ICIC Express Letters
Part B: Applications
Volume **12**, Number **1**, January **2021**

ICIC International ©2021 ISSN 2185-2766
pp. **35–42**

TRANSFORMER ENCODERS INCORPORATING WORD TRANSLATION FOR RUSSIAN-VIETNAMESE MACHINE TRANSLATION

Thien Nguyen[1,*], Trang Nguyen[2], Huu Nguyen[3] and Phuoc Tran[4]

[1]Faculty of Information Technology
Ton Duc Thang University
19 Nguyen Huu Tho Street, Tan Phong Ward, District 7, Ho Chi Minh City, Vietnam
*Corresponding author: nguyenchithien@tdtu.edu.vn

[2]Faculty of Natural Sciences
Novosibirsk State University
Pirogova Street, Novosibirsk 630090, Russia
trangtulgu@gmail.com

[3]Faculty of Information Technology
Ho Chi Minh City University of Food Industry
140 Le Trong Tan Street, Tay Thanh Ward, Tan Phu District, Ho Chi Minh City, Vietnam
huunt@hufi.edu.vn

[4]NLP-KD Lab
Faculty of Information Technology
Ton Duc Thang University
19 Nguyen Huu Tho Street, Tan Phong Ward, District 7, Ho Chi Minh City, Vietnam
tranthanhphuoc@tdtu.edu.vn

Received June 2020; accepted September 2020

Abstract. *Neural machine translation systems including the latest Transformer models represent translation units in the form of embeddings – vectors of real numbers. Such continuous representations of translation units lead to smoother translation results, but do not always guarantee better results due to wrong word translations, compared to statistical machine translation systems. Moreover, for low-resource language pairs, such as Russian-Vietnamese, the errors of word translations in neural machine translation systems are more aggravated. In order to solve the problem, we try different ways of concatenating source word embeddings with embeddings of their corresponding word translations, when building a Transformer-based translation system for the Russian-Vietnamese language pair. As a result, we create two novel Transformer models: Transformer with Long Encoder and Transformer with Short Encoder. In the Transformer with Long Encoder source word embedding and translation embedding of single size are concatenated to form a vector of double size. The Long Encoder reduces the size of the concatenated embedding to single size with a linear layer, and then adds it with positional embedding of the source word to create a final embedding. The Short Encoder resembles the Long Encoder except for the linear layer. Instead, the Short Encoder creates word embedding and translation embedding of half-size, and then concatenates them to form a concatenated embedding of single size. The experimental results show that the proposed models provide better translation quality compared to the baseline Transformer model.*

Keywords: Word translation, Transformer, Neural networks, Neural machine translation, Russian-Vietnamese machine translation

1. **Introduction.** Machine translation is the task of translating texts from one language to another with computers. On a very basic intuitively level, words from sentences of a source language are replaced by words of a target language. Words as translation unit

DOI: 10.24507/icicelb.12.01.35

first come to mind naturally thanks to the fact that words are the smallest unit of meaning in sentences. The first works of machine translation which were done at IBM are word-based statistical models [1]. Obviously, using word-based statistical models has a disadvantage of translation quality, because stable phrases of source languages are broken down into pieces before translating. In order to solve the problem, in years 2000s phrase-based statistical machine translation [2] (phrase-based SMT) was developed to replace word-based statistical models. However, phrase-based SMT has its own problem. Each phrase is used independently not considering its meaning relative to other phrases. Developments of neural machine translation (NMT) [3, 4, 5, 6] have successfully handled the problem by representing translation units as embeddings – vectors of real numbers. Now translation units can be compared to each other. Recently a new NMT paradigm with the name of Transformer [7, 8] has revolutionized the task of machine translation by providing self-attention mechanism. Translation units can be compared and aligned not only between the source and target side, but also in the same side. Thanks to that advantage, Transformer models have replaced SMT and other NMT models to become the state-of-the-art translation paradigm. However, Transformer models have a flaw. Their continuous representation of translation units gives smoother translation results, but does not always guarantee better results due to wrong word translations. In [9] the researchers have addressed the problem by introducing Transformer with Gated Encoder incorporating word translation. They also propose dictionary approach to obtaining one-to-one word translation table. For a source word, its translation is a word in the target side with the highest lexical translation probability among all possible translations available in a word translation dictionary. The lexical translation probability is obtained from the alignment of parallel corpus. In Gated Encoder the embedding of source word is added by its translation embedding of the same dimension weighted with a certain amount. They reported translation quality improvement when translating from Chinese into English. Inspired by their work, we have applied their method for the Russian-Vietnamese language pair. Unfortunately, our preliminary experiment results have showed an insignificant improvement compared to the baseline Transformer model without deploying word translation. Such results urge us to find other ways to incorporate word translation into the Transformer model. In [10] the authors have incorporated linguistic features, such as lemmas, part-of-speech tags into an attentional encoder-decoder network [3] with recurrent neural networks [11] by concatenating their embeddings with embeddings of source words. In [9] the authors state that the reason they do not use concatenation of embeddings of source word and corresponding word translation is that Transformer multi-head attention layers and feed forward network sub-layers require input sequence and output sequence to have the same dimension. In this work, we try different ways to concatenate source word embeddings and their corresponding word translation embeddings, while keeping the dimension of concatenated input sequence equal the dimension of output sequence. According to ways of concatenating embeddings, we create different Transformer encoders to improve incorporation of word translation.

The rest of the paper is divided into three sections. We propose Transformer encoders in Section 2. Section 3 presents and analyzes the results of our experiments. Finally, Section 4 summarizes our work and gives our main conclusions.

2. **Transformer Encoders Incorporating Word Translation.** Transformer is a sequence to sequence model. The Transformer encoder encodes a sequence of Russian words $\mathbf{x} = \{x_i, \text{ for } i = 0, \dots, |\mathbf{x}| - 1\}$ with the length $|\mathbf{x}|$, where x_i is a word at position i in a sequence of context vector $\mathbf{D} = \{\mathbf{d}_i, \text{ for } i = 0, \dots, |\mathbf{x}| - 1\}$ of the same length, where $\mathbf{d}_i \in \mathbb{R}^d$ has d dimensions. The Transformer decoder decodes the sequence of context vector $\mathbf{D}$ into a sequence of Vietnamese words $\mathbf{y} = \{y_i, \text{ for } i = 0, \dots, |\mathbf{y}| - 1\}$ of the

length $|\mathbf{y}|$. In the following we detail variations of Transformer encoder with the aim of incorporating word translation.

2.1. **Long Encoder.** The encoder first looks up each Russian word x_i in the word translation dictionary to find its translation w_i. If the Russian word does not exist in the dictionary, the encoder will return token <unk>. The Russian word, and its translation are then passed through embedding layers $\mathbf{embedder}_x$ and $\mathbf{embedder}_w$, respectively, to get word embedding $\mathbf{t}_i \in \mathbb{R}^d$ of dimension d and word translation embedding $\mathbf{e}_i \in \mathbb{R}^d$ of dimension d. Embeddings $\mathbf{t}_i$ and $\mathbf{e}_i$ are then concatenated to form a vector $\mathbf{u}_i \in \mathbb{R}^{2d}$. Meanwhile, the position i of the Russian word x_i is passed through a positional embedding layer $\mathbf{embedder}_i$ to get a positional embedding $\mathbf{p}_i \in \mathbb{R}^d$. The encoder projects embedding $\mathbf{u}_i$ to embedding $\mathbf{v}_i \in \mathbb{R}^d$ with a **linear** layer, and then adds embedding $\mathbf{v}_i$ weighted by a factor $\sqrt{d}$ with positional embedding $\mathbf{p}_i$ to create an embedding $\mathbf{c}_i$. Mathematically, a **linear** layer is represented by the following formula:

$$\mathbf{v}_i = W\mathbf{u}_i + b, \tag{1}$$

where $W \in \mathbb{R}^{d\times 2d}$ and $b \in \mathbb{R}^d$ are parameters to be learned. The encoder repeats the previous steps for all Russian words x_i in sentence to create a sequence $\mathbf{C} = \{\mathbf{c}_i, \text{ for } i = 0, \dots, |\mathbf{x}| - 1\}$ of embeddings $\mathbf{c}_i$, where $\mathbf{c}_i \in \mathbb{R}^d$, and $\mathbf{c}_i = \sqrt{d} \times \mathbf{t}_i + \mathbf{p}_i$. Then we apply a dropout layer **dropout** to the sequence $\mathbf{C}$ to get a sequence $\mathbf{D} = \{\mathbf{d}_i, \text{ for } i = 0, \dots, |\mathbf{x}| - 1\}$, where $\mathbf{d}_i \in \mathbb{R}^d$. Finally, the sequence $\mathbf{D}$ is passed through a sequence of N encoder sublayers $\mathbf{encoder\_layer}_n$, for $n = 0, \dots, N - 1$ to get a sequence of context vectors. Essential parts of an $\mathbf{encoder\_layer}_n$ are a self-attention layer and a position-wised feed forward layer. Step-by-step procedure of the encoder is presented in **Algorithm 1**.

Algorithm 1 – Long Encoder: Encodes a sequence of Russian words in a sequence of context vectors with the help of a linear layer

```
Input: A sequence of Russian words $\mathbf{x} = \{x_i, \text{ for } i = 0, \dots, |\mathbf{x}| - 1\}$
Output: A sequence of context vectors $\mathbf{D} = \{\mathbf{d}_i, \text{ for } i = 0, \dots, |\mathbf{x}| - 1\}$
for $i = 0$ to $|\mathbf{x}| - 1$ do
    $w_i = \mathbf{lookup}(x_i)$
    $\mathbf{t}_i = \mathbf{embedder}_x(x_i)$
    $\mathbf{e}_i = \mathbf{embedder}_w(w_i)$
    $\mathbf{u}_i = \mathbf{concat}(\mathbf{t}_i, \mathbf{e}_i)$
    $\mathbf{v}_i = \mathbf{linear}(\mathbf{u}_i)$
    $\mathbf{p}_i = \mathbf{embedder}_i(i)$
    $\mathbf{c}_i = \sqrt{d} \times \mathbf{v}_i + \mathbf{p}_i$
end for
$\mathbf{D} = \mathbf{dropout}(\mathbf{C})$, where $\mathbf{C} = \{\mathbf{c}_i, \text{ for } i = 0, \dots, |\mathbf{x}| - 1\}$
for $n = 0$ to $N - 1$ do
    $\mathbf{D} = \mathbf{encoder\_layer}_n(\mathbf{D})$
end for
```

2.2. **Short Encoder.** The Short Encoder bears a close resemblance to the Long Encoder except for the linear layer. The Short Encoder does not apply the linear layer to project embeddings to d-dimension space. Instead, the encoder passes Russian words, and their translations through embedding layers $\mathbf{embedder}_x$ and $\mathbf{embedder}_w$ to get word embedding $\mathbf{t}_i \in \mathbb{R}^{\frac{d}{2}}$ of dimension $\frac{d}{2}$ and word translation embedding $\mathbf{e}_i \in \mathbb{R}^{\frac{d}{2}}$ of dimension $\frac{d}{2}$. Embeddings $\mathbf{t}_i$ and $\mathbf{e}_i$ are then concatenated to form a vector $\mathbf{u}_i \in \mathbb{R}^d$ of dimension d. Step-by-step procedure of the encoder is presented in **Algorithm 2**.

Algorithm 2 – Short Encoder: Encodes a sequence of Russian words in a sequence of context vectors

```
Input: A sequence of Russian words $\mathbf{x} = \{x_i,\ \text{for } i = 0, \ldots, |\mathbf{x}| - 1\}$
Output: A sequence of context vectors $\mathbf{D} = \{\mathbf{d}_i,\ \text{for } i = 0, \ldots, |\mathbf{x}| - 1\}$
for $i = 0$ to $|\mathbf{x}| - 1$ do
    $w_i = \mathbf{lookup}(x_i)$
    $\mathbf{t}_i = \mathbf{embedder}_x(x_i)$
    $\mathbf{e}_i = \mathbf{embedder}_w(w_i)$
    $\mathbf{u}_i = \mathbf{concat}(\mathbf{t}_i, \mathbf{e}_i)$
    $\mathbf{p}_i = \mathbf{embedder}_i(i)$
    $\mathbf{c}_i = \sqrt{d} \times \mathbf{u}_i + \mathbf{p}_i$
end for
$\mathbf{D} = \mathbf{dropout}(\mathbf{C})$, where $\mathbf{C} = \{\mathbf{c}_i,\ \text{for } i = 0, \ldots, |\mathbf{x}| - 1\}$
for $n = 0$ to $N - 1$ do
    $\mathbf{D} = \mathbf{encoder\_layer}_n(\mathbf{D})$
end for
```

3. Experiments.

3.1. Experiment corpus.

3.1. Experiment corpus. Our parallel corpus consists of 33,027 Russian-Vietnamese sentence pairs. The numbers of tokens in Russian sentences are from 10 to 20, exclusively. Sentences contain only alphabetic, numeric and punctuation characters. The Russian sentences are extracted from News Commentary data[1] of Shared Task: Machine Translation of ACL 2013 Eighth Workshop on Statistical Machine Translation. By translating the Russian sentences we create a set of corresponding Vietnamese sentences. Our parallel corpus is divided randomly into three parts: training, development and testing with the sizes 30,027, 1,500, 1,500, respectively. Our corpus devision approach proceeds very much in the same way as indicated in works studying machine translation for low-resource language pairs [12, 13].

Table 1 shows the number of sentences, the number of tokens, the number of tokens per sentence, and the number of unique tokens (dictionary size) in the training, development and testing datasets. In sentences tokens are separated by white spaces.

Table 1. Summary of the experiment corpus

Number of	Russian			Vietnamese		
	training	development	testing	training	development	testing
Sentences	30,027	1,500	1,500	30,027	1,500	1,500
Tokens	438,875	21,820	21,941	693,681	34,436	34,651
Tokens per sentence	14.6	14.5	14.6	23.1	23.0	23.1
Unique tokens	46,789	7,520	7,450	5,402	1,985	2,058

3.2. Experiment setup.

3.2.1. *Building lexical translation dictionary for the Russian-Vietnamese language pair.* We obtain Lexical Translation Dictionary from the training dataset of the bilingual corpus which sentences are already tokenized. Russian sentences in the corpus are tokenized simply by split operation, given that Russian words are delimited by white spaces. As in the case of Chinese language [17], Vietnamese is an isolated language with white spaces seperating not words, but syllables, so we tokenize all Vietnamese sentences in the corpus into sequences of words by an external tool provided in [14]. After that, we use GIZA++

[1] Download at: http://www.statmt.org/wmt13/training-parallel-nc-v8.tgz

toolkit [15] to align Russian and Vietnamese words. Based on these alignments we create lexical translation table with the help of Moses toolkit [16] which was initially used for phrase-based SMT. Among translations of a Russian word in the lexical translation table, we select the one with the highest translation probability. We do similar lookup for all Russian words in the lexical translation table, and thus, create a one-to-one word translation dictionary. The head of the dictionary is presented in Table 2. An example of a Russian sentence translated to a sequence of word translations with the use of the one-to-one dictionary is presented in Example 3.1.

Example 3.1. ***Russian sentence:*** *“жизнеспособная медицина также требует сравнения расходов на здравоохранение с расходами на другие социально важные нужды”.*

Word translations: *“khả_ thi y_ học huy_ hoàng đòi_ hỏi so_ sánh chi_ tiêu chợ y_ tế jaradua chi_ tiêu chợ khác bất_ công perminov nguyên”.*

Table 2. Head of word translation dictionary

Index	**Source word**	**Word translation**
1	метро	tàu_điện_ngầm
2	постсоветская	hậu
3	администрация	chính_quyền
4	прилагая	cần
5	типичного	bình_thường
6	аппаратом	tàu_vũ_trụ
7	насладиться	tận_hưởng
8	традицию	truyền_thống
9	проверил	thử_nghiệm
10	обладающие	chuyên_nghiệp

3.2.2. *Evaluating the proposed Transformer encoders for translating Russian into Vietnamese.* To evaluate the proposed models we perform four experiments. In each experiment we build and evaluate a word-to-word Transformer model from a set of models: baseline Transformer model, Transformer with Gated Encoder [9], proposed Transformer with Long Encoder and Transformer with Short Encoder. Transformer with Long Encoder has a greater number of parameters than the other models. Because it has parameters $W \in \mathbb{R}^{d\times 2d}$ and $b \in \mathbb{R}^{d}$ of the **linear** layer mentioned in Equation (1) as well as standard parameters of a Transformer model.

We base our Transformer implementations on the work of Ben Trevett[2] with the following hyper-parameter values: the dimension of context vectors – 256, the number of encoder/decoder sublayers equal to the number of decoder sublayers – 3, the number of heads in multi-head attention layer – 8, the dimension of encoder feedforward layer equal the dimension of encoder feedforward layer – 512, the level of dropout layers – 0.1. Regardless of the number of parameters in each Transformer model, we apply the same procedure to training all Transformer models. For each Transformer model, first we perform Xavier uniform initialization for all model parameters. Based on the former values of model parameters we then calculate their present values which provide the least cross-entropy loss in an epoch of the training dataset using Adam optimizer with a fixed learning rate $5e^{-4}$ [19]. We repeat the calculation for 20 epochs of the training dataset, and save the values of model parameters each time. Finally, among 20 sets of parameter

[2] Download at: https://github.com/bentrevett/pytorch-seq2seq

values we select the one which provides the least cross-entropy loss in an epoch of the development dataset.

After the model is built, we use the testing dataset to assess the model. Each Russian sentence in the testing dataset is tokenized and fed to the model. The model generates a predicted Vietnamese sentence. After the translation is completed, all predicted Vietnamese sentences are detokenized and compared with the corresponding Vietnamese sentences of the testing dataset. We evaluate the translation results in terms of BLEU score. We use the natural language toolkit NLTK [18] to calculate the lowercase BLEU score.

3.3. **Experiment results and analysis.** The BLEU scores of translation results by the models are presented in Table 3.

TABLE 3. The BLEU scores of translation results

Experiment	**NMT model**	**BLEU score**
1	Baseline Transformer	34.45
2	Transformer with Gated Encoder	34.69
3	Transformer with Long Encoder	35.76
4	Transformer with Short Encoder	36.03

Compared to the baseline Transformer, an insignificant improvement of translation quality ($34.69 - 34.45 = 0.24$ BLEU) by Transformer with Gated Encoder is revealed. The result is the motivation for us to propose other translation models. Our experiment results support our proposal, in fact, we succeed in raising BLEU score by $35.76 - 34.45 = 1.31$, using Transformer with Long Encoder. Surprisingly, the best model is Transformer with Short Encoder, although it has fewer parameters, compared to Transformer with Long Encoder. An improvement of $36.03 - 34.45 = 1.58$ BLEU by Transformer with Short Encoder is reported, compared with the baseline Transformer. Moreover, our proposed Transformer with Long Encoder and Transformer with Short Encoder outperform Transformer with Gated Encoder proposed in [9] by 1.07 and 1.34 BLEU, respectively.

Further analysis, which takes human judgement into account, is undertaken to verify our findings. Let us consider the following examples.

Example 3.2. ***Russian sentence:*** *"действительно, религиозная напряжённость между суннитами и шиитами ирака возросла после свержения саддама".*

Meaning in English: *"indeed, the religious tension between the sunnis and shiites of iraq increased after the overthrow of the saddam".*

Vietnamese gold reference: *"thật vậy, căng thẳng tôn giáo giữa người sunni và người shiite ở iraq đã tăng lên sau khi saddam bị lật đổ".*

Translation by baseline Transformer: *"thật vậy, căng_ thẳng tôn_ giáo giữa người shiite và người shiite đã tăng lên sau cuộc tấn_ công saddam".*

Translation by Transformer with Gated Encoder: *"thật vậy, căng_ thẳng tôn_ giáo giữa người shiite và người shiite đã tăng lên sau iraq lật_ đổ saddam".*

Translation by Transformer with Long Encoder: *"thật vậy, căng_ thẳng tôn_ giáo giữa người sunni và người shiite đã tăng lên sau khi lật_ đổ saddam lật_ đổ".*

Translation by Transformer with Short Encoder: *"thật vậy, căng_ thẳng tôn_ giáo giữa người shiite và người shiite ở iraq đã tăng sau khi lật_ đổ saddam".*

Example 3.2 shows that all NMT models, except for Transformer with Long Encoder, give the same wrong translation "shiite" for the key Russian word "суннитами" (sunnis) in the source sentence. However, we find that Transformer with Long Encoder and the baseline model fail to keep the word "ирака" (iraq) in translation, while the other models successfully do it. Besides, all models with word translation succeed in translating the word “свержения” (overthrow) into “lật_đổ”. Comparing the models with word translation to each other, we note that our proposed Transformer with Long Encoder and Transformer with Short Encoder have better translation in overall meaning. This can be explained by the fact that the word orders of translations by our models are more natural, similar to the word order of the gold reference.

Example 3.3. ***Russian sentence:*** *“они рассматривают любую критику саддама как поддержку агрессивных действий америки”.*

Meaning in English: *“they see any criticism of saddam as supporting america’s aggressive actions”.*

Vietnamese gold reference: *“họ thấy bất kỳ lời chỉ trích nào về saddam là sự ủng hộ các hành động gây hấn của mỹ”.*

Translation by baseline Transformer: *“họ coi bất_kỳ lời chỉ_trích nào về sự hỗ_trợ của saddam là những hành_động tích_cực của mỹ”.*

Translation by Transformer with Gated Encoder: *“họ coi bất_kỳ sự chỉ_trích nào đối_với sự hỗ_trợ của saddam đã bị ủng_hộ bởi những hành_động nổi_tiếng của mỹ”.*

Translation by Transformer with Long Encoder: *“họ coi bất_kỳ sự chỉ_trích nào của saddam như ủng_hộ hành_động của mỹ”.*

Translation by Transformer with Short Encoder: *“họ coi bất_kỳ lời chỉ_trích nào của saddam là sự ủng_hộ hành_động của nước mỹ”.*

Example 3.3 shows the power of NMT models. In Example 3.3 all NMT models translate “рассматривают” (see/consider) into “coi” (see/consider). Although the gold reference has a different word “thấy”, the meanings are almost identical. Moreover, Example 3.3 proves the usefulness of our proposed Transformer with Long Encoder and Transformer with Short Encoder. The models predict a short translation, but keep the meaning of the source sentence almost the same. On the other hand, translation results by the baseline Transformer and Transformer with Gated Encoder are lengthier, contain many words in the gold reference, but the meaning is changed, compared to the source sentence.

4. Conclusions.

In this paper we have customized the baseline Transformer Encoder for the Russian-Vietnamese language pair. We adopt different approaches to concatenate source word embeddings with their corresponding word translations. The resulting Long Encoder and Short Encoder have the capacity to incorporate word translation. Evidences from analysis of experiment results by machine and human judgements support our idea. The Transformer models with Long/Short Encoders outperform both the baseline Transformer, and the Transformer model with Gated Encoder, which was previously proposed to incorporate word translation. Our study provides the framework for new ways to combine word translation into NMT models. Nevertheless, our work clearly has some limitations. Here we investigate word-to-word NMT models in spite of the fact that subword-based NMT models have proved to be better models for many other language pairs. We are now in the process of building subword-based Transformer models incorporating word translation.

REFERENCES

[1] P. Koehn, *Statistical Machine Translation*, Cambridge University Press, 2009.

[2] R. Zens, F. J. Och and H. Ney, Phrase-based statistical machine translation, in *Advances in Artificial Intelligence. KI 2002. Lecture Notes in Computer Science*, M. Jarke, G. Lakemeyer and J. Koehler (eds.), Berlin, Heidelberg, Springer, 2002.

[3] D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate, in *The 3rd International Conference on Learning Representations. ICLR 2015. Conference Track Proceedings*, Y. Bengio and Y. LeCun (eds.), San Diego, CA, USA, 2015.

[4] K. Cho, B. van Merriënboer, D. Bahdanau and Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, *Proc. of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, Doha, Qatar, pp.103-111, Association for Computational Linguistics, 2014.

[5] M.-T. Luong, H. Pham and C. D. Manning, Effective approaches to attention-based neural machine translation, *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp.1412-1421, 2015.

[6] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes and J. Dean, Google's neural machine translation system: Bridging the gap between human and machine translation, *CoRR*, abs/1609.08144, 2016.

[7] 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.

[8] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. Gomez, S. Gouws, L. Jones, Ł. Kaiser, N. Kalchbrenner, N. Parmar et al., Tensor2Tensor for neural machine translation, *Proc. of the 13th Conference of the Association for Machine Translation in the Americas*, vol.1, pp.193-199, 2018.

[9] D. Han, J. Li, Y. Li, M. Zhang and G. Zhou, Explicitly modeling word translations in neural machine translation, *ACM Trans. Asian and Low-Resource Language Information Processing (TALLIP)*, vol.19, no.1, pp.1-17, 2019.

[10] R. Sennrich and B. Haddow, Linguistic input features improve neural machine translation, *Proc. of the 1st Conference on Machine Translation*, vol.1, pp.83-91, 2016.

[11] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.1724-1734, 2014.

[12] P. Tran, D. Dinh and H. T. Nguyen, A character level based and word level based approach for Chinese-Vietnamese machine translation, *Computational Intelligence and Neuroscience*, 2016.

[13] P. Tran, D. Dinh and L. H. Nguyen, Word re-segmentation in Chinese-Vietnamese machine translation, *ACM Trans. Asian and Low-Resource Language Information Processing (TALLIP)*, vol.16, no.2, pp.1-22, 2016.

[14] T. Vu, D. Q. Nguyen, M. Dras, M. Johnson et al., VnCoreNLP: A Vietnamese natural language processing toolkit, *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp.56-60, 2018.

[15] F. J. Och and H. Ney, A systematic comparison of various statistical alignment models, *Computational Linguistics*, vol.29, no.1, pp.19-51, 2003.

[16] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens et al., Moses: Open source toolkit for statistical machine translation, *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pp.177-180, 2007.

[17] W. Gan and X. Yu, Automatic understanding and formalization of natural language geometry problems using syntax-semantics models, *International Journal of Innovative Computing, Information and Control*, vol.14, no.1, pp.83-98, 2018.

[18] E. Loper and S. Bird, NLTK: The natural language toolkit, *Proc. of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pp.63-70, 2002.

[19] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol.1, pp.4171-4186, 2019.