A Lightweight Ensemble Method for Sentiment Classification Task

Minh Quang Nhat Pham, Tran The Trung
FPT Technology Research Institute (FTRI)
8 Ton That Thuyet, My Dinh 2, Nam Tu Liem, Ha Noi, Viet Nam
{minhpqn, trungtt}@fpt.edu.vn

***Abstract*—In this report, we describe our system for sentiment classification task at VLSP 2016 evaluation campaign. Sentiment classification is to classify documents, articles, or product reviews into classes the reflect their sentiments about some subject matters. We propose a lightweight ensemble method for the task. Our ensemble model combines three classification models, namely Random Forests, Support Vector Machines, and Multinomial Naive Bayes by the majority voting strategy. In feature extraction, we use *n*-gram features ($n = 1, 2, 3$) with TF-IDF weighting scheme to train three classifiers. Our system obtained 69.6% accuracy over all classes, and 70% F1-score on average.**

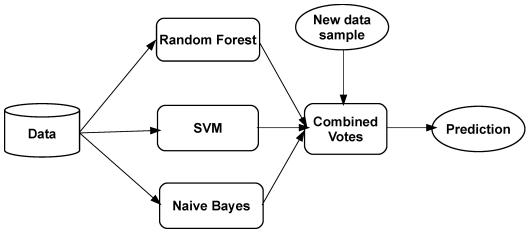

Fig. 1. System architecture

I. Introduction

Sentiment analysis is the task of mining opinions in sentences, users' reviews, or articles according their sentiment towards some subject matters such as products [1]. Sentiment analysis is useful in many business intelligence applications. For instance, sentiments of product reviews give us a quick summary of users' opinions about products. The task has received extensive attentions of natural-language-processing and data mining communities since early 2000s.

For Vietnamese language, VLSP 2016 (Vietnamese Language and Speech Processing) evaluation campaign is the first effort to provide the benchmark data and to perform a systematic comparison between Vietnamese sentiment analysis systems. This year, the scope of the campaign is polarity classification in which participant systems need to classify Vietnamese reviews/documents into one of three categories: "positive", "negative", or "neutral."

In this report, we describe the sentiment classification system which we used to produce our submission for the VLSP 2016 evaluation campaign. We proposed a lightweight ensemble method which uses the majority voting strategy to combine three classification models trained with Random Forests algorithm [2], Support Vector Machine (SVM) [3] and Multinominal Naive Bayes [4]. We also give an analysis on errors that our system made and discuss some directions for further improvements.

The rest of the paper is organized as follows. In section II, we describe our participant system. In section III, we present our evaluation results on the test set. In section IV, we give the error analysis. Finally, section V gives conclusions about the work.

II. System Description

Figure 1 shows the architecture of our participant system. After preprocessing data, we train three classifiers using the training data set and combine three classifiers with majority voting strategy. We use three algorithms: Random Forests [2], Support Vector Machines (SVM) [3] with linear kernel, and Multinomial Naive Bayes (MNB) [4]. Random Forests is an ensemble method that combine multiple tree predictors. Random Forests algorithm have been shown to be an effective method for classification problems [2]. We choose SVM and Multinomial Naive Bayes with the same reason. That is because previous work has shown that they are very effective for sentiment classification task [5]. In our system, since the number of features is much larger than the number of training examples and by some preliminary experiments, we decided to use linear kernel in SVM.

To train three classification models, we use TF-IDF weighted *n*-grams features, in which $n = 1, 2, 3$. In our implementation, we use python scikit-learn library [6] for both extracting features and training models.

An important step to improve the prediction accuracy of a machine-learning system is model selection. We tune parameters for classification models by using grid search and evaluating F1-score of 5 folds in 5-fold cross validation. Parameter tuning is done on the training set provided by the VLSP 2016 evaluation campaign. More specifically, for Random Forest algorithm, we tune the number of trees in the forest. For the linear SVM model, we select the parameter *C*.

In preprocessing step, we just perform word segmentation by using tool vnTokenizer [7]. To produce our submitted results, we re-used the word-segmented data provided by the organizer of VLSP 2016 evaluation campaign.

TABLE I
DISTRIBUTION OF CLASSES IN DATA SETS

	positive	negative	neutra	Totall
Training set	1700	1700	1700	5100
Test set	350	350	350	1050

TABLE II
EVALUATION RESULTS OF OUR SYSTEM

	precision	recall	f1-score	support
positive	0.75	0.71	0.73	350
negative	0.72	0.67	0.70	350
neutral	0.63	0.71	0.67	350
avg/total	0.70	0.70	0.70	1050

TABLE III
CONFUSION MATRIX 1. ROWS ARE NUMBERS OF INSTANCES WITH ACTUAL LABELS. COLUMNS ARE NUMBER OF INSTANCES WITH PREDICTED LABELS

Predicted Actual	negative	neutral	positive	all
negative	235	75	40	350
neutral	59	247	44	350
positive	32	69	249	350
All	326	391	333	1050

III. EVALUATION RESULTS

A. Data sets

We use the training data provided by the organizer to train our ensemble model. The trained model is used to predict the labels for examples in the test data. Table I shows the distribution of classes in the training and test data. The table indicated that the data sets are perfectly balanced.

B. Evaluation measures

VLSP 2016 evaluation campaign uses accuracy, precision, recall, and F1 score as evaluation measures. In this report, we report precision, recall, F1 score for all three classes in the data set.

C. Results

Table II shows our evaluation results (precision, recall, f1 score for each class) on the test set. We obtained 69.62% accuracy.

The results indicated that in the evaluation data, the class "positive" is the easiest class and the class "neutral" is the most difficult to predict.

IV. ERROR ANALYSIS

A. Confusion Matrix

Table III shows the confusion matrix between gold standard labels and predicted labels. Table IV is the confusion matrix with normalized values (probabilities).

We can see that our system predicted "neutral" with the highest number of instances. However the precision for the class "neutral" is lowest. We hypothesize that it is because "neutral" reviews may contain both positive and negative opinions and "neutral" reviews are cases where even annotators find they are difficult to decide whether it is positive or negative.

TABLE IV
CONFUSION MATRIX 2. CONFUSION MATRIX WITH PROBABILITY VALUES

Predicted Actual	negative	neutral	positive	all
negative	0.22	0.07	0.04	0.33
neutral	0.06	0.23	0.04	0.33
positive	0.03	0.06	0.24	0.33
All	0.31	0.37	0.32	1.0

id	Gold	Predicted	Review
1	POS	NEG	Nước_miếng rơi rơi ... . nước_mắt rưng_rưng !
2	NEG	NEU	Nếu vẫn thế_này thì R.I.P Apple . Đây là cơ_hội cho Samsung có những bước_ngoặt lớn
3	NEG	POS	Mình dùng suface_pro_3 thấy chạy tốt mà đẹp hơn con này
4	POS	NEG	ZenFone_3 không làm tui thất_vọng . Đang trên tay , ngất_ngây con gà_tây !
5	POS	NEU	con này dùng hay phết gprs chát_chít vèo_vèo . ai có game nào hay post lên đây đi . mình tìm được mấy cái game nhung chơi chán quá
6	NEG	NEU	Mắc kinh , cơ_mà kính đeo vào cái phá được kỷ_lục Olympic thì 120.000 $ ngta cũng mua
7	POS	NEU	Mình đang dùng 1 con này rất ngon , mạng 1-2 vạch , dùng nó kéo lên 5 vạch , mạng miếc nhanh hơn hẳn (dĩ_nhiên cũng chỉ hòm_hòm khi lại gần router thôi) . Mình lên công_ty , chỗ mình ngồi wifi công_ty chỉ 2-3 vạch là kịch , mà phải đúng chỗ ngồi , còn đi khuất tí là rất khó vào mạng . Từ hồi có nó có_thể lỉnh vào 1 góc , chui xuống gầm bàn xem phim , chơi game ... mà thằng này lại khá mạnh , mình đi cách nó hơn 10m , xuyên qua 1 bức tường (toalet) mà vẫn 4 vạch . Nhưng nó có 1 nhược_điểm đang cố tìm cách khắc_phục là nó chỉ nhớ 1 mạng . Nghĩa_là nếu cài mạng ở công_ty thì khi về nhà hay đi công_tác , gặp mạng khác phải reset nó và cài lại từ đầu .

Fig. 2. Some miss-classified instances made by our system

B. Some phenomena

For each class (positive, negative, neutral), we get samples of 30 examples of that class, for which the system failed to predict their gold labels. In total, we analyze 90 examples to investigate phenomena that our system could not capture. We found some phenomena that are difficult as follows. Figure 2 shows some instances extracted from our sample.

In some cases, we need to use common sense knowledge to infer the sentiment of a review. Examples 1 and 2 in the Figure 2 belong to this kind. Solving this problem is difficult because inference using common sense knowledge is still a challenging problem in natural language processing [8].

We find that an user may use comparison to express her/his opinion. In example 3, the user compare a product with its rival product to convey her/his negative opinion.

One of problems in analyzing product reviews on social medias is dealing with slang, rare words, *teen codes* or *trolling* comments of users. Example 5 and 6 are two examples of that phenomena.

Another phenomenon we found by error analysis is that a review may contain the main part that discusses about the product in the question and other parts discuss related things. Example 7 is an example of that phenomenon. In that case, the sentiment of the main part decides the sentiment of the whole

review. Other parts of review may give noises to the machine learning algorithm. In such cases, we need to identify the main phrase or sentence that shows the sentiment about the product.

V. Conclusion

In this report, we present our participant system for the sentiment classification task at VLSP 2016 evaluation campaign. We adopted a very lightweight ensemble method that combined three classification models trained on the training data using Random Forests, Support Vector Machines, and Multinomial Naive Bayes. The method is lightweight and easy to implement with the library scikit-learn. Despite its simplicity, the system obtained 0.70 F1-score on average over all classes. We also analyse errors made by our system and discuss some difficulties of the task.

References

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, ser. EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 79–86. [Online]. Available: http://dx.doi.org/10.3115/1118693.1118704

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

[4] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752. Citeseer, 1998, pp. 41–48.

[5] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ser. ACL '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 90–94. [Online]. Available: http://dl.acm.org/citation.cfm?id=2390665.2390688

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] L. Hong Phuong, N. Thi Minh Huyen, A. Roussanaly, and H. T. Vinh, *A Hybrid Approach to Word Segmentation of Vietnamese Texts*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 240–249. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88282-4_23

[8] P. LoBue and A. Yates, "Types of common-sense knowledge needed for recognizing textual entailment," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 329–334. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002736.2002805