Variants of Long Short-Term Memory for Sentiment Analysis on Vietnamese Students' Feedback Corpus

Vu Duc Nguyen*, Kiet Van Nguyen† and Ngan Luu-Thuy Nguyen‡
University of Information Technology
Vietnam National University – Ho Chi Minh City, Vietnam
Email: *14521100@gm.uit.edu.vn, †kietnv@uit.edu.vn, ‡ngannlt@uit.edu.vn

***Abstract*—The Long Short-Term Memory (LSTM) and Dependency Tree-LSTM have shown the state-of-the-art results for the sentiment analysis task for the English language. Despite many studies of LSTM approach, there are no studies of Dependency Tree-LSTM approach for Vietnamese sentiment analysis. In this paper, we conducted experiments with LSTM, Dependency Tree-LSTM, and our proposed models on Vietnamese Students' Feedback Corpus. According to the experimental results, the Dependency Tree-LSTM were not better than the LSTM model. However, when combining final hidden state vectors of LSTM and Dependency Tree-LSTM models with a Support Vector Machine classifier, we achieved the F1-score of 90.2%, which is higher than the performance of the LSTM model.**

I. Introduction

In recent years, sentiment analysis is one of the emerging research topics in the field of Natural Language Processing (NLP). Sentiment analysis is used as a tool to analyze and evaluate users' opinion automatically. There are three primary levels of sentiment analysis which are sentence-level, document-level, and aspect-level [1]. Sentence-level performs the task that classifies a sentence into negative, positive, or neutral class. Document-level is used to determine the whole document whether that document sentiment is negative, positive, or neutral. Aspect-level is used to determine sentiment polarity for each aspect of the entity mentioned in a review.

Our study concentrate sentiment analysis at sentence-level on the Vietnamese language. At the first stage in studying of this field, Kieu and Pham [2] addressed sentiment analysis at the sentence-level and built a rule-based system using the GATE framework [3]. They conducted experiments on a corpus of computer products reviews and obtained 61.16% of precision, 64.62% of recall. Duyen *et al.* [4] investigated linguistics feature aspects for sentiment analysis on a corpus of hotel reviews.

At the later stage in sentiment analysis at sentence-level on the Vietnamese language, many studies have been done with the objective of sentiment analysis using Deep Learning models. Pham *et al.* [5] used word embedding technique [6] and BI model [7] to propose Learning Semantic Representations for Rating Comments (LSRRCs) models. They demonstrated that their proposed LSRRCs performs slightly better than neural network combined bag-of-words model and neural network combined word vector averaging model on rating Vietnamese comments. Besides, Vo *et al.* [8] built a Vietnamese Sentiment corpus containing 17,500 reviews from Vietnamese e-commercial sites. They introduced a novel model which integrates the advantages of Convolutional Neural Network (CNN) [9] and Long Short-Term Memory (LSTM) [10]. Their model gave better results than CNN, LSTM and Support Vector Machine (SVM) [11] on Vietnamese Sentiment corpus.

In our work, we concentrate sentiment analysis on the vital domain which is education. At present, education is being devoted much attention in Vietnam. The feedback from learners about educational aspects are crucial language resources. Except for Vo *et al.* [12] researched sentiment analysis on the educational domain, there are few works on that domain. We realized the analysis of learners' opinion are objectives of both the field of education and natural language processing.

In short, our main problem in this paper is as follows. Given a feedback sentence, which usually contains opinion about a specific aspect (e.g. lecturer, facility), the goal is to determine whether that sentence expresses positive, neutral or negative. For example, the sentence "Cách truyền đạt kiến thức dễ hiểu, sự hài hước vui vẻ." (means: "The way to present knowledge is easy to understand, humorous and fun.") is annotated with positive polarity. Another example, the sentence "Thầy thường xuyên đi dạy trễ." (means: "Lecturer usually went to the classroom late.") is assigned to the negative polarity.

In this paper, we study with the aim of sentiment analysis on students' feedback using Deep Learning models. LSTM model is effective for sentiment analysis at sentence-level, such as Vo *et al.* [8] studied on Vietnamese Sentiment corpus, and Tai *et al.* [13] conducted experiments on Stanford Sentiment Treebank. Besides, there are many studies for Vietnamese dependency parsing [14] [15] [16] which has inspired us to applied dependency struct into Vietnamese sentiment analysis area. Therefore, we used LSTM [10] and Dependency Tree-LSTM [13] model as primary methods. Moreover, we proposed the method which combining outputs of LSTM and Dependency Tree-LSTM model for the SVM classifier. Lastly but not least, we analyzed and compared methods using deep neural network models with traditional classification models.

II. Related Work

Sentiment analysis on Vietnamese Students' Feedback Corpus is connected to two different areas of research: NLP and education. We describe remarkable points in two subsections below.

978-1-5386-6113-0/18/$31.00 ©2018 IEEE

A. Sentiment Analysis in NLP Research

Naive Bayes (NB) [17], Maximum Entropy [18], and SVM are effective sentiment classifiers when using bag-of-words features on Movie Review Dataset [19]. Exploiting efficient syntactic structures over the sentence can improve the sentiment models better. The evidence is Matsumoto *et al.*'s study [20] used text mining techniques to extract frequent word sub-sequences and dependency sub-trees from sentences in a document dataset and use them as features of SVM. They conducted experiments on the corpora of movie reviews with the same evaluation criteria as [19], [21]. Their classifiers had achieved the best results at the time published.

The Stanford Sentiment Treebank (SST) corpus for sentiment analysis with five labels is introduced, and Recursive Neural Models are proposed by Socher *et al.* [22]. That work has been motivation for many research about SST and variants of Recursive Neural Models later.

B. Sentiment Analysis in Education Research

Kechaou *et al.* [23] applied sentiment analysis to examine the nature and the structure of web forums and e-learning blogs. Their corpus contains 1,000 positive texts and 1,000 negative ones. They achieved the best result by applying Information Gain feature [24] on hybrid classifier Hidden Markov Models [25] and SVM. Altrabsheh *et al.* [26] presented methods for collecting feedback from students. They used features such as term presence and frequency, n-grams, part-of-speech, syntax, and negation on NB and SVM classifiers.

III. Vietnamese Students' Feedback Corpus

In this section, we introduce the Vietnamese Students' Feedback Corpus (UIT-VSFC) which is used for evaluating sentiment analysis in this paper. UIT-VSFC corpus is publicly available for research [27], consisting of 16,175 sentences from students' feedback which are human-annotated for two different tasks: sentiment-based and topic-based classifications. In this paper, we focus sentiment-based task. UIT-VSFC corpus has the inter-annotator agreement measure of 91% which is significant for our work. The baseline model with the Maximum Entropy classifier achieved 88% in F1-score on the test set. Table I shows three example sentences taken from the corpus with three sentiment polarity labels.

TABLE I
Annotated Examples With Sentiment

No.	Sentence	Sentiment
1	Thầy thường_xuyên đi_dạy trễ. (Lecturer usually went to the classroom late.)	Negative
2	Không có gì đặc_biệt. (Nothing special.)	Neutral
3	Cách truyền_đạt kiến_thức dễ_hiểu, sự hài_hước vui_vẻ. (The way to present knowledge is easy to understand, humorous and fun.)	Positive

Prominent points of the sentiment corpus are presented as follows. Both positive and negative polarities appear in the dataset with the significant proportion of over 45% each; conversely, neutral polarity accounts for a small percentage (see Figure 1(a)).

Sentence lengths are dependent on the writing style of students and the content. Thus, analysis of the dataset according to sentence length provides shallow observations. Figure 1 shows the normalized label distributions at each sentence length. Many short sentences are positive; long sentences tend to be negative. The neutral class rarely appears in the dataset.

The dataset is categorized into three classes based on length of sentence (b), (c) and (d) which have approximate quantities to each other (see Figure 1). The aim of categorizing is to allow easy observation of the ratios of negative labels to positive labels and analyze performance between models in Section V. The numbers of sentences belonging to class (b), (c), and (d) are 4,809, 6,056, and 5,310 respectively. Another observation is that the ratios of negative labels to positive labels are 0.56, 0.73, and 1.83 for class (b), (c) and (d), respectively. Lastly, distributions of sentiment values for (b), (c) and (d) show that longer sentences are negative.

Besides, the UIT-VSFC corpus has the average sentence length of 11. The vocabulary size is 4,239, which is five times smaller than the Stanford Sentiment Treebank' [22].

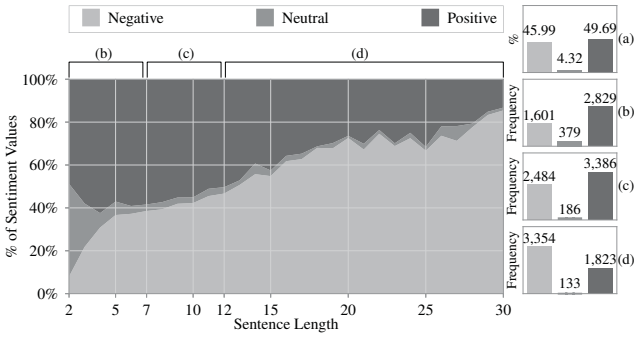


Fig. 1. Normalized histogram of sentiment annotations at each sentence length (the point 30 denote sentences length greater than or equal to 30). The subfigure (a) denote for an overall dataset and (b), (c), and (d) indicates for sets of sentences have length fall in half-open intervals $[2, 7)$, $[7, 12)$, and $[12, +\infty)$ respectively.

Lastly, we provide properties of training, development, and test sets. It is essential for training and testing the sentiment models in Section V. In UIT-VSFC corpus, the portions of training, development, and test sets are approximately 70%, 10%, and 20% of the whole corpus respectively. Table II shows information about the number of sentiment labels of the training, development, and test datasets.

TABLE II
Number Of Sentiment Labels Of Training, Development, And Test Sets

	Number of labels			
	Negative	**Neutral**	**Positive**	**Overall**
Train	5,325	458	5,643	11,426
Dev	705	73	805	1,583
Test	1,409	167	1,590	3,166

IV. Methods

A. Long Short-Term Memory

Recurrent neural networks or RNNs [28] are a family of neural networks for processing an input sequence of arbitrary length of vectors $x_1, ..., x_\tau$ (τ denotes for the last time step). RNNs are suitable for tasks that require predicting the output of time step t based on its previous one $t - 1$. The network

typically uses hidden state h_t as a kind of lossy summary to represent output at time step t. Commonly, hidden state $h_t \in \mathbb{R}^d$ is an affine transformation followed by a pointwise nonlinear function:

$$h_t = \sigma(Wx_t + Uh_{t-1} + b) \tag{1}$$

In practice, the traditional RNNs are still hard to apply for long sequences because of the vanishing and exploding gradient problems [29].

The Long Short-Term Memory (LSTM) architecture [10] is a essential variation of RNNs, has been proposed for dealing with vanishing and exploding gradient problems. The critical difference in LSTM architecture is the *memory cell* that can store and get information over long-term dependencies. In this subsection, we describe the LSTM architecture [30] that is one of the LSTM variations that have been proposed.

As an extension of ordinary RNNs, LSTM recurrent networks have *LSTM cells*. Each cell has the same inputs and outputs as a vanilla recurrent network, but has more parameters and a system of gating units that controls the flow of information. We define d as the memory dimension of the LSTM, the *input gate* $i_t \in \mathbb{R}^d$ (for time step t) fall in the closed interval $[0, 1]$ via the logistic sigmoid function:

$$i_t = \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\right), \tag{2}$$

where $x_t \in \mathbb{R}^n$ is the input at the current time step, h_{t-1} is the previously hidden layer vector, and $W^{(i)}, U^{(i)}, b^{(i)}$ are input weight, recurrent weight and bias for the input gate respectively . Besides, the extent to which the previous memory cell is forgotten is controlled by a *forget gate* f_t:

$$f_t = \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\right), \tag{3}$$

where $W^{(f)}, U^{(f)}, b^{(f)}$ are forget weight, recurrent weight and bias into the LSTM cell. The information extracted from current input x_t is *output gate* u_t:

$$u_t = \tanh\left(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}\right). \tag{4}$$

Thus, the most important component is the *memory cell* c_t. The *input gate* i_t decides which information from *output gate* u_t to store in *memory cell* c_t as follows with a condition *forget gate* f_t:

$$c_t = i_t \odot u_t + f_t \odot c_{t-1}, \tag{5}$$

where the operation $\odot$ denotes the element-wise vector product. Lastly, the output h_t of the LSTM cell can also be obtained by o_t deciding which information is read from the memory cell c_t:

$$o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\right), \tag{6}$$

$$h_t = o_t \odot \tanh\left(c_t\right). \tag{7}$$

B. Dependency Tree-LSTM

Tai *et al.* [13] have proposed a natural extension to the LSTM architecture which is named *Dependency Tree-LSTM*. The common LSTM architecture composes information at each LSTM cell from only one child unit at previous time step. As a generalization of vanilla LSTM, the Dependency Tree-LSTM allows richer network topologies where each LSTM unit can incorporate information from multiple child units.

Each Dependency Tree-LSTM unit (indexed by j), contains an input i_j and an output gates o_j, a memory cell c_j and hidden state h_j. As in standard LSTM units, gates perform tasks that control the flow of information. Additionally, instead of a single forget gate, Dependency Tree-LSTM unit contains a forget gate f_{jk} for each child unit k. This allows the Dependency Tree-LSTM unit to selectively incorporate information from the child nodes.

Given a tree, let $C(j)$ denote the set of children of node j. The Dependency Tree-LSTM unit is expressed as the following recursive formula [13]:

$$\tilde{h}_j = \sum_{k \in C(j)} h_k, \tag{8}$$

$$i_j = \sigma\left(W^{(i)}x_j + U^{(i)}\tilde{h}_j + b^{(i)}\right), \tag{9}$$

$$f_{jk} = \sigma\left(W^{(f)}x_j + U^{(f)}h_k + b^{(f)}\right), \tag{10}$$

$$o_j = \sigma\left(W^{(o)}x_j + U^{(o)}\tilde{h}_j + b^{(o)}\right), \tag{11}$$

$$u_j = \tanh\left(W^{(u)}x_j + U^{(u)}\tilde{h}_j + b^{(u)}\right), \tag{12}$$

$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k, \tag{13}$$

$$h_j = o_j \odot \tanh\left(c_j\right), \tag{14}$$

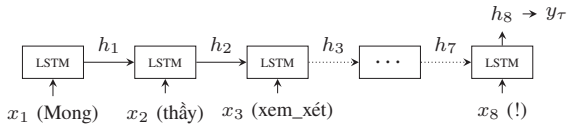
where in Eq. 10, $k \in C\left(j\right)$.

C. Sentiment Models

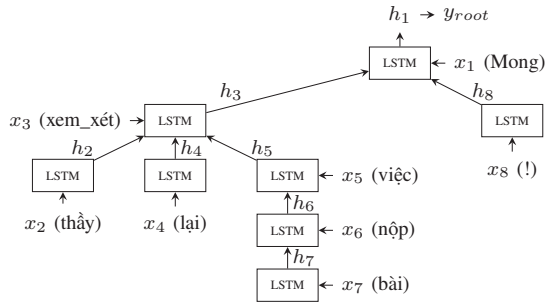
Both standard LSTM unit and Dependency Tree-LSTM unit take an input vector $x_j \in \mathbb{R}^n$. In our implementations, x_j is a vector representation of a word in a sentence. The input word at each node depends on the structure of the network. Figure 2 shows the chain structure and the tree structure of the example sentence "Mong thầy xem_xét lại việc nộp bài!" (means: "Hoping lecturer reconsider submission of assignments!") with branching factor.

We construct sentiment classification models over the corpus in which each sentence has only one label on the root node. Therefore, the Dependency Tree-LSTM model predicts the label $\hat{y}_{root}$ from a discrete set of classes Y for the root node. In the standard LSTM model we want to predict the label $\hat{y}_\tau$ for the last node in a sequence.

We use $\hat{y}$ to indicate that a result holds if we let $\hat{y} = \hat{y}_{root}$ (for Dependency Tree-LSTM) or $\hat{y} = \hat{y}_\tau$ (for LSTM). Similarly, we use h to denote that a result holds if we let $h = h_{root}$ or $h = h_\tau$. For each sentence, we use a softmax classifier to predict the label $\hat{y}$ given the input x observed over the whole



(a) A vanilla LSTM model.



(b) A Dependency Tree-LSTM model over a dependency tree.

Fig. 2. Examples for applying the vanilla LSTM and Dependency Tree-LSTM models. The sentence "Mong thầy xem_xét lại việc nộp bài!" means "Hoping lecturer reconsider submission of assignments!".

sentence. The classifier computes via the hidden state h as follows [13]:

$$\hat{p}_\theta\left(y \mid \{x\}\right) = \text{softmax}(W^{(s)}h + b^{(s)}), \quad (15)$$

$$\hat{y} = \arg\max_y \hat{p}_\theta\left(y \mid \{x\}\right). \quad (16)$$

The cost function is a negative log-likelihood function of the true class labels $y^{(k)}$ of a sample k [13]:

$$J(\theta) = -\frac{1}{m}\sum_{k=1}^{m} \log \hat{p}_\theta\left(y^{(k)} \mid \{x\}^{(k)}\right) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \quad (17)$$

where m is the number of samples in the training set, and λ is an L2 regularization hyper-parameter.

D. Proposed Method

We use final hidden states of LSTM and Dependency Tree-LSTM models as features for training the SVM classifier. In the subsection above which describes sentiment models, the final hidden state vectors of LSTM model, and Dependency Tree-LSTM model is h_τ, and h_{root} respectively. For each sample in the dataset, the input vector $x_{feature}$ of the SVM classifier is computed in three ways which temporarily called L-SVM, D-SVM, and LD-SVM respectively as follow (the operation $\oplus$ denotes the concatenation operator):

- $x_{feature} = h_\tau$,
- $x_{feature} = h_{root}$,
- $x_{feature} = h_\tau \oplus h_{root}$.

V. Experiments

We evaluated LSTM, Dependency Tree-LSTM (DT-LSTM), L-SVM, D-SVM, and LD-SVM models on UIT-VSFC corpus for the sentiment analysis task mentioned in Section III. Besides, we compared them with commonly used methods that use bag-of-words features Naive Bayes, and Support Vector Machine with unigram and bigram features. We abbreviated these methods with BiNB, and BiSVM, respectively. We ran the experiment 100 times. For each run of the experiment, deep neural network models achieved the slight difference result because word representations are initialized randomly. Therefore, we used the average result to compare the performance of models.

A. Data Pre-processing

The data pre-processing is a pipeline process of word segmentation, POS tagging, and dependency parsing. We used the Vietnamese processsing tool VnCoreNLP [16] for word segmentation, POS tagging, and dependency parsing.

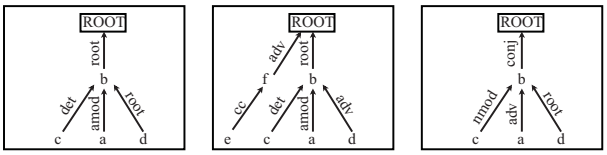


Fig. 3. Three examples illustrate dependency tree do not satisfy the criterion of dependency tree structure.

After the automatic dependency parsing phase, with VnCoreNLP [16], we analyzed structures of the output dependency trees of all sentences in UIT-VSFC corpus. Among 11,426 dependency trees in training set, there are 341 cases which do not satisfy the criteria of a dependency tree structure. We used rules to detect these errors as follows:

- (a) More than one *roots* relation,
- (b) More than one nodes have the arcs connected to *ROOT*,
- (c) The *root* dependency is not between *ROOT* (the head) and the *main verb* (the dependent).

Examples of the three (a), (b), and (c) are shown in Figure 3. We got rid of 341 sentences above from the training set for the Dependency Tree-LSTM model.

B. Hyper-parameters and Training

The hyper-parameters for our models were tuned on the development set, except for the NB classifier.

1) Bag-of-words Models: Word bi-gram features were used for the baseline models: NB, and SVM. For the SVM classifier, we used (soft-margin) linear Support Vector Machine, the loss function used was hinge loss with L2 regularization, and α value was 0.0001.

2) LSTM and Dependency Tree-LSTM Models: Both models were trained using AdaGrad [30] with learning rates of {0.1, 0.01, 0.001}, L2 regularization strengths of {1e-4, 1e-5, 1e-6} and a minibatch size of 20. Word representations were updated during training with a learning rate of {0.1, 0.01, 0.001}. Word vector dimension sizes of {50, 100, 300} were tested. For each word vector dimension, we have chosen memory vector dimension equal half of word vector dimension. Lastly, we have chosen the word vector dimension

size of 300. The memory vector dimension size was 168 (as Tai *et al.* [13] have used) instead of 150. According to the experimental results, LSTM model was trained with the learning rate of 0.05 and word representations' learning rate 0.1; the Dependency Tree-LSTM model was trained with the learning rate of 0.003 and word representations' learning rate 0.005 gave the best result. Both models were trained for five epochs.

3) Our proposed Models: The kernel used was *sigmoid* for SVM classifier.

4) Evaluation Metrics: We used *accuracy* and, *F1-score* metric to evaluate the performance of models. The average F1-score of the model was computed based on the F1-score of three classes negative, neutral, and positive via the number of classes:

$$\text{Average F1-score} = \frac{\sum_{c}^{C} N_c \cdot \text{F1-score}_c}{\sum_{c}^{C} N_c}, \qquad (18)$$

where C is set of classes, F1-score_c denotes the F1-score of class c, and N_c denotes the quantity of c.

C. Experimental results

TABLE III
AVERAGE EVALUATION FOR SENTIMENT CLASSIFICATION MODELS ON TEST SET (%)

Model	Negative	Neutral	Positive	F1-score	Accuracy
BiNB	89.01	2.35	90.39	85.13	87.37
BiSVM	90.35	32.67	90.97	87.62	88.21
LSTM	92.13	41.29	92.67	89.72	90.25
DT-LSTM	91.57	30.99	92.18	88.68	89.67
L-SVM	92.20	42.42	92.73	89.84	90.34
D-SVM	91.62	35.39	92.20	88.94	89.72
LD-SVM	**92.52**	**43.37**	**93.06**	**90.20**	**90.74**

Our results were summarized in Table III. The LD-SVM obtained the highest performance with an F1-score of 90.20% and an accuracy of 90.74%, followed by the L-SVM, LSTM, Dependency Tree-LSTM, D-SVM, BiSVM, and BiNB.

D. Discussion

TABLE IV
DISTRIBUTION FOR EACH SENTIMENT CLASS OF THREE SUBSETS OF TEST SET

Subset	Sentence lengths	Negative	Neutral	Positive	Total
A	$[2, 7)$	306 (09.67%)	89 (02.81%)	593 (18.73%)	988 (31.21%)
B	$[7, 12)$	452 (14.27%)	43 (01.36%)	641 (20.25%)	1136 (35.88%)
C	$[12, +\infty)$	651 (20.56%)	35 (01.11%)	356 (11.24%)	1042 (32.91%)

We divided the test set into three test subsets A, B, and C according to sentence lengths to analyze the performance of models in detailed. The test subsets A, B, and C consisted of short, medium, and long sentences respectively. The details of three test subsets can be seen in Table IV.

TABLE V
AVERAGE EVALUATION FOR SENTIMENT CLASSIFICATION MODELS ON TEST SUBSET A (%)

Model	Negative	Neutral	Positive	F1-score	Accuracy
BiNB	87.33	4.35	91.78	82.53	86.23
BiSVM	87.42	42.73	91.78	86.01	86.46
LSTM	91.61	**56.14**	**93.87**	**89.77**	**90.12**
DT-LSTM	91.08	42.08	93.63	88.20	89.40
L-SVM	91.64	47.39	91.92	89.09	89.65
D-SVM	91.33	34.72	91.40	87.92	88.97
LD-SVM	**92.02**	46.68	92.18	89.34	89.98

1) Comparison of models over short sentences: The LSTM mode was better than the rest models on short sentences by an F1-score of 89.77% and an accuracy of 90.12% (see Table V). All our proposed models had the lower results than LSTM, and DT-LSTM model.

TABLE VI
AVERAGE EVALUATION FOR SENTIMENT CLASSIFICATION MODELS ON TEST SUBSET B (%)

Model	Negative	Neutral	Positive	F1-score	Accuracy
BiNB	86.94	0.00	91.59	86.27	87.85
BiSVM	89.15	21.13	92.28	88.34	88.97
LSTM	90.41	24.70	**93.45**	89.64	90.27
DT-LSTM	89.82	22.30	92.93	89.02	89.86
L-SVM	91.92	46.52	92.48	89.80	90.14
D-SVM	91.02	40.02	91.76	88.69	89.27
LD-SVM	**92.22**	**48.10**	92.87	**90.21**	**90.58**

2) Comparison of models over medium and long sentences: Table VI shows the results over medium sentences. The LSTM model had the result of F1-score of 89.64%, which is lower than LD-SVM by F1-score of 90.21%. According to the accuracy metric, both L-SVM and D-SVM still achieved a result lower than LSTM, and DT-LSTM respectively.

TABLE VII
AVERAGE EVALUATION FOR SENTIMENT CLASSIFICATION MODELS ON TEST SUBSET C (%)

Model	Negative	Neutral	Positive	F1-score	Accuracy
BiNB	91.26	0.00	85.54	86.24	87.91
BiSVM	92.49	12.97	87.16	88.00	89.04
LSTM	**93.58**	11.76	89.23	89.35	90.35
DT-LSTM	93.05	7.96	88.31	88.57	89.73
L-SVM	93.09	30.14	93.69	90.56	91.20
D-SVM	92.58	29.83	93.37	90.16	90.92
LD-SVM	93.36	**32.37**	**94.05**	**90.97**	**91.63**

Table VII shows the results over long sentences. The LD-SVM model obtained the highest F1-score of 90.97% and accuracy of 91.63%. All our proposed models had the better results than LSTM and DT-LSTM model over long sentences.

3) The impact of unbalanced training dataset on models efficiency: BiNB model completely cannot predict the neutral class, as can be seen in Table III. All methods did not well solve unbalanced training dataset problem. Therefore, we assumed that when the balanced training set for the neutral sentiment. The number of neutral opinion in training set is 446 (not counting for samples which do not satisfy criteria of a dependency tree structure), and we trained models with the sub-training sets with 1,338 (446×3) sentences (the negative and positive sentences were sampled from the training set). We conducted an experiment with this assumption over 100 runs for classification models, and average results are shown in Table VIII.

TABLE VIII
AVERAGE EVALUATION FOR CLASSIFICATION MODELS OVER 100 RUNS ON TEST SET WITH BALANCED TRAINING SETS ASSUMPTION(%)

Model	Negative	Neutral	Positive	F1-score	Accuracy
BiNB	82.69	27.83	**87.05**	81.99	**80.94**
BiSVM	81.24	29.22	85.54	80.65	77.48
LSTM	81.89	30.69	86.20	81.35	78.60
DT-LSTM	81.49	30.89	86.83	81.50	78.33
L-SVM	82.09	30.62	85.93	81.30	78.25
D-SVM	80.23	29.13	86.00	80.43	76.22
LD-SVM	**83.00**	**31.77**	86.84	**82.23**	79.13

We evaluated models which were trained from sub-training sets on the test set. In this case, the Dependency Tree-LSTM model had the average F1-score of 30.89% for neutral class, which is slightly better the LSTM model with the F1-score of 30.69%. And we can see the performance of Dependency Tree-LSTM which were slightly better than the LSTM model.

The LD-SVM model obtained the highest mean F1-score of negative and neutral sentiment, as can be seen in Table VIII. Amazingly, the Naive Bayes model obtained the highest mean accuracy, but this model was unbalanced between classes sentiment. To summarize, the effective of LD-SVM model was better than for the rest model with the F1-score of 82.23%.

VI. CONCLUSION AND FUTURE WORK

We introduced the UIT-VSFC corpus. We presented the LSTM model and Dependency Tree-LSTM and applying them to the UIT-VSFC corpus. We proposed the method which combines final hidden states of LSTM and Dependency Tree-LSTM model for SVM classifier. The best model for UIT-VSFC corpus with average F1-score 90.2% and accuracy 90.7%. And, the Dependency Tree-LSTM was not better than the LSTM model in our experiments.

We analyzed the performance of the models according to sentence length and the impact of an unbalanced training dataset on the performance of models. According to analyzing results, the LSTM model had good performance over short sentences, and the LD-SVM model was better than others model over medium and long sentences.

In future work, we are going to use Vietnamese pretrained word vector for deep neural networks on UIT-VSFC corpus. Besides, we are also planning to conduct experiments with variants of Tree-Structured Long Short-Term Memory Networks. Furthermore, we are planning to evaluate LD-SVM for sentiment analysis on others copurs.

ACKNOWLEDGEMENTS

This research is funded by the University of Information Technology - Vietnam National University Ho Chi Minh City under grant number D1-2017-06.

REFERENCES

[1] B. Liu, "Opinion Mining and Sentiment Analysis", in *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, 459–526.

[2] B. T. Kieu and S. B. Pham, "Sentiment Analysis for Vietnamese", in *2010 Second International Conference on Knowledge and Systems Engineering (KSE)*, 2010, 152–157.

[3] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: A framework and graphical development environment for robust NLP tools and applications", in *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[4] N. T. Duyen, N. X. Bach, and T. M. Phuong, "An empirical study on sentiment analysis for Vietnamese", in *2014 International Conference on Advanced Technologies for Communications (ATC 2014)*, 2014, 309–314.

[5] D. H. Pham, A. C. Le, and T. K. C. Le, "Learning Semantic Representations for Rating Vietnamese Comments", in *2016 Eighth International Conference on Knowledge and Systems Engineering (KSE)*, 2016, 193–198.

[6] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality", in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13, 2013, 3111–3119.

[7] K. M. Hermann and P. Blunsom, "Multilingual Models for Compositional Distributed Semantics", *CoRR*, vol. abs/1404.4641, 2014. arXiv: 1404.4641.

[8] Q. H. Vo, H. T. Nguyen, B. Le, and M. L. Nguyen, "Multi-channel LSTM-CNN model for Vietnamese sentiment analysis", in *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, 2017, 24–29.

[9] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences", *CoRR*, vol. abs/1404.2188, 2014. arXiv: 1404.2188.

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, 1735–1780, 1997.

[11] C. Cortes and V. Vapnik, "Support-Vector Networks", *Machine Learning*, vol. 20, no. 3, 273–297, 1995.

[12] H. Vo, H. C. Lam, D. Dung Nguyen, and N. Huynh Tuong, "Topic classification and sentiment analysis for Vietnamese education survey system", vol. 6, 27–34, 2016.

[13] K. S. Tai, R. Socher, and C. D. Manning, "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks", in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2015, 1556–1566.

[14] K. V. Nguyen and N. L.-T. Nguyen, "Vietnamese transition-based dependency parsing with supertag features", in *2016 Eighth International Conference on Knowledge and Systems Engineering (KSE)*, 2016, 175–180.

[15] D. Q. Nguyen, M. Dras, and M. Johnson, "An empirical study for Vietnamese dependency parsing", *CoRR*, vol. abs/1611.00995, 2016. arXiv: 1611.00995.

[16] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, "VnCoreNLP: A Vietnamese Natural Language Processing Toolkit", *CoRR*, vol. abs/1801.01331, 2018. arXiv: 1801.01331.

[17] I. Rish, "An empirical study of the naive Bayes classifier", in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, 2001, 41–46.

[18] A. L. Berger, S. D. Pietra, and V. J. D. Pietra, "A Maximum Entropy Approach to Natural Language Processing", *Computational Linguistics*, vol. 22, 39–71, 1996.

[19] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques", in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2002, 79–86.

[20] S. Matsumoto, H. Takamura, and M. Okumura, "Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees", in *Advances in Knowledge Discovery and Data Mining*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, 301–311.

[21] B. Pang and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts", in *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ser. ACL '04, Association for Computational Linguistics, 2004.

[22] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2013, 1631–1642.

[23] Z. Kechaou, M. B. Ammar, and A. M. Alimi, "Improving e-learning with sentiment analysis of users' opinions", in *2011 IEEE Global Engineering Education Conference (EDUCON)*, 2011, 1032–1038.

[24] Y. Yang and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization", in *Proceedings of the Fourteenth International Conference on Machine Learning*, ser. ICML '97, Morgan Kaufmann Publishers Inc., 1997, 412–420.

[25] Z. Kechaou, A. Wali, M. B. Ammar, and A. M. Alimi, "Novel Hybrid Method for Sentiment Classification of Movie Reviews", in *Proceedings of the 2010 International Conference on Data Mining*, CSREA Press, 2010, 415–421.

[26] N. Altrabsheh, M. Gaber, and M. Cocea, "SA-E: Sentiment Analysis for Education", vol. 255, 353–362, 2013.

[27] K. V. Nguyen, V. D. Nguyen, P. X.-V. Nguyen, T. T.-H. Truong, and N. L.-T. Nguyen, "UIT-VSFC: Vietnamese Students' Feedback Corpus for Sentiment Analysis", in *2018 10th International Conference on Knowledge and Systems Engineering (KSE) (KSE'18)*.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back Propagating Errors", vol. 323, 533–536, 1986.

[29] S. Hochreiter, "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, 107–116, 1998.

[30] W. Zaremba and I. Sutskever, "Learning to Execute", *CoRR*, vol. abs/1410.4615, 2014.