TEXT CLASSIFICATION BASED ON SUPPORT VECTOR MACHINE

Le Thi Minh Nguyen*

*aThe Faculty of Information Technology, Hochiminh City University of Foreign Languages - Information Technology, Hochiminh City, Vietnam
*Corresponding author: Email: nguyentm@huflit.du.vn

Article history
Received: December 15th, 2018
Received in revised form: January 29th, 2019 | Accepted: February 14th, 2019

Abstract

The development of the Internet has increased the need for daily online information storage. Finding the correct information that we are interested in takes a lot of time, so the use of techniques for organizing and processing text data are needed. These techniques are called text classification or text categorization. There are many methods of text classification, but for this paper we study and apply the Support Vector Machine (SVM) method and compare its effect with the Naïve Bayes probability method. In addition, before implementing text classification, we performed preprocessing steps on the training set by extracting keywords with dimensional reduction techniques to reduce the time needed in the classification process.

Keywords: Feature vector; Kernal; Naïve Bayes; Support Vector Machine; Text classification.

DOI: http://dx.doi.org/10.37569/DalatUniversity.9.2.536(2019)
Article type: (peer-reviewed) Full-length research article
Copyright © 2019 The author(s).
Licensing: This article is licensed under a CC BY-NC-ND 4.0
PHÂN LỚP VĂN BẢN DỰA TRÊN SUPPORT VECTOR MACHINE

Lê Thị Minh Nguyễn*

*Khoa Công nghệ Thông tin, Trường Đại học Ngoại ngữ - Tin học TP. Hồ Chí Minh, TP. Hồ Chí Minh, Việt Nam
*Tác giả liên hệ: Email: nguyenltm@huflit.du.vn

Lịch sử bài báo
Nhận ngày 15 tháng 12 năm 2018
Chỉnh sửa ngày 29 tháng 01 năm 2019 | Чấp nhận đăng ngày 14 tháng 02 năm 2019

Tóm tắt

Sự phát triển của Internet làm cho thông tin lưu trữ trực tuyến hàng ngày tăng nhanh chóng. Do vậy, để tìm đúng thông tin mà chúng ta cần quiero thời gian nên cần phải dùng những kỹ thuật tổ chức và xử lý dữ liệu để văn bản. Kỹ thuật này được gọi là phân loại văn bản hay nói cách khác là phân loại văn bản. Đã có rất nhiều phương pháp nghiên cứu về phân loại văn bản nhưng trong bài viết này chúng tôi tìm hiểu và áp dụng phương pháp Support Vector Machine và so sánh hiệu quả của nó với phương pháp phân loại theo xác suất Naïve Bayes. Ngoài ra, trước khi thực hiện phân loại chúng tôi thực hiện các bước tiền xử lý bằng cách trích xuất các từ khóa đặc trưng với kỹ thuật giảm chiều tập huấn luyện nhằm làm giảm thời gian trong quá trình phân loại.

Từ khóa: Hàm nhất; Naïve Bayes; Phân loại văn bản; Support Vector Machine; Vector đặc trưng.

DOI: http://dx.doi.org/10.37569/DalatUniversity.9.2.536(2019)
Loại bài báo: Bài báo nghiên cứu gốc có bình duyệt
Bản quyền © 2019 (Các) Tác giả.
Cấp phép: Bài báo này được cấp phép theo CC BY-NC-ND 4.0
1. INTRODUCTION

Text classification is not a new problem because it is widely used to classify documents. For example, in financial market analysis, an analyst needs to synthesize and read a lot of articles and documents related to this field in order to make economic predictions for his business to know what to do in the incoming stages. However, with the ever-increasing amount of information available on the Internet, the analyst can no longer read to classify which document belongs to the group he is interested in so that he can read more carefully for his intended purpose. Therefore, text classification is becoming more of a hot topic in modern information processing. Moreover, today's textual information is stored in servers and different databases and most of them are semi-structured text data. Thus, the purpose of the text classification is to determine the category for each document in a set of documents according to the predefined topic category (Jiang, Li, & Zheng, 2011, as cited in Xue & Fengxin, 2015). Through the process of text categorization, texts can be classified and help users’ information searching to be greatly improved and the analyst can quickly read the documents that he cares about.

In the learning machine there are text classification models based on methods such as: Decision trees, Naïve Bayes, k-nearest neighbors, neural networks, random forest (Kim, Han, Rim, & Myaeng, 2006; Xue & Fengxin, 2015), but the Support Vector Machine (SVM) algorithm is of great interest and is used in text classification as it gives better classification results than other classification methods. Studies and applications using SVM are presented in Section 2; Section 3 presents the definition and the generalized classification model; Section 4 presents the feature extracting techniques in texts; Section 5 presents the classification method based on the Naïve Bayes theorem and SVM; Section 6 presents the experimental results of the Naïve Bayes and SVM models and compares the classification efficiency of the two models on the dataset collected from www.vnexpress.net news sites and, finally, is the conclusion and direction of future research.

2. RELATED WORKS

There have been many studies on text processing in the World using the SVM achieving many positive results, such as: In the education data mining technique, (Umair & Sharif, 2018) predicted students’ performance on the basis of their habits and comments; Stock price prediction (Madge & Bhatt, 2015) used daily closing prices for 34 technology stocks to calculate price volatility and strong momentum for individual stocks and for the overall sector. Ehrentraut, Ekhholm, and Tan (2018) built a surveillance system that reliably detects all patient records of who have potentially hospital-acquired infections to reduce the burden of having the hospital staff manually check patient records. In addition, Text Categorization with Support Vector Machine is available not only in English but also in German (Leopold & Kinermann, 2002) or Chinese (Lin, Peng, & Liu, 2006) and many other languages. Leopold and Kinermann (2002) studied different weight schemes for the representation of texts in input space. Each of the mappings of text to input space consists of three parts: First the term
frequencies are transformed by a bijective mapping, then the resulting vector is multiplied by a vector of importance weights, and this is finally normalized to unit length because the text content is of different lengths. Long texts can contain thousands of words while short texts only contain a few dozen words, so using the frequency to make the text of different lengths comparable. Lin et al. (2006) used the SVM algorithm to perform question classification in Chinese to aim at predicting the answer from question features.

In addition, the Vietnamese classification issue has also been studied by research agencies and very feasible and significant results have been achieved, such as Vietnamese classification by SVM model (Nguyen & Luong, 2006) with data collected from vnexpress.net pages achieving a classification accuracy up to 80.72%. Vietnamese language classification based on neural network method (Pham & Ta, 2017) with data collected from Websites: vnexpress.net, tuoitre.vn, thanhnien.vn, teleport-pro.softonic.com, and nld.com.vn have achieved great results with accuracy up to 99.75%.

3. CLASSIFICATION MODEL

3.1. Define

Given a set of documents \( D = \{d_1, d_2, ..., d_n\} \) and a set of classes \( C = \{c_1, c_2, ..., c_n\} \). The goal of the problem is to determine the classification model, which means finding the function \( f \) so that:

\[
 f : D \times C \rightarrow Boolean
 f(d,c) = \begin{cases} 
 true & \text{if } d \in c \\
 false & \text{if } d \notin c 
\end{cases}
\]  

(1)

3.2. General model

There are many approaches to the text classification problem that have been studied, such as: Approaches based on graph theory, rough set theory, statistics, supervised learning, unsupervised learning and reinforcement learning. In general, the text classification method usually consists of three stages:

- **Stage 1**: Preparing the dataset, including data loading process and performing basic pre-processing such as deleting HTML tags, and standardizing spelling. Then, splitting the processed data into two parts: The training dataset and the test set;

- **Stage 2**: The next step is to extract the features from the raw dataset by selecting representative keywords as input datasets and then transforming them into flat features for use with the classification model;
• **Stage 3:** The final step is to build the model from the labelled training dataset.

4. **FEATURE EXTRACTION**

After the pre-processing stage we apply some natural language processing techniques to translate the dataset into feature vectors as input attributes for classification.

4.1. **Word segmentation**

In this article, we apply the SVM method to Vietnamese text. Unlike English, the boundary between words in Vietnamese is not always separated by character spacing because Vietnamese is an East Asian language. In Vietnamese character spacing is used to separate syllables rather than words (Nguyen, Ngo, & Jiamthapthaksin, 2016). The syllable in Vietnamese does not make any sense. However, it is also explained in structural features such as "quốc kỹ". Here, "quốc" means nation, "kỹ" means "flag", so "quốc kỹ" means national flag. The basic unit in Vietnamese is the phoneme. Phonemes are the smallest units but are not used independently in the syntax. Vietnamese words can be classified into two types: i) One syllable with full meaning and ii) n syllables in the fixed token group. Thus, the extract feature section is the word segmentation stage. So the word segmentation in Vietnamese is to combine the adjacent syllables into a meaningful phrase. For example, “các phương pháp phân loại văn bản” is separated into các phương pháp phân loại văn bản. After performing word segmentation, for words that have many syllables, the syllables are joined to each other by underscores, e.g. "văn_bản". But in other cases, a sentence is separated into several different meanings. For example, "đêm hôm qua câu gãy". It is split into (1) đêm hôm qua câu gãy or (2) đêm hôm qua câu gãy. We see a clear difference between the two meanings of a sentence. So, the accuracy of word segmentation is very important. If the word segmentation is incorrect, the classification is wrong. To choose the good features, it is necessary to remove words that are not meaningful to the classification, i.e. remove the word-stop. In the removal of the word stop, we identify common words that are not specific or make no sense when participating in the text categorization, such as “cửa, cái, tất cả, từ đó, từ ấy, bố cuối, bằng đúng, bởi thế, etc” The number of popular stop words in Vietnamese we retrieved from (vietnamese-stopwords, n.d.) is about 3800 words.

4.2. **Feature keyword extraction**

Users are capable of knowing which documents will be classified into categories, however, they do not know what to do because they don’t know which keywords play the role of classification. Therefore, we created a dictionary that extracts the appropriate keywords to describe the categories. For example, a category includes health articles in which the description is “information and computer, or information and technology”. That is, if a text includes the words “information” and “computer”, this text must belong to the Information Technology category. If the text contains both
"information and technology", the text also belongs to the Information Technology category.

The major difficulty of text classification is the featured space with large dimensions. An and Chen (2005) found that the distance between each pair of data points is almost the same in a large dimensional space. For example, there are three data points \((A, B, C)\) in a space, the distance of: \(d(A, B) = 100.32, d(A, C) = 99.83, \) and \(d(B, C) = 101.23\). It can be said that the data point \(C\) is closer to \(A\) than \(B\). However, Pham and Ta (2017) suggested that keywords should be extracted in the text as unique words, meaning that the words are not repeated and non-existent in the list of stop-words. Based on the method of Pham and Ta (2017), we reduce the dimensionality of the input space in the text-classification problem. Here we extract keywords by taking 30% of the content in a text, which means taking the keywords in the first line in a text. The frequency of the keywords will be sorted by descending weight, we select keywords with higher weight and build a dictionary to store all the keywords that have been extracted from all the text in the file training data.

4.3. Feature vector construction

Before applying any classification model, it is important to transform the text into numeric features called feature vectors. The feature vector is simply a series of numbers. In this paper, we select the bag-of-words model because of its simplicity and popularity in classification, where the frequency of occurrence of each word is used as a feature for training the classifier (Ninh & Nguyen, 2017). The idea of this model is that each word in the text is represented by a vector with zero and not zero, depending on whether the word is in the dictionary or not. If the word is not in the dictionary, that position has a value of zero; If it is in the dictionary, the position receives a value of 1, and depending on the frequency of occurrence of that word it will adjust the frequency of the value at that position.

For example, there are two texts quoted from two newspaper as follows: i) “Messi hiện cùng là cầu thủ nước ngoài giành nhiều chức vô địch nhất. Anh cũng là cầu thủ ghi nhiều bàn nhất trong lịch sử giải đấu, đồng thời là cầu thủ ghi nhiều bàn nhất trong một mùa” and ii) “Trụ sở chính đặt tại Ủy ban chứng khoán tại địa chỉ 164 Trần Quang Khải - Hà Nội; Phòng giao dịch nghiệp vụ đặt tại sân tổng một của Trung tâm giao dịch chứng khoán Hà Nội và Chi nhánh Trung tâm lưu ký tại TP. Hồ Chí Minh đặt ở Số 1 Nam Kỳ Khởi Nghĩa”. Assuming that the storage dictionary list is the following 18 words: [Messi, ủy ban, cầu thủ, nước ngoài, giải đấu, ghi bàn, mùa, giành, trụ sở, chứng khoán, Hà Nội, trung tâm, đặt, lưu ký, giao dịch, lịch sử, sân]. We will create a feature vector with dimension numbers of 18 in each text, with each element representing the corresponding number of words that appear in the text. With these two texts, we will have two the feature vectors: (1) \([1, 0, 3, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]\) and (2) \([0, 1, 0, 0, 0, 0, 1, 2, 2, 1, 0, 2, 0, 1]\). As shown in the vector (1) the word "ủy ban" does not appear in the text so the second position of the dictionary vector length has a value of 0, while the word “cầu thủ” appears three times in the text, so the third position of the vector has a value of 3. After transferring the words into the bag-of-
words, we create the feature vector for each file in the dataset. Each vector has the same length as the number of words in the dictionary.

5. **CLASSIFICATION METHODS**

   After completing the above processing steps, we apply some supervised learning algorithms to solve the text classification: Naïve Bayes and SVM.

5.1. **Naïve Bayes**

   Naïve Bayes is a one of the popular classification methods based on Bayes’ theorem in probability theory to make predictions and classify data. This theorem assumes that the features \( X = \{x_1, x_2, ..., x_n\} \) are probabilistically independent of each other (Kim et al., 2006). According to Bayes’ theorem probability \( P \) is calculated as:

   \[
   P(Y | X) = \frac{P(X | Y)P(Y)}{P(X)}
   \]  
   \[
   (2)
   \]

   For the text classification problem, Bayes’ theorem is stated as:

   \[
   P(C_i | X) = \frac{P(X | C_i)P(C_i)}{P(X)}
   \]  
   \[
   (3)
   \]

   Where \( D \) dataset has been vectorized as \( \vec{x} = x_1, x_2, ..., x_n \). \( C_i \) is the dataset \( D \) of class \( C_i \) with \( i = \{1,2,3,...,n\} \). The attributes \( x_1, x_2, ..., x_n \) are independent probabilities.

   - Conditional independence:

   \[
   P(C_i | X) = P(x_1 | C_i) \times P(x_2 | C_i) \times ... \times P(x_n | C_i) = \prod_{k=1}^{n} P(x_k | C_i)
   \]  
   \[
   (4)
   \]

   - New rules for text classification:

   \[
   X_{map} = \max(P(C_i) \prod_{k=1}^{n} P(x_k | C_i))
   \]  
   \[
   (5)
   \]

   In particular, \( n \) is the vocabulary set of the training set, \( P(C_i) \) is the frequency of documents appearing in the training set, \( P(x_k | C_i) \) depends on the type of data. There are three commonly used types: Gaussian Naïve Bayes, Multinomial Naïve Bayes, and Bernoulli Naïve (Vu, 2018).

   - *Gaussian Naïve Bayes*: Usually applied for continuous data types:

   \[
   P(x_k | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_k - \mu_y)^2}{2\sigma_y^2}\right)
   \]  
   \[
   (6)
   \]
Multinomial Naïve Bayes: Usually used in text categorization, where vector features are measured in bag of words:

\[
P(x_k | y) = \frac{N_{yk} + \alpha}{N_y + d\alpha}
\] (7)

Bernoulli Naïve: Usually applied for binary data types:

\[
P(x_l | y) = P(i | y)x_i + (1 - P(i | y))(1 - x_i)
\] (8)

5.2. Support Vector Machine (SVM)

The SVM was proposed by Cortes and Vapnik (1995). Its use grew significantly in 1995 and still continues to grow. It is still a highly efficient algorithm even by today’s standards. The idea of SVM is to find a hyperplane to divide the space into different domains such that each domain contains a data type. The hyperplane is represented by the function: \(< w.x >= b\) (w, x are vectors). But the problem is that there are many possible hyperplanes. Figure 1 shows three hyperplanes separating two classes, illustrated by circle and square nodes. Which hyperplane should we choose for optimization? The hyperplane separates the two classes \(H_0\) by the following formula:

\[<w.x> + b = 0\] (9)

This hyperplane divides data into two half spaces. The half space of the negative class \(x_l\) satisfies (9a) and the half-space of the positive class \(x_j\) satisfies (9b):

\[<w.x_l> + b \leq -1\] (9a)
\[<w.x_j> + b \geq 1\] (9b)

Figure 2 shows two hyperplanes \(H_1\) and \(H_2\). Hyperplane \(H_1\) passes through the negative points and \(H_2\) passes through the positive points. Both margins are parallel to \(H_0\).

\[H_1 : <w.x> + b = -1\] (10a)
\[H_2 : <w.x> + b = 1\] (10b)

Margin rate m can be calculated by:

\[m = d_- + d_+ = \frac{2}{\|w\|}m = d_- + d_+\] (11)

\(d_-\) is the distance from \(H_1\) to \(H_0\), \(d_+\) is the distance from \(H_2\) to \(H_0\).
\[ d_+ = \frac{|\langle w, x_i \rangle + b|}{\|w\|} = \frac{1}{\|w\|} \]  
\[ d_- = \frac{|\langle w, x_j \rangle + b|}{\|w\|} = \frac{1}{\|w\|} \]  
\[ \|w\| = \langle w, w \rangle = \sqrt{w_1^2 + w_2^2 + \ldots + w_n^2} \] \hspace{1cm} (11c)

In reality, observation, as well as Vladimir (1999) showed that the hyperplane classification is optimal if the hyperplane is separated from a wide margin.

![Figure 1. Hyperplanes](image1)

However, in Figure 2, the data points are given in ideal conditions so it is easy to find the hyperplane \( H_0 \). But in the case of noisy data, it is necessary to loosen the margin conditions by using the variable \( \varepsilon_i \geq 0 \). This is called the soft margin of SVM.

![Figure 2. Margins of hyperplane](image2)
\[
<w_i x_i> + b \leq -1 + \varepsilon_i
\]  \hspace{1cm} (12a)

\[
<w_i x_i> + b \leq -1 - \varepsilon_i
\]  \hspace{1cm} (12b)

The search for the optimal hyperplane solution can be extended in the case of non-linear data by representing the initial X space as F space through a nonlinear mapping function: \( \emptyset: X \rightarrow F, x \rightarrow \emptyset(x) \). The transform function \( \emptyset(x) \) will change non-linear data into distinct linear. However, functions \( \emptyset(x) \) often produce data with a larger dimension than the original dimension. If it is calculated directly, its memory cost and performance would be very expensive, but fortunately the SVM includes some kernel functions developed in the Hilbert-Schmidt theory and Mercer conditions (Courant & Hilbert, 1953; Liu & Xu, 2013). There are three common kernel function as follows:

- **Gaussian kernel function formula**

  \[
  k(x_i, x_j) = \exp \left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right), \sigma(0, +\infty)
  \]  \hspace{1cm} (13)

- **Polynomial kernel function formula**

  \[
  k(x_i, x_j) = \left( <x_i, x_j> + r \right)^d, r > 1, r \in +
  \]  \hspace{1cm} (14)

- **Linear kernel function formula**

  \[
  k(x_i, x_j) = <x_i, x_j>
  \]  \hspace{1cm} (15)

The standard kernel function corresponding to \( k(x_i, x_j) \) is defined as follows:

\[
 k(x_i, x_j) = \frac{k(x_i, x_j)}{\sqrt{k(x_i, x_i)k(x_j, x_j)}}
\]  \hspace{1cm} (16)

### 5.3. Evaluating a classification Model

After completing the classification, we need to evaluate the test dataset to assess the effectiveness of the model. In the classification problem, datasets between classes can be confused, one for another. Assuming that the binary classification problem has positive and negative classes, three parameters are used in the evaluation, as follows:
• Precision: $P = \frac{TP}{TP + FP}$

(17)

• Recall: $R = \frac{TP}{TP + FN}$

(18)

• Measurement: $F1 = 2 \frac{P \times R}{P + R}$

(19)

$TP$ is the dataset of the positive class that is correctly classified as positive, $FP$ is the dataset of the negative class that is incorrectly classified as positive and $FN$ is the dataset of the positive class that is incorrectly classified as negative. For the multi-class problem, the class being considered is viewed as the positive class, while the rest is viewed as the negative class. So, the precision and recall are calculated as follows:

$$Precision = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FP_i)}$$

(20)

$$Recall = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FN_i)}$$

(21)

With $TP_i$, $FP_i$, and $FN_i$, respectively, as $TP$, $FP$, and $FN$ of the corresponding $i$ class. The $F1$ measure is also calculated based on precision and recall, correspondingly. A good classification model is a model with both high precision and recall, i.e. as close to one as possible.

6. EXPERIMENT

We used eight categories from the dataset of the www.vnexpress.net news Website: Technology, business, law, health, world, sports, culture, and society. The eight topics include 21,407 articles. Each article has a different length, but we estimate by taking 400 documents in eight categories the average of each article is about 502 words. We divided the articles into two datasets, a training set and a test set, as shown in Table 1.

In this experiment, we used the Python language and the ViTokenizer library for word segmentation, genism natural language processing and the scikit-learn machine learning library. After the word segmentation process and the removal of stop-words, we used the training dataset to build a dictionary. However, to reduce the number of dimensions in the dictionary we divided news articles so that the training set is smaller than the test set. The training set includes approximately 30% of the total number of texts (details in Table 1). We only scanned up to 30% of the content of each document.
and selected the words that appear the least in 20 articles in a category to store into the dictionary. Specifically, in our experiment, the dictionary received 5000 words, which means that the size of the vector has a dimension of 5000. Before using the Naïve Bayes and SVM classification models for training, we used the bag-of-words model to generate a feature vector for the training set based on the dictionary to count the frequency of appearance of each word in each text. Finally, we used the testing set to evaluate results based on precision, recall and F1 score for Naïve Bayes and SVM classification methods. Here we applied Naïve Bayes with the Gaussian NB and Multinomial probabilities; SVM was implemented with the linear and the RBF kernel functions. The classification results of these four methods are shown in details in Table 2 and its average is shown in Figure 3. In particular, Figure 3 shows that the SVM classification method with Linear and RBF kernel functions has a higher value than the Naïve Bayes probability methods. Vectorization based on the bag-of-words model as an input is suitable for the Multinomial probability model in Naïve Bayes, so the assessments of Naïve Bayes (Multinomial) are much better than Naïve Bayes (Gaussian).

Table 1. News articles from vnexpress.net pages by topic

| Category    | Total | Training set | Test set |
|-------------|-------|--------------|----------|
| Technology  | 1739  | 774          | 965      |
| Business    | 3060  | 983          | 2077     |
| Law         | 2546  | 874          | 1672     |
| Health      | 1784  | 652          | 1132     |
| World       | 3519  | 775          | 2744     |
| Sport       | 2644  | 644          | 2000     |
| Culture     | 2800  | 690          | 2110     |
| Society     | 3315  | 902          | 2413     |
| Total       | 21407 | 6294         | 15113    |

In each type of classification, we choose a classification method of a higher value to compare the rating of each news topic, for SVM we choose RBF kernel and for Naïve Bayes we choose Multinomial as visually shown in Figures 4, 5, and 6. Evaluation of the precision in Figure 4 shows that the Naïve Bayes classification method is higher than SVM in only the Health category. The highest precision belongs to the Sport category, both classifications’ ratings are 99% and the lowest is the Social category in which Naïve Bayes reaches 87% while SVM is about 89%. Evaluation of the recall in Figure 5 shows that the best rating belongs to the Technology category in
which SVM is about 97% and Naïve Bayes is about 96%, but in the World category Naïve Bayes' recall is better than SVM. According to evaluation of the F1 score in Figure 6, the rating of SVM is almost always better than Naïve Bayes.

**Table 2. The results of the measurement of the method of each category**

| Category | SVM          | Naïve Bayes |  |
|----------|--------------|-------------|---|
|          | Kernal: RBF  | Kernal: Linear | Gaussian | Multinomial |
|          | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 | Prec | Rec | F1 |
| Technology | 0.920 | 0.970 | 0.940 | 0.910 | 0.960 | 0.930 | 0.750 | 0.880 | 0.810 | 0.890 | 0.960 | 0.920 |
| Business  | 0.930 | 0.950 | 0.930 | 0.900 | 0.950 | 0.920 | 0.830 | 0.780 | 0.810 | 0.920 | 0.910 | 0.910 |
| Law       | 0.930 | 0.930 | 0.930 | 0.910 | 0.930 | 0.920 | 0.760 | 0.840 | 0.800 | 0.910 | 0.930 | 0.920 |
| Health    | 0.930 | 0.950 | 0.940 | 0.930 | 0.960 | 0.950 | 0.910 | 0.830 | 0.870 | 0.940 | 0.950 | 0.940 |
| World     | 0.980 | 0.930 | 0.950 | 0.970 | 0.930 | 0.950 | 0.920 | 0.910 | 0.920 | 0.950 | 0.940 | 0.940 |
| Sport     | 0.990 | 0.950 | 0.970 | 0.980 | 0.960 | 0.970 | 0.970 | 0.930 | 0.950 | 0.990 | 0.930 | 0.960 |
| Culture   | 0.950 | 0.950 | 0.950 | 0.960 | 0.950 | 0.960 | 0.900 | 0.830 | 0.870 | 0.930 | 0.950 | 0.940 |
| Society   | 0.890 | 0.900 | 0.890 | 0.900 | 0.870 | 0.880 | 0.650 | 0.690 | 0.670 | 0.870 | 0.870 | 0.870 |
| Avg/total | 0.940 | 0.941 | 0.938 | 0.933 | 0.939 | 0.935 | 0.836 | 0.836 | 0.838 | 0.925 | 0.930 | 0.925 |

**Figure 3. Evaluating the classifications’ average measurements**
Figure 4. Evaluating the Precision of SVM and Naïve Bayes

Figure 5. Evaluating the Recall of SVM and Naïve Bayes

Figure 6. Evaluating the F1 score of SVM and Naïve Bayes
We used a test set of 15,113 texts, independent of the training set provided by vnexpress.net pages, which showed better results from using the SVM method as opposed to the Naïve Bayesian. Moreover, we also verified that the dataset of 20,446 independent texts from tuoitre.vn and vnexpress.net pages also show SMV having higher results than the Naïve Bayes (Table 3). This means the SVM text classification method is stable.

| Classification                | Testing dataset: 20,446 | Testing dataset: 15,113 |
|------------------------------|-------------------------|-------------------------|
|                              | Precision | Recall  | F1 score | Precision | Recall  | F1 score |
| Naïve Bayes (Gaussian)       | 0.800     | 0.793   | 0.798    | 0.836     | 0.836   | 0.838    |
| Naïve Bayes (Multinomial)    | 0.905     | 0.913   | 0.906    | 0.925     | 0.930   | 0.925    |
| SVM (RBF)                    | 0.898     | 0.820   | 0.848    | 0.940     | 0.941   | 0.938    |
| SVM (Linear)                 | 0.914     | 0.918   | 0.918    | 0.933     | 0.939   | 0.935    |

Returning to the test dataset of 15,113 texts, when the feature vector of the training set is changed with different lengths, as shown in Table 4, the two SVM and Naïve Bayes classification methods show that the higher the number of vectors, the higher the precision of SVM, but when the dimension is reduced to about 1/2 compared to the initial reduced dimension, the accuracy of SVM and Naïve Bayes (Multinomial)
is reduced and the evaluation of these two methods becomes similar. If its dimension continues to decrease to 1700, the rating of Naïve Bayes (Multinomial) is higher than SVM. This means that SVM is perfectly suitable for large data sets with many features.

7. CONCLUSION AND FUTURE

In this paper we have researched and presented text classification techniques. The experimental part of the text classification problem has a relatively large feature space. Therefore, cooperation between the natural language and dimensional reduction not only helps lower storage space below what it originally was, but also makes the performance time of the classification faster. When comparing Naïve Bayes (Multinomial) and Naïve Bayes (Gaussian), it is clear that the bag-of-words vectorization technique is suitable for research Naïve Bayes (Multinomial) because it always has a much better rating. However, when a large data set requires a lot of features, SVM always has much higher accuracy than Naïve Bayes (Multinomial).

Our experimental results come out better than research results of (Nguyen & Luong, 2006) with the same SVM classification algorithm. In addition, our results are also better than the experimental results of (Phan & Nguyen, 2015) with the same algorithm but different datasets. The experimental dataset of Phan and Nguyen (2015) had 2,114 texts in total, of which 1,000 texts belonged to the training set. Whereas our dataset had a total of 21,407 texts, of which 6,294 texts belong to the training set and the accuracy reached 94%.

We are still continuing to study and improve the data pre-processing techniques for classification by using the TF-IDF feature vector technique and applying word embedded technology, based on word2vec or doc2vec, instead of the bag-of-words method with the hope of adjusting and improving the accuracy in the SVM classification.

REFERENCES

An, J., & Chen, Y. P. P. (2005). Keyword extraction for text categorization. Paper presented at The International Conference on Active Media Technology, Japan.

Cortes, C., & Vapnik, V. (1995). Support vector networks. Machine Learning, 20(3), 273-297.

Courant, R., & Hilbert, D. (1953). Methods of mathematical physics. New Jersey, USA: John Wiley & Sons.

Ehrentraut, C., Ekholm, M., & Tan, H. (2018). Detecting hospital-acquired infections: A document classification approach using Support Vector Machines and gradient tree boosting. Health Informatics Journal, 24, 24-42.

Github. (2017). Vietnamese stopwords. Retrieved from https://github.com/stopwords/vietnamese-stopwords/blob/master/vietnamese-stopwords.txt.
Kim, S., Han, K., Rim, H., & Myaeng, S. (2006). Some effective techniques for Naive Bayes text classification. *Transactions on Knowledge and Data Engineering, 18*(11), 1457-1466.

Leopold, E., & Kinermann, J. (2002). Text categorization with Support Vector Machines. How to represent texts in input space? *Machine Learning, 46*(1-3), 423-444.

Lin, D., Peng, H., & Liu, B. (2006). *Support Vector Machines for text categorization in Chinese question classification*. Paper presented at The International Conference on Web Intelligence, China.

Liu, Z., & Xu, H. (2013). Kernel parameter selection for Support Vector Machines classification. *Journal of Algorithms & Computational Technology, 8*(2), 163-177.

Madge, S., & Bhatt, S. (2015). *Predicting stock price direction using Support Vector Machines*. Retrieved from https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf.

Nguyen, G. L., & Luong, M. T. (2006). Phân loại văn bản tiếng Việt với bộ phân loại vector hỗ trợ SVM. Retrieved from http://ictvietnam.vn/files/_layouts/biznews/uploads/file/Uploaded/admin/CS15012_bai_anh_Linh_Giang.pdf.

Nguyen, S. D., Ngo, H. Q., & Jiamthaphaksin, R. (2016). *State-of-the-art Vietnamese word segmentation*. Paper presented at The International Conference on Science in Information Technology, Indonesia.

Ninh, D. K., & Nguyen, Q. V. (2017). Biểu diễn ngữ cảnh trong khai triển chữ viết tắt dùng tiếp cận học máy. *Tạp chí Khoa học và Công nghệ Đại học Đà Nẵng, 5*(114), 31-35.

Pham, T. V., & Ta, T. M. (2017). *Vietnamese news classification based on BoW with keywords extraction and neural network*. Paper presented at The Asia Pacific Symposium on Intelligent and Evolutionary Systems, Vietnam.

Phan, T. H., & Nguyen, Q. C. (2015). Automatic classification for Vietnamese news. *Advances in Computer Science: An International Journal, 4*(4), 126-132.

Umair, S., & Sharif, M. (2018). Predicting students grades using artificial neural networks and Support Vector Machines. In M. K. Pour (Ed.), *Encyclopedia of Information Science and Technology (4th ed. p. 14)*. Pennsylvania, USA: IGI Global USA.

Vladimir, V. (1999). *The nature of statistical learning theory* (2nd ed.). Berlin, Germany: Springer Publishing.

Vu, T. H. (2018). *Bài 32: Naive Bayes classifier*. Retrieved from https://machinelearn.ingcoban.com/2017/08/08/nbc/.

Xue, D., & Fengxin. (2015). *Research of text categorization model based on random forests*. Paper presented at The IEEE International Conference on Computational Intelligence & Communication Technology, India.