A New Context-Based Sentence Embedding and the Semantic Similarity of Sentences

Dinh-Minh Vu, Thanh-Trung Trinh, and Masaomi Kimura

***Abstract*—Calculating semantic similarity between sentences is a difficult task for computers due to the complex structure and syntax of a sentence. Typically, in order to represent a sentence, there are numerous significant characteristics which need to be alternatively considered, for example, ambiguity, words' order, the context of sentences, etc. Various methods have been proposed to construct a language model for computing the similarity, such as average words embedding or sentence embedding based on auto-encoder architecture. However, these methods usually focus on the sentence and skip the influence of the previous sentences. In the paper, we introduce a novel approach to transform from sentences with context to embedding vectors based on auto-encoder architecture. Experiment results showed that the proposed method could find a better result for estimating similarity sentences in a certain scenario.**

***Index Terms*—Semantic similarity, word embedding, sentence embedding, language model, auto-encoder.**

I. Introduction

Estimating the semantic similarity is a difficult task in natural language processing. Unlike humans, it not easy for a machine to recognize the same meaning between similarity sentences because the structure and syntax of a sentence in human language are too complex. To completely get the intention of a sentence, a language model is required to cover the meaning of words, the arrangement of these words, and the context of this sentence. For instance, sentences with different words usually do not have the same meaning, and even the ones with same words might not either. For example, the sentence "Peter hit Tom" and "Tom hit Peter" share the same words, but the meanings of the two sentences are different. Moreover, the meaning of a sentence is also affected by the context of its conversation. For instance, if a question in the conversation is: "Are you hungry?", the answer could be: "Yes, I am hungry" which is an indication about the condition of the speaker. On the other hand, if the conversation has a question such as "Do you want something to eat?", the purpose of the answer: "Yes, I am hungry" can be considered as a confirmation that he needs food, same as the answer: "Yes, I want something to eat.". A language model is often constructed to help a machine solve the complicated problem mentioned above. There are two main research directions: word embedding and sentence embedding which aim to determine the representation of sentences.

In the former direction, the success of various word embedding models were developed in the past years such as Word2vec [1], Glove [2], FastText [3] and more recently ELMO [4], the approaches in this direction assume that the representation of sentence is simply calculated by the average of words embedding. With a word embedding model, a baseline method is computing the Bag-of-words (BoW) of word vectors to get sentence representation but does not really have an effective result because BoW rarely ponders the weight of words in the sentence. Consequently, some recent methods, especially smooth-inverse-frequency (SIF) [5], demonstrated the importance of using weighted average and modify them using singular-value decomposition (SVD). However, ignoring the ordering of words in a sentence is still a disadvantage of the methods in this direction. As an example, although "Tom hit Peter" and "Peter hit Tom" consist the same words, the meanings of them are different.

In the latter direction, sentence embedding models based on auto-encoder architecture were developed and has demonstrated that it can address the words ordering. To solve the problem, the methods in this direction usually encode full sentence to the vector by putting words consecutively to a neural network. In this direction, there are two types of approaches such as supervised and unsupervised learning-based method. Firstly, the method based on a supervised training task like InferSent [6] or Universal Sentence Encoder (USE) [7] use the Stanford Natural Language Inference (SNLI) labeled dataset to predict entailment/contradiction. By using the same encoder for two sentences with the gold score which is the semantic similarity value manually defined by humans, these above methods have received an impressing result. Nonetheless, the drawback is that these methods must require high quality dataset. Secondly, Skip Thought Vector [8], [9] or Quick Thought [10] – the unsupervised training-based methods are proposed with the idea using encoder-decoder models for the sentence, and the surrounding sentences of the given sentence. In this approach, the drawback of high quality dataset cost is solved because those applied on the unstructured dialogues.

In our best knowledge, all prior researches are performing passable representation of sentences but missing an important point about context. In fact, context plays an important role in the represented sentence, and the meaning of a sentence can only be fully understood within the context. Thus, we propose a model with two encoders for the previous sentences and the current sentence, a decoder to express the next sentence. By cover the influence of previous sentences, our model outperformed the previous studies in the unsupervised approach.

The remainder of this paper is organized as follows. In Section II, we review the approaches based on word embedding and sentence embedding. Next, the proposed

Manuscript received November 10, 2019; revised May 20, 2020.

All authors are with the Shibaura Institute of Technology, Koto-ku, Tokyo, Japan (e-mail: nb17502@shibaura-it.ac.jp, nb18503@shibaura-it.ac.jp, masaomi@sic.shibaura-it.ac.jp).

doi: 10.18178/ijke.2020.6.1.124

Context based Sentence Embedding (CSE) model is defined formally in Section III. Section IV describes an experiment of the proposed model to numerous datasets of similar sentences. Experimental results are discussed in Section V. Finally, open issues and potential future work are discussed in Section VI.

II. Related Work

A model language is essential, playing an integral role in numerous applications in natural language processing. As the heart of the application, a model language usually helps the machine understand and carry out the requirement of human. Specifically, for solving the different languages between human and machine, this model is an intermediary to transforms human language into the information understandable for machine and vice versa. In the last decade, with the advent of strong calculating devices, there are many proposed methods [1]-[10], which exist and has acquired impressive achievements. These methods often represent words, sentences based on the embedding vector to capture their semantic and syntax. Therefore, by computation of the combination of vector, machines can find out a representation of the requirement of humans.

One of the first popular approaches in word embedding is Word2vec [1] which aims to compute the semantic relationship adjacent words within a sentence. This approach relied on either Continuous Bag of Words (CBOW) architecture which uses the bag-of-words context to predict a target word or Skip-gram architecture which predicts the context word based on given a target word. Another approach similar to Word2vec is Glove [2], which also uses a context window to simulate word representation. The difference between them is the way of using the context of words. While Word2vec use context window to make a training set for neural network, Glove utilizes it to create a co-occurrence matrix. In term of this aspect, Glove might be more comprehensive than Word2vec because it computes based on all dataset, instead of only the context word in Word2vec. However, both models have the drawback in out of vocabulary problem.

This is one reason why FastText [3] is introduced by Facebook. They suggested a method which divided words into n-gram and training based on these tokens. For instance, the word “apple” will transform to “app”, “ppl” and “ple”. Thus, this model works well for the rare words and solves the about out of vocabulary problem.

Although word embedding operates well in some tasks, it still has a few issues when performs sentences. Hence, another research direction was proposed using sentence embedding to learn language representation. Based on auto-encoder architecture, some models attempt to represent sentence by supervised or unsupervised learning. A major advantage of this research direction is keeping the order of word in the sentence.

First of all, Skip-Thought Vector [8] model includes a Recurrent Neural Network (RNN) to map words to sentence vector in the encoder and two RNNs generate the surrounding sentences (previous and next sentence) in the decoder. A major advantage of Skip-Thought and other models in sentence embedding approach compared to word embedding is the order of word in the sentence.

After Skip-Thought, InferSent – a recent approach in sentence embedding has been introduced by Facebook [6] with the same ideology about the representation of sentences. The large difference between the two approaches is that InferSent was a supervised learning approach, instead of unsupervised learning as Skip-Thought. They applied a BiLSTM model to a labeled dataset to predict entailment/contradiction. This approach proves its efficiency with much better results on various tasks than other methods. However, a disadvantage of this approach is the requirement of the high-quality labeled dataset for training. The author used SNLI dataset which includes 570k pair of sentences in English to build the semantic sentence.

Finally, the latest proposal approach recently is Universal Sentence Encoder (USE) [7] which was provided by Google. Like InferSent, this approach also trains on the SNLI dataset but integrated with the unsupervised learning tasks in two different encoders for making sentence representation. The first one is the Transformer-based encoder model which required a high computational resource and got a better accuracy. In which, Transformer [11] is a novel architecture which uses only attention mechanism, instead of a Recurrent neural network. The second one is Deep averaging network (DAN) which utilizes a mechanism where words and n-gram are averaged as the input in a deep neural network.

Although the above approaches highly successfully performed the representation of sentences, there is a major issue in these models. It is a missing of the influence of context on the sentence. For example, machines cannot select the meaning between two sentences in Table I if missing the helpful information of the previous sentence. Therefore, we propose a novel approach which covers the sentence and its context for the best performing sentence.

TABLE I: The Influence of the Previous Sentence to the Current Sentence

	Previous sentence	Current Sentence	Meaning
Example	I have just bought a new telescope.	I saw a man in the building with a telescope.	There was a man in the building, and I saw him with my telescope.
	I came to my office.	I saw a man in the building with a telescope.	There was a man in the building, who I saw and he had a telescope.

III. Approach

Fig. 1 depicts the architecture of the proposed model. Based on Encoder-Decoder architecture, our model aims to construct the context vector which is a representation of the input sentences. Then, the next sentence will be generated using this context vector.

As we mentioned it in Section II, our model considers both the current sentence and the previous sentences as the input to obtain the context vector before insert to the decoder for generating the next sentence. The proposed model is divided into three parts: encoder, context vector extraction and decoder. Firstly, our model utilizes a Gate recurrent unit (GRU) neural network [12] as the encoder to encode the

sentences. Throughout GRU, the next word is predicted by the previous words in the sentence. The last hidden state of GRU will be considered as a representation of the input sentence. Furthermore, according to the difference of the effect contributed by the sentences on the next sentence, the current sentence and the past sentences are encoded by two different GRUs. While the meaning of the current sentence is performed to the hidden state by a GRU, another GRU extracts the additional information from the previous sentences to other hidden states for increasing the quality of the representation. In the second part, our model uses the attention mechanism to construct a context vector which represented the meaning of input sentence with its context. Specifically, these hidden states are computed by attention mechanism to get the weight of the input sentences. The average value of weights and hidden states are used to make a context vector. In the last part, the context vector is inserted to eventually generate the next sentence under the operation of the decoder. After the training phrase, the context vector can be considered as a representation of the sentence.

More specifically, assume that we have given a list of sentences (Si-k … Si, Si+1) where k is the number of previous sentences. Let W_i^t denote the tth word for sentence Si and X_i^t denote for its word embedding. The operation of CSE model will be expressed in three parts: encoder, context extraction and decoder.

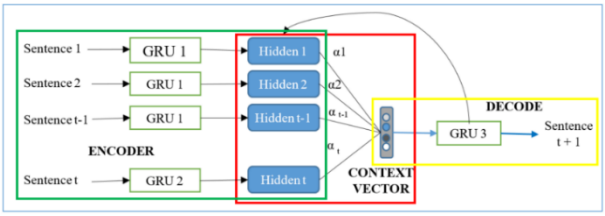

Fig. 1. Context based sentence embedding model.

A. Encoder

In the encoder step, we separate the current sentence and the previous sentences. Then, the numerous of preceding sentences are encoded by a GRU1. At the same time, GRU2 will transform the current sentence to the hidden states. Regarding the parameter, both GRUs share the same vocabulary matrix V, but have separate parameters. More specifically, given a sentence, words $W_i^1 \ldots W_i^N$ in sentence ith will be encoded to the N hidden state by GRU and the last step of the encoding produce h_i^N is considered as the representation of sentences. This produce can be illustrator such as follow:

$$r^t = \sigma\left(W_r x^t + U_r h^{t-1}\right). \quad (1)$$

$$z^t = \sigma\left(W_z x^t + U_z h^{t-1}\right). \quad (2)$$

$$\bar{h}^t = \tanh\left(Wx^t + U\left(r^t \odot h^{t-1}\right)\right). \quad (3)$$

$$h^t = \left(1 - z\right) \odot h^{t-1} + z^t \odot \bar{h}^t. \quad (4)$$

where W and U are the weight matrix, x^t is word at t^{th} in sentence, h^t is the hidden state of the word x^t, r^t is the reset gate, z^t is the update gate, ($\odot$) denotes a component-wise product and $\bar{h}^t$ is the proposed state update at time *t*. All update gates take values between zero and one.

B. Context Extraction

In this step, the given last hidden states h_i^N from encoder step are used to get the context vector. With the definition of the last hidden state which is considered as a representation, the attention mechanism is applied to help the model focusing on the important information.

More specifically, after encoding the input sentences, h_i^N will represent the first sentence in the dialogue and h_T^N is the hidden state of the current sentence where *N* is the max length of sentence in the dataset, *T* is the sum of the number of previous sentences and the current sentence. Using Bahdanau's attention mechanism, the weight of hidden states will be calculated as follows:

$$\alpha_j = align\left(h_T^N, h_j^N\right) \quad (5)$$
$$= v_a^T \tanh\left(W_{a^{h_T^N}} + U_{a^{h_j^N}}\right).$$

From (5), the context vector is expressed such as the following:

$$c = \sum_{i=1}^{T} \alpha_i h_i^N. \quad (6)$$

where *c* is the context vector, α_i is the weight of the sentence i^{th}.

C. Decoder

At the last step, the operation in decoder is the same as in encoder, except only the first input. To generate the next sentence, instead of token <start>, the context vector is firstly inserted to the decoder. Then, the hidden state of words in the next sentence h_{i+1}^tcan be computed as follow:

$$r^t = \sigma\left(W_r^d x^t + U_r^d h^{t-1}\right). \quad (7)$$

$$z^t = \sigma\left(W_z^d x^t + U_z^d h^{t-1}\right). \quad (8)$$

$$\bar{h}^t = \tanh\left(W^d x^t + U\left(r^t \odot h^{t-1}\right)\right). \quad (9)$$

$$h_{i+1}^t = \left(1 - z^t\right) \odot h^{t-1} + z^t \odot \bar{h}^t. \quad (10)$$

where W_r^d, U_r^d, W_z^d, U_z^d are weight matrix of GRU in decoder. Hence, given *t*-1 previous words and the context vector of encoder, we can recognize the word t^{th} in the next sentence.

IV. Experiment

Our experiment is divided into two parts: training and evaluation on the CSE model. In the training part,

Skip-Thought and our model were applied to the DailyDialog which was recently a popular dataset included 13118 daily conversations.

For comparison between Skip-Thought and our model, we utilized the same setting in training with the mini batch size 64, the embedding vector is 512 and learning rate is 5e-4 with the Adam optimization algorithm in 1000000 steps to predict the next sentence. The loss was computed by the comparison between the prediction and the target sentence.

Besides, to evaluate the influence of the previous sentences, we implemented two instances of our model, CSE1 is a model using one previous sentence and CSE2 is a model using two previous sentences.

After the training part, we evaluate Skip-Thought and our model on 1379 pairs of sentences of the STS benchmark data and the SICK dataset with 10000 records following the idea proposed by Tai *et al* [13] such as shown in Fig. 2 instead of calculating based on cosine similarity. Compared to cosine similarity, this method allows the weights to be learned while cosine similarity applies the same weight for all features. The procedure of this method is carried out as follows. Firstly, a given sentence pair will be encoded to the representation vector u and v. Then, we extract the relation between these two sentences. In order to do that, the concatenation of the component-wise product u v and the absolute difference |u − v| are regarded as the features for the given sentence pair. Finally, we train a logistic regression to predict a semantic relatedness for the given sentence pair.

For comparison, we evaluate two models each 50000 steps by Pearson and Spearman correlation which is the measure to estimate the quality of sentence embedding.

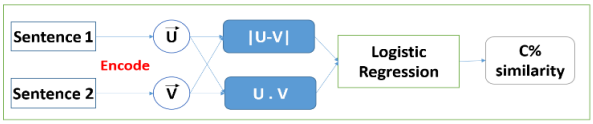

Fig. 2. The operation of calculating semantic relatedness.

V. Results and Discussion

The primary contribution of the proposed model is to find the influence of previous sentences to the meaning of the current sentence. This influence is equivalent to the context of the current sentence which reinforces the clarity of its meaning. This is an important task to separate the sentences having identical wording but different meaning. Hence, the representation of sentences is considerably more accurate.

TABLE II: The Best Pearson and Spearman Correlations in 20 Checkpoints

Channels	Skip-Thought	CSE-1 previous sentence	CSE-2 previous sentence
SICK Pearson	61,66 %	65,5 %	**65,8 %**
SICK Spearman	55,09 %	**58,88 %**	58,67 %
STS Pearson	42,19 %	48,98 %	**52,01 %**
STS Spearman	41,07 %	48,59 %	**51,74 %**

In our experiment, our model was shown that it outperformed Skip-Thought model. Fig. 3 shows that the loss of both models was almost convergence. However, while the prediction of the proposed model was approximated to the target sentence (the loss computed by the predicted and the target sentence was 0,33), the prediction of Skip-Thought model still need more time to get the best result. In addition, according to the semantic relatedness task, the results in Table II indicated that the proposed model also is better. All values of Pearson and Spearman correlation coefficient of our model were higher, especially the difference between the two models in the STSBenchmark dataset is approximate 10%. Besides, Fig. 4 shows that the worse result in the proposed model also is even better than the best result of Skip-Though.

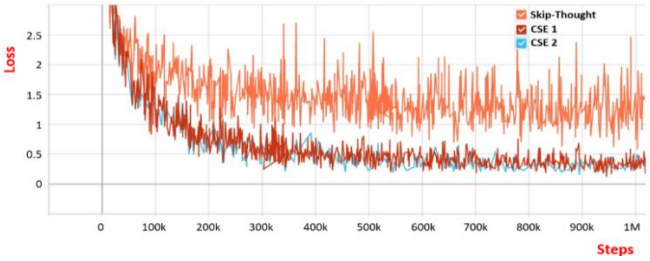

Fig. 3. Comparison loss of CSE model and skip-thought.

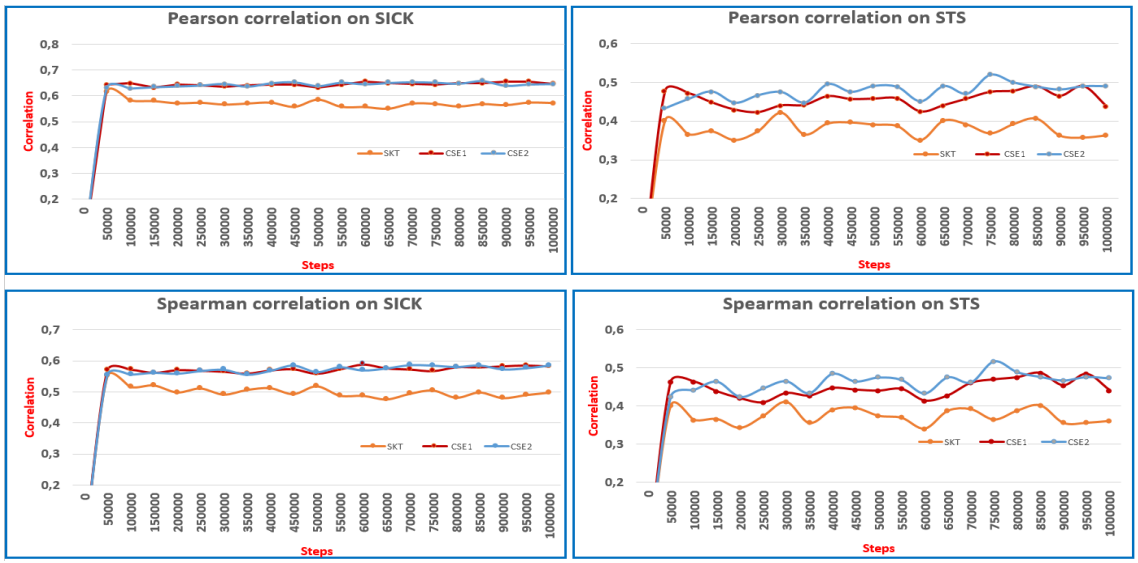

Fig. 4. Comparison of evaluation of CSE and skip-thought.

Regarding the training process, Fig. 4 also indicates that the early stop can be applied after 100000 steps because the proportion between the highest and the lowest is not too different. In addition, Table II shows that those models include more information from history which often perform better. With more information, CSE2 achieved three favorable results except for the Spearman correlation on SICK dataset where CSE1 achieved a better result (58,88%). As observed from Fig. 4, with a large dataset like SICK dataset, the variety in the results of our proposed model's two instances are not significantly different. However, with a small dataset such as STS, the model CSE2 is moderately better than CSE1. This can be understood that the model with more information will be stable.

Although the proposed model demonstrated the efficiency of embedding sentence based on context, the result still needs to be improved. As presented in [8], Skip-Thought could achieve a Pearson correlation 86,6% and Spearman correlation 80,83% on the SICK dataset which higher than our result. However, their experiment was applied on a much higher configuration: a bigger training dataset with 74 million sentences, embedding vector 2400 dimensions, and takes a much larger training time. At this moment, because of the limitation of equipment and time, we cannot make an experiment with a large training dataset. Consequently, the proposed model need to verify with a large training dataset.

VI. Conclusion

In this paper, we introduced a new approach to perform the representation of sentences. To do this, the proposed model simulated the dialogue with two GRU for the current sentence and previous sentences in an encoder, one GRU in a decoder to predict the next sentence. Then, the evaluation of the quality of sentence embedding is processed on the SICK and STSBenchmark datasets. The experiment result shows that CSE model can transform a sentence to a good sentence embedding to calculate the semantic similarity.

However, the used dataset for training is too small so there are some obstacles in transforming sentences to embedding. In the future, we will apply the proposed model to a large dataset and evaluate model again.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Dinh-Minh VU did programing, designed the methodology and setup experiment. Thanh-Trung TRINH and Masaomi KIMURA evaluated the methodology and experiment. All authors contributed to the preparation of the manuscript. All authors had approved the final version.

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, 2013, *arXiv Preprint arXiv:1301.3781*.

[2] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. the 2014 Conf. on Empirical Methods in Natural Language Processing* (EMNLP), 2014, pp. 1532-1543.

[3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135-146, 2017.

[4] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, *Deep Contextualized Word Representations*, 2018, *arXiv preprint arXiv:1802.05365*.

[5] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *Proc. ICLR 2017*, 2017, pp. 1-16.

[6] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, *Supervised Learning of Universal Sentence Representations from Natural Language Inference Data*, 2017, *arXiv preprint arXiv:1705.02364*.

[7] D. Cer, Y. Yang, S. Y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, and Y. H. Sung, *Universal Sentence Encoder*, 2018. *arXiv preprint arXiv:1803.11175*.

[8] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Proc. the 28th International Conf. on Neural Information Processing Systems*, 2015, pp. 3294-3302.

[9] S. Tang, H. Jin, C. Fang, Z. Wang, and V. de Sa. "Rethinking skip-thought: A neighborhood based approach," in *Proc. the 2nd Workshop on Representation Learning for NLP*, 2017, pp. 211-218.

[10] L. Logeswaran and H. Lee. *An Efficient Framework for Learning Sentence Representations*, 2018, *arXiv preprint arXiv:1803.02893*.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, pp. 5998-6008, 2017.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*, 2014. *arXiv preprint arXiv:1412.3555*.

[13] K. S. Tai, R. Socher, and C. D. Manning, *Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks*, 2015, *arXiv preprint arXiv:1503.00075*.

Vu Dinh Minh received the B.E in 2009 in information technology from Hanoi Open University, M.S in 2014 in computer and communication engineering from Hanoi University of Science and Technology (HUST). He has worked as a full stack developer for HUST since 2011. Currently, he is a PhD student at Shibaura Institute of Technology (SIT), Japan. His research interests include data mining, machine learning, and natural language processing.

Trinh Thanh Trung received the B.E in information technology from Le Quy Don Technical University (VN) in 2009, and received the M.Sc. in intelligent computer games development from the University of South Wales in 2011. Since 2013, he has been working at Hanoi University of Science and Technology (VN) as a lecturer. Currently, he is a PhD student at Shibaura Institute of Technology (SIT), Japan. His research interests include data engineering, computer agents and simulation.

M. Kimura is a full professor and the head of the Department of Computer Science and Engineering at Shibaura Institute of Technology, Japan. His research interests include text mining, databases, data mining, information extraction, natural language processing and machine learning. His contact address is at Dept. of Computer Science and Engineering, Shibaura Institute of Technology. 3-7-5 Toyosu, Koto-Ku, Tokyo 135-8548, Japan.