A Vietnamese Sentiment Analysis System Based on Multiple Classifiers with Enhancing Lexicon Features

Bich-Tuyen Nguyen-Thi and Huu-Thanh Duong(✉)

Faculty of Information Technology, Ho Chi Minh City Open University,
97 Vo van Tan, Ward 6, District 3, Ho Chi Minh City, Vietnam
{1551010145tuyen, thanh.dh}@ou.edu.vn

Abstract. Today, a customer is easy to express his opinions about a bought products thanks to accelerated development of social networks and ecommerce websites. These opinions are very useful indicators to evaluate the degree of the customers' real satisfaction. From that, the traders will emerge the strategies and predict the trends to get the directions for their products and businesses in the future. In this paper, we have built a dataset and executed many experiments based on the multiple classifiers with complementing lexicon features to increase the accuracy of sentiment polarities. The experimental section shows good results and proves our approach is reasonable.

Keywords: Machine learning · Text mining · Natural language processing · Sentiment analysis · User behavior

1 Introduction

Sentiment analysis task is a hot trend in natural language processing applied widely in many domains, especially in ecommerce system, this identifies the sentiment polarity between positive and negative comments. Through sentiment, managers of the brand and product can quickly get an overview of customer attitudes towards their brand at a certain time or within a certain period of time. Moreover, sentiment analysis also points to brand or product strengths and weaknesses in the eyes of the customers, which aspects are being appreciated and what are the focal points of negative discussion. In addition to monitor sentiment changes over a long period of time will tell brand health, which helps the brand managers and marketers to re-evaluate the performance and giving directions for the future campaigns. This not only improves the effectiveness of business, but also helps to be more proactive in preventing and handling crisis.

In this paper, we have proposed the sentiment analysis solution based on multiple classifiers with enhancing lexicon features. Our main contribution is to build the Vietnamese dataset from reputation ecommerce websites for sentiment analysis, apply preprocessing techniques for the dataset such as word segmentation, lowercase transformation, punctuation removal and feed more features to the feature vectors such as the adjective phrases, negation and emotional icons replacement. The experiments of

T. Q. Duong et al. (Eds.): INISCOM 2019, LNICST 293, pp. 240–249, 2019.
https://doi.org/10.1007/978-3-030-30149-1_20

various aspects are executed to evaluate the well-known classifiers and various features to find the suitable solution for real-time applications.

In the rest of this paper is organized as follows: Sect. 2 presents the related works, Sect. 3 shows background and our approach. Next, Sect. 4 shows the experimental results. Finally, the conclusions and future works are presented in Sect. 5.

2 Related Works

Sentiment Analysis is a potential field attracted many research groups having studied with various experiences and approaches.

Medhat et al. [1] showed the main approaches for sentiment analysis problem including machine learning, lexicon-based and hybrid approaches. Actually, sentiment analysis is a kind of text classification problem identifying the user's comments as positive or negative polarities, so it can totally apply the machine learning algorithms with the linguistic features to sentiment polarities. About lexicon-based approach relies on the emotional lexicons and is divided into dictionary-based or corpus-based approaches built by statistical or semantic methods to find the emotional polarities. Besides, the hybrid approach combines these two approaches to utilize the advantages and limit the disadvantages of them in the hope of increasing more accuracy.

Although the lexicon-based approach also has drawbacks such as reliability of the lexicons, dependencies on their contexts or languages. But the emotional lexicons are important factors to recognize the customers' emotions in their comments, so it has many lexicon-based studies in particular languages, especially adjective, adjective phrases, verb and verb phrases. The adjective and verb phrases are the essential clues for sentiment analysis problem to enhance the accuracy of the adjective and verb phrases, also build the sentiment wordnet. Trinh et al. [2] proposed the lexicon-based approach for sentiment analysis with facebook data in Vietnamese and built the Vietnamese emotional lexicon dictionary, including noun, verb, adjective, adverb based on the English emotional analysis applied to the Vietnamese language and used support vector machine classifier to identify the emotions. Tran et al. [3] proposed a fuzzy language computation based on Vietnamese linguistic characteristics to provide an effective method for computing the sentiment polarity of verb phrases.

Deep learning is also an trending solution mentioning a lots in the recent years. Vo et al. [4] integrated the advantages of CNN (Convolutional Neural Network) and LSTM (Long ShortTerm Memory) for sentiment analysis with their Vietnamese proposed corpus as comments/reviews in ecommerce websites. Araque et al. [5] enhanced deep learning sentiment analysis with ensemble techniques, including ensemble classifiers and ensemble features.

Our approach has based on the multiple classifiers and complemented the lexicon features. According to the experiments of Duong and Truong Hoang [6], we choose logistic regression, SVM (Suport Vector Machine), random forest, OVO, OVR which are classfiers obtains the best score in current.

Next, it presents the theory background of our approach and the experiments to evaluate the well-known classifiers and also the dataset.

3 Background

3.1 Feature Extraction

Each dimension of comment vectors is *tf* × *idf* weight of a term, where *tf* is the number of the terms appearing in a document, *df* is the number of the documents containing a term, *idf* is the inversion of *df* weight. This weight is widely used in natural language processing because *tf* shows the importance of the term, but if that term also appears many times in other documents, it may be a less meaning word, so incorporating with *idf* to punish that one. Deng et al. [7] executed the exhaustive experiments showing *tf* × *idf* has still gotten the high score in text classification, our approach also chooses this one for dimensions of the feature vectors. This weight is calculated as follows:

$$(tf \times idf)_{w_i} = freq(w_i) log \frac{N}{1 + df} \quad (1)$$

Subject to N is the number of the comments, $freq(w_i)$ is the frequency of w term in the i-th comment. Besides, it also incorporates unigram and bigram for feature vectors.

3.2 The Vietnamese Phrases

Woking in Vietnamese processing will face the first challenge as word segmentation. It's different from English which the words are the tokens divided by the space characters, Vietnamese words may one token or two tokens such as tốt (good), xuất sắc (excellent), tuyệt vời (wonderful), hoàn hảo (perfect). The second one is Part of Speech (POS) tagging which is used to assign parts of speech to each words such as noun, verb, adjective, adverb helps to increase more semantic to texts. They are important problems in natural language processing, this study has used pyvi[1] library for Vietnamese word segmentation (F1 score 0.979) and POS (F1 score 0.925), and relies on them to get the adjective phrases to increase the semantic dimensions for feature vectors. The adjective phrases are the essential indicators in sentiment analysis to determine the degrees of customers' satisfaction. An adjective phrase in Vietnamese has the structure:

$$\mathbf{P1} < center\ adj > \mathbf{P2}$$

The previous (**P1**) and post (**P2**) sub-sections may be lack, but the center adjective is required. **P1** is often the adverbs of complementing for the center adjectives. For example, "*rất tốt*" (very good): "*tốt*" (good) is an adjective and "*rất*" (very) is the adverb of complementing for "tốt". **P2** is often adverbs of degree. For example "đẹp quá" (so beautiful): "*đẹp*" (beautiful) is an adjective and "*quá*" (so) is the adverb of degree. It also may be noun, adjective, verb making more clear the features of the center adjective. For example, "*anh ấy khó thuyết phục*" (it is difficult to convince him): "khó" (difficult) is the adjective and "thuyết phục" (convince) is the verb.

[1] https://pypi.org/project/pyvi/.

The adjective phrases are especially important to distinguish strong satisfied and only satisfied polarities. The special point is to distinguish two of them as the adverb complementing for the adjective to increase the emotion of the adjective word. For example, a user's said: "*tôi rất hài lòng về sản phẩm*" (I'm very pleased about the product) is more satisfied than "*tôi hài lòng về sản phẩm*" (I'm pleased about the product) or "*tôi tạm hài lòng về sản phẩm*" (I'm a little bit pleased about the product), "*sản phẩm đẹp quá*" (the product is so beautiful) is more satisfied than "*sản phẩm đẹp*" (the product is beautiful). Moreover, comments are the short texts, so the study focuses on two forms of the adjective phrases as **P1 <center adj>** and **<center adj> P2**.

3.3 SVM

SVM (Support Vector Machine) is a strong classifier using both regression and classification problem. The main idea is to find a hyperland (calling M_0) to divide the dataset into various groups in the multi-dimensions space. Firstly, SVM is used for the binary problem and the linearly separable dataset, it means M_0 divides the dataset into two groups and gets the same distance with two support vector hyperlands of those two groups (see Fig. 1 is the dashed lines). The support vectors of groups are the data points having the nearest distance to M_0. Clearly, these data points are more important than other ones in finding M_0-. The cost function of M_0 forms as follows:

$$y = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b \tag{2}$$

Where $y_n(\boldsymbol{w}^{\boldsymbol{T}}x_n + b) \geq 1, \forall n = 1, 2, \ldots, N$ (N is the number of data points) and b is the bias. It needs to find $\boldsymbol{w}$ and b have satisfied as below

$$(\boldsymbol{w}, b) = argmin_{\boldsymbol{w},b}\frac{\|\boldsymbol{w}\|_2^2}{2} \tag{3}$$

This is an optimal problem with constraints which can solve by Lagrange function. It means to need find the roots of the following equation, where α_n are Lagrange multipliers $\forall n = 1, 2, \ldots, N$

$$\mathcal{L}(\boldsymbol{w}, b, \alpha) = \frac{\|\boldsymbol{w}\|_2^2}{2} - \sum_{n=1}^{N}\alpha_n\left(y_n\left(\boldsymbol{w}^{\boldsymbol{T}}x_n + b\right) - 1\right) \tag{4}$$

After solving this, the category of a data point is calculated by $f(x) = sign(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b)$.

If the dataset has any noise data points (only nearly separable linear dataset), the set of the data points will be divided into safe, unsafe and wrong data point areas (see Fig. 1). It combines with slack variables (ξ) for this division, slack variable of the i-th data point ($\boldsymbol{d_i}$) is calculated: $\xi_i = \left|\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x_i} + b - y_i\right|$, it means

- $\xi_i = 0 \rightarrow \boldsymbol{d}_i$ is the safe data point (belongs to right category).
- $0 < \xi_i < 1 \rightarrow \boldsymbol{d}_i$ is unsafe data point (still belongs to right category, but in area between M_0 and support vectors hyperland of that category).
- $\xi_i > 1 \rightarrow \boldsymbol{d}_i$ is the wrong data point (belongs to wrong category).

It needs to find $\boldsymbol{w}$ and b of the following equation satisfying $y_i(\boldsymbol{w}^T x_i + b) \geq 1 - \xi_i$

$$(\boldsymbol{w}, b, \xi) = argmin_{\boldsymbol{w},b,\xi} \frac{\|\boldsymbol{w}\|_2^2}{2} + C \sum_{i=1}^{N} \xi_i \tag{5}$$

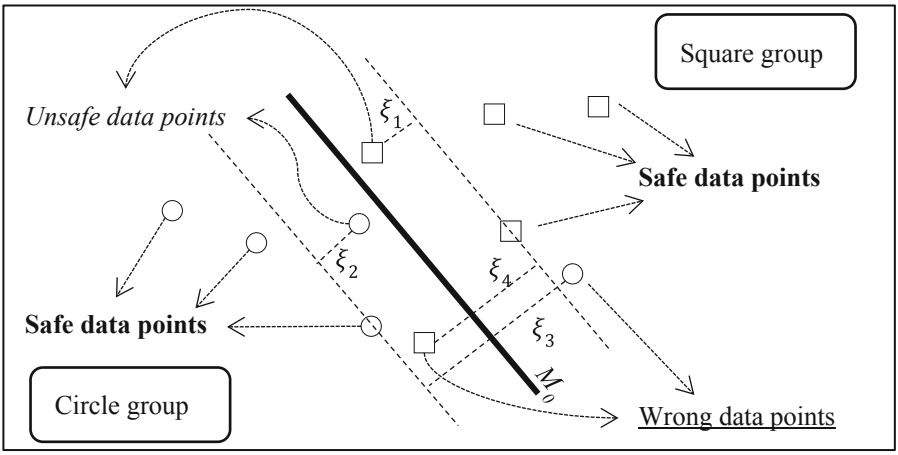

Fig. 1. Illustration for the nearly separable linear dataset.

For the non-linear separable dataset needs to find the transformation to transform the dataset from the non-linear separable space to the linear separable space.

3.4 Logistic Regression

Logistic regression is a statistical approach to determine the relationship between the dependent variable y and a set of independent variables $\boldsymbol{x}$. The prediction of a data point is a probability of each category by logistic function and uses a threshold $\in [0, 1]$ to determine it belongs to that category or not, the form of logistic function is as follows:

$$y = logistic\left(\boldsymbol{w}^T \boldsymbol{x} + b\right) \tag{6}$$

It looks like the linear models which needs to estimate the coefficients $\boldsymbol{w}^T$ and b from training phase. However, the logistic function is a non-linear function, the most often using is sigmod function as follows:

$$f(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{x}}} \tag{7}$$

Although the name contains "regression" word, but it's used more in classification problems. This approach is high score and good performance, easy to implement, so it's widely used in various fields of machine learning.

3.5 Random Forest

Random forest is one of the most used algorithms today applied to both regression and classification problem developed Breiman in 2001. The main idea is to combine the decision trees into a single model, each decision tree is built from a random subset of the features in the training dataset using classification and regression trees (CART) technique. For the original random forest, the label of an unlabeled data point is determined via majority voting. Afterward, there are many studies to boost its performance, our context uses average of probabilistic prediction via scikit-learn[2].

The decision tree is built by splitting the training dataset recursively from the root node, CARD uses Gini Index to measure the impurity of data for the split feature in a decision node, this one is calculated as:

$$G = 1 - \sum_{i=1}^{N} (p_i)^2 \tag{8}$$

Where N is the number of categories, p_i which is probability of a data point in a subset belongs to the i-th category. The split process continues recursively until the decision tree reaches the max deep or can't split subsets anymore.

Random forest is a simple algorithm, obtains great result and avoid the big problem in machine learning as overfitting. However, its main limitation is slow and not effective in real-time prediction when the number of subtrees is large.

3.6 OVO and OVR

OVO (One-vs-One) classifier has got each pairwises of the categories and applied the binary classifiers, the final category of the document is decided by the majority voting of them. So, if having c categories, it decomposes $c(c - 1)/2$ iterations for pairwises of the categories to execute the binary classifier. This is a simple approach and obtains the good experiments, but gets high computational cost and takes a long time for training.

Another approach has the less computational cost than OVO, namely OVR (One-vs-Rest) classifier, this approach has executed c binary classifiers of c categories, each i-th one defined whether the documents belong c_i category or a probability of the document belongs to that category. The final category based on the probability.

In our experiments, we use linear SVM for each iterations of OVO and OVR.

4 Approach and Experiments

Our dataset has collected 7200 comments on the ecommerce websites such as tiki.vn, thegioididong.com, fptshop.com.vn grouped into StrongSatisfied (2380 comments), Satisfied (2440 comments), UnSatisfied (2380 comments) sentiment polarities manually. This dataset shows more challenges than the original sentiment analysis problem

[2] https://scikit-learn.org/stable/.

which only contains two sentiment polarities as Positive and Negative. For evaluation, the dataset is preprocessed and complemented the useful lexicons, the comments is vectorized with each dimension of feature vectors is *tf* × *idf* weight of terms or phrases, they are feeded to the multiple classifiers for sentiment polarities.

Based on a comparative evaluation of preprocessing techniques of Symeonidis et al. [8] by measuring their accuracy in twitter sentiment classification, we also applied some preprocessing techniques including lowercase transformation, number removal, punctuation removal to improve the accuracy. Next, we have complemented the adjective phrases by using the label of the part of speech to the comments and the adjective structure to indicate the adjective phrases. Then, it replaces negation lexicons and emotional icons, a list of negation lexcions is manually determined such as "không" (not), "chưa" (not yet), "chẳng" (not) and two lists of positive lexicons and negative lexicions based on Vietnamese SentiWordNet built by Vu et al. [9] and our collected dataset. For those ones, each negation lexicon which follows as a positive lexicon is replaced by "not_positive" lexicon or a negative lexicon is replaced by "not_negative" lexicon. Similarly, it also prepares a list of emotional positive and negative icons, positive icons are replaced by "positive" lexicion and negative icons are replaced by "negative" lexicon.

In order to prove our approaches, we executes various experiments with the well-known classifiers. Since the dataset hasn't been much big enough yet, so we have used *k*-fold cross-validation method to evaluate, this divides the dataset into *k* subsets and executes *k* iterations. For each iteration, one of the subsets is used for the testing data, the remaining ones are used for the training data. It uses *k* as 5 for the experiments and all of the experiments are executed on Macbook Pro (2017) 2.8 GHz Intel Core i7, RAM 16 GB 2133 MHz. The first experiment executes with the original features, Table 1 shows the average of F1 scores, average of training and testing time (in seconds) when executing the multiple classifiers without preprocessing and feeding the proposed feature indicators. Where F1 score is the weighted average of precision and recall, this reaches the best score at 1 and worst score at 0.

Table 1. Executing the multiple classifiers with *tf* × *idf* weight of features.

Classifiers	Parameters	Average of F1 score	Training time (s)	Testing time (s)
Logistic regression	*multi_class = ovr; solver = lbfgs*	0.801	**0.210**	**0.001**
	multi_class = multinomial; solver = lbfgs	0.809	0.523	**0.001**
Random forest	*subtrees = 10*	0.734	0.346	0.005
	subtrees = 50	0.787	1.683	0.022
	subtrees = 80	0.792	2.684	0.035
	subtrees = 100	0.802	3.343	0.043
SVM	*kernel = linear; C = 1e5*	0.784	4.432	0.913
	kernel = rbf; gamma = auto	0.766	4.752	0.826
OVR	*linear SVM*	0.804	10.088	2.300
OVO	*linear SVM*	**0.817**	4.299	2.964

Table 2. Executing the multiple classifiers with preprocessing the dataset and implementing the lexicon features.

Classifiers	Parameters	Average of F1 score	Training time (s)	Testing time (s)
Logistic regression	*multi_class = ovr; solver = lbfgs*	0.829	1.289	0.003
	multi_class = multinomial; solver = lbfgs	0.837	2.032	**0.002**
Random forest	*subtrees = 10*	0.765	**1.070**	0.007
	subtrees = 50	0.807	5.377	0.034
	subtrees = 80	0.812	8.468	0.060
	subtrees = 100	0.818	10.694	0.078
SVM	*kernel = linear; C = 1e5*	0.839	9.750	2.103
	kernel = rbf; gamma = auto	0.847	10.067	2.097
OVR	*linear SVM*	0.842	20.246	4.936
OVO	*linear SVM*	**0.850**	8.887	6.324

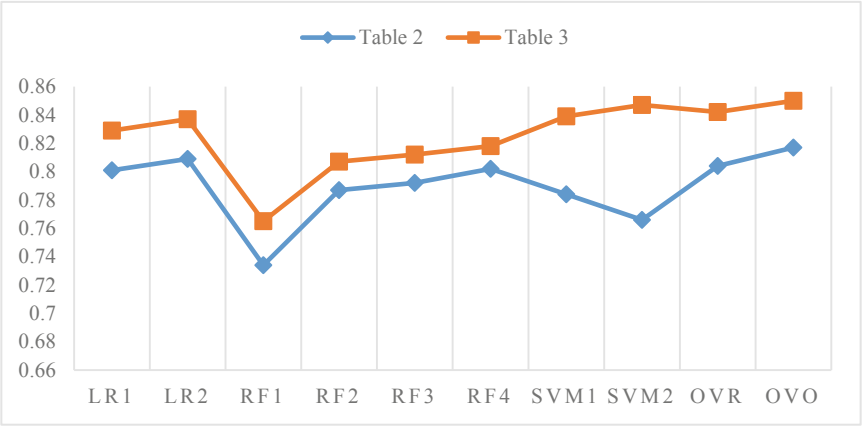

Fig. 2. The F1 scores stats of the experimental results

Table 2 shows the results in the same strategies as Table 1, but applying the proprocessing techniques and complementing the useful lexicons. Observing the F1 score stats of Fig. 2 and the time stats of Table 2, the results are improved and OVO classifier still obtains the highest score for now, but training and testing time are also rather high, this is a big disadvantage for the bigger dataset and difficult to apply the real-time prediction. In other hands, logistic regression classifier obtains a little bit

lower than OVO classifier, but training and testing time are much faster, so it's a nice option to deploy in the real-time applications.

5 Conclusions and the Future Works

In this paper, we have built the sentiment dataset and used the multiple classifiers approach with enhancing lexicon features to the comment's vectors to improve the accuracy of sentiment polarities and executed various experiments to prove our approach and suggest the suitable solution for the real-time prediction. In the experiment, it has obtained the good results.

In future works, we will build a Vietnamese sentiment dictionary, investigate misspelling words, wrong grammars of the sentences, synonymy words, antonymous words, slang, also more for emotional icons/symbols, negation. Besides, the dataset will continue to broaden in more fields.

References

1. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: a survey. Ain Shams Eng. J. **5**, 1093–1113 (2014). https://doi.org/10.1016/j.asej.2014.04.011
2. Trinh, S., Nguyen, L., Vo, M., Do, P.: Lexicon-based sentiment analysis of Facebook comments in vietnamese language. In: Król, D., Madeyski, L., Nguyen, N.T. (eds.) Recent Developments in Intelligent Information and Database Systems. SCI, vol. 642, pp. 263–276. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31277-4_23
3. Tran, T.K., Phan, T.T.: Computing sentiment scores of adjective phrases for vietnamese. In: Sombattheera, C., Stolzenburg, F., Lin, F., Nayak, A. (eds.) MIWAI 2016. LNCS (LNAI), vol. 10053, pp. 288–296. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49397-8_25
4. Vo, Q.-H., Nguyen, H.-T., Le, B., Nguyen, M.-L.: Multi-channel LSTM-CNN model for Vietnamese sentiment analysis. In: 2017 9th International Conference on Knowledge and Systems Engineering (KSE), pp 24–29. IEEE, Hue (2017)
5. Araque, O., Corcuera-Platas, I., Sánchez-Rada, J.F., Iglesias, C.A.: Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Expert Syst. Appl. **77**, 236–246 (2017). https://doi.org/10.1016/j.eswa.2017.02.002
6. Duong, H.-T., Truong Hoang, V.: A survey on the multiple classifier for new benchmark dataset of vietnamese news classification. In: 2019 11th International Conference on Knowledge and Smart Technology (KST). IEEE, Phuket, pp 23–28 (2019)
7. Deng, X., Li, Y., Weng, J., Zhang, J.: Feature selection for text classification: a review. Multimed. Tools Appl. **78**, 3797–3816 (2019). https://doi.org/10.1007/s11042-018-6083-5
8. Symeonidis, S., Effrosynidis, D., Arampatzis, A.: A comparative evaluation of pre-processing techniques and their interactions for Twitter sentiment analysis. Expert Syst. Appl. **110**, 298–310 (2018). https://doi.org/10.1016/j.eswa.2018.06.022
9. Vu, X.-S., Song, H.-J., Park, S.-B.: Building a vietnamese SentiWordNet using vietnamese electronic dictionary and string kernel. In: Kim, Y.S., Kang, B.H., Richards, D. (eds.) PKAW 2014. LNCS (LNAI), vol. 8863, pp. 223–235. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13332-4_18

10. Mirończuk, M.M., Protasiewicz, J.: A recent overview of the state-of-the-art elements of text classification. J. Expert. Syst. Appl. **106**, 36–54 (2018)
11. Hussein, D.M.E.-D.M.: A survey on sentiment analysis challenges. J. King Saud Univ.-Eng. Sci. **30**, 330–338 (2018). https://doi.org/10.1016/j.jksues.2016.04.002
12. Devika, M.D., Sunitha, C., Ganesh, A.: Sentiment analysis: a comparative study on different approaches. Procedia Comput. Sci. **87**, 44–49 (2016). https://doi.org/10.1016/j.procs.2016.05.124
13. Ban, D.V., Thung, H.V.: Ngữ pháp tiếng Việt, "Vietnamese Grammar". Vietnam Education Publisher, Hanoi (1998)