Regular Article

Attention-Based BiGRU Model for Real-time Sign Language Translation Applications

Sam X. Nguyen[1], **Tien T. Tran**[2]

[1] Faculty of Information Technology, HCMC University of Technology and Education, Ho Chi Minh city, Vietnam
[2] Department of Computing, Greenwich University (FPT University), Ho Chi Minh city, Vietnam

Correspondence: Sam, X. Nguyen, samnx@hcmute.edu.vn
Communication: received 01 February 2024, revised 13 June 2024, accepted 25 June 2024
Online publication: 04 July 2024, Digital Object Identifier: 10.21553/rev-jec.364

Abstract– **Sign language applications provide an important key to solving communication problems for deaf community and normal hearing people. Current research problem usually focuses on improving communication access between deaf and hearing people. In this study, we consider real-time communication context from deaf to hearing people, and thus we propose an attention-based bidirectional gated recurrent unit (A-BiGRU) model which demonstrates on trading-off among an precision performance, and computational efficiency which includes training time, testing time, and system resources on extended the American Sign Language Gloss (E-ASLG-PC12) dataset. The results shown that our proposal has a significant performance improvement in term of training time, testing time, system resources, comparing to attention-based bidirectional long-short term memory (A-BiLSTM), and the other moderns of sequence to sequence models. Moreover, precision performance of our proposal model achieve closer to that of the complex architecture, A-BiLSTM. Thus, we believe that our proposed model is a suitable and potential candidate for real-time translation applications as well as and lower computational devices when they solve the communication problems from deaf to normal hearing people direction.**

Keywords– **computational efficiency (CE), attention-based bidirectional gated recurrent unit (A-BiGRU), sign language translation applications (SLTA).**

1 Introduction

Recently, sign language translation applications (SLTA) have become an important key to helping deaf community and normal hearing people communicate in two ways and the communication gap can be solved with the development of sign language translation applications on smart devices. In order to enable real-time interaction between deaf and normal hearing people communicate in two ways, sign language translation applications will be installed or embedded on smart devices. By this way, it can bring the translators to every deaf person to communicate with normal people at anytime and anywhere. Though many SLT applications and researches have been developed many real-time sign language translators on smart devices, there are several challenges in current research related to improvement of automatic sign language production (SLP), and thus, it is necessary to doing research and investigating solutions for the issues.

Advanced studies on sign language translation tasks, such as SLR and SLP [1]. Key components of the tasks not only focus on the advanced architecture of neural machine translation for improving precision performance but also pay attention on computational efficiency in terms of training time, testing time, system resources, and etc. While SLR [2, 3] focuses on detecting gestures and recognizing the signs, then convert them to the text form, SLP [4, 5] concentrates on translating sign language into natural language. In particle, SLP is a difficult and challenging task, because producing sign language to spoken language has either lack of a specific grammar and structure standard or the target devices may have not strong resources and datasets to synthesis spoken sentence from sequence of sign glosses.

The amount of studies on SLP are mainly based on artificial neural networks (ANN), and recurrent neural network (RNN) [6], a type of ANN, is widely used to solve the SLP problems. In RNN, the most significant important features are hidden states, which allow RNN simply estimate ahead prediction of small input sequence through conditional probability. This explain the reason why RNN can solve the short sequence text. In additional, long short-term memory (LSTM) and gated recurrent unit (RGU) [7], special types of RNN, can process long sequence text and preserve amount of information with smaller parameters and lower computational cost. The models are knows as sequence-to-sequence (seq2seq) models. The most significant important features of the seq2seq models that they perform the SLP tasks efficiently and the performance of the seq2seq models is better than the other machine learning models, especially in term of bilingual evaluation understudy (BLUE) score [8, 9]. In [10], a survey on precision performance for sign language machine

translation tasks shown that performance of the seq2seq models are also higher *BLEU* score than performance of machine learning–based methods.

Recently, an attention mechanism [11] was born to solve the long sentence problem for SLTA. Instead of accessing entire the long sentence, the attention mechanism allow seq2seq model pay attention on relevant information by using an alignment process to compare source and target. The attention mechanism create a shortcut to solve fixed length context vector problem in the traditional seq2seq architecture.

In this study, we propose an attention-based Bidirectional GRU (A-BiGRU) model to solve the real-time and computational cost problems for the translating task, gloss-to-text, for SLP. The main contributions of this paper are summarized as follows:

- ASLG-PC12 [12] dataset is well known for translation applications because it is constructed on pairs of set of English written text (EWT) and set of American sign language gloss (ASLG). However, a major bottleneck comes from a lack of parallel corpus pairs when we map from ASLG to natural language EWT. We solved the problem by enriching ASLG, as well as representing corpus between the sets to meet the specific requirements for building real-time applications.
- We implemented an BiGRU architecture, where the first GRU layer processes the input sequences in forward direction, and the other GRU layer in a backwards direction for both for encoder decoder. The architecture, namely BiGRU, allows it capture and handle long sentences. It is worth to note that the extension architecture of BiGRU allows improving precision performance as well as reducing computational cost when they work with the attention mechanism.
- We investigated and analyzed performance of our proposal architecture with the other similar powerful seq2seq architectures, including A-BiLSTM, BiGRU, and BiLSTM to evaluate performance in terms of precision, and computational efficiency. As results, A-BiGRU offers potential trade-off between the precision performance in *BLEU* score and computational cost in training time, testing time, and system resources.

The rest of the paper is organized as follows: in section 2, we present theoretical backgrounds of previous studies related to using datasets, seq2seq models and AL. Section 3 describes in detail our proposal. Next, we show the experimental results and discussions in section 4. Finally, we draw conclusions and future works in section 5.

2 Related Works

2.1 ASLG-PC12 Dataset

Various aspects of SLT must be considered to solve translating sign language into a natural language problems. The first studies was proposed a seq2seq model for extracting "Gloss" features from video frames to the corresponding spoken or text with the PHOENIX 2014T dataset [13] and ASLG-PC12 dataset [12]. The ASLG-PC12 dataset involves 887710 pairs of EWT and ASLG, thus, it is the most popular for translation applications. However, the limitation of data in ASLG comparing to natural spoken language in EWT may reduce performance of seq2seq models. In order to solve the problem, we enrich ASLG set from various sources, including *Handspeak*, *StartASL*, and *Lifeprint* [14]. Hence, the extended glosses are embedded into dataset [12]. On the other hand, we prepare the best pairs of the sets by preprocessing steps, then we extract text from EWT into sentences. Moreover, we remove punctuation marks and preposition to minimize size of the set. The In order to keep maximum of information for translating models, a process of normalization and tokenization for both sets are proposed, thus we search and merge the words in the sets if they are widely used. It is necessary to note that mapping parallel corpus from the glosses to text may need a grammatical rules to improve performance of learning models. However, a basic structure in written English, including subject (S), verb (V), and object (O) can be represented in different ways in ASLG, such as, OSV, OVS, SOV, VOS, VSO, and etc. [15], it may yield amount of parallel data. Hence, we filtered and kept SVO and OSV formats for ASLG set. As a result, an extended ASLG-PC12 (E-ASLG-PC12), including 810000 high parallel corpus can support the real-time translation task.

2.2 BiLSTM

Long-short term memory (LSTM) architectures are proposed for long term learning dependence data [7]. Because LSTM can hold information for an extended period, and thus, it is well-suited for sequential data. LSTM networks include the chain of memory blocks, which is called LSTM cells. Each LSTM cell consists of three gates, which are the input gate, the forget gate, and the output gate. The "input/update" gate controls the flow of information and decides what information is added to LSTM cell. Forget gate allows what information will be removed from LSTM cell. The output gate controls which information will go out of LSTM cell. The Architecture of the LSTM cell is described in Figure 1.

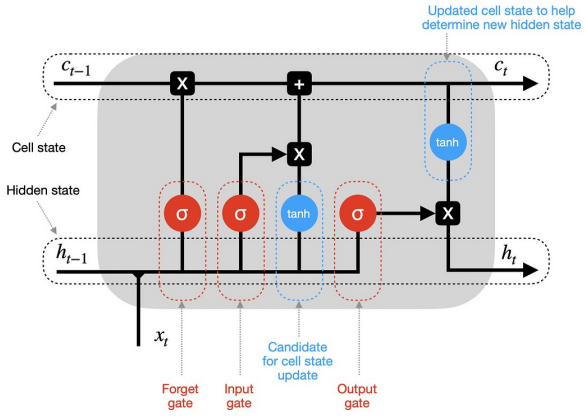

Figure 1. The LSTM cell.

At the time t, the process of LSTM can be expressed as follows: first, the forget gate f_t is computed in equation (1) which take previous memory h_{t-1} as an input. Because of the sigmoid activation function σ, the outcome of f_t is bounded between 0 and 1, thus, if value of f_t is close to 0, it removes the previous state h_{t-1} or if value of f_t is close to 1, it keeps the previous state h_{t-1}. Mathematically, the calculation of LSTM cell can be summarized as follows

$$f_t = \sigma([W_t * (h_{t-1}, x_t] + b_f) \tag{1}$$

Then, the "input/update" gate i_t is computed in equation (2). The outcome value tells the cell which new information to store in the internal cell state

$$i_t = \sigma(W_t * [h_{t-1}, x_t] + b_i). \tag{2}$$

We compute k_t, and present the new input:

$$k_t = tanh(W_k * [h_{t-1}, x_t] + b_k). \tag{3}$$

Next, the cell state c_t is calculated from the forget gate f_t and the previous cell state c_{t-1}. The result is summed with update state k_t

$$c_t = c_{t-1} \odot f_t + i_t \odot k_t. \tag{4}$$

Next, output gate o_t is computed in equation

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o). \tag{5}$$

Finally, output of $LSTM$ cell h_t is computed in equation

$$f(n) = \begin{cases} o_t \odot tanh(c_t) & \text{if } t > 0 \\ 0 & \text{if } t < 0 \end{cases}. \tag{6}$$

where, $\odot$ is *Hadamard* multiplication, x_t is the input sequence at the time t, and b_f, b_i, b_k, b_o are biases. $tanh$ and σ are activate functions, respectively.

Bidirectional long-short term memory (BiLSTM) is a extended version of LSTM. Unlike LSTM, input sequences in BiLSTM are transmitted in both directions, allowing it to use information from both sides. The architecture of BiLSTM consists of two LSTM that process input sequences in both forward and backward directions. The first one receives the input sequences in one direction, while the other one gets input sequences in the opposite direction. BiLSTM returns a probability vector as output, and the final output is a combination of both of these probabilities in various ways, such as mean, sum, multiply which can be represented as follows

$$p_t = p_t^f + p_t^b, \tag{7}$$

where, p_t, p_t^f, and p_t^b are the final probability vector of the network, the probability vector from the forward LSTM network and the probability vector from the backward LSTM network. The architecture of BiLSTM are presented in Figure 2.

2.3 BiGRU

GRU is an advanced model which introduced as an alternative method to learn patterns from sequential data [7]. The GRU model has the same design concept

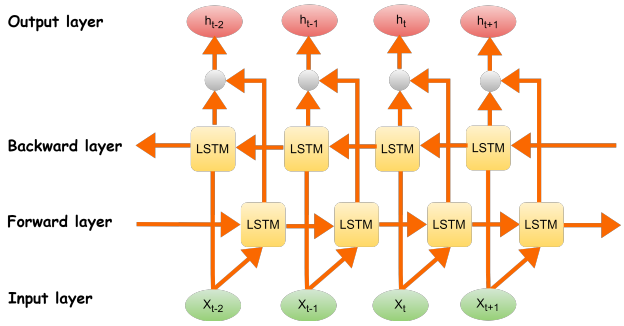

Figure 2. The architecture of BiLSTM.

as the LSTM model but it has simple architecture of cell where it only has an update gate and a reset gate as shown in Figure 3. This helps it focuses on filtering unimportant information and organizing the flow of information efficiently. Moreover, the simplified architecture makes it enable in computing. The amount of information received is essentially determined at each time step, so the GRU is capable of remembering time patterns over a longer period of time than other models.

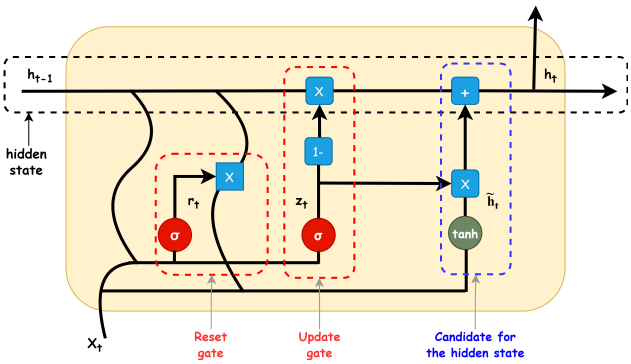

Figure 3. The GRU cell.

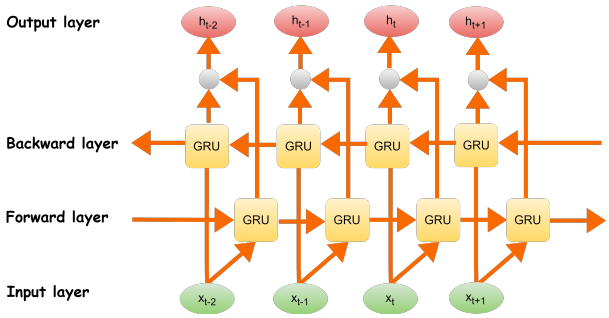

Figure 4. The architecture of BiGRU.

GRU consists of two types of gates, including update gate and reset gate. The update gate decides to keep and let information through it. It takes the current input x_t at the each time step and the hidden state of the previous h_{t-1} to produce outputs as a value either between 0 and 1. on the other hand, the reset gate responsible for controlling the past information s relevant to the computation of the current output. Like update gate, it takes the input x_t and the previous hidden state h_{t-1} to produce outputs as a value between 0 and 1. Mathematically, the calculation of GRU cell can be summarized as follows:

$$z_t = \sigma([W_z * (h_{t-1}, x_t] + b_z), \quad (8)$$

$$r_t = \sigma(W_r * [h_{t-1}, x_t] + b_r), \quad (9)$$

$$\hat{h_t} = tanh(W_h * [t_t \odot h_{t-1}] + b_h), \quad (10)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h_t}, \quad (11)$$

where, W_z, W_r, and W_h are the weighted matrices, and b_z, b_r, and b_h are the bias. $\odot$ is *Hadamard* multiplication. The *sigmoid* activation function is specified by σ and it is used to calculate the update gate from equation (8) and the reset gate from equation (9). The *sigmoid* activation function to be in the range [0,1]. If the output from the *sigmoid* function is 1, we do not use past information, whereas if the output is 0, then, only new information is kept. On the other hand, if the value of the reset gate is 0, we do not use the previous value to calculating h_t, and if the value of the reset gate is 1, we use the previous value for the calculating h_t.

BiGRU, a variant of GRU, is designed with two GRU units running in parallel. Each GRU unit processes data in a separate direction. the first one forwards and the other one reverses. This allows BiGRU to capture information from both sides of the sequential data and information, thereby the architecture can improve accuracy. Moreover, the structure of BiGRU can connect the forward hidden layer and the backward hidden layer to the same output, thus it fully extracts the temporal characteristics of the language data sequence. The architecture of BiGRU are described in Figure 4. Mathematically, the computing of the backward hidden layer can be summarized as follows:

$$\overrightarrow{h} = \phi(W^f_{xh} * x_t + W^f_{hh} * \overrightarrow{h}_{(t-1)} + b^f_h), \quad (12)$$

$$\overleftarrow{h} = \phi(W^b_{xh} * x_t + W^b_{hh} * \overrightarrow{h}_{(t-1)} + b^b_h), \quad (13)$$

$$h_t = [\overrightarrow{h}; \overleftarrow{h}], \quad (14)$$

where, the hidden states of the forward and backward layers can be represented as $\overrightarrow{h}$ and, $\overleftarrow{h}$, respectively. [;] denotes the merging of hidden states of the forward and backward layers.

2.4 Attention Mechanisms

The attention mechanism is a best way to enhance the performance of seq2seq architecture. In general, the attention mechanism is located between the encoder layer and the decoder layer, and the first goal of the attention mechanism is to align the hidden state of the encoder and decoder in RNN. Recently, the attention mechanism has been proposed by Bahdanau et al. [11]. As illustrated in Figure 5, *context vector* Z_t, $t \in (1, 2, ..., n)$, depends on a sequence of (h_1,h_2,...,h_n) and hidden status s_{t-1}. Mathematically, it can be calculated as follows

$$Z_t = \Sigma^n_{j=1} s_{tj} * h_j. \quad (15)$$

At each time step, *weight* s_{tj} of each h_j is calculated by equation (16) as follows

$$s_{tj} = \frac{exp(d_{tj})}{\Sigma^n_{k=1} exp(d_{tk})}, \quad (16)$$

where, $d_{jt} = a(s_{t-1}, h_j)$ is an *alignment* model that shows the degree of correspondence between inputs in position j and output in position $t-1$. By calculating attention weights at every time, the attention mechanism can computer a separate context vector. Thus, the attention mechanism enables to pay attention on relevant information and ignore the other information.

In [16], attention scores are computed by Luong et al., the attention scores of the work are different to the previous work because they measure the attention scores by using multiplicative attention directly. The attentions cores are express as follows:

$$d_{jt} = h^n_j * s_{t-1}, \quad (17)$$

$$d_{jt} = h^n_j * W^n_m * s_{t-1}, \quad (18)$$

$$d_{jt} = W^n_\alpha tanh(W_h * h_j + U_h * s_{t-1}), \quad (19)$$

where, W_α, W_h, W_m, and U_h are weight parameters.

In order to trade-off between the *BLEU* score and system resources, we implement the first one in our proposed model.

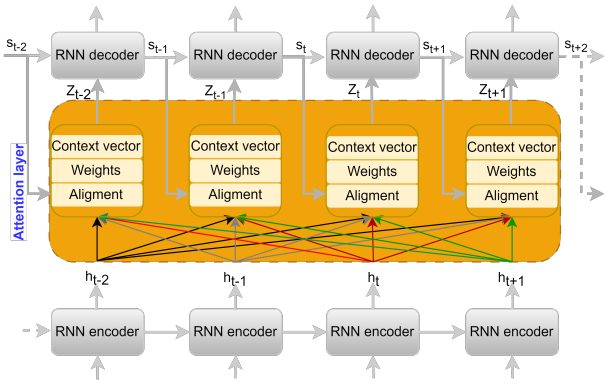

Figure 5. The architecture of attention mechanism.

3 Methodology

3.1 Study Approach

In order to study on gloss to text translation in SLP, we applied the powerful seq2seq models to show how to implement the approach to real-time sign language translation applications. First, We investigate and analyze performance of A-BiGRU. Then, we compare our proposal with the other models in term of *BLEU* score, training time $Tr(S)$, testing time $Te(s)$, and system resources $Re(GB)$. The other models include BiLSTM, BiGRU, and A-BiLSTM. As shown in Figure 6, a study on the gloss to text architecture is investigated. The encoder layer has input vectors x_{t-2}, x_{t-1}, x_t, and x_{t+1}. At each time step t, encoder output, vector Z_t, is passed to decoder layer. The context vector is calculated in equation (15) and it is used with the previous hidden state s_{t-1} of the decoder to compute the new hidden state s_t and output y_t. It is worth to note that all

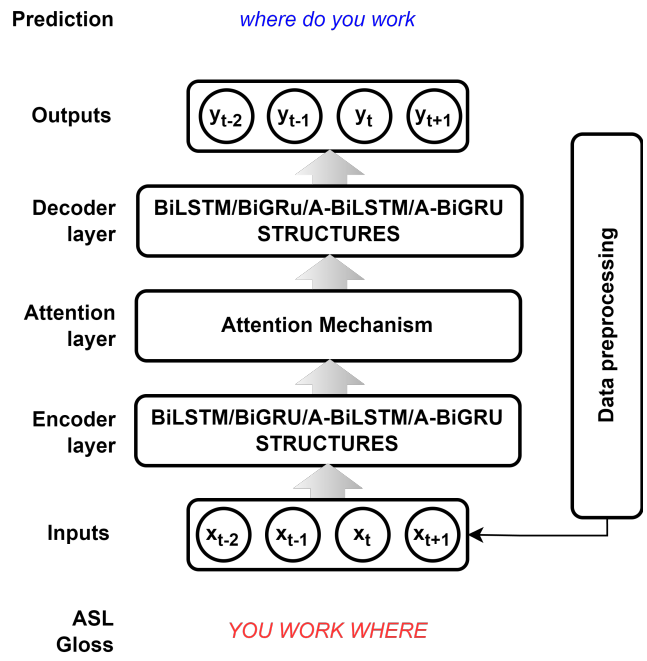

Figure 6. The investigated gloss to text architecture.

the mathematical equations for the models with the attention mechanism have been solved in [11].

The investigated models, including BiLSTM, BiGRU, A-BiLSTM, and A-BiGRU, have been described in previous sections, where either the LSTM cell or GRU cell is configured as the core component. In general, there are many activation functions such as *ReLU*, *tanh*, *sigmoid*, and etc., may use for the models as well as variant type of attention mechanisms [11, 16] have been introduced to the attention layer. Because we are looking for a simple architecture where computational efficiency as well as *BLEU* score are taken into account. Therefore, we implemented a model based on BiGRU architecture and the attention mechanism [11] to achieve both computational efficiency as well as *BLEU* score.

3.2 Metrics

The *BLEU* [8, 9] is a metric for measurement of the different between predicted translation of sentence to references of sentences. Thus, it is used to evaluate machine translation systems. Traditionally, *BLEU* score has a translation scale of 0 to 1, while the score closes to 1 means that the translated sentence is exactly the same as the reference sentence, the score closes to 0 means that the translated sentence is different to reference sentence. In general, the *BLEU* score is expressed on a scale of 1 to 100 and it uses multiple of precision scores of n-gram (n = 1, 2, 3, 4). Mathematically, the *BLEU* score are calculated as follows

$$BLEU = min(1, \frac{l_{candidate}}{l_{reference}})\Pi_{n=1}^{4}(p_i)^{\frac{1}{4}}, \qquad (20)$$

where, $l_{candidate}$ and $l_{reference}$ are the length of translated sentence and the length of reference sentences, and p_i is the n-gram precision where n is up to a maximum order of four. The precision metric measures the number of words in the candidate sentence that also occur in the reference sentences.

4 Experimental Results and Evaluations

4.1 Experimental Environment

In this study, we used Google Colab Pro as the primary environment for model training. It offers a Tesla T4 GPU with 16GB of VRAM, along with a fast Intel(R) Xeon(R) CPU that speeds up the model training process. All deep learning methods are implemented using Python language version 3.10.7, Tensorflow library version 2.3.1, and Keras version 2.4.3. In order to investigate and analyze performance of A-BiGRU, the E-ASLG-PC12 dataset is divided in two parts which 80 percents were used for training, and the remaining 20 percents were used to test translation results. To examine the models, We use the $categorical\ cross-entropy$ function to optimize the model to predict the probability distribution. Regularization are proposed in the study to retaining part of the training data for the test set. We chose a batch size of 128 during training to optimize GPU memory usage while ensuring model accuracy. The training was conducted over 200 epochs.

4.2 Experimental Results and Discussions

As shown in Table I, few translation results of A-BiGRU are presented. There are several parameters that refer to 1) ground truth (ground tru.) is the truth sequence text that can be used to compare with outcome, 2) gloss sequences (gloss seque.) is a sequence of glosses than can be translated to sequence text, and 3) translation as target outcome of the model. The sample tests in Table I shown that mapping among gloss sequence, ground truth and outcome of proposed model is highly match to original data.

While the model can reconstruct the simple context, the complex grammatical structure must be investigated. Intuitively, the gloss to text translations not only base on architecture of models but also depend on ground truth where variety and available datasets can give close meaning for translating. Larger and more available datasets enable the algorithms as well as impact on the performance of models.

Precision performance and computational cost are the core issues of the work. In the first one, BLUE scores of of BiLSTM, BiGRU, A-BiLSTM, and A-BiGRU are investigated. It is worth to noted that the higher BLUE score value, the better translation quality. As show in table II, The BLUE score of A-BiGRU is significant better than BiLSTM and BiGRU but it is slight lower than A-BiLSTM. It means that the attention mechanism allows our proposed model to pay more attention to the relevant information and words in long sentences, and hence improve the BLUE score. Because A-BiGRU has a simpler architecture than A-BiLSTM, the explain that BiGRU is slight lower BLUE score than A-BiLSTM.

However, a simpler architecture, which can make A-BiGRU faster to train and test, as well as less computational cost. As shown in Table II, training time

Table I
Samples of Text to Gloss A-BiGRU Model with E-ASLG-PC12 Dataset

Parameters	The generated output samples
gloss sqe.	BATH NEED YOU
ground tru.	you need to take a bath
translation	you need to take a bath
gloss sqe.	LONG SEE NO HOW YOU
ground tru.	long time no see you been doing time
translation	long time no see you been doing time
gloss sqe.	YOU WORK WHERE
ground tru.	where do you work?
translation	where do you work?
gloss sqe.	YOUR NEW CAR COLOR WHAT
ground true	what color is your new car?
translation	what color is your new car?

Table II
Performance of Investigated Models

Models	BLEU	Tr	Re	Te	Wgt
BiGRU	37.52	1215	2.4	30	14.7
BiLSTM	38.26	1533s	3.0	35	15.8
A-BiLSTM	46.34	1948	4.2	67	16.8
A-BiGRU	45.63	1662	3.7	55	15.8

Tr(seconds), system resources *Re(Gigabyte)*, and time for testing samples *Te(seconds)* of A-BiGRU achieved significant improvement. Especially, both time for testing samples and training time of A-BiGRU are significant reduced, comparing to A-BiLSTM. In order to support fully for the core idea that the fewer weights (Wgt) are produced, the training and testing time will be improved. We also present comparison the *Wgt(Millions)* among the models in Table II.

In order to show overall performance comparison among the models, we visualize the comparisons in Figure 7, where system resources *Re(Gigabyte)* and the *BLEU* score of each model is evaluated in separate presentation. It is worth to note that overall performance of each model can be observed the gaps related to *BLEU* scores and computational efficiency, and thus, a complete evaluation of performance could be drawn. For our purpose, the simple architecture, A-BiGRU, make it overall better than the other models. The results are meaning to practical applications of machine translation.

4.3 Android Applications

In order to test our proposed model, A-BiGRU, a prototype of real-time translation application is generated. Because we just test how does it work and does it provide real-time capabilities, We focus on effectively real-time translator. However, it can provide graphic user interface so that we can receive real-time feedback performance in several scenarios.

5 Conclusion and Future Works

In this study, we presented and compared the performance of seq2seq models, including BiGRU, BiLSTM, A-BiLSTM, and A-BiGRU. In particular, we focus on

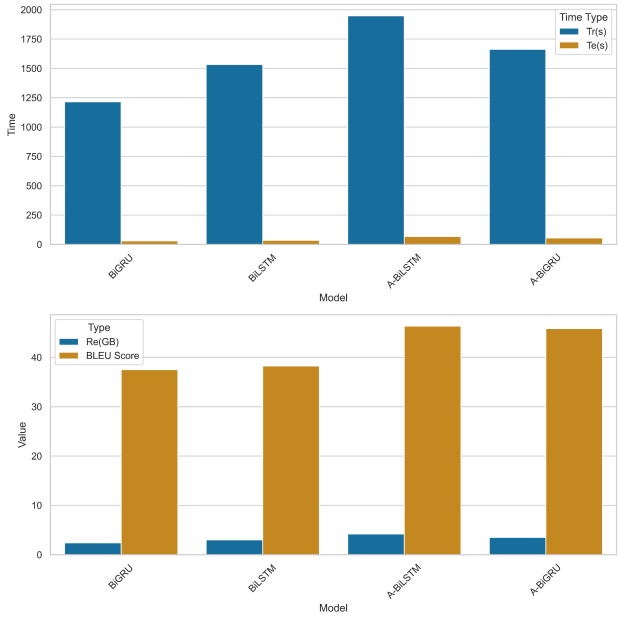

Figure 7. Comparison with the state of the art.

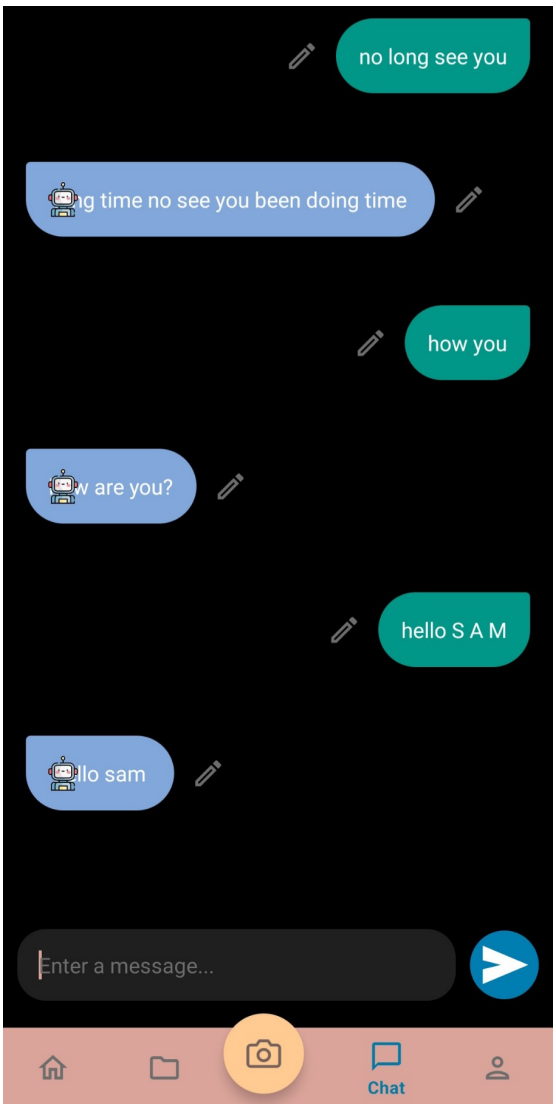

Figure 8. SLTA on Android.

the A-BiLSTM and A-BiGRU to find out the best solutions for real-time translations applications. Experimental results shown that A-BiGRU has simpler architecture than A-BiLSTM and it can handle significant improvement than A-BiLSTM in terms of training time *Tr(Seconds)*, system resources *Re(Gigabyte)* , and time for testing samples *Te(Seconds)*, while still providing nearly the same *BLEU* score comparing to A-BiLSTM. Thereby, it has the potential to be integrated into real-time sign language translation applications.

In the future works, it is worth to noted that a proposed model must be observed the gaps, which are important features to improve overall models for real-time applications and thus, we will still focus on the simple architecture of models with suitable datasets to provide the best alternative for improving real-time sign language translation applications.

Acknowledgment

The paper was published as part of REV-ECIT 2023 conference.

References

[1] R. Rastgoo, K. Kiani, S. Escalera, V. Athitsos, and M. Sabokrou, "All You Need In Sign Language Production," *arXiv preprint arXiv:2201.01609*, 2022.

[2] U. Farooq, M. S. M. Rahim, N. Sabir, A. Hussain, and A. Abid, "Advances in machine translation for sign language: approaches, limitations, and challenges," *Neural Computing and Applications*, vol. 33, no. 21, pp. 14 357–14 399, 2021.

[3] 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," *arXiv preprint arXiv:1406.1078*, 2014.

[4] D. Guo, W. Zhou, A. Li, H. Li, and M. Wang, "Hierarchical Recurrent Deep Fusion Using Adaptive Clip Summarization for Sign Language Translation," *IEEE Transactions on Image Processing*, vol. 29, pp. 1575–1590, 2020.

[5] J. Huang, W. Zhou, H. Li, and W. Li, "Attention-Based 3D-CNNs for Large-Vocabulary Sign Language Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, p. 2822–2832, sep 2019. [Online]. Available: https://doi.org/10.1109/TCSVT.2018.2870740

[6] 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," *arXiv preprint arXiv:1406.1078*, 2014.

[7] T. Ananthanarayana, P. Srivastava, A. Chintha, A. Santha, B. Landy, J. Panaro, A. Webster, N. Kotecha, S. Sah, T. Sarchet *et al.*, "Deep learning methods for sign language translation," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 14, no. 4, pp. 1–30, 2021.

[8] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[9] C. Callison-Burch, M. Osborne, and P. Koehn, "Re-evaluating the role of BLEU in machine translation research," in *Proceedings of the 11th conference of the european chapter of the association for computational linguistics*, 2006, pp. 249–256.

[10] A. Núñez-Marcos, O. Perez-de Viñaspre, and G. Labaka, "A survey on Sign Language machine translation," *Expert Systems with Applications*, vol. 213, p. 118993, 2023.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[12] A. Othman and M. Jemni, "English-asl gloss parallel corpus 2012: Aslg-pc12," in *Proceedings of the sign-lang@ LREC 2012*. European Language Resources Association (ELRA), 2012, pp. 151–154.

[13] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, pp. 108–125, 2015.

[14] Q. Feng, "Automatic American Sign Language Imitation Evaluator," Ph.D. dissertation, The Ohio State University, 2016.

[15] X. Zhang and K. Duh, "Approaching sign language gloss translation as a low-resource machine translation task," in *Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL)*, 2021, pp. 60–70.

[16] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

Sam X. Nguyen received the Bachelor of engineering in Communication Engineering from PTIT, Hanoi, Vietnam in 2002, the Master of science in Information and Communications Engineering from the Andong National University, and the Doctor of Philosophy in Computer Engineering from Korea University (Seoul campus), Republic of Korea in 2009 and 2016, respectively. He is currently a faculty member of FIT, Ho Chi Minh City University of Technology and Education. His research interests include Distributed Computing, Real-time Embedded Systems, Artificial Intelligence for Internet of Things, and Cyber Security.

Tien T. Tran will receive 1st rank of the Bachelor of degree in Information Technology from Greenwich University (FPT University), Ho Chi Minh City campus, Vietnam in 2024, respectively. His research interests include artificial intelligence, machine and deep learning, and natural language processing. He attended several domestic and global competitive in AI and robotics and involved in student and scientific associations.