithm

The frequent set finding algorithm is applied for finding the frequent wordsets in the document which has multiple lines of text. Each text line is considered as a transaction. An itemset $\{i_1, i_2, \ldots, i_k\}$ has items of $i_1, i_2, \ldots, i_k$ which will become sets of words $i_1 i_2 \ldots i_k$. Note that $i_1, i_2, \ldots, i_k$ are words separating by a space; or following by a full stop before or after those words.

Step 1: Generating F_1 wordsets which have only one word and frequency is greater than *minsupp.*

Step 2: Using Apriori algorithm to find frequent itemsets in the database. At step *k-th*, Apriori uses Breadth-First Search (BFS) and a Hash tree structure to count candidate itemsets efficiently. It generates candidate itemsets of length *k* from itemsets of length *k-1*, the candidate itemsets contains all frequent *k*-length itemsets. After that, it scans the transaction database to determine frequent itemsets among the candidates.

Based on the aforementioned frequent wordsets, we construct the maximal frequent wordsets of the text.

D. *Clustering and extracting main idea of the text based on the maximal frequent wordsets*

On the basis of maximal frequent wordset, the text clustering algorithm is designed as follows:

Step 1: Identifying the last total number cluster of the data block that contains the text manned as c_end.

Step 2: Creating *C* which is the set of feature vectors of initial clusters. Each vector represents the sentences in the document that is needed to cluster.

Repeat the following steps:

Step 3: Computing the distance matrix among feature vectors of clusters in C relied on Hamming distance calculation algorithm.

Step 4: Finding two clusters which have the minimum distance between any two cluster feature vectors in C. They are named as c_min_i and c_min_j.

Step 5: Merging two cluster c_min_i and c_min_j to render a new cluster named as c_min_ij.

Step 6: Removing c_min_i and c_min_j out of the vector space *C* and inserting c_min_ij.

Step 7: If $|C| \leq c\_end$, exiting and returning to Step 3.

The algorithm is re-written as follows:

```
procedureextract_clustering(c_end)
---------------------------------------------
Input:
The number of clusters of data block that
contains entire text: c_end
Set C of feature vectors of initial cluster
Output:
The vector spce C after clustering
---------------------------------------------
Begin
 do
  iMin = maxint
  for I from 1 to c.length- 1do
    for j from I + 1 to c.length do
      Hamming=hamming_distance(c,I,j)
      if(iMin<iHamming) then
       iMin = hamming_distance(c,i,j)
         c_min_i = i
         c_min_j = j
      endif
    endfor
  endfor

//Merge two c_min_i and c_min_j
//and removing out of C
   c_min_ij = or_vector(C,i,j)
   remove_vector(C, c_min_i, c_min_j)
     add_vector_c(C, c_min_ij)
 loop‖C‖ ≤ c_end
end
```

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

The algorithm of *hamming_distance(C, i, j)* is used for computing the minimum distance between two clusters according to Hamming algorithm:

```
   function hamming_distance(C, i, j)

   input:
    + Set of feature vectors of clusters:
C
    + The indexes of two vectors needed to
calcualate the distance: i, j
   output:
    + Hamming distance of two vectors i and
j

   Begin
    iHamming = 0
   for k from 1 to c.length do
    if (c[i][k] = c[j][k] then
        iHamming++
    endif
   endfor
     return iHamming
   end
```

The algorithm of *or_vector(C, i, j)* is used for merging two given vectors to render a new vector under the operation OR.

The algorithm of *remove_vector(C, c_min_i, c_min_j)* is applied for removing two vectors of *c_min_i* and *c_min_j* out of vector *C*.

The algorithm of *add_vector(C, c_min_ij)* is applied for adding the vector *c_min_ij* into the vector *C* space.

Relying on the total number of clusters obtained from clustering phase, we look for the key sentences of the original document based on the space of the feature vector to achieve the summary of the document. In this paper, we do not describe in details the algorithms that allow us to find the key sentences and summarise the main idea of the document. This is done by using the *iToolSVM* tool which enables to summarise the document, supports to assign labels and creates training data file based on input standard which supports *SVMLin* tool.

E. Text classification model

Text classification is a process of analysing and mapping a document into one or more given classes according to a classification model. This model is built by basing on a set of documents which are labelled (are determined the class) named as training documents.

The text classification problem can be stated as follows. Given a set of documents $U = \{u_1, u_2, \ldots, u_n\}$ and a set of titles $C = \{c_1, c_2, \ldots, c_m\}$. The objective of the problem is to properly classify the document u_i containing in set *C*. This problem can be considered as the finding of function *f* problem:

$f : U \times C \rightarrow$ Boolean

$f(u,c)$ = true if u has the title in c

$f(u,c)$ = false if u has no title in c

There are many data classification problems such as binary classification (identifying that a document is whether it belongs to a given class or not), multi-class classification (a document belongs to a class in a given class list), multi-value classification (a document belongs to more than one class in a given class list, e.g., a document can belong to both sport class and news class).

The general model for text classifying is described as follows:

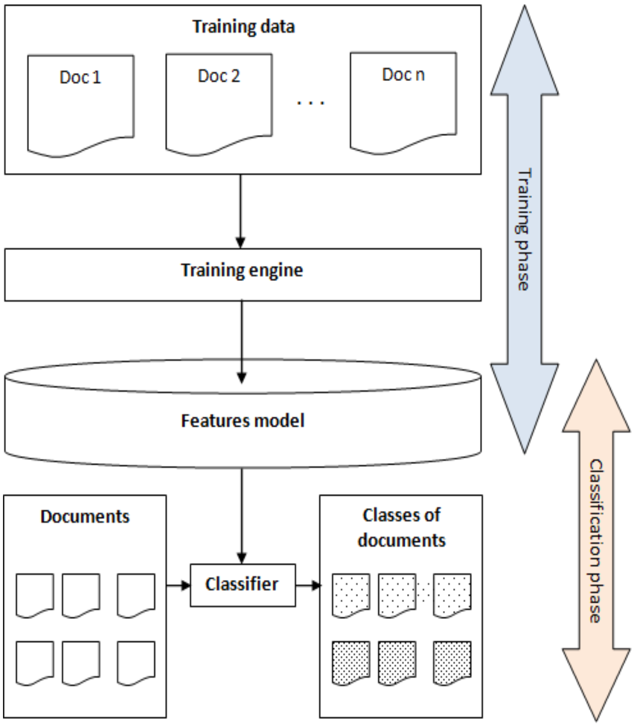

Fig. 1. Text classification model

F. Self-training algorithm

Self-training algorithm is semi-supervised learning technique in which the initial classifier is trained by a small amount of labelled data [28]. Then, this classifier is used for labelling to unlabelled data. Labelled data which are highly reliable (i.e., the reliability of the labelled data is above a given threshold) will be added into the training data set. The classifier will repeat to learn based on the new training data set. In each loop, the highest reliable samples will be move into the training data set.

Objective: Extending the training data set which has been labelled by using unlabelled data set *U* and title (label) set *C* [18][20].

Input:

L: Labelled training data set

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

U: Unlabelled training data set

C: Title set (label)

Output: Labels of elements that is subset of U, i.e., U′, having the highest reliability.

Algorithm [20]

Step 1. Input L, U, C

Step 2. Generating U′ is a subset of U by randomly choosing P elements in U.

Step 3. Loop for iterations

Use L to individually train the classifiers C_i and label the examples in U'.

For each classifiers C_i select G most confidently examples and add them to L, while maintaining the class distribution in L. Confidently can be assessed by calculating the precision of classification carried out with L. We have L = L + G; U′ = U′ − G.

Refill U′ with examples from U, to keep U′ at a constant size of P examples.

III. THE PROPOSED MODEL

We propose to summarise the document with a given compression ratio before training the system by applying the model that has been introduced in our previous work [34] for text classifying. The proposed model is shown in the following figure:

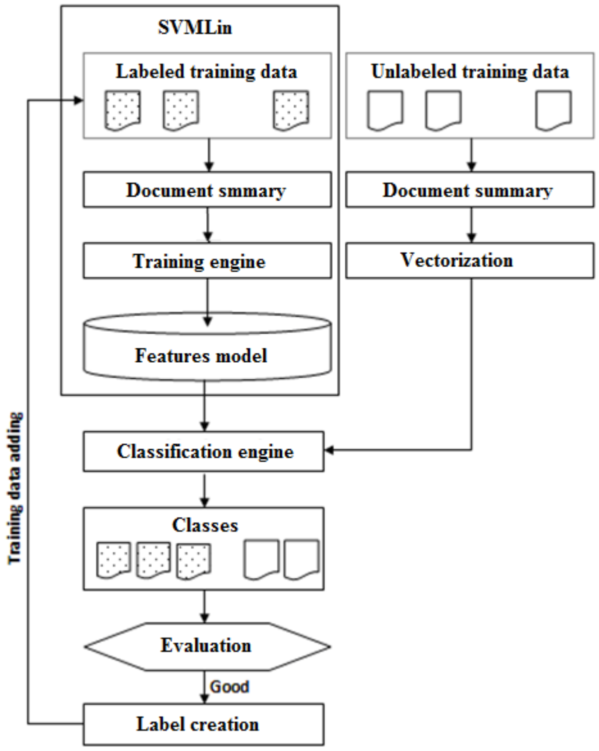

Fig. 2. The proposed model for text classification with the support of text summarization

Our proposed model includes two stages:

Stage 1: Applying *iToolSVM* tool for summarising of input documents with a given compression ratio. In this paper, this ratio is set to 70%.

Stage 2: Employing *SVMLin* for training and classifying text based on the proposed model.

IV. EXPERIMENTS AND EVALUATIONS

A. Objectives

To apply the proposed model for classifying documents into different subjects: sports, entertainment and education from the input gathered on the online newspaper.

B. Implementation

In the scope of this research, we have summarised the document with the compression ratio of 70%. The implementation is described as follows:

Step 1: Applying *iToolSVM* tool for summarising document and generating training data set which includes labelled documents.

Step 2: Using *SVMLin* for building the feature model to each class.

Step 3: Testing the classification for 600 random documents.

Step 4: Adding data to unlabelled training data.

Step 5: Utilising *SVMLin* to re-generate the feature model.

Step 6: Testing the classification for 600 documents which have been previously tested in Step 3.

Step 7: Comparing results obtained in Step 3 and Step 6.

Consequently, we employ *SVMLin* tool [21] for training and evaluating as well as comparing achieved results with the results obtained by using our model [34] which has been proposed previously.

C. Evaluation of test results

To evaluate the efficiency achieved by applying the clustering algorithm during the process of text classifying, we compare the proposed model with the semi-supervised learning SVM model. In this experiment, we apply both supervised learning method with the *Regularized Least Squares Classification* (RLS) algorithm and semi-supervised learning SVM with the *Multi-switch Transductive L2-SVMs* algorithm [32] to evaluate the efficiency based on the dimensions of labelled and unlabelled data sets. In each case, the experiment has been run with 200 documents extracted from *vnexpress.net*. The achieved results are compared to that of the previously proposed model [34] to evaluate the model that combines the former model with the algorithm of document summarising having the compression rate of 70%.

1) Efficiency of the semi-supervised learning model with respect to the dimension of unlabelled training data set

The experiment has been repeated 10 times with the 610 document training data set. In each experiment, 10 labelled documents have been randomly selected, and the dimension of training data set has been increased from 100 to 600 documents. The achieved results are compared to that of the previous work [34] as follows:

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

TABLE 1. THE ACCURACY OF THE SEMI-SUPERVISED LEARNING MODEL VS. THE DIMENSION OF UNLABELLED DATA

#Test	Accuracy (%)		
	RLS	*L2-SVM*	*SL2-SVM*
1	94.60	97.00	97.14
2	93.00	94.50	95.06
3	73.50	96.00	95.36
4	78.30	95.50	96.02
5	78.40	95.50	95.67
6	91.00	96.50	97.12
7	87.50	93.00	95.63
8	89.00	94.50	95.67
9	83.50	96.50	96.34
10	82.40	97.00	97.05
Average	85.10	95.70	96.11

Where the columns of RLS, L2-SVM and SL2-SVM represent the accuracy of the supervised learning method, semi-supervised learning method and the proposed method respectively.

Table 1 shows that the semi-supervised learning model has a much higher accuracy than the supervised learning model. Also, the proposed model achieves a better classification results than the semi-supervised learning model. Figure 3 is plotted based on the results described in Table 1.

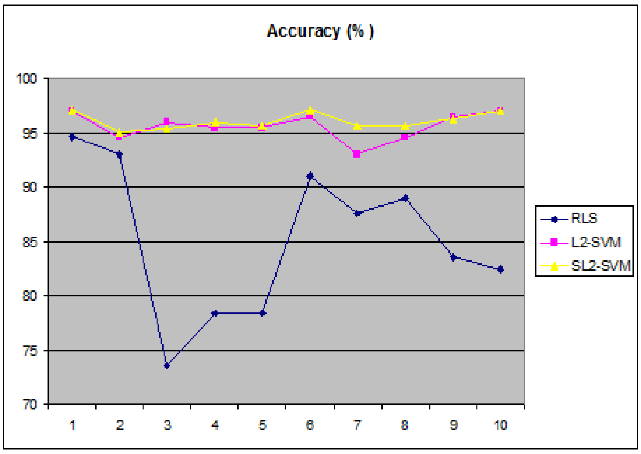

Fig. 3. The accuracy of the semi-supervised learning model vs. the dimension of unlabelled data

2) Efficiency of the semi-supervised learning model with respect to the dimension of labelled training data set

The experiment has been repeated 10 times with the 610 document training data set. In each experiment, 510 unlabelled documents have been randomly selected, and the dimension of training data set has been increased from 10 to 100 documents. The obtained results are compared to that of the previous work [34] as follows:

TABLE 2. THE ACCURACY OF THE SEMI-SUPERVISED LEARNING MODEL VS. THE DIMENSION OF LABELLED DATA

#Test	Accuracy (%)		
	RLS	*L2-SVM*	*SL2-SVM*
1	85.62	95.30	96.39
2	92.70	96.50	97.62
3	84.06	95.80	95.06
4	87.32	97.50	97.12
5	93.17	94.70	95.67
6	89.22	94.00	95.06
7	86.37	96.80	97.43
8	91.23	93.50	95.04
9	83.26	98.50	97.49
10	89.72	97.00	98.36
Average	88.23	95.96	96.22

Table 2 reveals that when the dimension of the training data set increases, the accuracy of the supervised learning method also increases. However, the accuracy of the semi-supervised learning method [34] is still much higher than that of the supervised learning method. Futhermore, the proposed model, which further applies the document summarization method, has a better accuracy than others. Fig. 4 illustrates the results described in Table 2.

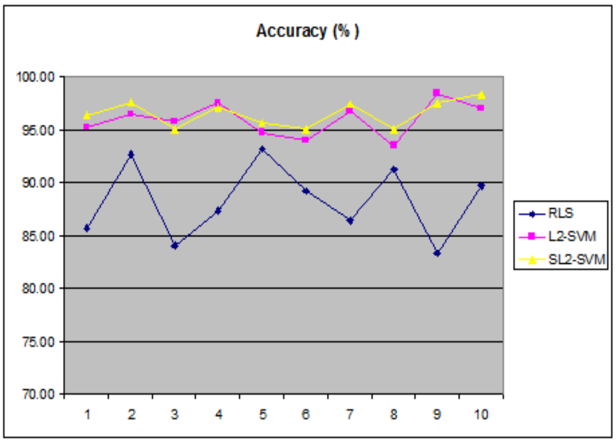

Fig. 4. The accuracy of the semi-supervised learning model vs. the dimension of labelled data

V. CONCLUSION

The experiment results show that in both situations when the dimension of the labelled and unlabelled training data sets increase, the semi-supervised learning method has achieved a better classification result and a higher accuracy than others. As can be seen that the model which has been proposed in this paper has a better classification result and a higher accuracy than the model that has been previously proposed in [34], the improvement is observed as non-significant.

To improve the efficiency of the semi-supervised learning model with text summarization, we keep going on with the methods of separating Vietnamese word technique. This technique helps to increase the accuracy of the main idea extraction method. Besides, the experiment will be conducted with different compression rates to find the optimal one in

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

order to make a lot of further improvements about the performance of the proposed model.

REFERENCES

[1]. M. F. Balcan, and A. Blum, "An augmented pac model for semi-supervised learning", In O. Chapelle, B. Sch¨olkopf and A. Zien (Eds.), Semi-supervised learning. MIT Press, 2006.

[2] K. P. Bennett, "Semi-Supervised Support Vector Machines", Department of Mathematical Sciences Rensselaer Polytechnic Institute Troy, 1998.

[3] A. Blum, and T. Mitchell, "Combining labeled and unlabeled data with co-training. COLT", Proceedings of the Workshop on Computational Learning Theory, 1998.

[4] A. Carlson, "Coupled Semi-Supervised Learning", Machine Learning Department School of Computer Science, Carnegie Mellon University Pittsburgh, 2010.

[5] W. B. Cavnar, "Using An N-Gram-Based Document Representation With A Vector Processing Retrieval Model", Proceeding TREC 1994, Gaithersburg, Maryland, 1994.

[6] J. Carbonell, J. Goldstein, "The Use of MMR and Diversity-Based Reranking for Reodering Documents and Producing Summaries", Proceedings of the 21st meeting of International ACM SIGIR Conference, Melbourne, Australia, August 1998, p. 335-336.

[7] O. Chapelle, B. Sch¨olkopf, and A. Zien, "Semi-Supervised Learning", The MIT Press Cambridge, Massachusetts London – England, 2006.

[8] A. Chinatsu, "A scalable summarization system using robust NLP", Proceedings of a Workshop Sponsored by the Association for Computational Linguistics, 1997.

[9] M. Collins, Y. Singer, "Unsupervised models for named entity classification", Methods in Natural Language Processing EMNLP/VLC-99, 1999.

[10] F. G. Cozman, I. Cohen, "Unlabeled data can degrade classification performance of generative classifiers", Int' l Florida Artificial Intell. Society Conf, pp. 327-331, 2002.

[11] D. Dien, H. Kiem, N. V. Toan, "Vietnamese Word Segmentation", Proceedings of the NLPRS2001, Tokyo (Japan, 27-30 November 2001, p. 749-756).

[12] Đ. Phúc, T. T. Lân, "Phân loại văn bản tiếng Việt dựa trên tập thô", Hội thảo Quốc gia về CNTT, Đà Nẵng, 2004.

[13] Đ. Phúc, "Nghiên cứu ứng dụng tập phổ biến và luật kết hợp vào bài toán phân loại văn bản tiếng Việt có xem xét ngữ nghĩa", Tạp chí Phát triển Khoa học và Công nghệ, Tập 9, Số 2-2006.

[14] J. Goldstein, M. Kantrowitz, V. Mittal, J. Carbonell, "Summarizing text documents: Sentence selection and Evaluation Metrics", Proceedings of SIGIR-99, Berkeley, CA, 1999.

[15] Y. Gong, X. Liu, "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis", Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, USA, 2001.

[16] F. Glenn, O.L.Mangasarian., "Semi-supervised Support Vector Machines for Unlabeled Data Classification", Optimization Methods and Software, pp. 1-14, 2001.

[17] C. C. Kemp,T.L.Griffiths, S.Stromsten, J.B. Tenenbaum, "Semi-supervised learning with trees", Advances in Neural Information Processing System 16th, 2003.

[18] K. P. Bennett, A. Demiriz., "Semi-supervised Support Vertor Machines", Advances in neural information processing systems, pp. 368-374, 1998.

[19] L. Hamel, "Knowledge Discovery With Support vector machines", University of Rhode Island, 2008.

[20] H. Kiem, D. Phuc, "Phân loại văn bản dựa trên cụm từ phổ biến", Kỷ yếu hội nghị khoa học lần 2, Trường Đại học Khoa học Tự nhiên TP. HCM, 2002, p109-p113.

[21] J. Larocca Neto, A. D. Santos, A. A. Kaestner, A. A. Freitas, "Generating Text Summaries through the Relative Importance of Topics", Proceedings of 15th Brazilian Symposium on Artificial Intelligence (SBIA'00). Lecture Notes in Artificial Intelligence, No. 1952, Springer-Verlag (2000) 300-309.

[22] G. S. Man, A. McCallum, "Simple, robust, scalable semi-supervised learning via expectation regularization", The 24th International Conference on Machine Learning, 2007.

[23] T. Mitchell, "The discipline of machine learning", Technical Report CMUML-06-108, Carnegie Mellon University, 2006.

[24] T. Mitchell, "The role of unlabeled data in supervised learning", Proceedings of the 6th International Colloquium on Cognitive Science, San Sebastian, Spain, 1999.

[25] A. McCallum, &K. Nigam, "A comparison of event models for naïve bayes text classification", AAAI-98 Workshop on Learning for Text Categorization, 1998.

[26] K. Nigam, A. K. McCallum, S. Thrun, &T. Mitchell, "Text classification from labeled and unlabeled documents using EM", Machine Learning, vol. 39, pp. 103–134, 2000.

[27] D. Radev, "Text Summarization Tutorial", Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR), 2000.

[28] R. Mihalcea, "Co-training and self-training for Word sense disambiguation", Conference onNatural Language Learning (CoNLL), Massachusets, U.S 2004.

[29] T. M. Huyen Nguyen, "A case study in POS Tagging of Vietnamese texts", TALN 2003, Batz-sur-Mer, 11-14 Juin 2003.

[30] V. Sindhwani, S. S. Keerthi, "Large Scale Semi-supervised Linear SVMs", Proceedings of the 29th annual International ACM SIGIR conference on Research and development in Information retrieval, pp. 477 – 484, 2006.

[31] V. Sindhwani, S. S. Keerthi., "Newton Methods for Fast Solution of Semisupervised Linear SVMs", MIT Press, 2005.

[32] T. Joachims, "Transductive inference for text classification using support vertor machines", Proc. 16th International Conf. on Machine Learning, pp. 200–209, Morgan Kaufmann, San Francisco, CA., 1999.

[33] V. Sindhwani, "SVMLin – Fast Linear SVM Solvers for Supervised and Semi-supervised Learning", Department of Computer Science, University of Chicago, 2006.

[34] V. D. Thanh, V. T. Hung, P. M. Tuan, and D. V. Ban, "Text classification based on semi-supervised learning", Proceeding of the SoCPaR 2013, IEEE catalog number CFP1395H-AR, ISBN 978-1-4799-3400-3/13/$31.00, pp. 238-242, 2013.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)