Extracting Aspects in Product Reviews of Vietnamese e-Commerce Websites

Ngoc Truong[1†] and Truong-Thang Nguyen[††],

Institute of Information Technology, Vietnam Academy of Science and Technology, Hanoi, Vietnam

Thai-Le Luong[††],

University of Transport and Communications, Hanoi, Vietnam

Abstract
In earlier times, there are many studies to extract from customers' product reviews on social networks, e-commerce websites, etc. However, in the Vietnamese market with different characteristics of culture and language have only a few studies published. Thus, in this paper, we present a deep convolutional neural network (D-CNN) architecture using features and linguistic patterns of the Vietnamese language for the aspect extraction task. The archived results are acceptable with F1 scores for three domain *book*, *female fashion,* and *mobile accessories* are 76.92%, 71.74%, and 74.54%, respectively.

Key words:
Aspect extract; product review; sentiment analysis; customer profile

1. Introduction

In the recent decade, technical developments in Nature Language Processing (NLP) as Deep Learning (DL) and computing ability supported researchers to personalize customer profiles. In which, catching main interested factors or aspects of products from customers who bought and reviewed on e-commerce websites is necessary to work.

For clearly, let consider a product domain (for example, book or fashion domain) referring to a set of product entities $P = \{p_1, p_2, \dots, p_{|P|}\}$, each *entity* $p \in P$ has a review set $C_p = \{c_1, c_2, \dots, c_{|C_p|}\}$. A review corpus $c \in C_p$ is a document that contains a review of a customer for the product p and can include several sentences express opinions of some aspects. *Aspects* of an entity $p \in P$ are elements or properties of p, denoted $\{A_i\}$ with $i = 1, \dots, k$, where k is the number of aspects of the product. The aspects usually are terms which are expressed in words or a phrases, for instance, in the book domain data has some aspect word such as '*nội dung*' or '*giấy*' or '*tác giả*' (respectively '*content*' or '*paper*' or '*author*' in English). Moreover, each aspect has its variations, for example, '*bìa*', '*bìa sách*' and '*mẫu mã*' also similar to '*hình thức ngoài*' ('*appearance'* or '*book cover'*). Thus, they are extracted, then grouped into the same catalog. Then can be used that information to recommend for other people who have the same concern with the high persuasive ability.

Many studies have attempted to solve this problem for English text, but these are still little for less popular languages such as Vietnamese. Then in this paper, we used the Deep Learning (DL) technique that is D-CNN to solve the aspect extraction task from Vietnamese reviews. We collected data from three domains included *book*, *female fashion*, and *mobile accessories* domain on some popular e-commerce websites such as Tiki, Lazada, and Shopee.

This paper is organized in sections as follows: the next section summarizes the literature methods and especially for the DL approach of aspect extraction. Section 3 describes some background framework for our architectural model in section 4. Section 4 explains the architectural model, dataset, features, and rules for our experiment; and the results for three domains in section 5. Finally, section 6 concludes the work and next enhancement direction to improve and apply for our recommender system.

2. Related work

Aspect extraction is one of three tasks on the Aspect-based sentiment analysis and studied during the last decade because of its applications such as online user's views and comments. This task had some approaches as using sequential learning (sequential labeling) technique are Hidden Markov and Conditional Random Field models [1], frequently count [2], or topic modeling [3], phrase dependency parsing [4], or dependency-based propagation [5].

In the few last years, several emerged approaches towards deep learning as using deep bidirectional LSTMs (Long-

Manuscript received September 5, 2020
Manuscript revised September 20, 2020
https://doi.org/10.22937/IJCSNS.2020.20.11.11

short Term Memory) for joint extraction of opinion entities and the IS-FORM and IS-ABOUT relationships to connect the entities, Katiyar and Cardie [6]. Li [7] also used the LSTM-based method and then improved on [8] that used history attention and selective transformation. Zhang et al. [9] presented a method that used an architectural model combined between LSTM and CNN, used a context gate to encode the relationship of syntax-based interaction between words in the same context. Wang et al. [10] [11] co-extracted the aspects and opinion terms based on a joint model integrating RNN and Conditional Random Fields (CRF) and improved by using CMLA. Besides, Zhang et al. [12] extended a CRF model using a neural network to jointly extract aspects and corresponding sentiments by using a CRF variant to replace original discrete features with continuous word embeddings. Also, Yin et al. [13] learned word embedding by a dependency path connecting words and used more embedding features considered linear context and dependency context information for CRF-based aspect extraction. A presentation learning method of Poria et al. [14] used word embedding and a deep convolutional neural network combined with linguistic patterns.

Additionally, an unsupervised approach of He et al. [15] used an attention-based model that utilized the attention mechanism to focus on the aspect-related words, while de-emphasizing aspect-irrelevant words during the learning of aspect embedding. Xiong et al. [16] also used an attention-based model to learn a feature representation of contexts, used both aspect phrase embedding and context embedding to learn a deep feature subspace metric for K-means clustering.

In particular, the aspect extraction task for the Vietnamese language has distinctive characteristics about syntax, structure, expression manner. There are few works related, Le et al. [17] proposed a semi-supervised learning GK-LDA and using a dictionary to extract better for noun-phrases. Another, a sequence labeling scheme associated with bidirectional recurrent neural networks (BRNN) and CRF to extract opinion targets and detect its sentiment simultaneously, was proposed by Mai et al. [18]. They collected and constructed specifically for the smartphone domain and outperformed CRF with hand-designed features. Besides, related studies use the CNN model can be mentioned as Vo et al. [19] combined CNN and LSTM to create a multi-channel model and Chi et al. [20] used CNN architecture for aspect detection.

3. Background on Deep CNN

3.1. Deep neural network

A deep neural network (DNN) is a simple composite of unsupervised models such as restricted Boltzmann machines (RBMs), where each RBM's hidden layer servers as the visible layer for the next RBM. An RBM is a bipartite graph comprising of two layers of neurons: a visible and a hidden layer; connections between neurons in the same layer are not allowed.

When a multi-layer system is trained, needs to compute the gradient of the total energy function E with concerning weights in all the layers. To learn these weights and maximize the global energy function, therefore, need to use the approximate maximum likelihood contrastive divergence approach. This method employs each training sample to initialize the visible layer. Next, it uses the Gibbs sampling algorithm to update the hidden layer and then reconstruct the visible layer consecutively, until convergence occurs. Each visible neuron is assumed to be a sample from a normal distribution. The continuous state $\widehat{h_j}$ of the hidden neuron j, with bias b_j, is a weighted sum over all continuous visible neurons $\boldsymbol{v}$:

$$\hat{h}_j = b_j + \sum_i v_i w_{ij} \quad (1)$$

where w_{ij} is the weight of connection from the visible neuron v_i to the hidden neuron j. The binary state h_j of the hidden neuron can be defined by a sigmoid activation function:

$$h_j = \frac{1}{1+e^{-\hat{h}_j}} \quad (2)$$

Similarly, at the next iteration, the continuous state of each visible neuron v_i is reconstructed. Here, to determine the state of the visible neuron i, with bias b'_i, as a random sample from the normal distribution where the mean is a weighted sum over all binary hidden neurons:

$$v_i = b'_i + \sum_j h_i w_{ij} \quad (3)$$

where w_{ij} is the weight of connection from the visible neuron i to the hidden one j. This continuous state is a random sample from a normal distribution $N(v_i, \sigma)$, where σ is the variance of all visible neurons. Unlike hidden neurons, in a Gaussian RBM the visible ones can take continuous values.

Then, the weights are updated as the difference between the original data $\boldsymbol{v}_{data}$ and reconstructed visible layer $\boldsymbol{v}_{recon}$:

$$\Delta w_{ij} = \alpha(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon}) \quad (4)$$

where α is the learning rate and $\langle v_i h_j \rangle$ is the expected frequency with which the visible neuron i and the hidden neuron j are active together, when the visible vectors are sampled from the training set and the hidden neurons are calculated according to (1) –(3) , after some k iterations.

Finally, the energy of a DNN can be determined from the final layer (the one before the output layer) as:

$$\mathrm{E} = -\sum_{i,j} v_i h_j w_{ij} \quad (5)$$

To extend the deep neural network to a deep CNN, one simply partitions the hidden layer into Z groups. Each of the Z groups is associated with an $n_x \times n_y$ filter, where n_x is the height of the kernel and n_y is the width of the kernel. Assume that the input has dimensions $L_X \times L_Y$, which in our case is given by L_X words in the sentence and L_Y features, such as word embedding, of each word. Then the convolution will result in a hidden layer of Z groups, each of dimension $(L_X - n_x + 1) \times (L_Y - n_y + 1)$.

The learned weights of these kernels are shared among all hidden neurons in a particular group. The energy function of the layer l is now a sum over the energy of individual blocks:

$$\mathrm{E}^l = -\sum_{z=1}^{Z} \sum_{i,j}^{(L_X - n_x + 1),\ (L_Y - n_y + 1)} \sum_{r,s}^{n_x, n_y} v_{i+r-1, j+s-1} h_{ij}^z w_{rs}^l \quad (6)$$

3.2. Training CNN for sequential data

A special training algorithm is suitable for sequential, proposed by Collobert et al. [21]. We will describe as follows:

The algorithm trains the neural network by back-propagation in order to maximize the likelihood of training sentences. Consider the network parameter θ, h_y is the output score for the likelihood of an input x to have the tag y. Then, the probability to assign the label y to x is calculated as

$$p(y|x,\theta) = \frac{\exp(h_y)}{\sum_j \exp(h_j)} \quad (7)$$

Define the logadd operation as

$$\operatorname*{logadd}_i h_i = \log \sum_i \exp h_i \quad (8)$$

then for a training example, the log-likelihood becomes

$$\log p(y|x,\theta) = h_y - \operatorname*{logadd}_i h_i \quad (9)$$

In aspect term extraction, the terms can be organized as chunks and are also often surrounded by opinion terms. Hence, it is important to consider sentence structure on a whole in order to obtain additional clues. Let it be given that there are T tokens in a sentence and y is the tag sequence while $h_{t,i}$ is the network score for the t-th tag having i-th tag. We introduce $A_{i,j}$ transition score from moving tag i to tag j. Then, the score tag for the sentence c to have the tag path y is defined by

$$c(x,y,\theta) = \sum_{t=1}^{T} (h_{t,y_t} + \mathrm{A}_{y_{t-1}, y_t}) \quad (10)$$

This formula represents the tag path probability over all possible paths. Now, from (8) we can write the log-likelihood

$$\log p(y|x,\theta) = c(x,y,\theta) - \operatorname*{logadd}_{p,j} c(x,y,\theta) \quad (11)$$

The number of tag paths has exponential growth. However, using dynamic programming techniques, one can compute in polynomial time the score for all paths that end in a given tag. Let y_t^k denote all paths that end with the tag k at the token t. Then, using recursion, we obtain

$$\begin{aligned} \delta_t(k) = & \operatorname*{logadd}_{p \in y_t^k} c(x,y,\theta) \\ & = h_{t,k} + \operatorname*{logadd}_j \delta_{t-1}(j) + A_{j,k} \end{aligned} \quad (12)$$

For the sake of brevity, we shall not delve into details of the recursive procedure. The next equation gives the logadd for all the paths to the token T

$$\operatorname*{logadd}_{p,y} c(x,y,\theta) = \operatorname*{logadd}_i \delta_T(i) \quad (13)$$

Using these equations can maximize the likelihood of (11) all training pairs. For inference, need to find the best tag path using the Viterbi algorithm; e.g., need to find the best tag path that minimizes the sentence score (10)

4. Our experiment

4.1. Preprocessing

We collected the set of raw data from some e-commerce popular websites such as Shopee, Tiki and Lazada and then handled with some preprocessing steps:

- De-noise and remove irrelevant documents (for example, advertisement), special character, icon.
- Use some regular expressions to standardize similar errors, such as syntax, stick words.
- Convert the text to lowercase. Remove punctuation and additional white space.
- Normalize acronym words, teen-code.
- Tokenize and POS tag (part-of-speech tag) based on VnCoreNLP [22].

4.2. Method

We define the issue as follows:

Input: sets of review documents C_p

Output: Aspect set $A = \{A_1, A_2, \dots A_p \dots, A_{|P|}\}$, where A_p is aspect set of product entity *p.*

Solved method: D-CNN (deep convolutional neural network).

Our architecture is inspired by Poria et al. [14] for expecting aspects. The network contained one input layer, two pairs of convolution layer – max-pool layer continued, and a fully connected layer with softmax output. The first convolution layer has 100 feature maps with filter size 2, the second convolution layer has 50 feature maps with filter size 3, the stride is 1. The max-pooling layers followed each convolution layer have the pool size is 2. The penultimate layer uses dropout regularization with a constraint on L2-norms of the weight vectors, with 30 epochs. The output of each convolution layer is computed using a non-linear function, hyperbolic tangent.

The features of an aspect term based on its surrounding words, and because specific Vietnamese review documents are usually short, hence, we selected a window of 5 words around each word in a sentence. The local features of that window were considered to be features of the middle word.

Besides, we used word embedding to train corpora, also used some additional features and rules to increase the accuracy, described in section 4.4. The CNN creates local features around each word and then combining into a global feature vector. The kernel size of two convolution layers had the dimensionality $L_X \times L_Y$ (mentioned section 3), is 3×300 and 2×300, the input layer is 30×300, where 30 is the maximum number of words in a sentence, 300 is the dimensionality of the word embedding used per each word.

The process was deployed for each word in a sentence. We trained the model using propagation after convolving all tokens in the sentence. Then we stored the weights, biases, and features for each token after convolution and back-propagated error to correct them once all tokens were processed by using the training scheme described in section 3.2

4.3. Dataset and evaluation

We tagged for three domains included book, female fashion, and mobile accessories. Each domain has over 4000 tagged corpora that were randomly separated into datasets of training, validation, or testing by cross-validation, with a ratio is 60/20/20

Each document was presented by word embedding that encoded semantic and syntactic properties of words. We used a word embedding dataset is Word2VecVN [23] [24] for our experiments.

For each corpus in our training dataset, we annotated it according to IOB2 format, which is usually used to coding scheme for representation sequences. For example:

Đóng/B-A gói/I-A tốt/O và/O giao/B-A hàng/I-A nhanh/O ,/O giá/B-A cả/I-A có/O khuyến/B-A mãi/I-A ,/O nội/B-A dung/I-A chưa/O đọc/O

(It means "Good packaging and fast delivery, promotional prices, content have not read yet")

In this IOB2 format, */B-A* is tagged for the first word of each aspect term. */I-A* is for the continuation of the term. And */O* is for a word unrelated to any aspect of product.

4.4. Features and rules

In our experiment, we used the following features:

- **Word embedding**: we used word embedding described in section 4.3 as features of the network. Each word was encoded as a 300-dimensional vector.
- **Part of speech tags**: the aspect terms can be a noun, a noun phrase (chunk), a verb, or a verb phrase. For this reason, POS features have an important role and are used as additional feature. We used 8 basic POS in Vietnam syntax, included noun N, verb V, adjective A, expletive EX, pronoun PN, adverb AD, preposition PR, and conjunction C. They are encoded as an 8-dimensional binary vector.

Both feature vectors were concatenated and fed to CNN, hence, each word is represented by a 308-dimensional feature vector.

We used a set of linguistic patterns (LPs) based on the specific characteristics of the Vietnamese language. The five LPs used are listed below:

- R1: If a noun *dt* is a subject followed by a term *c* present in a large sentiment lexicon, Vietnamese part in Multilingualsentiment [25], VnEmoLex [26], and a self-built lexicon, *dt* is tagged as an aspect. For example, *"chất vải dày đẹp"* ("beautiful thick fabric") with '*chất vải*' is an aspect.
- R2: If a verb *v* stands in the first position in a shortened sentence and *v* is followed by a term *c* present in a large sentiment lexicon, *v* is tagged as an aspect. As an example, "*giao nhanh*" ("deliver quickly") has "*giao*" masked as an aspect.
- R3: If a noun *dt* stands next to "*của sản phẩm*" ("of the product"), *dt* is defined as an aspect. For instance, "*bìa của sách*" ("cover of the book") has "*bìa*" masked as an aspect.

- R4: If two nouns *dt1* and *dt2* stand next to each other are masked aspects, group them into an aspect noun phrase.
- R5: If two verb *v1* and *v2* stand next to each other are masked aspects, group them into an aspect verb phrase.

5. Results

In this section, we summarized a list of top extracted aspects on three domains in Table 1, showed that the Vietnamese customers pay more attention to "*chất lượng*" (quality), "*kiểu dáng*" (form), and "*giao hàng*" (delivery). Several pairs of aspects are usually reviewed together like "*giá / chất lượng*" (price /quality) and "*hỗ trợ / hậu mãi*" (support / post-sale).

Table 1: Summary of main extracted aspects

Main aspects	# reviews contain the aspect		
	Book	**Female fashion**	**Mobile accessories**
Giá (Price)	485	1286	897
Đóng gói (Package)	912	384	693
Giao hàng (Delivery)	1532	949	1413
Kiểu dáng (Form)	1121	1812	1274
Chất lượng (Quality)	709	1886	1314
Hỗ trợ (Support)	547	796	685
Hậu mãi (Post-sale)	514	887	714
Chính hãng (Genuine)	59	-	635
Nội dung (Content)	413	-	-
Other	2008	2169	1822

To evaluate our proposed model accuracy, we do some extra experiments. The results are shown in Table 2, where LP is the model that using only linguistic patterns [5], WE is the one that using only word embedding feature, and WE+POS is the one when combining WE and POS features. It can be seen that our proposed model achieved the best result among these ones.

Moreover, the experiment showed that up to over 60% of aspect terms for all three domains are phrases (67.8% for book domain, 65% for female fashion, and 61% for mobile accessories). Extracting aspect phrases is usually harder than a single word aspect, lead to the accuracy lower than expected. In this case, R4 and R5 of linguistic patterns help to correctly tag the aspect phrases.

Table 2: Performance on test dataset with impacting of features and linguistic patterns

		Book	**Female fashion**	**Mobile accessories**
LP	Recall	61.42	65.01	64.3
	Precision	63.45	66.4	62.37
	F1-score	**62.42**	**65.70**	**63.32**
WE	Recall	65.54	63.08	63.54
	Precision	63.4	62.76	62.5
	F1-score	**64.45**	**62.92**	**63.02**
WE+POS	Recall	73.3	70.53	68.9
	Precision	70.24	68.33	66.9
	F1-score	**71.74**	**69.41**	**67.89**
WE + POS + LP (our)	Recall	74.52	72.14	73.51
	Precision	79.48	71.34	75.6
	F1-score	**76.92**	**71.74**	**74.54**
# extracted aspects		*31*	*66*	*57*

Figure 1 below shows a visualization for Table 2

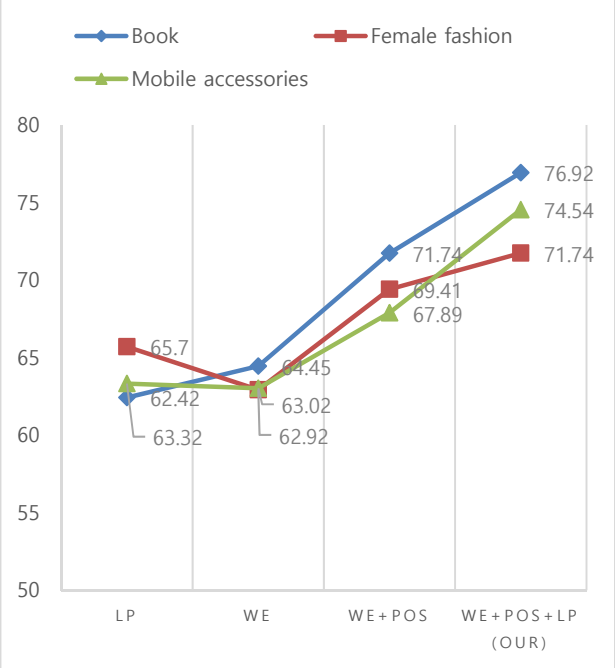

Fig. 1 Comparison of the performance of D-CNN with additional features and LP

Finally, the D-CNN architecture using word embedding and POS features combined with linguistic patterns achieved the promising results, they are 76.9% for the *book* domain, 71.7% for the *female fashion*, and 74.5% for the *mobile accessories*.

6. Conclusion

In this paper, we have presented a deep learning method to address the aspect extraction task from product reviews on the Vietnamese e-commerce websites. Our model used D-CNN architecture with word embedding feature, part-of-speech feature, and Vietnamese linguistic patterns. We chose three specific domains are *book*, *female fashion*, and *mobile accessories*, to do our experiments and archived acceptable results. In the future, we will extend data domains and refine patterns to improve performance and will use the outputs for our next task in our recommender system in the future.

Acknowledgments

We would like to thank CS'20.15 project of the Institute of Information Technology, Vietnam Academy of Science and Technology that has provided funding for the study.

References

[1] W. Lam and T.-L. Wong, "Hot Item Mining and Summarization from Multiple Auction Web Sites," 2014.
[2] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," 2004.
[3] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs. 2007.
[4] Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase Dependency Parsing for Opinion Mining," 2009.
[5] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," Comput. Linguist., vol. 37, no. 1, pp. 9–27, Mar. 2011.
[6] A. Katiyar and C. Cardie, "Investigating LSTMs for Joint Extraction of Opinion Entities and Relations," in Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2016), 2016, pp. 919–929.
[7] X. Li and W. Lam, "Deep Multi-Task Learning for Aspect Term Extraction with Memory Interaction *," in Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP 2017), 2017, pp. 2886–2892.
[8] X. Li, L. Bing, P. Li, W. Lam, and Z. Yang, "Aspect Term Extraction with History Attention and Selective Transformation," in Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 4194–4200.
[9] J. Zhang, G. Xu, X. Wang, X. Sun, and T. Huang, "Syntax-aware representation for aspect term extraction," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2019, vol. 11439 LNAI, pp. 123–134.
[10] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Recursive neural conditional random fields for aspect-based sentiment analysis," in EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings, 2016, pp. 616–626.
[11] W. Wang, Sinno, J. Pan, D. Dahlmeier, and X. Xiao, "Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms," in Proceedings of AAAI Conference on Artificial Intelligence (AAAI 2017), 2017.
[12] M. Zhang, Y. Zhang, and D.-T. Vo, "Neural Networks for Open Domain Targeted Sentiment," in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), 2015, pp. 17–21.
[13] Y. Yin, F. Wei, L. Dong, K. Xu, M. Zhang, and M. Zhou, "Unsupervised Word and Dependency Path Embeddings for Aspect Term Extraction," in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2016, 2016.
[14] S. Poria, E. Cambria, and A. Gelbukh, "Knowledge-Based Systems Aspect extraction for opinion mining with a deep convolutional neural network," Knowledge-Based Syst., vol. 108, pp. 42–49, 2016.
[15] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An Unsupervised Neural Attention Model for Aspect Extraction," in Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2017), 2017, pp. 388–397.
[16] S. Xiong, Y. Zhang, D. Ji, and Y. Lou, "Distance metric learning for aspect phrase grouping," in COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers, 2016, pp. 2492–2502.
[17] H. S. Le, T. Van Le, and T. V. Pham, "Aspect Analysis for Opinion Mining of Vietnamese Text," in Proceedings - 2015 International Conference on Advanced Computing and Applications, ACOMP 2015, 2016, pp. 118–123.
[18] L. Mai and B. Le, "Aspect-Based Sentiment Analysis of Vietnamese Texts with Deep Learning," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, vol. 10751 LNAI, pp. 149–158.
[19] H.-T. Nguyen, B. Le, H. Chi, Q.-H. Vo, and M.-L. Nguyen, "Multi-channel LSTM-CNN model for Vietnamese sentiment analysis Sequential Rule Mining View project Multi-channel LSTM-CNN model for Vietnamese sentiment analysis," in 9th international conference on knowledge and systems engineering (KSE), 2017, pp. 24–29.
[20] H. Chi et al., "Deep Learning for Aspect Detection on Vietnamese Reviews," in NAFOSTED Conference on Information and Computer Science, 2018, pp. 104–109.
[21] R. Collobert, J. Weston, J. Com, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," 2011.
[22] T. Vu, D. Quoc Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, "VnCoreNLP: A Vietnamese Natural Language Processing Toolkit."
[23] "GitHub - sonvx/word2vecVN: Pre-trained Word2Vec

models for Vietnamese." [Online]. Available: https://github.com/sonvx/word2vecVN. [Accessed: 13-Nov-2020].
[24] X.-S. Vu, T. Vu, S. N. Tran, and L. Jiang, "ETNLP: a visual-aided systematic approach to select pre-trained embeddings for a downstream task."
[25] "Multilingualsentiment - Data Science Lab." [Online]. Available: https://sites.google.com/site/datascienceslab/projects/multilingualsentiment. [Accessed: 12-Nov-2020].
[26] "VnEmoLex: A Vietnamese emotion lexicon for sentiment intensity analysis | Zenodo." [Online]. Available: https://zenodo.org/record/801610#.X6x6V2gzZqw. [Accessed: 12-Nov-2020].

Ngoc Truong received a Master degree in 2015 at University of Engineering and Technology, Vietnam National University. Currently working at the Institute of Information Technology, Vietnam Academy of Science and Technology. Research fields: NLP, data science.

Truong- Thang Nguyen received a Ph.D. in 2005 at the Japan Advanced Institute of Science and Technology (JAIST), Japan. Currently working at the Institute of Information Technology, Vietnam Academy of Science and Technology. Research fields: software quality assurance, software verification, program analysis.

Thai-Le Luong, being PhD at University of Engineering and Technology, Vietnam National University. Currently, working at the Faculty of Information Technology in the University of Transport and Communication. Research fields: NLP, data science.