Article

Improving the Performance of Vietnamese–Korean Neural Machine Translation with Contextual Embedding

Van-Hai Vu ¹, Quang-Phuoc Nguyen ^{2,*}, Ebipatei Victoria Tunyan ³ and Cheol-Young Ock ^{1,*}

¹ Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, Korea; cachua070717@ulsan.ac.kr

² User Data Processing Solutions, Seoul 04168, Korea

³ AIInsight, Incheon 21984, Korea; ebitunyan@aiinsight.io

* Correspondence: qp.nguyen@tqcs.io (Q.-P.N.); okcy@ulsan.ac.kr (C.-Y.O.)

Abstract: With the recent evolution of deep learning, machine translation (MT) models and systems are being steadily improved. However, research on MT in low-resource languages such as Vietnamese and Korean is still very limited. In recent years, a state-of-the-art context-based embedding model introduced by Google, bidirectional encoder representations for transformers (BERT), has begun to appear in the neural MT (NMT) models in different ways to enhance the accuracy of MT systems. The BERT model for Vietnamese has been developed and significantly improved in natural language processing (NLP) tasks, such as part-of-speech (POS), named-entity recognition, dependency parsing, and natural language inference. Our research experimented with applying the Vietnamese BERT model to provide POS tagging and morphological analysis (MA) for Vietnamese sentences, and applying word-sense disambiguation (WSD) for Korean sentences in our Vietnamese–Korean bilingual corpus. In the Vietnamese–Korean NMT system, with contextual embedding, the BERT model for Vietnamese is concurrently connected to both encoder layers and decoder layers in the NMT model. Experimental results assessed through BLEU, METEOR, and TER metrics show that contextual embedding significantly improves the quality of Vietnamese–Korean NMT.

Keywords: neural machine translation; part-of-speech; word-sense disambiguation; Korean–Vietnamese machine translation; bidirectional encoder representations from transformers

Citation: Vu, V.-H.; Nguyen, Q.-P.; Tunyan, E.V.; Ock, C.-Y. Improving the Performance of Vietnamese–Korean Neural Machine Translation with Contextual Embedding. *Appl. Sci.* **2021**, *11*, 11119. <https://doi.org/10.3390/app112311119>

Academic Editor: Valentino Santucci

Received: 22 September 2021

Accepted: 17 November 2021

Published: 23 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Bidirectional encoder representations for Transformers, abbreviated as BERT, is a pre-trained model introduced by Google [1]. BERT has greatly improved the quality of natural language processing (NLP) tasks [1–3]. This state-of-the-art context-based embedding model comprises two tasks: masked-language modelling (MLM) and next-sentence prediction (NSP). MLM uses context words surrounding a masked word to predict what the masked word should be. When two sentences are fed into the BERT model, NSP predicts whether or not the second sentence can follow the first sentence.

There are different variants of BERT for different languages: A Lite Bert (ALBERT) [4], Robustly Optimized BERT (RoBERTa) [5], and SpanBERT [6] are used for English; FlauBERT [7] for French; PhoBERT [3] for Vietnamese; GottBERT [8] for German; MacBERT for Chinese [9]; and KR-BERT [10] for Korean.

The ALBERT variant uses cross-layer parameter-sharing and factorized embedding-layer parameterization techniques to decrease the number of parameters, thus reducing training time. RoBERTa, a pre-training approach, is one of the most popular variants of the BERT model. To improve the original BERT model, RoBERTa has implemented several changes in pre-training, such as using byte-level byte-pair encoding (BBPE) and dynamic masking, instead of the static masking used in the MLM task.

PhoBERT is a pre-trained BERT variant for the Vietnamese language. It is based on the RoBERTa model. PhoBERT was trained on a huge Vietnamese corpus, including a 50 GB Vietnamese dataset.

Neural machine translation (NMT) directly translates a source sentence to a target sentence by using a neural network [11,12]. A series of publicly available toolkits, such as OpenNMT [13], Sockeye [14], and Fairseq [15], have made it easier for researchers to access NMT. NMT research is mostly performed on many rich-resource languages [2,16,17], but research in low-resource languages is receiving more attention [18–20].

The neural network models used in studies are diverse. For example, Wu et al. used Gap NMT (GNMT) [21]; Gerhring et al. used convolutional sequence-to-sequence learning (ConvS2S) [22]; Huang et al. used a hybrid translation model [23]; and Vaswani et al. applied the transformer model [24].

For Korean–Vietnamese machine translation (MT), Vu et al. [25] published a dataset of more than 412,000 sentence pairs for training with the NMT model. Nguyen et al. [26] performed a series of experiments with NMT by applying morphological analysis (MA) and word-sense disambiguation (WSD) to Korean sentences, before feeding language pairs into the MT system. In terms of TER [27] and BLEU [28] evaluation metrics, their system has led to the most significant improvement in the Korean-to-Vietnamese MT system by applying MA and WSD to Korean sentences and word segmentation to Vietnamese sentences. Cho et al. [29] used the Moses toolkit [30] to perform Korean–Vietnamese statistical MT (SMT). Before using a parallel corpus for training in the SMT model, they grouped component morphemes of adjectives and verbs in Korean sentences.

In this research, we apply the transformer model for Korean–Vietnamese MT, in which the MT model combines the pre-trained contextual-embedding Vietnamese BERT model and NMT. Before being included in the MT systems, the Korean–Vietnamese bilingual corpus is prepared by a series of pre-treatment steps to improve the quality of the MT. In particular, we tagged POS [3] to Vietnamese sentences, and we applied MA and WSD to Korean sentences. This POS tagging was based on a Vietnamese pre-trained contextual-embedding model. We used BLEU, METEOR [31], and TER scores to evaluate the accuracy of MT systems.

2. Transformer Model

A transformer [24,32] is a popular deep learning model that has often replaced RNN or LSTM for NLP tasks. In the language translation task, the transformer model adopts encoder–decoder architecture. For example, to convert a sentence from Vietnamese to Korean, the encoder learns the representation of the given Vietnamese sentence and feeds it into the decoder. The decoder then generates the corresponding sentence in the Korean language.

2.1. The Encoder of the Transformer

A transformer contains a stack of encoder blocks. Each encoder block involves two sub-layers, including a multi-head attention mechanism (which computes multiple attentions instead of a single attention) and a feed-forward network, as shown in Figure 1. Each encoder sends its output to the next encoder, and the last encoder generates a matching representation of the corresponding input sentence.

First, the input sentence is converted to input embedding. This embedding matrix is fed into the first encoder block (encoder 1). Encoder 1 receives the input and puts it into a multi-head attention layer, which generates the attention matrix as output. The attention matrix is then sent to the feed-forward network as the input for this layer, which returns the encoder representation as output. In the next step, the encoder representation of encoder 1 is fed into encoder 2 as input, and encoder 2 performs similar steps as encoder 1. This process ends when the final encoder block (encoder L) is computed and the representation of the given input sentence is generated.

Figure 1. A simple demonstration of the encoder.

As an example, suppose n is the number of the single attention matrix Z and the multi-head attention mechanism is the combination of all the attention matrices (attention heads) multiplied by a new weight matrix, W_0 , as follows:

$$\text{multi-head attention} = \text{Concatenate}(Z_1, Z_2, \dots, Z_n)W_0 \quad (1)$$

Each attention matrix Z_x can be computed as

$$Z_x = \text{softmax}\left(\frac{Q_x K_x^T}{\sqrt{d_k}}\right)V_x \quad (2)$$

where Q_x , K_x , and V_x are, respectively, the query, key, and value matrix of input word x . The query is the information we are looking for, the key represents relevance to the query, and the value is the actual content of the input; d_k represents the dimension of the key vector. Instead of computing a single attention head, multiple attention heads are calculated. As indicated in Equation (2), a single attention head is computed by the self-attention mechanism as follows: first, the dot product between the query matrix Q and the key matrix K^T is computed; the value of QK^T is then divided by the square root of the dimension of the key vector; next, the values obtained in the previous step are fed into the softmax function to normalize the scores and achieve the score matrix; finally, the attention matrix Z is computed by multiplying the score matrix by the value matrix V .

The feed-forward network in the encoder of the transformer contains two dense layers with a rectified linear unit (ReLU) activation function. The parameters of the feed-forward network differ in the encoder blocks, but they are the same over the various positions of the sentence.

In addition to the feed-forward network and multi-head attention, the add-and-norm process is also an important component of the encoder. The add-and-norm component is used to connect the input and output of each feed-forward network and multi-head attention layer. In addition, a positional encoding (an encoding announcing word order) provides information about the position of each word before feeding the information into the transformer model, as shown in Figure 2.

Figure 2. Encoder x with the add-and-norm component.

Another crucial component of the encoder is the add-and-norm component. It connects the input and output of the multi-head attention layer and the input and output of the feed-forward network layer. This component consists of a residual connection and a normalization layer. The add-and-norm component is fed into the encoder to make the training process faster, by avoiding heavy changes in value from layer to layer.

2.2. The Decoder of the Transformer

The decoder of the transformer receives the embedding representation of the encoder as an input and returns the target sentence as an output. Like the encoder, the decoder contains a stack of N sub-decoders. A decoder has two inputs: the output of the previous decoder and the representation of the encoder, as shown in the following Figure 3.

Figure 3. A simple demonstration of the decoder.

The $\langle \text{sos} \rangle$ and $\langle \text{eos} \rangle$ tokens, respectively, indicate the beginning and the ending of the sentence. Suppose, for example, that the output of the decoder is a sentence in the form $y = (y_1, y_2, \dots, y_m)$, where m is the length of the sentence. At the first time step, where $t = 0$, the $\langle \text{sos} \rangle$ output is $\langle \text{sos} \rangle$, and the output $\langle \text{eos} \rangle$ is y_1 ; at the next time step, $t = 1$, the $\langle \text{sos} \rangle$ output is $\langle \text{sos} \rangle y_1$, and the output $\langle \text{eos} \rangle$ is $y_1 y_2$; at the last time step, when full output is generated, the token $\langle \text{eos} \rangle$ is created and the output of the decoder is $y_1 y_2 \dots y_m \langle \text{eos} \rangle$.

There are three sub-layers on each decoder block, including masked multi-head attention, multi-head attention, and the feed-forward network. In addition, the output of each sub-layer is connected to an add-and-norm component. As with the encoder component, the positional encoder layer is added to the decoder component, as shown in Figure 4.

The masked multi-head attention is the first layer in the decoder. It operates in a manner similar to the multi-head attention mechanism in the encoder. For head i , the attention matrix Z_i is computed as follows:

$$Z_i = \text{softmax} \left(\frac{Q_i K_i^T}{\sqrt{d_k}} \right) V_i \quad (3)$$

where Q_i , K_i , V_i are the query, key, and value of the input matrix and d_k is the dimension of the key vector. The final attention matrix, M , is generated by concatenation of all attention matrices and a new weight matrix W_0 , as in the following equation:

$$M = \text{Concatenate} (Z_1, Z_2, \dots, Z_h) W_0 \quad (4)$$

The result of this final attention in the masked multi-head attention layer is fed into the multi-head attention layer. As shown in Figure 4, each multi-head attention layer has two inputs: the representation of the encoder (R) and the output of the masked multi-head attention (M). The key and the value of this layer are calculated in a manner similar to the method used for the encoder layer, but with a small difference. The weight matrix of the encoder representation is multiplied by the weight matrices of the query and the key, respectively.

Figure 4. A simple demonstration of the decoder.

2.3. Encoder and Decoder Architecture of the Transformer

A short demonstration of the interaction between an encoder and a decoder is provided in Figure 5. The input is fed into the encoder and the encoder generates the representation of the input text. This representation is sent as the decoder's input, and the decoder produces the output. In the MT task, the input and the output are, respectively, the source sentence and the target sentence.

Figure 5. Encoder and decoder architecture of the transformer.

3. Linguistic Annotation

Some recent studies have shown the benefit of applying linguistic annotation to input sentences for machine translation. The typical linguistic features, such as POS, morphological properties of words, WSD, and syntactic dependencies, can be added to input sentences. In this research, we present three types of annotation—POS, MA, and WSD—which we applied to enhance the quality of our MT system.

3.1. Vietnamese Parts of Speech

For Vietnamese sentences, we rely on the PhoNLP model [33] to tag POS, as shown in Figure 6. In this model, the PhoBERT-based variant is applied in the encoding layer to generate the embedding of each token of the given input. If we assume that the given input is a sequence in the form $x = (x_1, x_2, \dots, x_n)$, where n is the length of the sequence, the embedding (e_i) of the word (x_i) is calculated by:

$$e_i = \text{PhoBERT}_{\text{base}}(x_{1:n}, i) \quad (5)$$

Figure 6. Illustration of POS tagging with BERT.

The contextualized word-embedding (e_i) is fed into a feed-forward network; then, with a softmax predictor, the POS of each word in the given input is predicted as follows:

$$p_i = \text{softmax}(\text{FFN}(e_i)) \quad (6)$$

This experimental exercise is based on the benchmark datasets of the VLSP 2013 POS tagging corpus (<https://vlsp.org.vn/resources-vlsp2013>, (Accessed 20 October 2021)). The result of this method is superior to other research on POS for Vietnamese, as shown in the following table.

Table 1 shows that POS tagging using the BERT model provides the best results. BiLSTM-CNN-CRF, VnCoreNLP-POS, JPTDP-v2, and JointWPD use Bi-LSTM to predict output. The Bi-LSTM system learns words left-to-right and right-to-left separately; therefore, this system does not capture the meaning of words optimally. The BERT model, on the other hand, can learn the context from both directions simultaneously, thus enabling better learning of the word context.

Table 1. Performance of some Vietnamese POS tagging systems.

POS Tagging Systems	Accuracy (%)
BiLSTM-CNN-CRF [34]	95.4
VnCoreNLP-POS [33]	95.9
JPTDP-v2 [33]	95.7
JointWPD [35]	96.0
PhoBERT_{base}	96.8

The transformation in the form of Vietnamese sentences in the bilingual corpus is demonstrated as shown in the following Table 2.

Table 2. Form of Vietnamese sentences.

Form	Meaning
Initial	anh ấy làm việc tại trường đại học FPT (he works at the University of FPT)
Word Seg.	anh_ấy làm_việc tại trường đại_học FPT
Word Seg. and POS	anh_ấy N làm_việc V tại E trường N đại_học N FPT Ny

N, V, E, and Ny are the tagged POS indicating, respectively, noun, verb, preposition, and abbreviation of noun.

3.2. Korean Morphological Analysis and Word Sense Disambiguation

We add two kinds of linguistic annotation—MA and WSD—for Korean sentences, by using our Korean processing tool (UTagger) [26,36]. The token in Korean is analyzed and attached with the corresponding sense code. For example, the token “*a-peu-da*” is analyzed as two sub-words, “*a-peu*” and “*da*”. These two sub-words are connected by the character “+”. In addition, the homograph “*bae*” appears twice, with two different senses (“*stomachache*” and “*pear*”), which are defined, respectively, by the sense-codes “01” and “02”. The homograph “*bae*” will be altered into “*bae_01*” and “*bae_02*”, depending on the context. A simple example of defining MA and WSD by UTagger is shown below.

In Table 3, NNG, JKO, MAG, VV, EC, JKS, VA, EF, and SF are tagged using POS to indicate, respectively, noun, objective case, adverb, verb, connection, auxiliary, adjective, ending of a word, and the ending of the sentence. Compared with some other studies, UTagger showed outstanding results in both MA and WSD tasks, as shown in Table 4.

Table 3. Form of Korean sentences.

Form	Meaning
Initial	배를 너무 많이 먹어서 배가 아프다. <i>bae-leul neo-mu man-hi meog-eo-seo bae-ga a-peu-da.</i> (I have a stomachache because I ate too many pears.)
UTagger applied form	배__03/NNG + 를/JKO 너무__01/MAG 많이/MAG 먹__02/VV + 어서/EC 배__01/NNG + 가/JKS 아프__02/VA + 다/EF + ./SF <i>bae__03/NNG + leul/JKO neo + mu__01/MAG man-hi/MAG meog__02/VV + eo-seo/EC</i> <i>bae__01/NNG + ga/JKS a-peu__02/VA + da__02/VA</i>

Table 4. The result of the MA and WSD research.

Research	MA (%)	WSD (%)
Phrase-based statistical model [37]	96.35	
Bi-long short-term memory [38]	96.20	
Bidirectional recurrent neural network [39]		96.20
Statistical-based [40]		96.42
UTagger	98.20	96.52

4. Experimentations and Results

4.1. Vietnamese–Korean Parallel Corpus

To evaluate the accuracy of the system, we used over 412,000 Korean–Vietnamese sentence pairs that were published in our previous research [25]. In Vietnamese, we applied the word segmentation tool for sentences. The details of the corpus are shown in the following Table 5.

Table 5. Vietnamese–Korean bilingual corpus.

		Sentences	Average Length	Tokens	Vocabularies
Korean	Initial	412,317	11.6	4,782,063	389,752
	MA and WSD		20.1	8,287,635	68,719
Vietnamese			14.5	5,958,096	39,748

4.2. Machine Translation Experiments

From the bilingual corpus, we randomly extracted 2000 sentence pairs for development, 2000 sentence pairs for evaluation, and the rest for training. We built Vietnamese-to-Korean MT systems to translate Vietnamese sentences into Korean sentences, with different input forms, to compare the performance of NMT with contextual embedding using other methods, as follows:

- Baseline: building NMT without the BERT model, using Vietnamese sentences with word segmentation and original Korean sentences;
- KrUTagger: building NMT without the BERT model, using Vietnamese sentences with word segmentation and Korean sentences with MA and WSD;
- VnPOS: building NMT without the BERT model, using Vietnamese sentences with word segmentation and POS tagging and Korean sentences with MA and WSD; and
- VietBERT+NMT: building NMT with the Vietnamese BERT model, using Vietnamese sentences with word segmentation and POS tagging and Korean sentences with MA and WSD.

4.2.1. NMT

In our Vietnamese–Korean NMT systems, the baseline KrUTagger tool and the VnPOS MT system are implemented, based on the NMT model in OpenNMT [13]. NMT is a sequence-to-sequence encoder–decoder model with an attention mechanism. The encoder is a bi-directional RNN, which uses a context-free model (word2vec) to convert input sentences into input vectors. Using a bi-directional RNN, the encoder generates the corresponding representation of the input sentence. The decoder is also an RNN; it takes the representation of the input sentence, the previous hidden state, and the initial output word prediction, and provides a new output word prediction. The attention mechanism creates a shortcut that directly links the words in the target language with the corresponding words of attention in the source language [41].

4.2.2. BERT-Fused Model

Our VietBERT+NMT MT system is based on the BERT-fused model for supervised learning [3]. Unlike other free-based embedding models, such as word2vec, BERT is a context-based embedding model. Therefore, a word will be created by BERT in different contexts, using dynamic vector embedding. In the BERT model, the input sentence is fed into the encoder. The encoder uses a multi-head attention layer to understand the context of each given word and returns the representation of each word, based on context.

In this research, the structure of NMT with the BERT-fused model is demonstrated in Figure 7.

The BERT-fused translation model is based on the NMT model. The Vietnamese BERT-based model is used to extract the representations for an input sentence. Through the attention mechanism, the representations are then fused with the layers of both the encoder

and the decoder in the machine translation model. We used the Vietnamese BERT-based model with the dropout ratio set to 0.3; the embedding dimension, FNN dimension, and the number of layers in the model were set at 512, 1024, and 6, respectively. Before being inserted into MT models, all words in sentences of the parallel corpus were split into sub-words by BPE [42].

Figure 7. The BERT-fused model.

4.3. Results

We used BLEU, METEOR, and TER scores to evaluate the accuracy of the systems. BLEU is the most common method for assessing the accuracy of MT systems. It indicates how similar the candidate text is to the reference text. METEOR is another method of evaluating the similarity between candidate text and reference text. In addition, TER is used to measure the amount of correction the translator needs to produce a perfect translation. The ranges of BLEU and TER are from 0 to 100; the range of METEOR is from 0 to 1. A high BLEU score or METEOR score means that the MT system has higher accuracy, but a high TER score usually indicates lower accuracy.

The result of the experiment is shown in Table 6.

Table 6. The Vietnamese–Korean translation results.

Systems	BLEU	METEOR	TER
Baseline	13.52	0.115	71.46
KrUTagger	25.10	0.189	64.03
VnPOS	26.81	0.197	61.07
VietBERT+NMT	28.22	0.209	58.53

Table 6 shows that annotated WA and WSD for Korean sentences increased the quality of MT, as shown in the BLEU, METEOR, and TER scores. The quality of machine translation for Vietnamese also increased with tagging of POS. With WA and WSD tagging for Korean, POS tagging for Vietnamese, and the use of Vietnamese BERT to create representations for input sentences, Vietnamese–Korean MT produced the best results.

4.4. Discussion

In the KrUTagger system, the MA task reduces the number of rare words by parsing words into sub-words. For example, the sentence “*tôi chơi bóng đá giỏi*” (I play football well) in Vietnamese can be translated into “*na-neun chug-gu-leul jal-han-da*” or “*na-neun chug-gu-leul jal-hab-ni-da*” or “*na-neun chug-gu-leul jal-hae-yo*”. If the candidate sentence is “*na-neun chug-gu-leul jal-han-da*” and the reference sentence is “*na-neun chug-gu-leul jal-hae-yo*”, the meanings of the candidate and the reference are similar, but the evaluation method understands that “*jal-hae-yo*” and “*jal-han-da*” are different. Therefore, the result of the MT system is lower. With MA, the “*jal-hae-yo*” is split into “*jal*”, “*hae*” and “*yo*”; “*jal-han-da*” is split into “*jal*”, “*han*”, and “*da*”. When comparing two of these words, the first token of the two words is the same, thus leading to a better result under the MT system. In addition, the WSD task tags words with differing sense codes, depending on their meaning in different contexts. For instance, the word “*sseu-da*” in Korean has several meanings, such as “*write*”, “*wear*”, or “*bitter*”. When translating a sentence containing the word “*sseu-da*”, the MT system confuses the meaning of the word “*sseu-da*” in certain contexts. Our WSD tool converts “*sseu-da*” into “*sseu-da_01*”, “*sseu-da_02*”, or “*sseu-da_03*”, depending on the context. By adding more information to Korean words, the Vietnamese–Korean MT system provides better results.

In addition, in the VnPOS system, there are numerous polysemous words in Vietnamese. Many words have different meanings or diverse grammatical categories, depending on their context. For example, the word “*học*” can be a verb (“to learn”) or it can be a noun (“learning”), depending on the sentence. To overcome this ambiguity, annotation POS is used to add information to Vietnamese sentences. The word “*học*” is transferred into “*học | V*” or “*học | N*”. The POS tagging indicates the syntactic function of words, thus enabling the MT system to find words in the target language more easily, with the same meanings and functions. As a result, POS enables the machine to translate more accurately.

In the VietBERT+NMT system, the Vietnamese BERT model was used to extract representation for Vietnamese sentences. Unlike context-free models, such as word2vec, BERT can recognize the meaning of words according to context, and generate the corresponding embedding for different contexts. For example, when generating the representation of the word “*em bé*” (“baby”) and the word “*nó*” (“it”) in the sentence “*em bé thích ăn kem vì nó ngon*” (“the baby eats ice cream because it is delicious”), context-free models generate different representations for the words “*em bé*” and “*nó*” because they are different words. In contrast, contextual embedding models learn that, in terms of meaning, the word “*nó*” is the same as “*em bé*”. Therefore, contextual embedding models generate almost similar representations for the word “*em bé*” and the word “*nó*”. In addition, the output features of the BERT model are fully exploited, and BERT generates a better representation than the context-free word-embedding model of NMT. Further, the pre-trained features of BERT were fused with NMT, enabling the NMT model to take advantage of BERT’s features. Using the attention mechanism, this superior representation is then bridged to each layer of the encoder and decoder of the NMT model. As a result of clear input, the decoder of the NMT model generates better target sentences.

Our experiment also showed that using the Vietnamese BERT model in combination with NMT leads to a significant improvement in the quality of the Vietnamese–Korean translation system by 1.41 BLEU points and 2.54 TER points. Furthermore, using BERT also improves the results of POS tagging, which is an annotation of the Vietnamese data. For these reasons, the combination of the Vietnamese BERT model and NMT enhances the accuracy of Vietnamese–Korean machine translation.

5. Conclusions and Future Work

In this study, we have generally described a transformer model and a transformer-based language model (BERT). We applied POS tagging, which is significantly improved by BERT, to Vietnamese sentences in the Vietnamese–Korean bilingual corpus. We carried out experiments on Vietnamese–Korean machine translation based on supervised

NMT, with the BERT model for Vietnamese connected to the NMT model. MT results show that the BERT model significantly enhances the quality of the Vietnamese–Korean translation system.

In the future, because data related to low-resource languages such as Vietnamese or Korean are very rare, we plan to experiment with machines that are related to Korean or Vietnamese languages using unsupervised translation and semi-supervised translation. In addition, we plan to incorporate the Korean BERT model to enhance the quality of the Korean–Vietnamese MT systems.

Author Contributions: Data curation, V.-H.V. and Q.-P.N.; formal analysis, V.-H.V.; funding acquisition, C.-Y.O.; methodology, V.-H.V. and Q.-P.N.; project administration, C.-Y.O.; resources, V.-H.V. and Q.-P.N.; software, V.-H.V.; validation, V.-H.V., C.-Y.O., Q.-P.N. and E.V.T.; writing—original draft, V.-H.V., Q.-P.N. and E.V.T.; writing—review and editing, C.-Y.O. and E.V.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This work was supported by a grant from the Institute for Information & Communications Technology Planning & Evaluation (IITP), funded by the Korean government (MSIT) (No. 2013-0-00131, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186.
- Zhu, J.; Xia, Y.; Wu, L.; He, D.; Qin, T.; Zhou, W.; Li, H.; Liu, T.-Y. Incorporating BERT into Neural Machine Translation. *arXiv* **2020**, arXiv:2002.06823.
- Nguyen, D.Q.; Tuan Nguyen, A. PhoBERT: Pre-Trained Language Models for Vietnamese. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 1037–1042.
- Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. *arXiv* **2020**, arXiv:1909.11942.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
- Joshi, M.; Chen, D.; Liu, Y.; Weld, D.S.; Zettlemoyer, L.; Levy, O. SpanBERT: Improving Pre-Training by Representing and Predicting Spans. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 64–77. [[CrossRef](#)]
- Le, H.; Vial, L.; Frej, J.; Segonne, V.; Coavoux, M.; Lecouteux, B.; Allauzen, A.; Crabbé, B.; Besacier, L.; Schwab, D. FlauBERT: Unsupervised Language Model Pre-Training for French. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; European Language Resources Association: Marseille, France, 2020; pp. 2479–2490.
- Scheible, R.; Thomczyk, F.; Tippmann, P.; Jaravine, V.; Boeker, M. GottBERT: A Pure German Language Model. *arXiv* **2020**, arXiv:2012.02110.
- Cui, Y.; Che, W.; Liu, T.; Qin, B.; Wang, S.; Hu, G. Revisiting Pre-Trained Models for Chinese Natural Language Processing. In Proceedings of the Findings of the Association for Computational Linguistics, EMNLP 2020, online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 657–668.
- Lee, S.; Jang, H.; Baik, Y.; Park, S.; Shin, H. KR-BERT: A Small-Scale Korean-Specific Language Model. *arXiv* **2020**, arXiv:2008.03979.
- Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2016**, arXiv:1409.0473.
- Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 103–111.
- Klein, G.; Kim, Y.; Deng, Y.; Senellart, J.; Rush, A. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, British Columbia, Canada, 28 July 2017; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 67–72.

14. Hieber, F.; Domhan, T.; Denkowski, M.; Vilar, D.; Sokolov, A.; Clifton, A.; Post, M. Sockeye: A Toolkit for Neural Machine Translation. *arXiv* **2018**, arXiv:1712.05690.
15. Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; Auli, M. Fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 48–53.
16. Graça, M.; Kim, Y.; Schamper, J.; Khadivi, S.; Ney, H. Generalizing Back-Translation in Neural Machine Translation. In Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers), Florence, Italy, 1–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 45–52.
17. Johnson, M.; Schuster, M.; Le, Q.V.; Krikun, M.; Wu, Y.; Chen, Z.; Thorat, N.; Viégas, F.; Wattenberg, M.; Corrado, G.; et al. Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 339–351. [[CrossRef](#)]
18. Bojar, O.; Federmann, C.; Fishel, M.; Graham, Y.; Haddow, B.; Koehn, P.; Monz, C. Findings of the 2018 Conference on Machine Translation (WMT18). In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, Brussels, Belgium, 31 October–1 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 272–303.
19. Östling, R.; Tiedemann, J. Neural Machine Translation for Low-Resource Languages. *arXiv* **2017**, arXiv:1708.05729.
20. Tong, A.N.; Diduch, L.L.; Fiscus, J.G.; Haghpanah, Y.; Huang, S.; Joy, D.M.; Peterson, K.; Soboroff, I.M. Overview of the NIST 2016 LoReHLT Evaluation, Machine Translation Special Issue NLP in Low Resource Language. *Mach. Transl.* **2018**, *32*, 11–30. [[CrossRef](#)]
21. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv* **2016**, arXiv:1609.08144.
22. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional Sequence to Sequence Learning. In Proceedings of the 34th International Conference on Machine Learning, Volume 70, Sydney, NSW, Australia, 6–11 August 2017; pp. 1243–1252. [[CrossRef](#)]
23. Huang, J.-X.; Lee, K.-S.; Kim, Y.-K. Hybrid Translation with Classification: Revisiting Rule-Based and Neural Machine Translation. *Electronics* **2020**, *9*, 201. [[CrossRef](#)]
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: New York, NY, USA, 2017; pp. 5998–6008.
25. Vu, V.-H.; Nguyen, Q.-P.; Shin, J.-C.; Ock, C.-Y. UPC: An Open Word-Sense Annotated Parallel Corpora for Machine Translation Study. *Appl. Sci.* **2020**, *10*, 3904. [[CrossRef](#)]
26. Nguyen, Q.-P.; Vo, A.-D.; Shin, J.-C.; Tran, P.; Ock, C.-Y. Korean-Vietnamese Neural Machine Translation System with Korean Morphological Analysis and Word Sense Disambiguation. *IEEE Access* **2019**, *7*, 32602–32616. [[CrossRef](#)]
27. Snover, M.; Dorr, B.; Schwartz, R.; Micciulla, L.; Makhoul, J. *A Study of Translation Edit Rate with Targeted Human Annotation*; Association for Machine Translation in the Americas: College Park, MD, USA, 2006; p. 9.
28. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; Association for Computational Linguistics: Philadelphia, PA, USA, 2002; pp. 311–318.
29. Cho, S.W.; Lee, E.-H.; Lee, J.-H. Phrase-Level Grouping for Lexical Gap Resolution in Korean-Vietnamese SMT. In *Computational Linguistics*; Hasida, K., Pa, W.P., Eds.; Springer: Singapore, 2018; Volume 781, pp. 127–136. ISBN 978-981-10-8437-9.
30. Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; et al. Moses: Open Source Toolkit for Statistical Machine Translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, Prague, Czech Republic, 23–30 June 2007; Association for Computational Linguistics: Prague, Czech Republic, 2007; pp. 177–180.
31. Banerjee, S.; Lavie, A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; Association for Computational Linguistics: Philadelphia, PA, USA, 2005; pp. 65–72.
32. Sudharsan, R. *Getting Started with Google BERT*; Packt Publishing Ltd.: Birmingham, UK, 2021; ISBN 978-1-83882-159-3.
33. Nguyen, D.Q.; Vu, T.; Nguyen, D.Q.; Dras, M.; Johnson, M. From Word Segmentation to POS Tagging for Vietnamese. In Proceedings of the Australasian Language Technology Association Workshop 2017, Brisbane, Australia, 7–8 December 2017; pp. 108–113.
34. Nguyen, L.T.; Nguyen, D.Q. PhoNLP: A Joint Multi-Task Learning Model for Vietnamese Part-of-Speech Tagging, Named Entity Recognition and Dependency Parsing. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations, online, 6–11 June 2021; Association for Computational Linguistics: Philadelphia, PA, USA, 2021; pp. 1–7.
35. Nguyen, D.Q.; Verspoor, K. An Improved Neural Network Model for Joint POS Tagging and Dependency Parsing. In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Brussels, Belgium, 31 October–1 November 2018; Association for Computational Linguistics: Philadelphia, PA, USA, 2018; pp. 81–91.

36. Shin, J.C.; Ock, C.Y. Korean Homograph Tagging Model Based on Sub-Word Conditional Probability. *KIPS Trans. Softw. Data Eng.* **2014**, *3*, 407–420. [[CrossRef](#)]
37. Na, S.-H.; Kim, Y.-K. Phrase-Based Statistical Model for Korean Morpheme Segmentation and POS Tagging. *IEICE Trans. Inf. Syst.* **2018**, *E101.D*, 512–522. [[CrossRef](#)]
38. Matteson, A.; Lee, C.; Kim, Y.; Lim, H. Rich Character-Level Information for Korean Morphological Analysis and Part-of-Speech Tagging. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; Association for Computational Linguistics: Philadelphia, PA, USA, 2018; pp. 2482–2492.
39. Min, J.; Jeon, J.-W.; Song, K.-H.; Kim, Y.-S. A Study on Word Sense Disambiguation Using Bidirectional Recurrent Neural Network for Korean Language. *J. Korea Soc. Comput. Inf.* **2017**, *22*, 41–49. [[CrossRef](#)]
40. Kang, M.Y.; Kim, B. Word Sense Disambiguation Using Embedded Word Space. *J. Comput. Sci. Eng.* **2017**, *11*, 32–38. [[CrossRef](#)]
41. Luong, T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-Based Neural Machine Translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; Association for Computational Linguistics: Philadelphia, PA, USA, 2015; pp. 1412–1421.
42. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; Association for Computational Linguistics: Philadelphia, PA, USA, 2016; pp. 1715–1725.