BỘ GIÁO DỤC
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

NGUYỄN THỊ BÍCH ĐIỆP

DANH MỤC CÔNG TRÌNH CÔNG BỐ

LUẬN ÁN TIẾN SĨ
NGHIÊN CỨU VÀ PHÁT TRIỂN PHƯƠNG PHÁP TIẾP CẬN DỰA TRÊN CẤU TRÚC VÀ THỐNG KÊ TRONG DỊCH TỰ ĐỘNG NGÔN NGỮ KÝ HIỆU VIỆT NAM

Ngành Khoa học máy tính

Mã số: 9 48 01 01

Hà Nội, 2023

Special characters of Vietnamese sign language recognition System based on Virtual Reality Glove

Diep Nguyen Thi Bich[1], Nghia Phung Trung[1],
Thang Vu Tat[2], and Lam Phi Tung[2]

[1]Thai Nguyen University of Information and Communication Technology,
Thai Nguyen, Vietnam
[2]Institute of Information Technology, Vietnam Academy of Science and Technology,
Hanoi, Vietnam
{ntbdiep,ptnghia}@ictu.edu.vn,{vtthang,tunglam}@ioit.ac.vn

Abstract. In this paper, we introduce a method of recognition numbers and special characters of Vietnam sign language. We address a development of a glove-based gesture recognition system. A sensor glove is attached ten flex sensors and one accelerometer. Flex sensors are used for sensing the curvature of fingers and the accelerometer is used in detecting a movement of a hand. Depending on the hands postures, i.e., vertical, horizontal, and movement, sign language of numbers and special characters can be divided to group 1, 2, and 3, respectively. . Firstly, the hands posture is recognized. Next, if the hands posture belongs to either group 1 or group 2, a matching algorithm is used to detect a number or one of special characters. If the posture belongs to group 3, a dynamic time warping algorithm is applied. The use of our system in recognizing Vietnamese sign language is illustrated. In addition, experimental results are provided.

Keywords: Recognition, Vietnamese sign language, Number, Special Characters, Vituarl Reality Glove.

1 Introduction

There are about 360 millions of deaf people in the world, equivalent to 5% of the total world population [17]. Most of deaf people are poverty because of restricted educational opportunities and the poor communication. Today, researchers are increasingly paying attention to construction tools translate sign language - the language of the deaf, especially the field of investigation of hand shape and gesture recognition because it is so useful in several applications, e.g., tele-manipulation, sign language translation, robotics [12], etc. In this paper, we aim to develop a glove-based gesture recognition system that allows recognizing Vietnamese sign language (VSL), performed by a user with a single hand, using

a data glove as an input device. We focus on the classification and recognition of gestures that represent Number and special characters of Vietnamese sign language. Among the vast variety of existing approaches for hand shape and gesture recognition, methods using sensing gloves have proven to be remarkably successful [1][9]. A survey of glove-based system and their applications is presented in [5]. Mehdi and Khan [11] used a sensor glove to capture the signs of American sign language (ASL) performed by a user and translate them into sentences of English language. In addition, artificial neural networks (ANNs) are used to recognize the sensor values coming from the sensor glove. ANNs have been used for both (static) postural classification [4] and gesture classification [6][16]. A data glove is used for recognition the Japanese alphabets [13], for the Chinese language [3], etc. Vietnamese vocabulary is more complicated than English alphabet system because of more signs for VSL in comparison with ASL. Special characters are only available in Vietnamese. Bui and Nguyen [13][15] created 22 fuzzy rules to classify Vietnamese sign language postures. They used a sensing glove that is attached six accelerometers and a basic stamp microcontroller in recognizing Vietnam number and special characters sign language. In this paper, we aim to develop a glove-based gesture recognition system in which data glove are used in classification and recognition numbers and special characters in Vietnamese sign language. The glove has two main parts, i.e., sensors (flex sensors and an accelerometer) and a system of data processing and communication. Firstly, the hands posture is detected. Depending on the hands posture, i.e., vertical, horizontal, and movement, sign language of alphabets are divided into group 1, 2, and 3, respectively. In the next stage, if the hands posture belongs to either group 1 or 2, a matching algorithm is used to detect a letter. If the posture belongs to group 3, a letter is recognized by using a dynamic time warping algorithm (DTW). The system of data processing and communication (using microchip Atmega32U) handles data from sensors and then transfer results achieved to PC through USB port. Software running on a PC receives data and then displays an animation of a gloves gestures and a letter recognized. The paper is organized as follows: our data set and sensing glove are introduced in Section 2. In Section 3, our recognition system of Vietnamese sign language is described in detail. Experimental results are presented in Section 4. Finally, conclusions are drawn in Section 5.

2 The data set and sensing glove

2.1 The data set

Our data set are numbers and specials character in Vietnamese sign language (VSL). The numbers performances in Vietnamese sign language are different to other such as: American (ASL), Chinines (CSL). The expressing numbers in VSL, similar ASL and CSL with number 0 to 5, diffirent with number 6 to 9. Vietnamese alphabet system is more complicated than English alphabet system because more signs are needed for VSL in comparison with ASL. Some

Vietnamese typing tool as Unikey, if you want to type specail character, you must use some letters: w, s, f, r, x, j or use number 1 to 9.

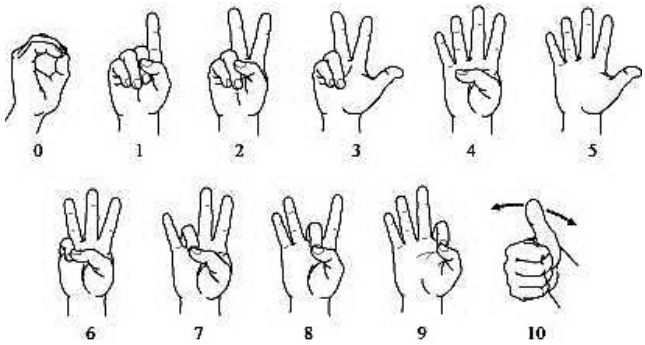


Fig. 1. Numbers in America sign language.

Fig. 2. Numbers in Vietnamese sign language.

Several specials character in Vietnamese are: acute (’), grave accent (‘), question mark(?), tilde (~). They are only available in Vietnamese.
In this paper, we are going to assess on dataset a list of the following : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and acute (’), grave accent (‘), question mark(?), tilde (~).

2.2 A sensing glove

The use of the sensor has been presented in [8]. Our sensing glove has two main parts, i.e., sensors (ten flex sensors and one accelerometer) and a system of data processing and communication. There are two flex sensors in one finger. Sensors are fixed in one point then they can move when fingers bent. An accelerometer and a system of data processing and communication (microchip Atmega32U is used) are assembled in one small board that can be immobilized with a users wrist. Flex sensors are passive resistive devices that can be used to detect bending or flexing. Flex sensors are analog resisters and work as analog voltage di-

viders. Inside the flex sensor are carbon resistive elements within a thin flexible substrate. When the substrate is bent, the sensor produces a resistance output relative to the bend radius. An output of a flex sensor is an analog. Ten outputs of flex sensors are connected to ten ADC channels of microchip Atmega32U.

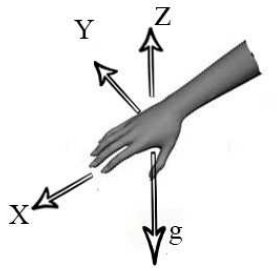


Fig. 3. X-Y-Z axis of an accelerometer in which the X-axis coincide with the direction of a hand, the Z-axis is taken to be vertical when the hand is in the horizontal plane, and g is a gravitational acceleration vector.

Here, we use an accelerometer, i.e., ADXL345. A function block diagram of ADXL345 is shown in [8]. The ADXL345 is a small, thin, low power, three-axis accelerometer with high resolution (13-bit) measurement up to 16g. The ADXL345 is well suited for mobile applications. It measures the static acceleration of gravity in tilt-sensing application, as well as dynamic acceleration resulting from motion or shock. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I2C digital interface. Fig. 3 depicts X-Y-Z axis of an accelerometer in which the X-axis coincide with the direction of a hand, the Z-axis is taken to be vertical when the hand is in the horizontal plane. An accelerometer returns magnitudes of the projection of vector g to X-Y-Z axis, respectively. These digital output data is accessible through a SPI of Atmega32U.

3 Recognition of Vietnamese numbers and special characters sign language

In this Section, we present our algorithm for classification and recognition of Vietnamese numbers and special characters sign language. The data set that we selected can be divided to three groups depending on the hands postures: i) Group 1: when the hands posture is vertical, which consists of the postures of numbers, i.e., 0, 1, 2, 3, 4, 5, 6, 8 and 9. ii) Group 2: when the hands posture is horizontal, i.e., 7. iii) Group 3: when the hand makes a move, which consists of the postures of letters, i.e., acute (’), grave accent (‘), question mark(?) and tilde

$\tilde{()}$. An accelerometer returns values of the projection of a gravitational acceleration vector, g, to 3-axis acceleration sensor. Let (Ax, Ay, Az) be magnitudes of the projection of vector g to X-Y-Z axis, respectively. Let S be a vector of 13 measurement parameters from sensors attached on the glove and is denoted by:

$$\mathbf{S} = [f_{11}\ f_{12}\ f_{21}\ f_{22}\ f_{31}\ f_{32}\ f_{41}\ f_{42}\ f_{51}\ f_{52}\ A_x\ A_y\ A_z]^T$$

where $i = \overline{1,5}$

are values measured from two flex sensors attached on finger i, starting from a thumb to a little finger. Based on signals from sensors attached on the glove, our system recognizes Vietnamese alphabet sign language by a user with a single hand, using the data glove as an input device. Here, flex sensors are used for sensing the curvature of fingers and the accelerometer is used in recognizing the movement of a hand. Firstly, the hands postures are divided into three groups. Next, if the posture belongs to either group 1 or group 2, the matching algorithm is used to detect a letter. Given a sampling measurement vector, we calculate a list of errors between the sampling measurement vector and a template vector of each letter belonging to group 1 (or group 2). An output is a letter corresponding to a letter that has the smallest error in the list. If the posture belongs to group 3, the DTW is applied to detect a letter. DTW is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed. Here, DTW is used to find an optimal alignment between the sequences of movement of the hand and the sequences of template movement of sign language of letters under certain restrictions. Our algorithm scheme for classification and recognition is presented in Fig. 6.

3.1 Classification

Assuming that we have n sampling measurement vectors that are recorded continuously from time t0 to tn, Tt, t = [t0,t1,,tn]. The variance of Ax is determined as follows:

$$\mathrm{Var}(A_x) = \frac{1}{n}\sum\nolimits_{h=t}^{t+n}\left(A_x^h - \bar{A}_x\right)^2 \tag{1}$$

where $\bar{A}_x$ is the expected value, i.e.,

$$\bar{A}_x = \frac{1}{n}\sum\nolimits_{h=t}^{t+n} A_x^h \tag{2}$$

If the variance of Ax, , is large than constant , the hand is movable. If the variance of Ax is smaller than , the hand is immobile and the hands posture is determined as follows:

$$Hand's\ posture = \begin{bmatrix} Horizontal\ if\ A_x \in (-60,\ 0] \\ Vertical \quad if\ A_x \in [-137,\ -100] \\ NULL \quad Otherwise \end{bmatrix} \tag{3}$$

In this paper, n = 8, =3. Fig. 5 presents an example of the hands postures depending on the values of Ax.

Fig. 4. An example of the hands gestures corresponding to special charactes of the Vietnamese sign language.

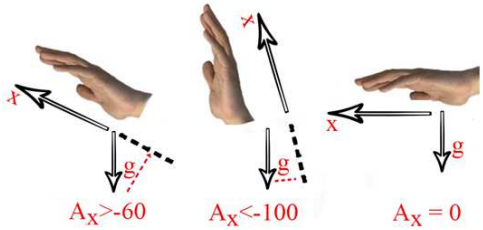


Fig. 5. The hands postures depending on the values of Ax.

3.2 Recognition

+ If the hand is immobile, we use the template matching method to detect a letter for both cases: the hands posture is vertical or horizontal. Here, we do not use parameters of an accelerometer because it is used for the classification stage. Let $\mathbf{T}^k = [f^k_{11}\ f^k_{12}\ f^k_{21}\ f^k_{22}\ f^k_{31}\ f^k_{32}\ f^k_{41}\ f^k_{42}\ f^k_{51}\ f^k_{52}\ 0\ 0\ 0]^T$ be a template vector of letter k-th in group 1, where is the number of letters in group 1 $f^k_{ij},\ i \in [1,5], j \in [1,2]$ is the value measured from a flex sensor. Let be a sampling measurement vector at time t and is denoted by $\mathbf{S}^t$ be a sampling measurement vector at time t and is denoted by

$$\mathbf{S}^t = [f^t_{11}\ f^t_{12}\ f^t_{21}\ f^t_{22}\ f^t_{31}\ f^t_{32}\ f^t_{41}\ f^t_{42}\ f^t_{51}\ f^t_{52}\ A^t_x\ A^t_y\ A^t_z]^T \tag{4}$$

Let $\Delta_{t,k}$ be the error of $\mathbf{S}^t$ and $\mathbf{T}^k$ and is calculated as follows:

$$\Delta_{t,k} = \frac{\sum_{i\in[1,5],j\in[1,2]} \left(f^t_{ij} - f^k_{ij}\right)}{10} \tag{5}$$

$\arg \min_{k\in[1,\,N_{C1}]} \Delta_{t,k}$ is calculated and then return letter k-th in group 1. The recognition of letters in group 2 is performed similarly. If the hand is movable, the DTW is applied to recognize a letter. Let $\hat{\mathbf{S}}^n = (S^0, ..., S^n)$ be a set of n sampling measurement vectors from time t_0 tot_n, where is a measurement vector at time $t \in (t_0, t_n)$

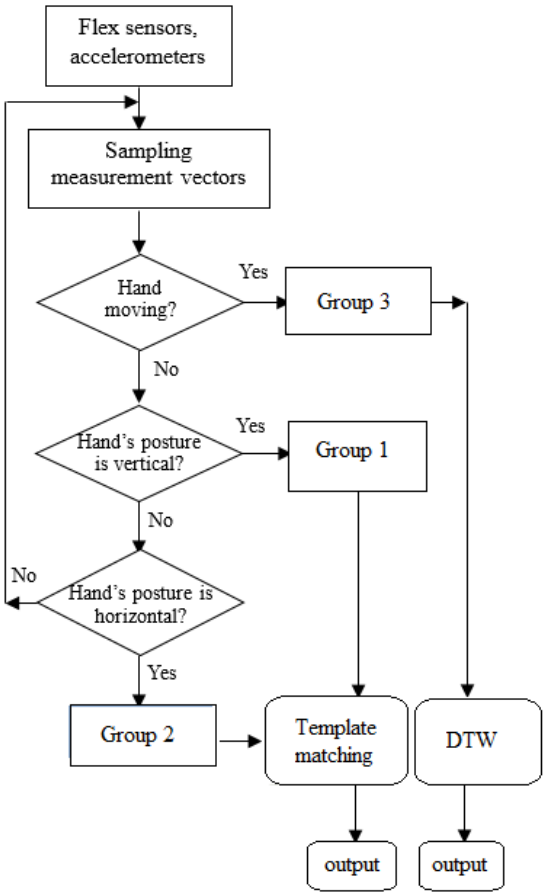


Fig. 6. An algorithm scheme for classification and recognition of Vietnamese numbers and special characters sign language.

$$\mathbf{S}^t = [f_{11}^t\ f_{12}^t\ f_{21}^t\ f_{22}^t\ f_{31}^t\ f_{32}^t\ f_{41}^t\ f_{42}^t\ f_{51}^t\ f_{52}^t\ A_x^t\ A_y^t\ A_z^t]^T \tag{6}$$

Let $\hat{\mathbf{T}}^{k,m} = (T^{k,t0}, .., T^{k,tm})$, $k = 1, ..., N_{C3}$, be a set of m template vectors from time t_0 tot_n, where is a template vector at time $t \in (t_0, t_n)$ of letter k-th in group 3, where N_{C3} is the number of letters in group 3.

$$\mathbf{T}^{k,t} = [f_{11}^{k,t}\ f_{12}^{k,t}\ f_{21}^{k,t}\ f_{22}^{k,t}\ f_{31}^{k,t}\ f_{32}^{k,t}\ f_{41}^{k,t}\ f_{42}^{k,t}\ f_{51}^{k,t}\ f_{52}^{k,t}\ A_x^{k,t}\ A_y^{k,t}\ A_z^{k,t}]^T \tag{7}$$

Let $\Delta(x, y)$ be the error of$\mathbf{S}^x$ and $\mathbf{T}^{k,y}$, $x \in (t_0, t_n)$,$y \in (t_0, t_m)$ and is calculated as follows:

$$\Delta(x,y) = \frac{\sum_{i\in[1,5],j\in[1,2]} \left(f_{ij}^{x} - f_{ij}^{k,y}\right)}{10} \tag{8}$$

Without lost of generality, assuming that t=1, we have $x = \overline{1,n}$ and $y = \overline{1,m}$. Time-normalized distance is determined as follows:

$$D(\hat{\mathbf{S}}^n, \hat{\mathbf{T}}^{k,m}) = \frac{g(n,m)}{n+m} \tag{9}$$

where g(n,m) is calculated recursively as follows:

$$g(1,1) = \Delta(1,1)$$

$$g(x,1) = g(x-1,1) + \Delta(x,1)$$

$$g(1,y) = g(1,y-1) + \Delta(1,y)$$

$$g(x,y) = \min \begin{pmatrix} g(x,y-1) + \Delta(x,y) \\ g(x-1,y) + \Delta(x,y) \\ g(x-1,y-1) + \Delta(x,y) \end{pmatrix} \tag{10}$$

Finally, $\arg \min_{k\in[1,\, N_{C3}]} D(\hat{\mathbf{S}}^n, \hat{\mathbf{T}}^{k,m})$ is calculated and then return letter k-th in group 3.

4 Experimental results

In this Section, the use of our system in recognizing Vietnamese numbers and special characters sign language is illustrated. We developed a soft-ware running on a PC in which an animation of the sensing glove and a character detected are shown. Several samples are tested for each letter of the Vietnamese alphabet. Precision rates of sign language recognition for letters are shown in Table 1. The testing process includes steps: Step 1: We had sign language expert that wear Virtual Reality Glove. Her hand movements under the sign language on our data set. Step 2: Our group monitoring process on step 1. Based on that we get 50 data types for each symbol is labeled. Data for samples run through the algorithm to obtain the labels. Step 3: Calculated% of the results obtained, coinciding with the label is correct, the difference with the wrong label available. Thus producing the results in table 1. Several characters are recognized with precision rate 100%, i.e., 2, 3, 4, 5, 7. Four characters, i.e., acute (’), grave accent (‘), question mark(?), tilde (˜) in category 3, have low precision rates because the hand is rotated around Z-axis.

Table 1. Precision rates of sign language recognition for numbers and special characters of Vietnam sign language

Character	Testing number	Precision number	Precision rate (%)
1	50	48	96
2	50	50	100
3	50	50	100
4	50	50	100
5	50	50	100
6	50	43	86
7	50	50	100
8	50	45	90
9	50	47	94
0	50	46	92
grave accent	50	30	60
acute	50	34	68
question mark	50	35	70
tilde	50	32	64

5 Conclusion

In this paper, we focus on recognition numbers and special characters in Vietnamese sign language. We design our system using a data glove that is attached ten flex sensors and one accelerometer. The recognition process has two stages, i.e., recognizing the hands posture and detecting numbers and special characters, respectively. Depending on the hands posture, either the matching algorithm or the DTW is used to detect a letter. The utility of our system in recognizing Vietnamese sign language is demonstrated. Precision rates of sign language recognition are reported. In future works, we aim to extend our glove-based gesture recognition system for complicated vocabulary in Vietnamese. In the future, we plan to develop the identification system is a large set of signs commonly used in Vietnam sign language. Thereby creating a complete system for the deaf aid.

References

1. B. Parton, Sign language recognition and translation: A multidiscipline approach from the field of artificial intelligence, Journal of Deaf Studies and Deaf Education, 11(11) pp. 94-101 (2006)
2. C. Wang, W. Gao, and S. Shan, An approach based on phonemes to large vocabulary Chinese sign language recognition, in Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. pp. 393398.(2002)
3. D. Xu, W. Yao, and Y. Zhang, Hand gesture interaction for virtual training of SPG, in Proc. Int. Conf. Artif. Reality Telexistence pp. 672676 (2006).
4. K. Murakami and H. Taguchi, Gesture recognition using recurrent neural networks, in Proc. Conf. Human Factors Comput. Syst pp. 237242 (1991).
5. L. Dipietro, A.M. Sabatini, and P. Dario, A survey of glove-based system and their applications, IEEE Trans. on Systems, Man, and CyberneticsPart C: Applications and Reviews, 38(4) pp. 461-482 (2008).

6. M.E. Cabrera, J.M. Bogado, L. Fermin, R. Acua, and D. Ralev, Glove-based gesture recognition system, Proc. of the 15th Int. Conf. on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Baltimore, MD, USA pp. 747-753 (2012).
7. M. Muller, Information Retrieval for Music and Motion, Chapter 4: Dynamic Time Warping, Springer Berlin Heidelberg pp. 69-84 (2007).
8. Lam T. Phi, T.T. Quyen Bui, Hung D. Nguyen, Thang T. Vu, A Glove-Based Gesture Recognition System for Vietnamese Sign Language, 15th International Conference on Control, Automation and Systems (ICCAS 2015) Oct. 13-16, in BEXCO, Busan, Korea (2015).
9. Ruiduo Yang, Sudeep Sarkar, Barbara L. Loeding, Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming. IEEE Trans. Pattern Anal. Mach. Intell. 32(3) 462-477 (2010).
10. R. Lockton, Hand-gesture recognition using computer vision, The 4th Year Project Report, Balliol College Oxford University, (2002).
11. S.A. Mehdi and Y.N. Khan, Sign language recognition using sensor gloves, Proc. of the 9th International Conference on Neural Information Processing, vol. 5 pp. 2204-2209 (2002).
12. T.T. Quyen Bui and K.-S. Hong, Evaluating a color-based active basis model for object recognition, Computer Vision and Image Understanding, 116 (11) 1111-1120 (2012).
13. T.D. Bui and L.T. Nguyen, Recognition of Vietnamese sign language using mems accelerometers, Proc. of the first International Conference on Sensing Technology, Palmerston North, New Zealand, Nov. 21-23 (2005) pp. 118-122.
14. T. Takahashi and F. Kishino, Hand gesture coding based on experiments using a hand gesture interface device, SIGCHI Bull., vol. 23, no. 2 pp. 6774, Apr (1991).
15. T.D. Bui and L.T. Nguyen, Recognition postures in Vietnamese sign language using mems accelerometers, The IEEE Sensors Journal, 7(5) pp. 707-712 (2007).
16. W. Gao, J. Ma, J.Wu, and C.Wang, Sign language recognition based on HMM/ANN/DP, Int. J. Pattern Recognit. Artif. Intell., vol. 14, no. 5 pp. 587602(2000).
17. `http://www.who.int/mediacentre/factsheets/fs300/en/`

A rule-based method for text shortening in Vietnamese sign language translation

Thi Bich Diep Nguyen[1], Trung-Nghia Phung[2], Tat-Thang Vu[3]

[1] Graduate University of Science and Technology, Ha Noi, Vietnam
[2] Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam
[3] Institute of Information Technology, Ha Noi, Vietnam
{ntbdiep, ptnghia}@ictu.edu.vn; vtthang@ioit.ac.vn

Abstract. Sign languages are natural languages with their own set of gestures and grammars. The grammar of Vietnamese sign language have significantly different features compared with those of Vietnamese spoken / written language, including the shortening, the grammatical ordering, and the emphasis. Natural language processing research on Vietnamese sign language including study on spoken / written Vietnamese text shortening into the forms of Vietnamese sign language is completely new. Therefore, we proposed a rule-based method to shorten the spoken / written Vietnamese sentences by reducing prepositions, conjunctions, and auxiliary words and replacing synonyms. The experimental results confirmed the effectiveness of the proposed method.

1 Introduction

The deaf communities in the world mostly communicate by performing gestures. The common used gestures were converted to sign languages since 18th century. After that, the sign languages have developed gradually and recognized as the official sign languages of the deaf communities of the countries. The sign language used by the deaf community of Vietnam is Vietnamese Sign Language (VSL).

Although sign languages share many similarities with spoken languages, there are some significant differences between sign and spoken / written languages in grammar and linguistic properties. As a result, VSL as well as other sign languages are natural languages with their own set of gestures and grammar. For instance, American Sign Language (ASL) has its own grammar system (its own rules for phonology, morphology, syntax, and pragmatics), separate from that of English [1].

Sign language translation (SLT) is the system translating written text to signs or / and signs to text. The adult deaf are quite easy to show what they mean to normal hearing people after their practice of sign and body languages every day. However, normal hearing people are very difficult to show their ideals to the deaf since they are rarely use body and sign languages. As a consequence, it seems that the translation side from text to signs is more helpful for the deaf than the side from signs to text.

Since sign languages are natural languages with their own grammar and linguistic properties, SLT requires researches in natural language processing (NLP). However,

there are just a few NLP researches in SLT in the world [2-6]. Especially, NLP research on VSL and Vietnamese SLT is completely new.

In [5], Humphries mentioned that shortening is one of the most important characteristics in ASL. What this means is text shortening is a critical step in NLP researches for SLT.

One recent research of Gouri Sankar Mishra and his colleagues proposed a NLP system to translate spoken English to Indian Sign Language (ISL) as shown in Figure 1, including a text shortening method as shown in Figure 2 [3]. The translation model follows a rule-based method in which a parser is used to parse the full English sentence and a dependency structure is identified from the parse tree. This structure represents the syntactic and grammatical information of a sentence. The shortened ISL sentence is generated from a bilingual ISL dictionary and a wordnet. From this shortened ISL sentence the corresponding ISL signs are displayed.

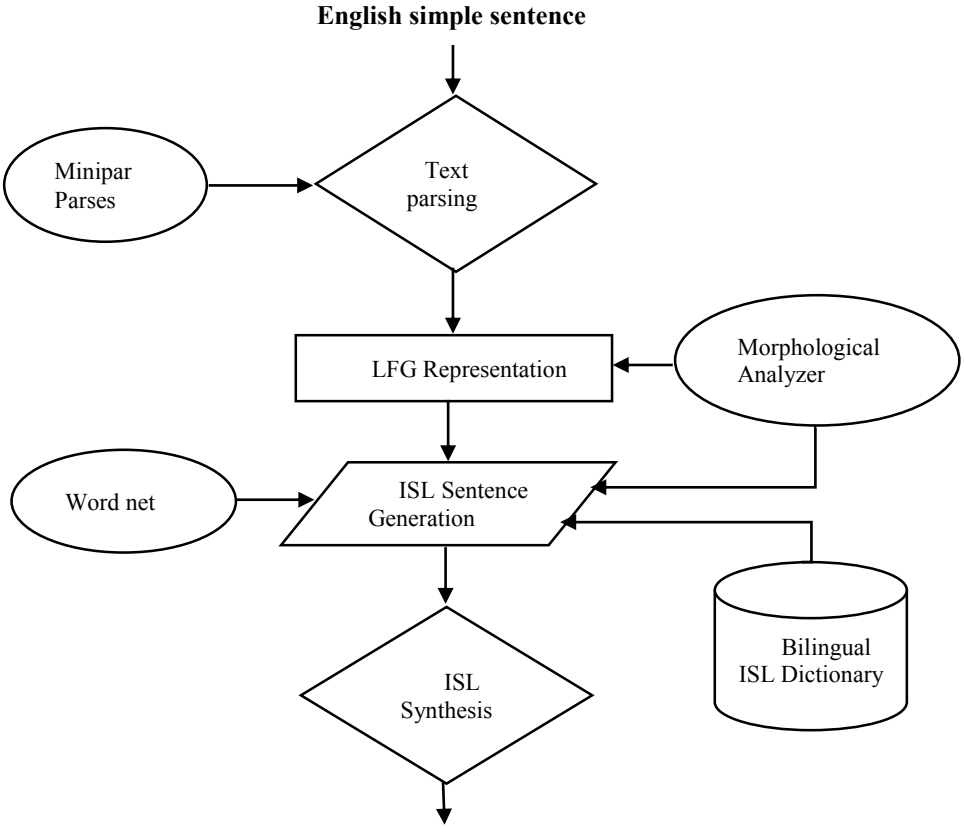


Fig. 1. Prototype machine translation system for ISL [3].

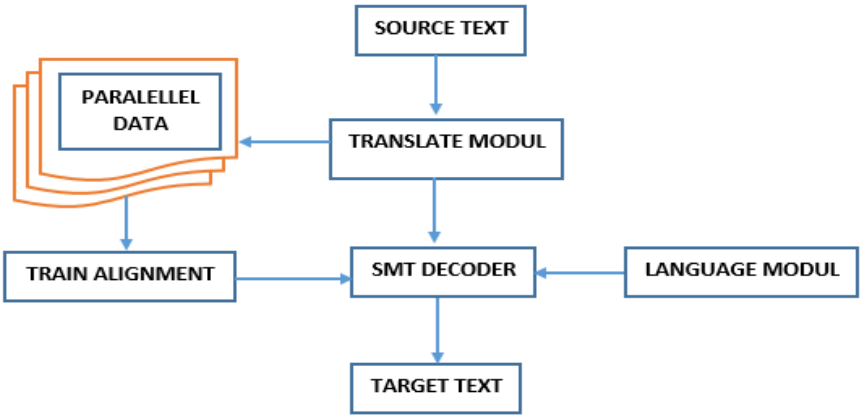


Fig. 2. : The architecture of the translation model for ISL [3].

Since NLP research on VSL and Vietnamese SLT is completely new, this paper proposed a method of spoken / written Vietnamese text shortening for Vietnamese SLT.

2 Linguistic fundamentals of the text shortening characteristic in VSL

VSL in the normal communication among the Vietnamese deaf has some basic characteristics. In [7], people show three most important characteristics of VSL. The first one is the shortening characteristic, in which the sentence of VSL is shorter than the sentence of the spoken / written Vietnamese language caused by the reduction of the prepositions and auxiliary words in the sentence. The second one is the grammatical ordering characteristic, in which grammatical ordering in VSL is different with that of spoken / written Vietnamese. The third one is the limited vocabulary characteristic of VSL. This characteristic is originated from the limited cognitive of the deaf. In [8], people show that there are three features in VSL, including the shortening, the grammatical ordering, and emphasis. The reduced components in VSL sentences are prepositions, conjunctions, and auxiliary words.

Table 1. Some examples of text shortening

Full sentences in spoken Vietnamese language (in Vietnamese)	**Reduced sentences in VSL (in Vietnamese)**
Viết bằng bút chì	Viết ~~bằng~~ bút chì
Tôi và anh đi học	Tôi ~~và~~ anh đi học
Anh ăn cháo hay ăn cơm?	Anh ăn cháo ~~hay~~ ăn cơm?
Mặc dầu trời mưa, tôi vẫn đi học	~~Mặc dầu~~ trời mưa, tôi ~~vẫn~~ đi học
Lấy hộ chị quyển sách	Lấy ~~hộ~~ chị quyển sách

Based on the characteristics of meaning and location appearing in the sentences, auxiliary words in Vietnamese can be divided into the following categories:

- Stressed auxiliary words: These words are used to emphasize words, phrases or sentences that accompany them. They are preceded by words or phrases that need emphasis. These are several Vietnamese words such as: cả, chính, đích, đúng, chỉ, những, đến, tận, ngay,...

These are some examples of stressed auxiliary words in Vietnamese:

+ Hai ngày sau, **chính** một số cảnh sát đã giải anh đi tối hôm trước lại quay về nhà thương Chợ Quán (Trần Đình Vân)

+ Nó mua **những** tám cái vé

+ Nó làm việc **cả** ngày lễ.

- Modal auxiliary words: These words are used to express emotions, moods, or attitudes. These words often indicate the purpose of the sentence (ask, order, exclaim ...). They stand at the end of the sentence to express the doubt, urge or exclamation. They also reveal the attitude or the feelings of the speaker, the writer. These are several Vietnamese words such as: à, ư, nhỉ, nhé, chứ, vậy, đâu, chăng, ừ, ạ, hả, hử....

These are some examples of modal auxiliary words in Vietnamese:

+ Chúng ta đi xem phim **nhé**?

+ Đã bảo **mà**!

+ Trời có mưa **đâu**?

- Exclamations words: These are words that directly express the emotions of the speaker. They cannot be used as emotional names, but as indications of emotions. They cannot be official part of a phrase or sentence, but can be separated from the sentence to form a separate sentence. They are often associated with an intonation or gesture, facial expressions or gestures of the speaker. Exclamation words can be used to call or answer (ơi, vâng, dạ, bẩm, thưa, ừ,...), can be used to express feelings of joy, surprise, pain, fear, anger,... (ôi ! trời ơi, ô, ; ủa, kìa, ái, ối, than ôi, hỡi ôi, eo ôi, ôi giời ôi,...). It can be said that exclamations words are used to express sudden, strong emotions of different types.

A text shortening algorithm involves grouping the above components into a dataset that is compared to the original text in the shortening process. With the linguistic bases analyzed above, we proposed a rule-based shortening method as present below.

3 The proposed text shortening method

The core ideal of the text shortening method proposed in this paper is the use of a vocabulary of all removable words in VSL including prepositions, conjunctions and modal auxiliary verbs.

Our currently VSL dictionary contains about 3000 words where each word corresponds to a distinct sequence of sign gestures. The number of words and phrases in this VSL dictionary is much smaller than that in spoken / written Vietnamese dictionary. The words appeared in Vietnamese dictionary but reduced in VSL dictionary are not only prepositions, conjunctions and modal auxiliary verbs but also synonyms. Therefore, in the proposed method, if words or phrases appeared in

Vietnamese dictionary but not in VSL dictionary, corresponding synonyms are searched in the VSL dictionary also.

The proposed method is described in Fig. 3 and explained step by step as follows:

Step 1. Use Vietnamese word segmentation with parsing tool Bikel, Vietnamese treebank, and VSL dictionary in the preprocessor to return the list of words and phrases.

Step 2. Find the labels of words and phrases.

- If the words and phrases are not found in the VSL dictionary, we find the corresponding synonyms and find the labels again.

- If the results are found, we move to Step 3, else we move to Step 4.

Step 3. Shorten the sentence by reducing the words or phrases with labels as prepositions, conjunctions and modal auxiliary verbs. Then, return the shortened sentence.

Step 4. Insert unfound words and phrases to a database. In the next steps for building a full VSL translator, each word or phrase in this database will be performed by pronouncing.

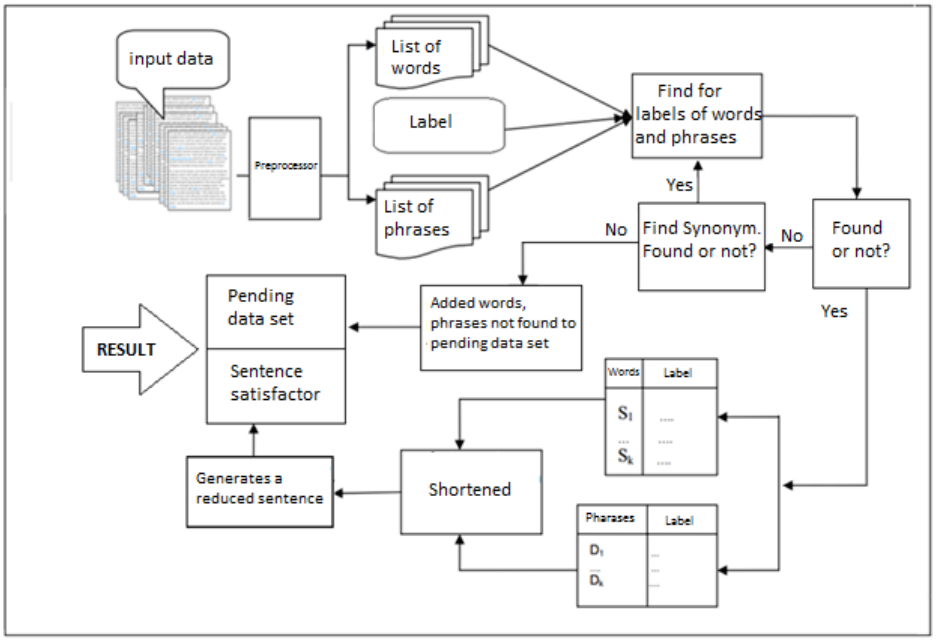


Fig. 3. Systematic diagram of the proposed method

4 Experiment Results

4.1 Evaluation method

BLEU is a method to evaluate quality of the documents automatically translated my machine, proposed by IBM in 2002 [9] and used as the primary evaluation measure for research in machine translation in [10]. The original ideal of the method is to compare two documents automatically translated by machine and manual translated by linguistic experts. The comparison is performed by statistical analyzing the coincidence of the words in the two documents that takes into account the order of the words in the sentences using n-grams. Specifically, BLEU scores are computed by statistically analyzing the degree of coincidence between n-grams of documents automatically translated by machine and the ones manual translated by high-quality linguistic experts [11].

BLEU score can be computed as follows [11]:

$$score = \exp\left\{\sum_{i=1}^{N} w_i \log(p_i) - \max\left(\frac{L_{ref}}{L_{tra}} - 1,0\right)\right\} \quad (1)$$

- $P_i = \frac{\sum_j NR_j}{\sum_j NT_j}$
- NR_j: the number of n- grams in segment *j* in the reference translation (by experts) with a matching reference co-occurrence in segment
- NT_j: the number of n- grams in segment *j* in the translation (by machine) being evaluated.
- $w_i = N^{-1}$
- L_{ref}: the number of words in the reference translation (by experts) that is closest in length to the translation being scored.
- L_{tra}: the number of words in the translation (by machine) being scored.

The value of *score* evaluates the correlation between the two translations by experts and machine, computed in each segment where each segment is the minimum unit of translation coherence. Normally, each segment is usually one or a few sentences. The n-gram co-occurrence statistics, based on the sets of n-grams for the test and reference segments, are computed for each of these segments and then accumulated over all segments. BLEU's output is always a number between 0 and 1. This value indicates how similar the candidate text is to the reference texts, with values closer to 1 representing more similar texts.

4.2 Evaluation results

We built a VSL dictionary with 3000 words and phrases. For evaluation, we used 200 simple sentences extracted from on the textbooks used in the schools for deaf children. After being translated (shortened) by using the proposed method, we

computed the BLEU scores between the translated sentences and the corresponding ones conducted by one expert in VSL.

The results of computed BLEU scores are shown in Fig. 3. The ratio of sentences correctly translated by using the proposed method (corresponding with BLEU score is one) is 97.5%. A few sentences incorrectly translated is caused by semantic ambiguity will be solved in our future researches.

Table 2. BLEU scores

ID sentence	L_{input}	NR_j	NT_j	L_{ref}	L_{tra}	BLEU Score
1	3	7	7	3	3	1.000
2	5	12	12	4	4	1.000
3	8	15	15	6	6	1.000
4	9	26	20	9	7	0.7515
5	5	14	14	5	5	1.0000
…	…	…	…	…	…	….
99	7	22	16	7	6	0.8465
100	8	24	24	8	8	1.0000
…	…	…	…	…	..	…
199	7	23	23	7	7	1.000
200	6	13	18	5	6	0.9762

5 Conclusions

NLP research on VSL and Vietnamese SLT is completely new. It is know that shortening is one of the most important features of VSL. In this paper, we proposed a rule-based method to shorten the spoken / written Vietnamese text into VSL forms by reducing prepositions, conjunctions, and modal auxiliary verbs and replacing synonyms. The experimental results show that the proposed method is efficient.

In the next researches, we will continue to study other issues on VSL and Vietnamese SLT.

Acknowledgments. This study was supported by the Ministry of Education and Training of Vietnam (project B2016-TNA-27).

References

1. Fromkin, V.: Sign language: Evidence for language universals and the linguistic capacity of the human brain. Sign Language Studies. 59 (1988) 115–127
2. Matthew P. Huenerfauth, “American Sign Language Natural Language Generation and Machine Translation Systems”, Technical Report Computer and Information Sciences University of Pennsylvania MS-CIS-03-32, September 2003.

3. Gouri Sankar Mishra, Ashok Kumar Sahoo and Kiran Kumar Ravulakollu, "Word based statistical machine translation from english text to indian sign language", ARPN Journal of Engineering and Applied Sciences, VOL. 12, NO. 2, 2017.
4. Dasgupta T., Basu A., "Prototype machine translation system from text-to-Indian sign language", Proceedings Of The 13th International Conference On Intelligent User Interfaces, Gran Canaria, Spain, pp. 313-316, 2008.
5. Humphries, T., & Padden, C. "Learning American sign language", Englewood Cliffs, N.J: Prentice Hall (1992).
6. Kar P., Reddy M., Mukherjee A. and Raina A.M. "INGIT: Limited Domain Formulaic Translation from Hindi Strings to Indian Sign Language", International Conference on Natural Language Processing (ICON), Hyderabad, India, 2007.
7. Pham Thi Coi, The process of language formation of the deaf children in Vietnam, PhD thesis, Institute of Linguistics, 1988 (in Vietnamese).
8. Vuong Hong Tam, Study the sign language of the deaf Vietnamese, Project report, Institute of Education Science of Vietnam, 2009 (in Vietnamese).
9. Papineni K., Roukos S., Ward T., Zhu Z-J, "BLEU: A method for Automatic Evaluation of Machine Translation", Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, p.p 311-318, 2001.
10. Hovy E.H.: Toward finely differentiated evaluation metrics for machine translation. Proceedings of the Eagles Workshop on Standards and Evaluation, Pisa, Italy, 1999
11. NIST report: Automatic evaluation of machine translation quality using N-gram co-occurrene statistics, 2002.

Some issues on syntax transformation in Vietnamese sign language translation

Thi Bich Diep Nguyen[†] and Trung-Nghia Phung[††],

Graduate University of Science and Technology, Ha Noi, Vietnam
Thai Nguyen University of Information and Communication Technology, Thai Nguyen,Vietnam

Abstract
Sign languages have their distinct syntax and grammar characteristics. In this paper, we summarized linguistic rules in syntax of Vietnamese sign language and proposed a rule-based algorithm for syntax transformation in Vietnamese sign language. The experimental results show that the proposed algorithm is efficient and useful for building automatic Vietnamese sign language translation.

Key words:
Vietnamese sign language, syntax transformation, sign language translation.

1. Introduction

Sign language is the daily language for the deaf performed by using unified hand gestures. Sign languages have been developed in several centuries and recognized as official languages with distinct vocabularies and grammars.

There are some translation services and products built to assist the deaf communicating with normal hearing people. The cores of these systems are the syntax transformation algorithms.

One of the most successful sign language translation systems up to now is the ViSiCAST for English [1]. This system uses an algorithm called as Head-driven Phrase Structure Grammar (HPSG) to convert English written documents into English sign language documents. The core of this system is the use of syntax analysis system CMU to analyze input English documents, and then transform them to correspondent English sign language documents by using declarations in Prolog [1].

TEAM project [2] is an American Sign Language (ASL) translation using other algorithm called as Synchronous Tree Adjoining Grammar (STAG). This system uses a bilingual dictionary between spoken / written English and English sign language [3]. This system also uses a syntax transformation in which input English sentences are analyzed and transformed.

The research in [4] aims to design a statistical machine translation from English written text to ASL. The system is based on the use of Moses tool with some modifications and the results are synthesized through a 3D avatar for interpretation.

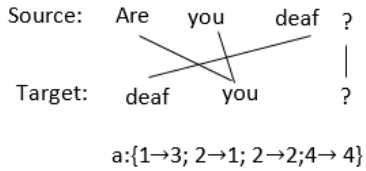


Fig. 1 Examples of IBM defines the translation probability for an English sentence.[4]

One recent research in [5] also proposed a translation between spoken / written English into Indian Sign Language (ISL). The objective of this work is to design a translation machine which can translate English text to ISL glosses. This approach is based on statistical machine translation for ISL by using a corpus. The corpus is prepared by collecting glosses and sentences used in Indian Railways for announcement and conversation in public assistance counters.

Sign : NAYA: ABHI: BACHA: ISKU: L SIKHA: NA:
ISL Gloss: new now child school teach
English : The new (thing is that) children are now taught (sign language) at school.'
Sign : PAHLE SIKHA: NA: MUSKIL.
ISL Gloss: before teach difficult
English : '(Signs) were not taught before; (it was too) difficult.'
Sign : ABHI: SURU: BACHA: SIKHA: NA:
ISL Gloss: now begin child teach
English : 'Now they have started to teach the children.'

Fig. 2 Examples of English to ISL gloss translation [5].

Language grammars are complex issues and largely different between languages. Therefore, researches on ASL or ISL cannot be directly applied for Vietnamese sign language (VSL). Therefore, in this paper, we proposed a syntax transformation algorithm applied in VSL translation system.

Manuscript received May 5, 2017
Manuscript revised May 20, 2017

2. Linguistic fundamentals of Vietnamese sign language

One of three most important features of VSL is the syntax. The syntax of VSL has distinct rules and is different with those of Vietnamese spoken / written language [6]. In this section, we present fundamental syntax rules of VSL.

2.1 Sentence forms

- *Simple sentences:*
The structure of spoken / written Vietnamese language is Subject → Predicate → Object. However, the structure of VSL is Subject → Object → Predicate.

Table 1: Simple sentence

	Vietnamese Spoken / Written Language	**Vietnamese Sign Language**
Structure	Subject → predicate → object	Subject → object → predicate
Example in Vietnamese	Cô ấy ăn táo	Cô ấy táo ăn

- *Question sentences:*
The structure of Vietnamese spoken / written sentences and Vietnamese sign sentences are completely different. There is no need to use words Yes / No in yes-no VSL questions. Instead of that, the expression on the face can be used to show the state of question sentences.

In Vietnamese spoken / written language, the positions of ask words in question sentences are not fixed. However, in VSL, ask words are always at the end of the sentences.

Table 2: Question sentences

	Vietnamese Spoken / Written Language	**Vietnamese Sign Language**
Structure	Subject → ask words → predicate → object?	Object → predicate → subject → ask words
Example	Ai ăn táo?	Táo ăn ai?
Structure	Subject → predicate → ask words → object?	Subject → object → predicate → ask words?
Example in Vietnamese	Cường ăn mấy quả táo?	Cường quả táo ăn mấy?

- *Negative sentences:*
Vietnamese has different types of negative sentences including full negative sentences and partial negative sentences.

In Vietnamese spoken / written language, verbal negatives in partial negative sentences precede the main verbs. However, in VSL, negative words always follow verbs and locate at the end of the sentence.

Table 3: Negative sentence forms

	Vietnamese Spoken / Written Language	**Vietnamese Sign Language**
Structure	Subject → predicate → negative word → object	Subject → object → predicate → negative word
Example in Vietnamese	Cường không ăn táo.	Cường táo ăn không.

2.2 Word orders

The word orders of Vietnamese spoken / written language and VSL have significant differences as shown in Table 4 and 5.

Table 4: Nouns and numerals

	Vietnamese Spoken / Written Language	**Vietnamese Sign Language**
Structure	Numeral → Noun	Noun → Numeral
Example in Vietnamese	Hai quả táo	Qủa táo hai

Table 5: Verbs and negative words

	Vietnamese Spoken / Written Language	**Vietnamese Sign Language**
Structure	Negative word → verb	Verb → negative word
Example in Vietnamese	Không ăn	Ăn không

3. Building rule-based syntax transformation trees

Based on the linguistic fundamentals of VSL, we built rule-based syntax transformation trees as the following.

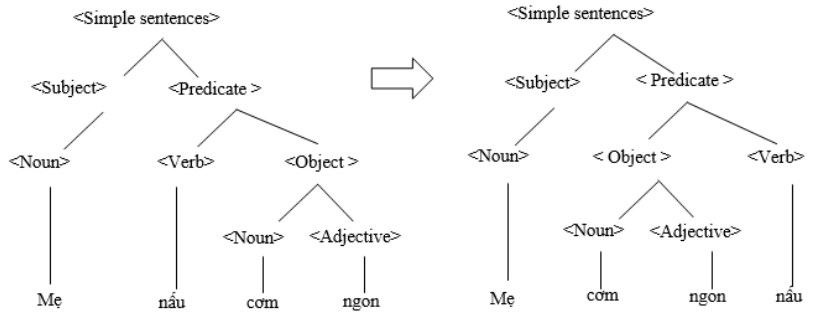


Fig. 1 Structure of syntax tree transforming simple sentences

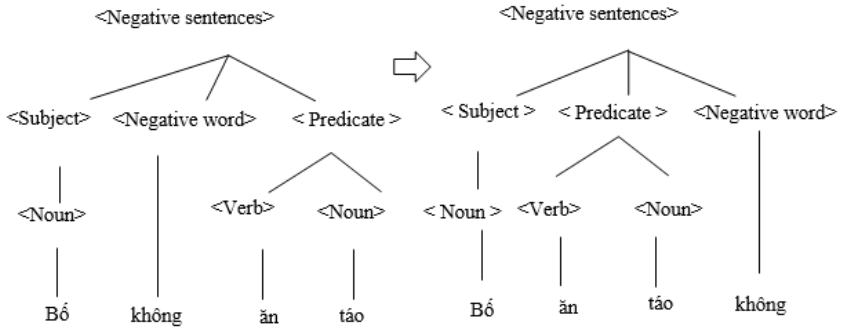


Fig. 2 Structure of syntax tree transforming type 1st negative sentences

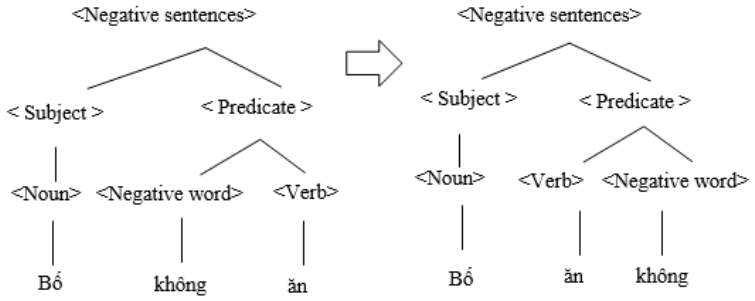


Fig. 3 Structure of syntax tree transforming type 2nd of negative sentences

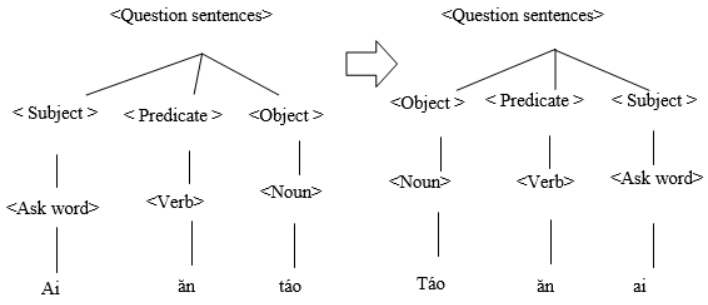


Fig. 4 Structure of syntax tree transforming type 1st of question

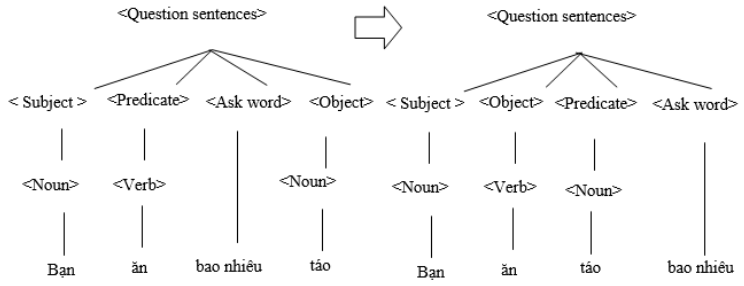


Fig. 5 Structure of syntax tree transforming type 2nd of question

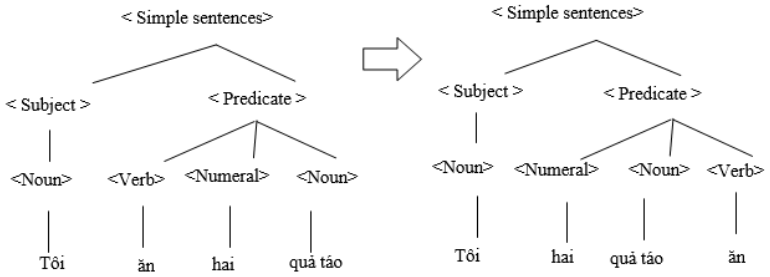


Fig. 6 Structure of syntax tree transforming sentences with numerals

4. Proposed syntax transformation algorithm

Based on the above syntax transformation trees, we proposed a rule-based syntax transformation algorithm for VSL as shown in Fig. 7.

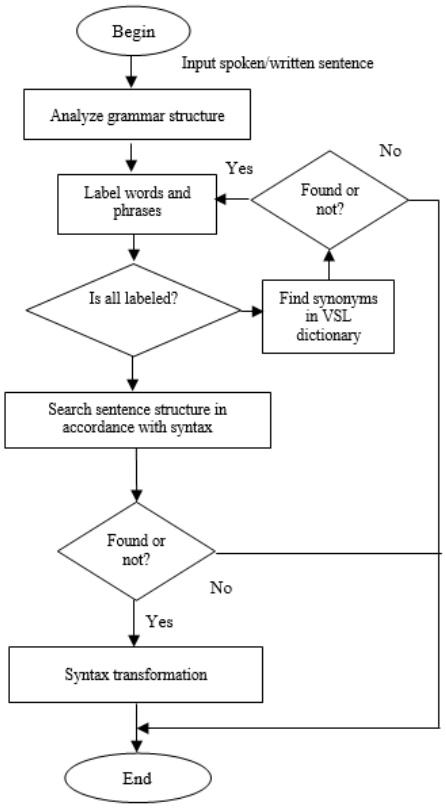


Fig. 7 Syntax transformation algorithm for VSL translation

5. Experiment Results

5.1 Evaluation method

BLEU is a method to evaluate quality of the documents automatically translated my machine, proposed by IBM in 2002 [7] and used as the primary evaluation measure for research in machine translation in [8]. The original ideal of the method is to compare two documents automatically translated by machine and manual translated by linguistic experts. The comparison is performed by statistical analyzing the coincidence of the words in the two documents that takes into account the order of the words in the sentences using n-grams. Specifically, BLEU scores are computed by statistically analyzing the degree of coincidence between n-grams of documents automatically translated by machine and the ones manual translated by high-quality linguistic experts [9].

BLEU score can be computed as follows [11]:

$$score = \exp\left\{\sum_{i=1}^{N} w_i \log(p_i) - \max\left(\frac{L_{ref}}{L_{tra}} - 1,0\right)\right\} \quad (1)$$

- $P_i = \frac{\sum_j NR_j}{\sum_j NT_j}$
- NR_j: the number of n- grams in segment *j* in the reference translation (by experts) with a matching reference co-occurrence in segment
- NT_j: the number of n- grams in segment *j* in the translation (by machine) being evaluated.
- $w_i = N^{-1}$
- L_{ref}: the number of words in the reference translation (by experts) that is closest in length to the translation being scored.
- L_{tra}: the number of words in the translation (by machine) being scored.

The value of score evaluates the correlation between the two translations by experts and machine, computed in each segment where each segment is the minimum unit of translation coherence. Normally, each segment is usually one or a few sentences. The n-gram co-occurrence statistics, based on the sets of n-grams for the test and reference segments, are computed for each of these segments and then accumulated over all segments. It is clear that the smaller the score, the better the co-occurrence statistics.

5.2 Evaluation results

We built a VSL dictionary with 3000 words and phrases. For evaluation, we used 200 simple sentences extracted from on the textbooks used in the schools for deaf children. After being translated (shortened) by using the proposed method, we computed the BLEU scores between the translated sentences and the corresponding ones conducted by one expert in VSL.

The results of computed BLEU scores are shown in Fig.7. The ratio of sentences correctly translated by using the proposed method (corresponding with BLEU score is zero) is 97.5%. A few sentences incorrectly translated is caused by semantic ambiguity will be solved in our future researches.

Table 5: BLEU Score

ID sentence	Linput	BLEU Score
1	5	1.000
2	3	1.000
3	7	0.253
4	5	1.000
…	…	…
…	…	…
196	6	0.2778
197	7	1.000
198	5	0.5250
199	4	1.000
200	3	1.000

Acknowledgments

This study was supported by the Ministry of Education and Training of Vietnam (project B2016-TNA-27).

References

[1] J. A. Bangham, S. J. Cox, R. Elliot, J. R. W. Glauert, I. Marshall, S. Rankov, and M. Wells, “Virtual signing: Capture, animation, storage and transmission - An overview of the ViSiCAST project.” IEEE Seminar on Speech and language processing for disabled and elderly people, 2000.

[2] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler, M. Palmer, "A Machine Translation System from English to American Sign Language", Envisioning Machine Translation in the Information Future, Vol. 1934, 2000, pp. 191-193.

[3] K. Liddell, Grammar, gesture, and meaning in American Sign Language, Cambridge University Press, 2003.

[4] Achraf Othman and Mohamed Jemni, “Statistical Sign Language Machine Translation: from English written text to American Sign Language Gloss”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3, September 2011.

[5] Gouri Sankar Mishra, Ashok Kumar Sahoo and Kiran Kumar Ravulakollu, Word base statistical machine translation from English text to Indian sign language, ARPN Journal of Engineering and Applied Sciences, vol. 12, no. 2, january 2017.

[6] Do Thi Hien, Sign language for deaf community in Vietnam: Problems and Solutions, Project report, Social Science Academy of Vietnam, 2012, p. 156 (in Vietnamese).
[7] Papineni K., Roukos S., Ward T., Zhu Z-J, “BLEU: A method for Automatic Evaluation of Machine Translation”, Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, p.p 311-318, 2001.
[8] Hovy E.H.: Toward finely differentiated evaluation metrics for machine translation. Proceedings of the Eagles Workshop on Standards and Evaluation, Pisa, Italy, 1999.
[9] NIST report: Automatic evaluation of machine translation quality using N-gram co-occurrene statistics, 2002.

Ứng dụng mô hình dịch máy Transformer trong bài toán dịch tự động ngôn ngữ ký hiệu Việt Nam

Nguyễn Thị Bích Điệp
Trường ĐH Công nghệ Thông tin và Truyền Thông
Đại học Thái Nguyên
Thái Nguyên, Việt Nam
ntbdiep@ictu.edu.vn

***Tóm tắt*: Dịch máy là một bài toán được quan tâm từ những năm 50 cho đến nay. Các mô hình dịch phát triển không giới hạn lĩnh vực. Dịch tự động ngôn ngữ kí hiệu Việt Nam là một bài toán mới với ngôn ngữ ít tài nguyên do những đặc điểm cú pháp của nó. Transformer là một mô hình dịch hoàn toàn chỉ dựa vào kĩ thuật self-attention cho phép thay thế hoàn toàn kiến trúc hồi quy của mô hình RNN bằng các mô hình kết nối đầy đủ với các lớp trước lớp ẩn. Việc ứng dụng mô hình dịch máy Transformer với bài toán đang đề xuất là một phương pháp phù hợp mang lại kết quả khả quan.** *(Abstract)*

***Từ khóa:* Trí tuệ nhân tạo; Xử lý ngôn ngữ tự nhiên; Dịch máy; mô hình Transformer; ngôn ngữ ít tài nguyên; ngôn ngữ ký hiệu Việt Nam.**

I. GIỚI THIỆU BÀI TOÁN

Ngôn ngữ ký hiệu đã trải qua nhiều thế kỷ hình thành và phát triển trên thế giới và đã được khẳng định đây là một ngôn ngữ thực thụ có hệ thống từ vựng và ngữ pháp riêng như bất cứ một ngôn ngữ thông thường nào. Điểm khác biệt của ngôn ngữ ký hiệu so với ngôn ngữ nói thông thường là nó là ngôn ngữ tượng hình dựa trên các biểu diễn, chuyển động của bàn tay, cơ thể, và sắc thái biểu cảm của khuôn mặt. Nhờ sự phát triển của khoa học công nghệ, hiện nay trên thế giới đã và đang nghiên cứu phát triển và đưa ra nhiều dịch vụ thông dịch và sản phẩm công nghệ nhằm hỗ trợ người khiếm thính trong giao tiếp xã hội. Trong đó việc nghiên cứu phương pháp dịch tự động mà trọng tâm là chuyển đổi cú pháp đúng trong ngôn ngữ kí hiệu là vấn đề được các nhà nghiên cứu về ngôn ngữ tự nhiên trên thế giới đặc biệt quan tâm.

Một trong những hệ thống dịch tự động thành công nhất hiện nay là chương trình dịch ViSiCAST. Đây là công cụ để dịch từ tiếng Anh sang ngôn ngữ kí hiệu Anh. Hệ thống này sử dụng HPSG (Head-driven phrase structure grammar) để thể hiện văn bản tiếng Anh thành ngôn ngữ kí hiệu Anh (BSL). Nó là một phần của dự án VisiCast của liên minh Châu Âu. Hệ thống này cũng được coi là phương tiện nghiên cứu để dịch sang ngôn ngữ kí hiệu tiếng Đức hoặc tiếng Hà Lan, tuy nhiên hiện nay khả năng đó vẫn chưa thể thực hiện được. Phương pháp tiếp cận ở chương trình dịch này là sử dụng bộ phân tích cú pháp liên kết CMU để phân tích một văn bản tiếng Anh đầu vào, sau đó sử dụng các quy tắc cú pháp khai báo Prolog để chuyển đổi cú pháp. Trong quá trình dịch ở pha đầu tiên, các nguyên tắc Phrase Structured Head Driven được sử dụng để tạo ra đại diện ngôn ngữ kí hiệu. Một lược đồ mã hóa các ngôn ngữ kí hiệu được yêu cầu để biểu diễn [1].

Dự án TEAM là một hệ thống dịch từ văn bản sang dạng ngôn ngữ kí hiệu Mỹ sử dụng kĩ thuật cây đồng bộ ngữ pháp liền kề (STAG - Synchronous Tree Adjoining Grammar) [2]. Đầu tiên từ một văn bản nguồn sử dụng kĩ thuật để chuyển sang dạng cấu trúc cú pháp của ASL. Hệ thống duy trì một vốn từ vựng song ngữ để xác định một cặp từ ứng với một từ tiếng Anh và một từ trong ngôn ngữ kí hiệu. Kết quả của một mô đun ngôn ngữ là một từ trong ngôn ngữ kí hiệu được thể hiện bằng văn bản [3]. Kết quả của một mô đun tổng hợp là hình ảnh thể hiện ngôn ngữ kí hiệu bằng một mô hình con người. Mặc dù cách tiếp cận TEAM có vẻ giống như một kiến trúc trực tiếp vì nó như là một bản đồ từ từ ngữ sang dạng kí hiệu, nhưng thực ra nó là một cách tiếp cận chuyển cú pháp. Văn bản tiếng Anh đầu vào cần phải được phân tích với trình phân tích cú pháp TAG trong quá trình dịch và thông tin về cú pháp sẽ giúp hướng dẫn quá trình tìm kiếm từ vựng song ngữ. "Quy tắc chuyển tiếp" trong hệ thống này sẽ là các trong từ điển song ngữ; Bằng cách xác định và áp dụng quá trình kết hợp này, họ chuyển đổi một phân tích cú pháp của câu tiếng Anh thành một cấu trúc cú pháp cho ngôn ngữ kí hiệu Mỹ.

Việc nghiên cứu xử lý ngôn ngữ ký hiệu trên máy tính ở Việt Nam còn rất mới mẻ. Chúng ta chưa thực sự có một hệ thống ngôn ngữ đồng nhất cho ngôn ngữ ký hiệu tiếng Việt [4]. Bên cạnh vấn đề ngôn ngữ học, việc phát triển sản phẩm ứng dụng công nghệ để phát huy ngôn ngữ ký hiệu nhằm nâng cao trình độ, tiếp nhận thông tin, khả năng giao tiếp cho người khiếm thính lại càng ít và kém hiệu quả. Một ví dụ điển hình là việc sử dụng một thông dịch viên trong một số chương trình truyền hình để

chuyển nội dung thông tin sang ngôn ngữ ký hiệu nhằm truyền tải tới người khiếm thính. Sau khi khảo sát, hầu hết người khiếm thính đều nhận xét họ thấy khó hiểu nội dung từ việc mô tả của thông dịch viên. Nguyên nhân chính của hiện trạng này là thông dịch viên đã cố gắng biểu diễn đầy đủ cả câu theo ngôn ngữ nói thông thường, trong khi vốn từ vựng của người khiếm thính rất ít, chủ yếu là động từ và các từ đơn giản, và trật tự của câu trong ngôn ngữ ký hiệu cũng thay đổi nhằm tạo ra điểm nhấn của thông tin quan trọng.

Việc nghiên cứu về ngôn ngữ kí hiệu nói chung và ngôn ngữ kí hiệu Việt Nam (VSL) nói riêng có 2 phần quan trọng. Đầu tiên là phần chuyển đổi cấu trúc cú pháp của câu ngôn ngôn ngữ nói thông thường sang dạng đúng cấu trúc cú pháp của VSL. Phần thứ hai là biểu diễn minh họa VSL. Trong nghiên cứu này tác giả tập trung vào phần thứ nhất là dịch tự động câu ngôn ngữ thông thường sang dạng đúng cấu trúc cú pháp trong VSL. Với một số kết quả nghiên cứu trước đây, tác giả tập trung vào việc xây dựng hệ thống dịch dựa trên các luật [5] [6]. Tuy nhiên khi đánh giá các bản dịch thì hiệu quả còn thấp. Hướng nghiên cứu mới của tác giả tập trung vào mô hình dịch mới, trong đó mô hình Transformer được đánh giá là phù hợp với bài toán bởi các phân tích cụ thể ở phần tiếp theo.

II. MÔ HÌNH TRANSFORMER

Mô hình Transformer là một mô hình nổi tiếng gần đây trong cộng đồng xử lý ngôn ngữ tự nhiên và tạo ra những bước ngoặt lớn trong những bài toán dịch máy. Đối với bài toán dịch tự động VSL, ta có thể coi là một bài toán dịch với ngôn ngữ ít tài nguyên bởi những đặc điểm hạn chế về ngữ pháp và từ vựng của loại ngôn ngữ này. Trước đây các tác vụ dịch máy (Machine Translation) sử dụng kiến trúc Recurrent Neural Networks (RNNs) là chủ yếu. Nhưng các nhà nghiên cứu dịch máy đều có thể nhận thấy nhược điểm của phương pháp này là rất khó bắt được sự phụ thuộc xa giữa các từ trong câu và tốc độ huấn luyện chậm do phải xử lý input tuần tự. Transformers đã giải quyết được 2 vấn đề này. Bởi vậy nó được cho là phù hợp với bài toán dịch tự động VSL. Cơ chế tổng quan của mô hình như sau:

A. Cơ chế sefl-attention

Self-attention là cơ chế giúp Transformer "hiểu" được sự liên quan giữa các từ trong một câu. Có thể tưởng tượng self-attention giống như cơ chế tìm kiếm. Với một từ cho trước, cơ chế này sẽ cho phép mô hình tìm kiếm trong các từ còn lại để xác định từ nào liên quan để sau đó thông tin sẽ được mã hóa dựa trên tất cả các từ trên. Đầu vào của self-attention là 3 vector query, key, value. Các vector này được tạo ra bằng cách nhân ma trận biểu diễn các từ đầu vào với ma trận học tương ứng.

• Query vector là vector dùng để chứa thông tin của từ được tìm kiếm, so sánh.

• Key vector là vector dùng để biểu diễn thông tin các từ được so sánh với từ cần tìm kiếm ở trên.

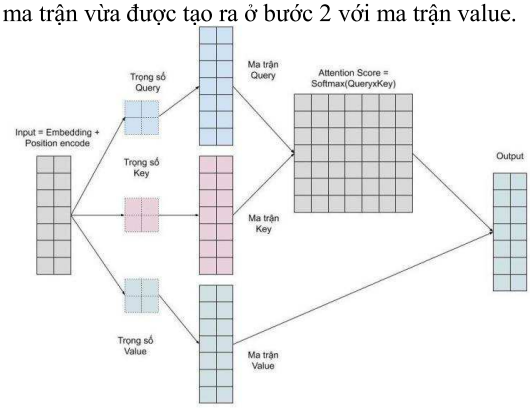
• Value vector là vector biểu diễn nội dung, ý nghĩa của các từ.

Vector attention cho một từ thể hiện tính tương quan giữa 3 vector này được tạo ra bằng cách nhân tích vô hướng giữa chúng và sau đó được chuẩn hóa bằng hàm softmax. Cụ thể quá trình tính toán như sau:

• Bước 1: Tính ma trận query, key, value bằng cách nhân input với các ma trận trọng số tương ứng

• Bước 2: Nhân hai ma trận query, key vừa tính được với nhau với ý nghĩa so sánh giữa câu query và key để học mối tương quan. Sau đó các giá trị sẽ được chuẩn hóa về khoảng [0-1] bằng hàm softmax với ý nghĩa 1 khi câu query giống với key ngược lại, 0 có nghĩa là không giống

• Bước 3: Output sẽ được tính bằng cách nhân ma trận vừa được tạo ra ở bước 2 với ma trận value.



Hình 1. Quá trình tính toán vecto attention

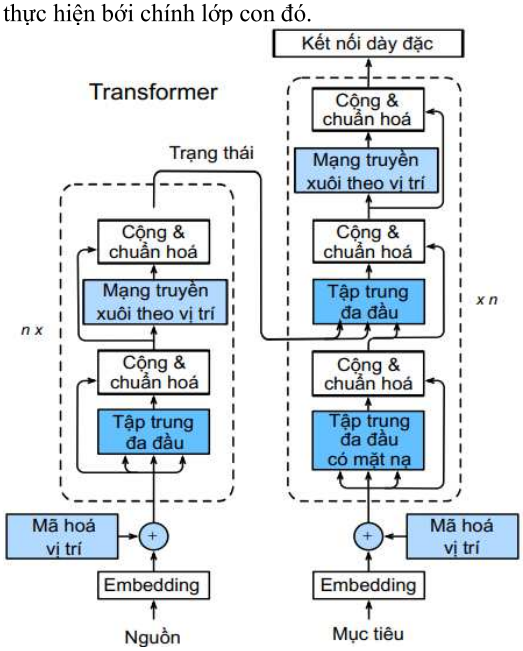
B. Tổng quan mô hình Transformer

Mô hình transformer bao gồm hai phần lớn là bộ mã hóa và bộ giải mã. Bộ mã hóa biểu diễn ngôn ngữ nguồn thành các vector, bộ giải mã sẽ nhận các vector biểu diễn này và dịch nó sang ngôn ngữ đích. Chi tiết các thành phần của bộ mã hóa và giải mã được thể hiện như hình 2.

Một trong những ưu điểm của transformer là mô hình có khả năng xử lý song song cho các từ. Đầu vào sẽ được đẩy vào cùng một lúc. Bộ mã hóa của mô hình transformer bao gồm một tập gồm N lớp giống nhau, mỗi lớp bao gồm 2 lớp con.

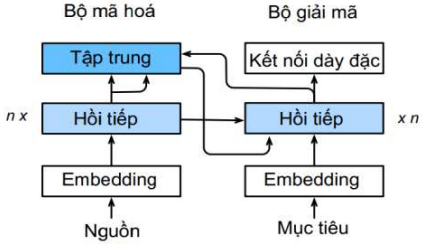
Lớp đầu tiên là cơ chế multi-head self-attention, và lớp thứ 2 là mạng feed-forward kết nối đầy đủ. Đầu ra của mỗi lớp con là LayerNorm(x +

Sublayer(x)), trong đó Sublayer(x) là một hàm được thực hiện bởi chính lớp con đó.



Hình 2. Kiến trúc Transformer

Bộ giải mã: cũng bao gồm tập gồm N lớp giống nhau. Ngoài hai lớp con giống như bộ mã hóa, bộ giải mã còn có một lớp để thực hiện multi-head attention trên đầu ra của lớp giải mã. Ở đây sẽ có thay đổi cơ chế self-attention trong bộ mã hóa [7].



Hình 3. Bộ mã hóa và giải mã

III. DỊCH TỰ ĐỘNG VSL BẰNG MÔ HÌNH TRANSFORMER.

Với khả năng tận dụng khả năng tính toán song song của GPU để tăng tốc độ huấn luyện cho các mô hình ngôn ngữ, đồng thời khắc phục điểm yếu xử lý câu dài thì mô hình Transformer là một trong những mô hình được xem xét là phù hợp đối với bài toán dịch tự động VSL.

Các bước của việc ứng dụng mô hình này trong bài toán được đặt ra ban đầu sẽ bao gồm các quá trình: mã hóa và giải mã dữ liệu, áp dụng mô hình dịch và đánh giá hiệu quả bản dịch.

A. Mã hóa và giải mã

Trước tiên bộ dữ liệu cần được chuyển đổi thành một biểu diễn số. Thông thường, ta cần chuyển đổi văn bản thành một chuỗi mã hóa, được sử dụng làm chỉ số thành một bản nhúng.

Dữ liệu cho mô hình huấn luyện bao gồm chứa hai dạng văn bản đã tách từ (tokenizer), một cho tiếng Việt thông thường và một cho VSL. Cả hai đều có các phương pháp giống nhau. Phương thức mã hóa chuyển đổi một loạt các câu thành các mã thông báo. Phương thức giải mã chuyển đổi các mã thông báo này trở lại thành văn bản mà con người có thể đọc được.

- Thiết lập đường dẫn đầu vào: Để xây dựng một đường dẫn đầu vào phù hợp cho việc huấn luận cần áp dụng một số biến đổi cho tập dữ liệu. Hàm sau đây sẽ được sử dụng để mã hóa các lô văn bản thô:

```
def tokenize_pairs(vsl, vi):
    pt = tokenizers.pt.tokenize(pt)
    # Convert from ragged to dense,
padding with zeros.
    vsl = vsl.to_tensor()
    vi = tokenizers.vi.tokenize(vi)
    # Convert from ragged to dense,
padding with zeros.
    vi = vi.to_tensor()
    return vsl, vi
```

- Mã hóa vị trí: Các lớp chú ý xem đầu vào là một tập hợp các vectơ, không có thứ tự. Mô hình này cũng không chứa bất kỳ lớp lặp lại nào. Do đó, một "mã hóa vị trí" được thêm vào để cung cấp cho mô hình một số thông tin về vị trí tương đối của các thẻ trong câu. Vectơ mã hóa vị trí được thêm vào vectơ nhúng. Véc tơ nhúng đại diện cho một mã thông báo trong không gian d chiều nơi các mã thông báo có ý nghĩa tương tự sẽ gần nhau hơn. Nhưng việc nhúng không mã hóa vị trí tương đối của các mã thông báo trong một câu. Vì vậy, sau khi thêm mã hóa vị trí, các mã thông báo sẽ gần nhau hơn dựa trên sự giống nhau về ý nghĩa của chúng và vị trí của chúng trong câu, trong không gian d chiều. Công thức tính toán mã hóa vị trí như sau:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

- Mặt nạ nhìn trước (look-ahead mask) được sử dụng để che dấu các mã thông báo trong tương lai theo một trình tự. Nói cách khác, mặt nạ cho biết mục nhập nào không nên được sử dụng. Điều này có nghĩa là để dự đoán mã thông báo thứ ba, chỉ mã thông báo đầu tiên và thứ hai sẽ được sử dụng. Tương tự như vậy để dự đoán mã thông báo thứ tư, chỉ mã thông báo đầu tiên, thứ hai và thứ ba sẽ được sử dụng, v.v.

- Hàm chú ý (attention function) được sử dụng bởi transformer có ba đầu vào: Q (truy vấn), K (phím), V (giá trị). Phương trình được sử dụng để tính toán là:

$$Attention(Q, K, V) = softmax_k\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Khi quá trình chuẩn hóa softmax được thực hiện trên K, các giá trị của nó quyết định mức độ quan trọng đối với Q. Đầu ra đại diện cho phép nhân của trọng số chú ý và vectơ V (giá trị). Điều này đảm bảo rằng các mã thông báo muốn tập trung vào được giữ nguyên trạng và các mã thông báo không liên quan sẽ bị loại bỏ.

B. Khởi tạo mô hình Transformer

Transformer bao gồm bộ mã hóa, bộ giải mã và một lớp tuyến tính cuối cùng. Đầu ra của bộ giải mã là đầu vào của lớp tuyến tính và đầu ra của nó được trả về.

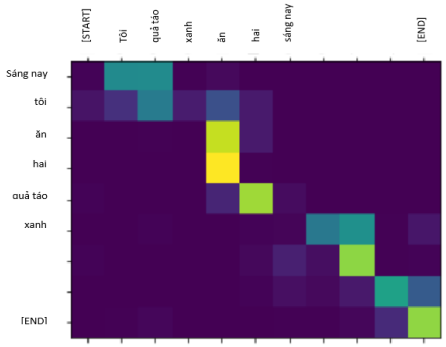
- Cài đặt siêu tham số: Mô hình cơ sở được mô tả trong bài báo được sử dụng: num_layers = 6, d_model = 512, dff = 2048.
- Trình tối ưu hóa: Sử dụng trình tối ưu hóa Adam với công cụ lập lịch tốc độ học tập tùy chỉnh (Thuật toán tối ưu hóa Adam là một phần mở rộng cho quá trình giảm độ dốc ngẫu nhiên mà gần đây đã được áp dụng rộng rãi hơn cho các ứng dụng học sâu trong thị giác máy tính và xử lý ngôn ngữ tự nhiên). [8]
- Huấn luyện và kiểm tra:

```
Transformer = Transformer(
    num_layers=num_layers,
    d_model=d_model,
    num_heads=num_heads,
    dff=dff,
    input_vocab_size=tokenizers.pt.ge
    t_vocab_size().numpy(),
    target_vocab_size=tokenizers.en.g
    et_vocab_size().numpy(),
    pe_input=1000,
    pe_target=1000,
    rate=dropout_rate)
```

Sau đó tạo đường dẫn checkpoint và trình quản lý checkpoint sử dụng để lưu các checkpoint sau mỗi n epochs.

Câu văn bản tiếng Việt thông thường được sử dụng làm ngôn ngữ đầu bào và VSL là ngôn ngữ đích.

- Các bước sau dùng để suy luận:
  - Mã hóa câu đầu vào bằng trình mã hóa tiếng Việt (tokenizers.pt). Đây là đầu vào của bộ mã hóa.
  - Đầu vào của bộ giải mã được khởi tạo thành mã thông báo (START).
  - Tính toán mặt nạ đệm (padding masks) và mặt nạ nhìn trước (look ahead masks).
  - Sau đó, bộ giải mã sẽ đưa ra các dự đoán bằng cách xem đầu ra của bộ mã hóa và đầu ra của chính nó (self-attention).
  - Nối mã thông báo được dự đoán với đầu vào của bộ giải mã và chuyển nó đến bộ giải mã. Trong cách tiếp cận này, bộ giải mã dự đoán mã thông báo tiếp theo dựa trên các mã thông báo trước đó nó đã dự đoán.
- Hiển thị Attention: Lớp Translator trả về từ điển bản đồ chú ý có thể sử dụng để hình dung hoạt động bên trong của mô hình.



Hình 4. Bản đồ Attention

IV. ĐÁNH GIÁ KẾT QUẢ VÀ KẾT LUẬN

Mô hình Transformer được đánh giá trong bài toán dịch tự động VSL trên dữ liệu là kho ngữ liệu được xây dựng bao gồm 600 cặp câu tiếng Việt và VSL đã được xem xét bởi chuyên gia ngôn ngữ VSL. Quy trình xây dựng kho ngữ liệu bao gồm việc thu thập từ điển VSL, thu thập luật ngữ pháp đúng trong VSL, quá trình kết nối các từ đồng nghĩa sau đó chuyển đổi tự động câu tiếng Việt sang dạng VL. Sau bước này, các cặp câu được đánh giá và chỉnh sửa một lần nữa bằng các chuyên gia sử dụng VSL thường xuyên trong cộng đồng người khiếm thính.

Việc đánh giá bản dịch của cặp câu tiếng Việt-VSL dựa trên thang điểm đánh giá BLEU. BLEU là một thuật toán để đánh giá chất lượng văn bản đã được dịch bằng máy từ ngôn ngữ tự nhiên này sang ngôn ngữ tự nhiên khác. Chất lượng được coi là sự tương ứng giữa đầu ra của máy và của con người: "bản dịch máy càng gần với bản dịch chuyên nghiệp của con người thì càng tốt" [9]. Điểm BLEU cho mô hình dịch máy Transformer với bài toán dịch tự động VSL hiện là 35,15 điểm. Đây là điểm cao cho một số mô hình dịch mới so với những cặp ngôn ngữ thông thường như Anh-Việt. Nhưng với bài toán VSL còn có nhiều hạn chế vì những đặc điểm đặc thù riêng và là ngôn ngữ ít tài nguyên hiện chỉ đánh giá trên bộ dữ liệu nhỏ.

Trong tương lai, tác giả hi vọng cải tiến mô hình và các thông số để đánh giá bài toán trên một kho ngữ liệu đủ lớn để đảm bảo sự đánh giá tốt hơn.

TÀI LIỆU THAM KHẢO

[1] [1] J. A. Bangham, S. J. Cox, R. Elliot, J. R. W. Glauert, I. Marshall, S. Rankov, and M. Wells, "Virtual signing: Capture, animation, storage and transmission – An overview of the ViSiCAST project." *IEEE Seminar on Speech and language processing for disabled and elderly people,* 2000.

[2] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler, M. Palmer, "A Machine Translation System from English to American Sign Language", *Envisioning Machine Translation in the Information Future*, Vol. 1934, 2000, pp. 191-193.

[3] K. Liddell, Grammar, *Gesture, and meaning in American Sign Language*, Cambridge University Press, 2003.

[4] Đỗ Thị Hiên, *Ngôn ngữ kí hiệu của cộng đồng người khiếm thính Việt Nam: thực trạng và giải pháp*, Báo cáo tổng hợp đề tài nghiên cứu khoa học cấp bộ, Viện Khoa học xã hội Việt Nam, 2012.

[5] Thi Bich Diep Nguyen and Trung-Nghia Phung, "*Some issues on syntax transformation in Vietnamese sign language translation". Sign Language Studies*. IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.5, 2017.

[6] Thi Bich Diep Nguyen, Trung-Nghia Phung, Tat-Thang Vu (2017) , "*A rule-based method for text shortening in Vietnamese sign language translation".* Springer AISC, Vol. 672, Proc. of INDIA-2017, Vietnam, 2017.

[7] Llion Jones, Aidan N. Gomez, Łukasz Kaiser*, Attention Is All You Need*, 31st Conference on Neural Information Processing Systems USA, 2017.

[8] Diederik P. Kingma, Jimmy Lei Ba*, "*Adam: a method for stochastic optimization", *International Conference on Learning Representations*, 2015.

[9] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, *BLEU: a Method for Automatic Evaluation of Machine Translation*, . Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.

Data Augmentation Techniques in automatic translation of Vietnamese Sign Language for the deaf

Thi Bich Diep Nguyen[1], Tuyen Ho Thi[1]

[1] Thai Nguyen University of Information and Communication Technology
ntbdiep@ictu.ed.vn

Abstract. Automatic translation of Vietnamese sign language is a new and meaningful problem. Automatic translation of Vietnamese sign language is a meaningful solution as it promises to eliminate communication obstacles and improve the lives of the deaf people. With the support of automatic translation technologies, there are many effective translation methods. However, the important problem in the current problem is that there is not a large enough amount of data available to evaluate and develop translation models. With wordnet, automatic data augmentation is possible. By applying the hyponym and hypernym in wordnet with the criteria of this study, we enrich the data based on the original data built. The measure of data similarity between the sentences generated from the original sentence is evaluated accordingly based on the combination of the cosine measure and meets the data requirements of the Vietnamese sign language automatic translation problem. Experiments show that BLEU scores on some translation models achieve high results after data augmentation.

Keywords: Natural Language Processing, Machine translation, Wordnet, Data Augmentation.

1 INTRODUCTION

Sign language has long been established as the official language of the deaf people in several countries. The language used by the Vietnamese deaf community is called Vietnamese Sign Language (VSL). Although sign language and spoken language have many similarities, there are substantial distinctions between spoken and written sign language [1]. For example, American Sign Language (ASL) has its own grammatical system (separate rules for phonemes, morphology, syntax, and semantics) that are different from those in English [2]. VSL is utilized as the official language in the approximately 7.5 million-member Vietnamese deaf population. Similar to other foreign languages, there are significant communication difficulties if sign language cannot be understood and comprehended.

There are two problems with the sign language interpretation process: converting sign language to regular language and vice versa. In which, the problem of translating from ordinary language is a major problem in order to transmit information and bring social knowledge to the deaf. Given the significant scientific and technological advancements in the field of information technology, there exist sign language translation systems in the world, such as TESSA - Translation from Speech to English Sign Language (BSL) [3]; ViSiCAST translator is a tool to translate from English to English sign language [4]; the SignSynth project is based on the ASCII-Stokoe model [5] ; the ASL workbench system is an automatic text translation system into American Sign Language [6]; the TEAM project is a text-to-American sign language translation system using the contiguous grammar tree technique. Recent research conducted by Gouri Sankar Mishra and colleagues presented a technique for translating spoken English into Indian Sign Language (ISL) [7]. Most of these research projects were initially based on structural translation models.

The process of translating ordinary language into sign language includes the following steps:

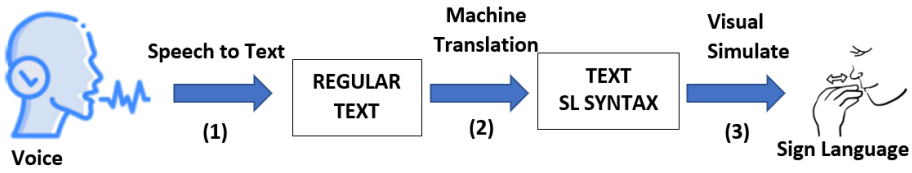


Figure 1: The process of translating speech into sign language

In which, (1) is the process of translating speech recognition into text. There have been many studies and applications that handle this part of the job well, such as Google's API. (2) is the process of converting plain text into a syntactically correct form in a sign language. (3) is the process of simulating from syntactically correct text in sign language into representations such as 3D models or videos and images of sign language. In this procedure, the second step gets the most attention due to the completion of the conveyed message. The basic challenge is that sign language, in general, has limited vocabulary compared to spoken/written language. If the machine translation is poorly performed, the complete message might not be successfully communicated, or in some cases, the conveyed message has a different meaning from the original [8]. Recent researches are making the most out of the technical advances in the fields of Natural Language Processing (NLP), Deep Neural Networks (DNN), and Machine Translation (MT), with the aim to develop systems that are able to translate between signed and spoken languages in order to fill the gap of communication between the SL speaking communities and the people using vocal language [9].

On a modest, custom-built dataset, we have used translation approaches and models to produce a number of successful translations. We have gone through a series of steps in our investigation of the situation. Initially, we proposed a rule-based translation technique using the syntactic rules of VSL [10] [11]. In this research, we propose a simple data augmentation method to apply to translation models. This is necessary for training models to improve the system's translation accuracy. Lastly, the detailed analysis and evaluation of the findings will be provided.

We currently have a bilingual dataset of Vie - VSL pairs including 10,000 sentence pairs that have been built and evaluated by a number of language experts and applied to the rule-based translation problem. This data is built on the domain of common communication sentences. We chose this data domain because it is closer and more meaningful to the hearing impaired. However, to apply some statistical machine translation methods, this data source is not large enough. We are studying this way of data augmentation based on wordnet to serve the statistical translation problem.

2 The proposed method

2.1 Data augmentation background

The WordNet semantic network is a lexical-level data collection describing the semantic link between words [12]. WordNet is comprised of three distinct tuples: one for nouns, one for verbs, and one for adjectives and adverbs. The English WordNet dataset, as of version 3.0, has around 117,000 nouns, 11,400 verbs, 22,000 adjectives, and 4,600 adverbs. As illustrated in Figure 2, the WordNet is structured as a tree, with each node containing a prototype word (lemma) and a collection of synonyms (synset). The WordNet only shows semantic relations, not phonetic or morphological relationships.

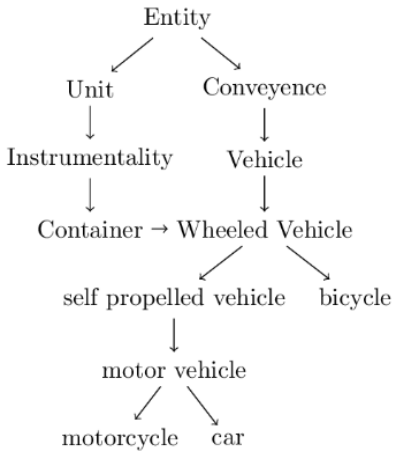


Figure 2: Hierarchical structure of WordNet

We use the wordnet's features to produce new data by changing words in sentences based on their semantic relationships. The newly generated sentence retains the same syntax and semantics, thus the same conversion rule is used to convert it to VSL. With the similarity assessments in the experimental section, the translation is therefore performed accurately and semantically.

Grammar parsing gives us the syntactic structure of a sentence. However, grammar

analysis can only determine grammatical accuracy and not semantic correctness. For example, by analyzing the phrase “the table eats the chicken,” we notice that it is absolutely grammatically accurate (“the table” acts as the subject, “eat” is the verb, and “the chicken” acts as a modifier for the verb). However, it is clear that the "table" cannot "eat" the "chicken", instead if it is changed to "the dog eats the chicken", it will be more reasonable. So how do you know if "the table" or "the dog" can "eat the chicken"? – one of the feasible solution is to use the hypernyms - hyponyms relationships in WordNet. Assume there is a heuristic stating that the verb "to eat" can only be performed by "animals" (i.e. only animals can eat). Thus, to determine if an object can eat or not, we will examine its hypernyms to see whether it is "animal" or not. By traversing back to the hypernyms, it is straightforward to determine that "dog" can execute the action "eat" but "table" cannot. Similarly, we may apply semantic restrictions to a statement to determine whether it is semantically accurate. From there, it is feasible to build new phrases by substituting identically hypernyms. Figure 3 depicts the structure of the hypernyms with the keyword "dog.".

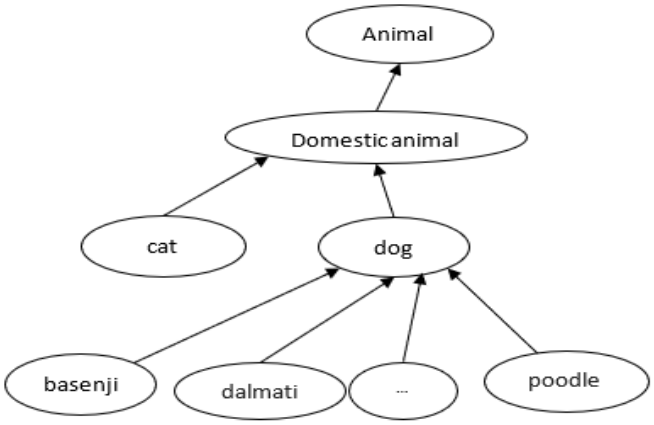


Figure 3: The structure of the hypernyms - hyponyms for the keyword “dog”.

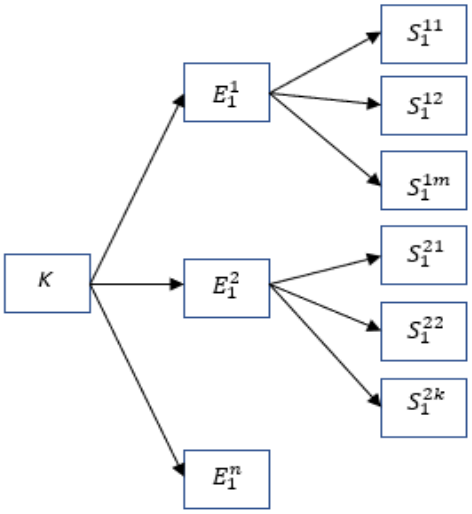


Figure 4. Illustration of criteria using the Synset E_i^j.

In our problem, we use 3 criteria:

Sibling criterion: applied when all synset sets S_i^j when all synset sets contain sibling synsets (with the same synset and hypernym). Then the synset $\{E_1^1, E_1^2, \dots\}$ is selected as sibling synsets

That is: $SV = \{S_i^{jk} / S_i^{jk} \in S_i^j (\forall j: 0 \le j \le n_i^j),\ (S_P\ is\_hyper\ S_i^{jk})\}$

Parent-child criterion: applied when the synset sets S_i^j contain a synset that is superior to the remaining synsets (as long as each remaining synset has a synset that is a subordinate of the above-mentioned superior synset). Then the synset $\{E_1^1, E_1^2, \dots\}$ is selected as sibling synsets.

That is:

$$SV = \{S_i^{jk} / \exists S_p \in S_i^h (h \in [1 \dots n_i^j]),\ S_i^{jk} \in S_i^h (\forall j: 0 \le j \le n_i^j, j \ne h), (S_P\ is\_hyper\ S_i^{jk})\}$$

Grandparent- grandchildren criterion: applied when in synset sets S_i^j contain a synset that is superior to the remaining synsets (as long as each remaining synset has a synset that is a subordinate of the above-mentioned superior synset). Then the synset $\{E_1^1, E_1^2, \dots\}$ is selected as these subordinate synsets.

$$SV = \{S_i^{jk} / \exists S_g \in S_i^h\ (h \in [1 \dots n_i^j]), S_i^{jk} \in S_i^j (\forall j: 0 \le j \le n_i^j\ , j \ne h), (S_g\ is\_dist\_hyper\ S_i^{jk})\}$$

Thus, when the word *W* appears in a phrase, *W* can be replaced with *W'* if *W* and *W'* satisfy the sibling, parent-child, and grandparent-grandchild criteria. Therefore, depending on the structure of the hypernyms and hyponyms and other characteristics of wordnet, we may construct fuzzy data by changing words in previous phrases according to predetermined criteria.

2.2 Data Augmentation Process

With the features of wordnet about semantic relations between words, we replace words in sentences to generate new data. The new phrase is syntaxically unchanged and semantically logical, so to translate it to VSL we keep the same conversion rule. As a result, the translation is done correctly and semantically with similarity assessments in the experimental part.

The base dataset is a bilingual data including 10,000 pairs of Vietnamese - Vietnamese sign language sentences that have been built and evaluated by a number of linguistic experts. The procedures involved in constructing bilingual data are outlined below:

Step 1: Crawl VSL dictionary from source: https://tudienngonngukyhieu.com/. This is a dictionary of sign language that is commonly used by the deaf population in Vietnam.

Step 2: Add to VSL Sign Language Dictionary: Since 2017, we have compiled a database of VSL dictionaries from the aforementioned website by collaborating with professionals and the deaf community. Because of sign language's brevity and simplicity, its lexicon has a small selection of words. We gathered a total of 3053 language units. As of 2022, the number of words and phrases is continuously added and there are 6304 characters/words/phrases represented in sign language.

Step 3: Construct a list of synonyms: This stage aims to optimize the representation of words and phrases in Vietnamese sentences into VSL, while the sign language dictionary is limited. All words that are not represented in sign language, including proper names and numerals, can be communicated in VSL through spelling.

Step 4: Construct a set of pairs of “bilingual” sentences. For our problem, we employed the parsing toolkit, which is the result of research by Nguyen et al. Preprocessing comprises input data normalization as well as a collection of techniques for extracting and labeling VietWS terms [13]. The data comprises of communication-related phrases that have been partially semi-automatically processed and then carefully examined. Finally, the data were re-evaluated by a number of sign language specialists. For the creation of a rule-based translation system, we accumulated a total of 10,000 pairs of Vietnamese-to-VSL bilingual phrases. Data published and shared at https://github.com/BichDiep/data-rules-VSL. We propose a method to enrich this data based on wordnet from this constructed 10,000-sentence original dataset.

Based on the properties and characteristics of wordnet for semantic constraints to check semantic correctness in sentences, we have a process of building new data through the following steps.

Algorithm: Data-Augment-VSL
1: Input: Sentences S
2: Output: Set of sentences S’ are generated based on S.
3: Split W word ∈ S
4: X ⇐W. hypernyms()
5: For i=1,n Do
    Xi⇐X.hyponyms()

Add Xi to set T
6: While !∃ Xi.hyponyms:
Yi ⇐ Xi.hyponyms()
Add Yi to set T
7: Replace replace each element in T, create new data S'

We proceed to produce new data based on part of the data that was initially built. Our data is evaluated by a community of deaf people and several language experts in the field, then the data is enriched using the proposed method.

Figure 5 depicts the creation of new data from an original sentence S. The sentence "I eat apples" is parsed and the noun 'apple' is separated from the words in the sentence. Using the preceding approach, we obtain the set T, which consists of replacement words, in order to construct the set S', which contains new sentences. This set of T consists of 92 words (excluding the initial words), hence it generates 92 new sentences.

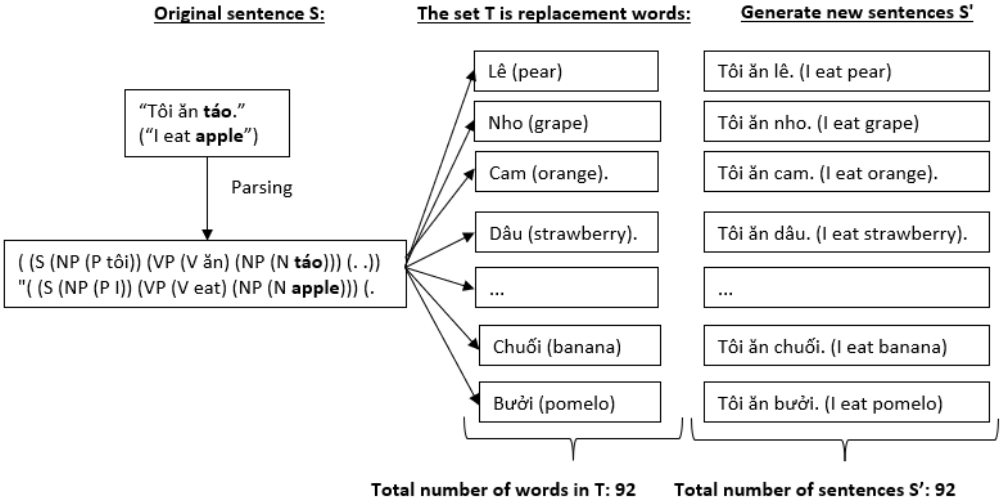


Figure 5. Example of generating new data from an original sentence.

3 Experiment and evaluate the results

3.1 Evaluation method

Bilingual Evaluation Understudy (BLEU) is a method to evaluate quality of the documents automatically translated my machine, proposed by IBM in 2002 and used as the primary evaluation measure for research in machine translation in [14]. The original ideal of the method is to compare two documents automatically translated by machine and manual translated by linguistic experts. The comparison is performed by statistical analyzing the coincidence of the words in the two documents that takes into account the order of the words in the sentences using n-grams. Specifically, BLEU scores are computed by statistically analyzing the degree of coincidence between n-

grams of documents automatically translated by machine and the ones manual translated by high-quality linguistic experts [11]. BLEU score can be computed as follows [1:

$$score = exp\left\{\sum_{i=1}^{N} w_i \, log(p_i) - \max\left(\frac{L_{\text{ref}}}{L_{\text{tra}}} - 1{,}0\right)\right\}$$

- $P_i = \frac{\sum_j NR_j}{\sum_j NT_j}$
- NR_j: the number of n- grams in segment j in the reference translation (by experts) with a matching reference co-occurrence in segment
- NT_j: the number of n- grams in segment *j* in the translation (by machine) being evaluated.
- $w_i = N^{-1}$
- L_{ref}: the number of words in the reference translation (by experts) that is closest in length to the translation being scored.
- L_{tra}: the number of words in the translation (by machine) being scored

The value of score evaluates the correlation between the two translations by experts and machine, computed in each segment where each segment is the minimum unit of translation coherence. Normally, each segment is usually one or a few sentences. The n-gram co-occurrence statistics, based on the sets of n-grams for the test and reference segments, are computed for each of these segments and then accumulated over all segments. This value indicates how similar the candidate text is to the reference texts, with values closer to 1 representing more similar texts. The BLEU was introduced by IBM and has since become the standard assessment metric for machine translation studies. In these experiments, we employ Multi-BLEU scripts to evaluate quality of the translation based on the BLEU score [15].

The method described above verifies the similarity of newly generated sentences based on their closeness to the structure of wordnet [16]. The method for measuring the similarity between two words is shown in Figure 6.

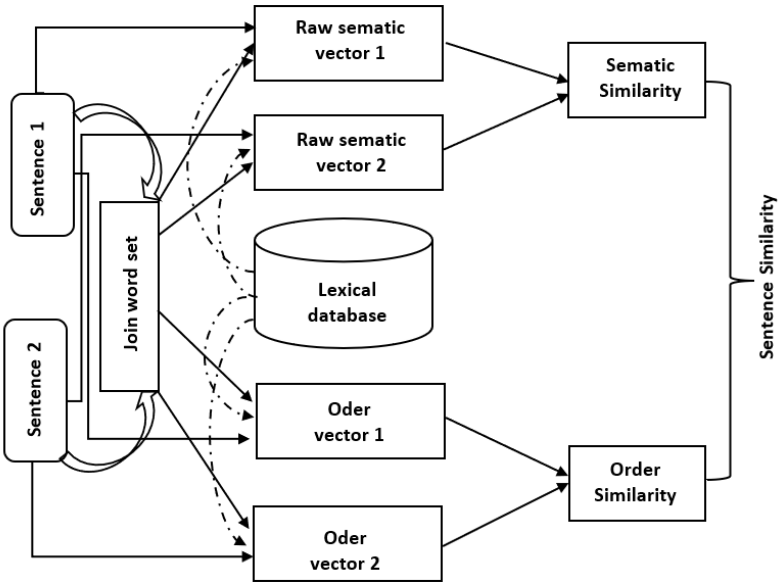


Figure 6. Model of the sentence similarity measuring approach utilizing WordNet.

3.2 Evaluation Results

We tested the method from initially built data. Our data is evaluated by a community of deaf people and several language experts. Then enrich the data using the proposed method. To evaluate our experiments, we base on data augmentation criteria. At the same time, we also score BLEU to evaluate the newly enriched data warehouse compared with the original data set on some machine translation models.

Table 1. Degree of data enrichment from the original data according to the proposed method.

ID	The word count of the set T	Number of original sentences	Number of new sentences
1	94	35	3290
2	12	4424	53088
3	183	5	915
4	471	3	1413
5	438	10	4380

With a pair of 10,000 sentences in a bilingual dataset originally constructed with 4626 lexical units and a context-based wordnet dataset that enriches the data with semantically meaningful and logical sentences. To evaluate the experiments with 63086 new data generated from the set of 10000 original data, we evaluate the BLEU scores of some machine translation methods for the problem of automatic translation of

Vietnamese sentences into VSL syntax correct sentences. The experimental process shows that, when re-evaluating the enriched bilingual sentences by the proposed method, the data increases by 14.4 times for some lexical groups which are nouns, pronouns and adjectives. As for the group of verbs, this method cannot be used because most of the newly born sentences are not semantically suitable. We evaluate the similarity between the nascent data and the original data using the similarity measurement method. With a range of values from 0-1 corresponding to 0 points is completely different, 1 point is exactly the same. The experimental process shows that the average similarity is 0.6.

Table 2. Comparison of BLEU scores on several translation models between original and augmentation data.

Translation model	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
Original data				
Rule-based translation	78.5	76.33	72.67	68.02
Seq2Seq	74.43	64.67	60.56	58.5
Transformer	76.25	70.33	68.25	65.2
Augmentation data				
Rule-based translation	78.5	76.33	72.67	68.02
Seq2Seq	92.5	89.25	85.4	81.11
Transformer	**94.87**	**92.16**	**90.15**	**89.23**

Through the experimentation process with several models, we observed that BLEU score do not change with the rules-based translation model, but have a significant change with the statistical translation models.

In general, the BLEU scores on the test sets are higher than the BLEU scores of certain other languages, such as in our case. The translation paradigm is mostly unchanged, as the majority of linguistic units in the two languages are equivalent. Only a few non-sign language terms are substituted with synonyms. In terms of sentence structure, VSL pairings are significantly less diversified than those of other language pairs. Thus the language model is simpler than the machine because the probabilistic model is convergent. Nevertheless, there are discrepancies between test sets. This variation depends mostly on the length, complexity, and vocabulary of each domain's sentences. The majority of sentences in the communication domain are short and simple, and the proportion of vocabulary from the VSL lexicon is larger than in other data domains. However, this BLEU score is suitable and has not too special value when referring to some sign language translation problems like ours such as German sign language translation with 82.87 points BLEU [17], Thai sign language [18], ..

4 Conclusion

VSL is a low resource language. While other low-resource languages have had some research results on the problem of machine translation, VSL is still an open problem and there are many approaches. Bilingual data of Vietnamese sentences - VSL syntax correct sentences has not been studied yet. With this translation problem, data is a very important part. In order to form modern translation models, it is necessary to construct enough data, so the proposed method of enriching this data on the wordnet is an option that has achieved good results. The evaluation results show that the BLEU scores in our test sets are much higher than other bilingual translation problems. For the reason in our problem, the translation model is almost unchanged with mostly the same language units. However, here we use this proposed method for a new and scientifically significant issue as well as to be applicable in practice.

References

1. Sandler W., Lillo-Martin. *Sign Language and Linguistic Universals*. J. Linguist. page 738-742, 2006.
2. Achraf Othman and Mohamed Jemni, *Statistical Sign Language Machine Translation from Englishwritten text to American Sign Language Gloss*, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3, 2011.
3. Stephen Cox, Michael Lincoln, Judy Tryggvason, Melanie Nakisa, Mark Wells, Marcus Tutt, "*Tessa, a system to aid communication with deaf people*", Proceedings of the fifth international ACM conference on Assistive technologies, 2002.
4. J. A. Bangham, S. J. Cox, R. Elliot, J. R. W. Glauert, I. Marshall, S. Rankov, and M. Wells, "*Virtual signing: Capture, animation, storage and transmission – An overview of the ViSiCAST project*" , IEEE Seminar on Speech and language processing for disabled and elderly people, 2000.
5. Angus Grieve-Smith, *SignSynth: A Sign Language Synthesis Application Using Web3D and Perl*, Conference: Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction, 2002.
6. Bernd Krieg-Brückner, Jan Peleska, Ernst-Rüdiger Olderog, Alexander Baer, *The Uniform Workbench, a Universal Development Environment for Formal Methods*, Lecture Notes in Computer Science 1709, Springer 1999.
7. L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler, M. Palmer, "*A Machine Translation System from English to American Sign Language*", Envisioning Machine Translation in the Information Future, Vol. 1934, pp. 191-193, 2000.
8. Quach, LD., Duong-Trung, N., Vu, AV., Nguyen, CN, *Recommending the Workflow of Vietnamese Sign Language Translation via a Comparison of Several Classification Algorithms*. In:Computational Linguistics, Communications in Computer and Information Science, vol 1215. Springer, 2020.
9. Akash Chintha, Ifeoma Nwogu, Shagan Sah, *Deep Learning Methods for Sign Language Translation*, ACM Transactions on Accessible Computing, DOI: 10.1145/3477498, 2021.
10. Thi Bich Diep Nguyen and Trung-Nghia Phung, "*Some issues on syntax transformation in Vietnamese sign language translation*". Sign Language Studies. IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.5, 2017.

11. Thi Bich Diep Nguyen, Trung-Nghia Phung, Tat-Thang Vu, "*A rule-based method for text shortening in Vietnamese sign language translation*". Springer AISC, Vol. 672, Proc. of INDIA, 2017.
12. https://www.nltk.org/howto/wordnet.html.
13. Tu- Bao Ho, Phuong-Thai Nguyen, et al. *https://vlsp.hpda.vn/*. VLSP research topic 2022.
14. Papineni K., Roukos S., Ward T., Zhu Z-J, "*BLEU: A method for Automatic Evaluation of Machine Translation*", Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, p.p 311-318, 2001
15. https://www.letsmt.eu/Bleu.aspx
16. Liu, Hongzhe, Wang Pengfei, *Assessing Sentence Similarity Using WordNet based Word Similarity*, Journal of software, Vol. 8, No. 6, DOI:10.4304/jsw.8.6.1451-1458, 2013.
17. Kayo Yin, Jesse Read, "*Better Sign Language Translation with STMC-Transformer*", Proceedings of the 28th International Conference on Computational Linguistics, 2020.
18. [30] Dangsaart. S, Naruedomkul .K, Cercone. N., & Sirinaovakul. B, *Intelligent Thai text – Thai sign translation for language learning*, Computers Education, 51(3), 1125–1141, 2008.

Journal of Computer Science and Cybernetics, V.39, N.2 (2023), 143–158
DOI no 10.15625/1813-9663/18025

A STUDY OF DATA AUGMENTATION AND ACCURACY IMPROVEMENT IN MACHINE TRANSLATION FOR VIETNAMESE SIGN LANGUAGE

THI BICH DIEP NGUYEN[1,2,*], TRUNG-NGHIA PHUNG[2], TAT-THANG VU[3]

[1] *Graduate University of Science and Technology, Vietnam Academy of Science and Technology, Ha Noi, Viet Nam*
[2] *Thai Nguyen University of Information and Communication Technology, Viet Nam*
[3] *Institute of Information Technology, Vietnam Academy of Science and Technology, Ha Noi, Viet Nam*

Abstract. Sign languages are independent languages of deaf communities. The translation from normal languages (i.e., Vietnamese Language - VL) as long as other sign languages to Vietnamese sign language (VSL) is a meaningful task that breaks down communication barriers and improves the quality of life for the deaf community. In this paper, we experimented with and proposed several methods for building and improving models for the VL to VSL translation task. We presented a data augmentation method to improve the performance of our neural machine translation models. Using an initial dataset of 10k bilingual sentence pairs, we were able to obtain a new dataset of 60k sentence pairs with a perplexity score no more than 1.5 times that of the original dataset. Experiments on the original dataset showed that rule-based models achieved the highest BLEU score of 68.02 among the translation models. However, with the augmented dataset, the Transformer model achieved the best performance with a BLEU score of 89.23, which is significantly better than that of other conventional approach methods.

Keywords. Natural language processing; Machine translation; Vietnamese sign language; Data augmentation.

1. INTRODUCTION

Sign language has been developed for a long time and is recognized as the official language of the deaf community in various countries. The sign language used by the deaf community in Vietnam is called Vietnamese sign language (VSL). Although sign language has many similarities with spoken language, there are significant differences between sign language and spoken/written language [24]. For example, in American sign language (ASL), there is a separate grammar system (separate rules for phonetics, morphology, syntax, and semantics) that differs from English [21]. Similarly, VSL is used as the official language in the deaf community of Vietnam with about 7 million people. Like other foreign languages, the

*Corresponding author.
E-mail addresses: ntbdiep@ictu.edu.vn (T.B.D.Nguyen); ptnghia@ictu.edu.vn (T.N.Phung); vtthang@ioit.ac.vn (T.T.Vu).

communication barrier is significant if one cannot understand and interpret sign language.

The sign language interpretation process involves two tasks, which are translating from sign language to spoken language and vice versa. Among them, the translation task from spoken language is an important task to convey information and provide social knowledge to the deaf.

The process of translating spoken language into sign language involves the following steps.

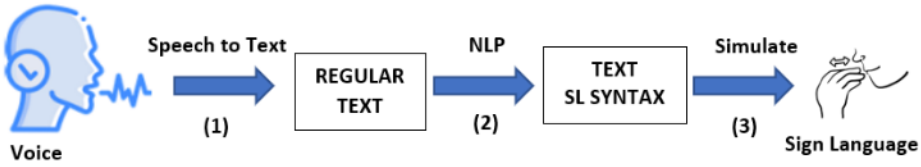


Figure 1: The process of translating speech into sign language

In which, (1) refers to the process of translating speech recognition into text. Many studies and applications have effectively handled this task, such as Google's API. (2) is the process of processing ordinary text into correct syntax in sign language. (3) is the process of simulating correctly syntaxed sign language text into representations such as 3D models, videos, or images of sign language.

In this procedure, the second step gets the most attention due to the completion of the conveyed message. The basic challenge is that sign language, in general, has a limited vocabulary compared to spoken/written language. If the machine translation is poorly performed, the complete message might not be successfully communicated, or in some cases, the conveyed message has a different meaning from the original [17].

The VSL translation task involves taking a regular Vietnamese sentence as input and producing an image, video, or 3D model as the final output. However, an important intermediate step in the translation process is to convert the regular Vietnamese sentence to a syntactically correct sentence in VSL. This is because VSL has some basic features such as reductionism, emphasis on focal points, and changes in word order compared to regular Vietnamese. In addition, there have been proposed technical methods for representing syntactically correct VSL sentences as images or 3D models that have produced good results. This means that the components of the sentence are separated, and we store them in a dictionary as a code that contains two components: the word/phrase and how it is represented using a 3D model. The soft-linkage motion between the sentence components is handled using interpolation techniques. Therefore, the scope of the task is focused on translating regular Vietnamese sentences into syntactically correct sign language sentences.

With remarkable advances in information technology, there have been the sign language translation systems developed worldwide, such as TESSA, which translates speech into British sign language (BSL) [4]; ViSiCAST, a tool for translating English into British sign language [2]; SignSynth project that employs the ASCII-Stokoe model [9]; ASL Workbench, an automated text-to-American sign language translation system [16]; and TEAM project, a system that translates text into American sign language using a contiguous bilingual parse tree technique [26]. Most of these research projects initially relied on structural-based translation models.

Recent studies have been maximizing the use of advances in natural language processing (NLP), deep neural networks (DNN), and machine translation (MT) to develop systems

that can translate between sign language and spoken language, to bridge the communication gap between the sign language community and the spoken language community [3]. A recent study of Gouri Sankar Mishra and colleagues proposed a system for translating spoken English into Indian sign language (ISL)[18]. The translation model follows a rule-based approach in which a parser is used to parse the full English sentence into a dependency structure representing the syntax and grammar information of a sentence. An ISL sentence is then generated from an ISL bilingual dictionary and a word network, with the ISL cues corresponding to the appropriate ISL signs being displayed.

Galian et al. experimented with two NMT architectures with optimized hyperparameters, various tokenization methods, and two data augmentation techniques (back-translation and paraphrasing). Through experimentation, they achieved significant improvements for models trained on the Phoenix 14T and DGS datasets for German sign language [1]. Following research on sign name adoption by Ka corri et al. [10], acceptance within the Deaf community is crucial for the application of sign language technologies. The perspective of Deaf users must be accurately analyzed, and the implementation of technology for the deaf community must be effective [5].

Currently, machine translation of Vietnamese sign language (VSL) is still a new and underexplored research field. Like other sign language translation problems in the world, many studies on VSL focus on the second step of the translation process - translating from regular text to the correct syntax in sign language. Therefore, there have been some studies on VSL related to the problem of translating Vietnamese to VSL with promising results, but there are also many limitations. The prominent limitation of these studies is the small database, which leads to low accuracy [6, 7, 26].

We have achieved certain results with the methods and translation models on a small dataset that we have constructed. Our research process has gone through several stages [19, 20]. Initially, we proposed a rule-based translation method based on the syntax rules of VSL. In this paper, we have experimented with some more advanced machine translation methods using a neural network approach and proposed a simple data enrichment method to apply to translation models. This is necessary for training models to help the translation system become more accurate. Section 3 presents the experimental results with some proposed modern translation models, and finally, a detailed analysis and evaluation results will be presented.

2. DATA AUGMENTATION

2.1. Data augmentation background

The base dataset is a bilingual corpus consisting of 10,000 sentence pairs in Vietnamese - Vietnamese sign language that we semi-automatically built and evaluated by language experts. The process of constructing the bilingual data is described in the following steps:

Step 1. Build the VSL-lexicon dictionary. The VSL-lexicon data stores lexical units with accompanying information such as word type, annotation code, synonyms, and corresponding animation models. Due to the difficulty of manually producing animation models with a large workload, currently, there are only 200 models in the VSL-lexicon. The models are saved in .FBX files. For the ".FBX" file format, 3D models can be exported with all animations, motions, rigging, and other parameters stored in the file. The ".FBX" file format is supported

by many different 3D software and is the standard file format used in Unity. Table 1 describes the structure of the VSL-lexicon data.

Table 1: Table describing the VSL-lexicon dictionary

ID	Lexical unit	Lexical category	Synonym	Tag code	Corresponding 3D animation model
1	a	Alphabet		VSL0001	M3D0001.FBX
2	ă	Alphabet		VSL0002	M3D0002.FBX
153	Tôi (I)	Pronoun (P)	tao, tớ	VSL0153	M3D0153.FBX
154	họ (They)	Pronoun (P)		VSL0154	M3D0154.FBX
296	chết (die)	Verb (V)	hi sinh, tử nạn	VSL0296	M3D0296.FBX
3035	trường học (school)	Noun (N)		VSL3035	M3D3035.FBX
3036	Nhà (house)	Noun (N)		VSL3036	M3D3036.FBX
6176	xương rồng (Cactus)	Noun (N)		VSL6176	Not in database yet

In this dictionary, there is a compilation of a set of synonyms to maximize the representation of words/phrases in Vietnamese sentences to VSL, as the lexicon of sign language is limited.

Step 2. Construct the Vie-VSL-10K dataset, which consists of bilingual sentence pairs. The data includes sentences in the communication domain, partially processed using automatic methods. We utilize a Vietnamese syntactic parsing toolkit, which is a research product by Dr. Nguyen Phuong Thai and colleagues, for our specific task. The preprocessing stage involves data normalization along with tokenization and part-of-speech tagging using the VietWS toolkit [11]. Subsequently, this dataset undergoes preliminary reviews and finally, the data is evaluated by a group of sign language experts. Finally, we have collected 10,000 bilingual sentence pairs in Vietnamese and VSL. The data is publicly available and shared at https://github.com/BichDiep/data-rules-VSL. We propose a method to augment this dataset based on Wordnet from the original 10,000 sentence pairs. Table 2 provides some examples of the different syntax between regular Vietnamese sentences and the correctly formatted VSL sentences in the Vie-VSL-10K dataset we have constructed.

Table 2: Syntax differences between regular Vietnamese sentences and correctly formatted VSL sentences.

ID	The Vietnamese sentence is syntactically analyzed.	The VSL sentence is syntactically analyzed.
1	SQ (NP (N Bạn) (N tên)) (VP (V là) (WHNP (P gì))) (? ?)	SQ (NP (N Bạn) (N tên) (P gì)) (? ?)
2	S (NP (P Tôi)) (NP (N tên)) (VP (V là) (NP (Np Hiếu))) (..)	S (NP (P Tôi)) (NP (N tên) (Np Hiếu)) (..)
3	S (NP (N Khế)) (C thì) (AP (A chua)) (..)	S (NP (N Khế)) (AP (A chua)) (..)
4	S (NP (P Tôi)) (NP (M 19) (N tuổi)) (..)	S (NP (P Tôi)) (NP (N tuổi) (M 19)) (..)
5	S (NP (P tôi)) (VP (R không) (V đi)) (..)	S (NP (P tôi)) (VP (V đi) (R không)) (..)
6	S (NP (P tôi)) (VP (R không) (V chơi)) (..)	S (NP (P tôi)) (VP (V chơi) (R không)) (..)
7	S (NP (P Tôi)) (VP (V thích) (NP (N mèo))) (..)	S (NP (P Tôi)) (VP (N mèo) (V thích)) (..)
8	SQ (NP (P Ai)) (VP (V biết) (VP (V bơi))) (? ?)	SQ (VP (V Biết) (VP (V bơi) (NP (P ai)))) (? ?)

The idea behind data augmentation is to substitute words in a sentence to generate new data. The newly generated sentences maintain the same syntax and logical coherence, so the translation to VSL (Visual sign language) follows the same conversion rules. This ensures accurate translation while preserving semantic similarity, as evaluated in the experimental phase. We have observed that the semantic relationships between words in Wordnet align perfectly with the concept of data augmentation. Therefore, we propose a data augmentation method based on Wordnet.

The Wordnet semantic network is a lexical dataset that represents semantic relationships between words. Wordnet only captures semantic relationships and does not encompass phonetic or morphological relationships [23].

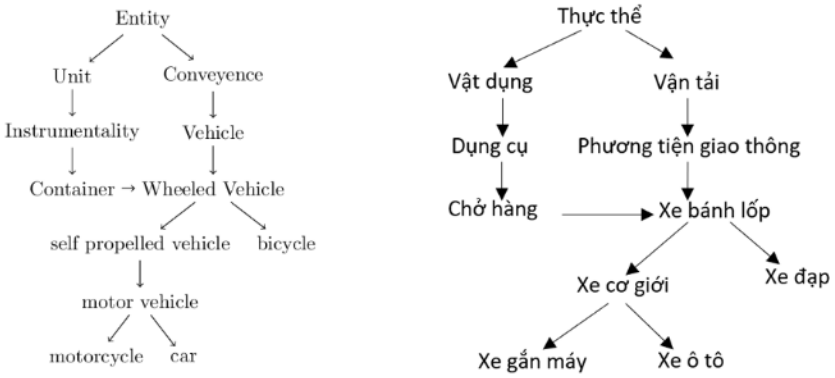


Figure 2: Hierarchical structure of Wordnet

Syntactic parsing provides us with the syntactic structure of a sentence. However, syntactic parsing only checks for grammatical correctness and does not verify semantic correctness. Take the sentence "cái bàn ăn con gà" (the table eats the chicken) as an example. If we analyze this sentence syntactically, we find that it is grammatically correct ("cái bàn" serves as the subject, "ăn" is the verb, and "con gà" functions as the object). However, it is evident that "cái bàn" cannot "ăn" "con gà". Instead, if we replace it with "con chó ăn con gà" (the dog eats the chicken), it becomes more logical. So, how can we determine if "cái bàn" or "con chó" can "eat" "con gà"? - By using the hyponym-hypernym relationship in Wordnet. Let's assume there is a heuristic that only "động vật" (animals) can perform the action of "ăn" (eating). Therefore, to check if an object can eat, we check if it is "động vật" by traversing its hypernyms. By traversing the hypernyms in reverse, we can easily determine that the "con chó" (dog) can perform the action of "ăn" (eating), whereas the "cái bàn" (table) cannot. Similarly, we can add semantic constraints to ensure semantic correctness in the sentence. This allows us to generate new sentences by replacing words with the same hypernym. The hierarchical structure with the keyword "con chó" is depicted in Figure 3.

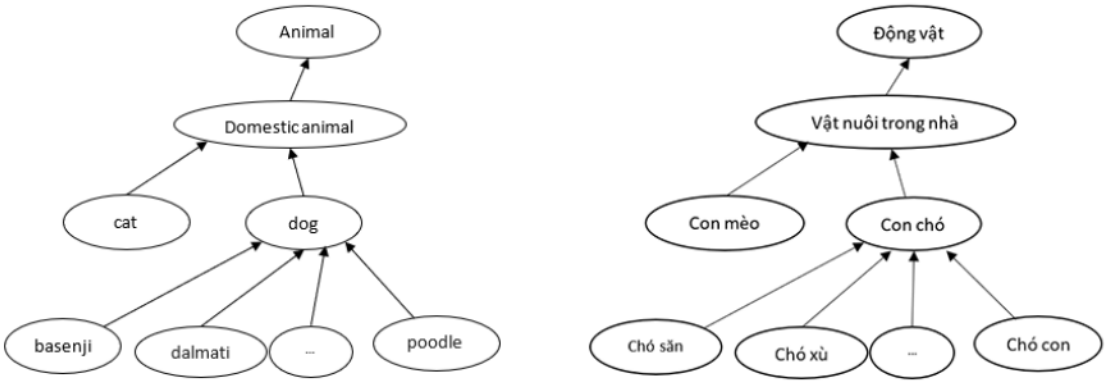


Figure 3: The structure of the hypernyms - hyponyms for the keyword "con chó"

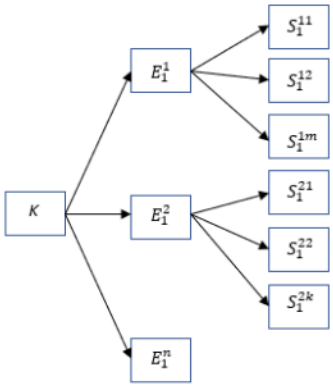


Figure 4: Illustration of criteria using the Synset E_i^j

In our problem, we use three criteria:

Sibling criterion: applied when all synset sets S_i^j when all synset sets contain sibling synsets (with the same synset and hypernym). Then the synset $\{E_1^1, E_1^2, \dots\}$ is selected as sibling synsets.

That is

$$SV = \{S_i^{jk} / S_g \in S_i^j (\forall j : 0 \leqslant j \leqslant n_i^j), S_p is\_hyper S_i^{jk}\}.$$

Parent-child criterion: applied when the synset sets S_i^j contain a synset that is superior to the remaining synsets (as long as each remaining synset has a synset that is a subordinate of the above-mentioned superior synset). Then the synset $\{E_1^1, E_1^2, \dots\}$ is selected as sibling synsets.

That is

$$SV = \{S_i^{jk} / \exists S_p \in S_i^h (h \in [1...n_i^j]), S_i^{jk} \in S_i^h), (\forall j : 0 \leqslant j \leqslant n_i^j, j \neq h), S_p is\_hyper S_i^{jk}\}.$$

Grandparent - grandchildren criterion: Applied when in synset sets S_i^j contain a synset that is superior to the remaining synsets (as long as each remaining synset has a synset that is a subordinate of the above-mentioned superior synset). Then the synset $\{E_1^1, E_1^2, \dots\}$ is selected as these subordinate synsets.

$$SV = \{S_i^{jk} / \exists S_g \in S_i^h (h \in [1...n_i^j]), S_i^{jk} \in S_i^j), (\forall j : 0 \leqslant j \leqslant n_i^j, j \neq h), S_g is\_dist\_hyper S_i^{jk})\}$$

Thus, when the word W appears in a phrase, W can be replaced with W' if W and W' satisfy the sibling, parent-child, and grandparent-grandchild criteria. Therefore, depending on the structure of the hypernyms and hyponyms and other characteristics of Wordnet, we may construct fuzzy data by changing words in previous phrases according to predetermined criteria.

2.2. Data augmentation process

Based on the characteristics and properties of Wordnet for semantic constraints to verify semantic correctness in a sentence, we integrate it with the Vietnamese Wordnet dataset from the VLSP (association for Vietnamese language and speech processing) community. This dataset comprises 10,000 core vocabulary units, each containing information such as English translations, synonyms, antonyms in Vietnamese, and hypernym-hyponym structure [11]. The data augmentation algorithm is described in pseudocode as follows.

From there, we have the process of constructing new data through the following steps.

Algorithm: Data-augment-VSL

Input: Sentences S
Output: Set of sentences S' are generated based on S
1: Split W word $\in S$
2: $X \leftarrow W.hypernyms()$
3: For $i = 1, n$ do
    $X_i \Leftarrow X.hyponyms()$
    Add X_i to set T
4: While !$\exists$ $X_i.hyponyms$:
    $Y_i \Leftarrow X_i.hyponyms()$
    Add Y_i to set T
5: Replace replace each element in T, create new data S'
6: Return Set of sentences S'

We proceed to construct new data based on a set of initially built data. Our data is evaluated by a community of individuals who are deaf and language experts in the field. Subsequently, we enrich the data using the proposed method.

Figure 5 illustrates the process of generating new data from an original sentence S. The sentence "tôi ăn táo" (I eat an apple) is syntactically parsed, and the noun "táo" (apple) is extracted from the sentence. Applying the algorithm, a set T is obtained, which consists of words that can be used as replacements to generate a new set of sentences, S'. This set T includes 92 words (excluding the root word), resulting in the generation of 92 new sentences.

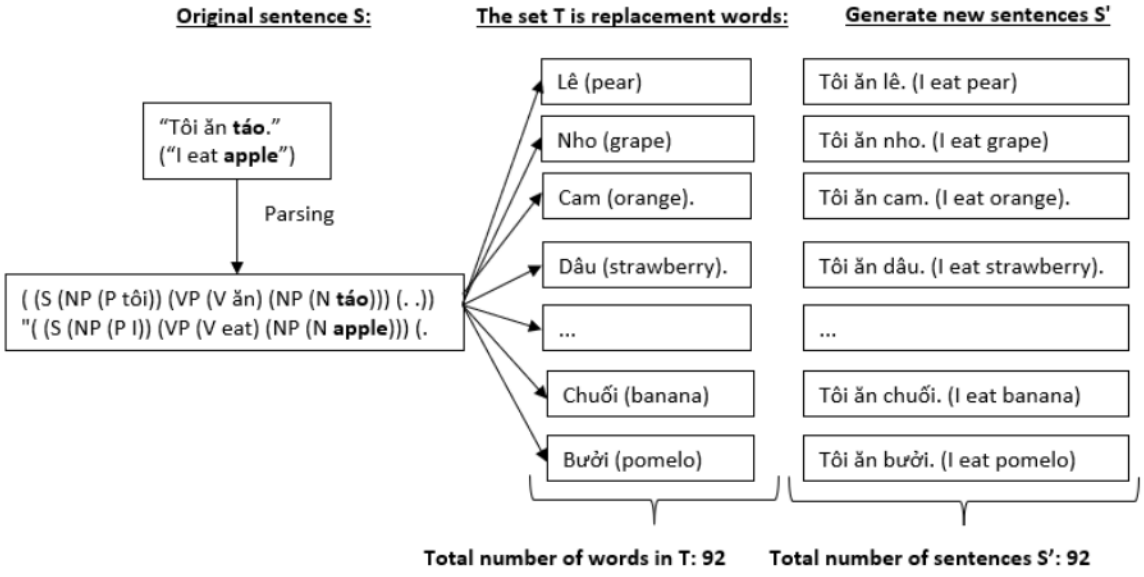


Figure 5: Example of generating new data from an original sentence

After experimenting with a set of data, it was observed that verb types, when using the method of searching for words with shared hypernyms based on sibling, parent-child, and grandparent-grandchild criteria, did not meet semantic requirements. Therefore, only pronouns, nouns, and adjectives were considered. Table 3 presents some sets T and summarizes the number of enriched sentences generated by the proposed algorithm (where T represents

the set of words with shared hypernyms based on the applied criteria for each word type, WS represents the number of original data sentences containing a word from the word type being considered, and W'S represents the number of enriched sentences from all original sentences containing a word from the word type being considered).

In the initial dataset of 10,000 sentences, due to the chosen domain of communication, pronouns constitute a significant portion of the vocabulary. Additionally, the categorization of nouns and adjectives is derived from their hypernym groups. This ensures that the replacement of words to generate new sentences maintains semantic similarity.

The similarity of the dataset before and after augmentation can be evaluated based on the language model's perplexity for each type. Perplexity is a measure used in probability and statistics to assess the effectiveness of a language model. In an n-gram language model, perplexity measures the model's ability to predict a new text segment based on the probability of n-grams in the model. Perplexity in an n-gram language model is calculated using the following formula

$$Perplexity(W) = \sqrt[n]{\frac{1}{P(w_1, w_2, ..., w_N)}}$$

where, N is the order of the n-gram model; $P(w_1, w_2, ..., w_N)$ is the probability of the test text segment in the n-gram language model; $\sqrt[n]{...}$ denotes taking the Nth root, where N is the number of words in the test text segment. This formula helps normalize perplexity to make it independent of the size of the text segment.

The smaller the perplexity, the better the model performs, indicating its ability to predict new word sequences. In n-gram language models, perplexity is often used to compare different models and evaluate their effectiveness in language prediction [8]. The lowest perplexity reported was in 1992 on the Brown Corpus dataset (1 million words of American English across various topics and genres), with an actual value of approximately 247, corresponding to a cross-entropy of $\log_2(247) = 7.95$ bits per word or 1.75 bits per character using a 3-gram model. Lower perplexity levels can often be achieved with more specialized datasets as they are easier to predict. The perplexity score of a dataset depends on various factors such as the size of the dataset, the complexity of the language structure, the vocabulary richness, and so on. In many cases, perplexity tends to increase with the size of the dataset, especially when the dataset size significantly grows. However, this increase does not always occur and can be limited by the complexity of the language structure or vocabulary richness. Table 4 presents the perplexity scores for the constructed datasets using a 3-gram language model, comparing them with some commonly used datasets.

Table 3: Perplexity scores of the datasets

Dataset	**Average perplexity score**
WikiText-103	109-113
Penn Treebank	110-120
Common Craml	600-800
Vie-VSL10k	**300-420**
Vie-VSL10k	**450-250**

Thus, we can observe that despite the dataset is more than six times larger than the original one, the perplexity score is only slightly higher, by no more than 1.5 times. This indicates

that the language model with a 3-gram approach performs well in terms of data efficiency. Additionally, the high similarity between the original and newly generated sentences, which preserves the syntactic structure, further supports this notion. In terms of semantics, the similarity is ensured by the hyponym relationship among words, as defined by the applied standards.

Table 4: Results of the data augmentation algorithm from Vie-VSL10K

Lexical category	Group	Example	T	WS	W'S
Noun	Plant 1 (fruits)	Bưởi, cam, nho, táo,.. (Pomelo, orange, grape, apple, etc.)	92	35	3220
	Plant 2 (flowers)	Hoa cúc, hoa hồng, hoa ly,... (Chrysanthemum, rose, lily, etc.)	183	5	915
	Plant 3 (general)	Cây, hoa, cỏ, lá, rau,... (Tree, flower, grass, leaf, vegetable, etc.)	438	10	2628
	Food	Bánh, kẹo, bia, thịt, rau... (Cake, candy, beer, meat, vegetable, etc.)	471	3	1413
	Animal 1 (pets)	chó, chó con, chó xù, gà, mèo,... (Dog, puppy, poodle, chicken, cat, etc.)	25	5	125
	Animal 2 (others)	Báo, hổ, hươu,.. (Tiger, lion, giraffe, etc.)	708	3	2124
	Object 1 (household items)	Bàn, ghế, tủ,.. (Table, chair, cabinet, etc.)	257	11	2827
	Object 2 (tools)	Buá, kéo, máy,.. (Hammer, scissors, machine, etc.)	1564	4	5056
	Object 3 (vehicles)	Xe máy, ô tô, xe chở hàng, .. (Motorcycle, car, truck, etc.)	78	7	546
	Weather	Nắng, mưa, gió,.. (Sun, rain, wind, etc.)	63	5	315
	Occupation	Giáo viên, công nhân,... (Teacher, worker, etc.)	21	8	168
	Body parts	Chân, tay, tóc, má, môi,... (Leg, arm, hair, cheek, lips, etc.)	231	4	924
	Geometric shapes	Tam giác, hình tròn, hình vuông,... (Triangle, circle, square, etc.)	134	3	402
Adjective	Color	Đỏ, xanh, vàng, tím,... (Red, green, yellow, purple, etc.)	12	36	432
	Material property	Nặng, nhẹ, Cứng, mềm,... (Heavy, light, hard, soft, etc.)	45	2	90
	Size	To, rộng, dài, ngắn,... (Big, wide, long, short, etc.)	15	4	60
	Emotions	vui, buồn, lo lắng,... (Happy, sad, worried, etc.)	279	7	1953
	Personality	hài hước, cục cằn, dễ thương... (Funny, grumpy, adorable, etc.)	23	4	92
Pronoun		Tôi, họ, chúng ta, .. (I, they, we, etc.)	12	3424	41088
				Total:	64378

3. STATE-OF-THE-ART MACHINE TRANSLATION MODELS FOR VSL

3.1. Sequence to sequence model

The "sequence to sequence" (Seq2Seq) model is one of the successful models in the field of natural language processing [12]. This model offers several advantages, including its applica-

bility to various tasks, especially in addressing natural language processing problems such as machine translation, text summarization, question answering, and many other applications. It possesses the capability to learn transformations from training data: Seq2Seq enables the learning of converting one type of data into another type. It is easily scalable, allowing the handling of input and output data of different sizes. Seq2Seq exhibits high accuracy, generating precise and natural outputs, particularly in machine translation and text summarization tasks. Furthermore, it can be combined with other models, such as the attention model, to enhance performance and accuracy. Therefore, for the translation of Vietnamese sentences into grammatically correct VSL sentences, utilizing the Seq2Seq model in combination with attention is a feasible approach. The Seq2Seq model consists of two main components: the encoder and the decoder. In the encoder, the input sentence, which is in Vietnamese, is transformed into a semantic vector using an LSTM model to encode information from each word in the sentence. In the decoder, the semantic vector is fed into the model to decode and generate the corresponding VSL output sentence using another LSTM model. The encoder and decoder components of the Seq2Seq model are illustrated in Figure 6.

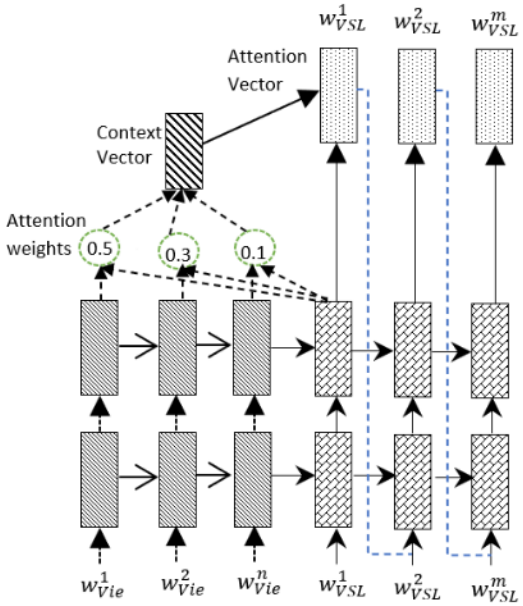


Figure 6: The encoder-decoder architecture of the Seq2Seq model in the Vietnamese-VSL translation task.

At each time step, the output of the decoder is combined with the weighted sum over the encoded input to predict the next word in the sentence. The decoder utilizes selective attention over parts of the input sequence. Attention takes a sequence of vectors as input and returns an attention vector. To train the Seq2Seq model, we need to use the input and output data in the form of parallel sentence pairs. In this case, with 60,000 sentence pairs, we used a simple yet effective Seq2Seq model with the following basic parameters:

- Batch size: 128;
- Number of epochs: 10;
- Learning rate: 0.001-0.01;

- Model architecture: LSTM with 3 hidden layers with a dimension of 256;
- Training time: 4.5 hours with a training speed on CPU of approximately 30-40 samples/second;
- GPU: NVIDIA Tesla T4.

3.2. Transformer model

The transformer is a recent and well-known model in the natural language processing community that has made significant breakthroughs in machine translation tasks since its introduction in 2017 [13]. With the ability to leverage the parallel computing power of GPUs to accelerate training speed for language models and overcome the issue of handling long sentences, the transformer model is considered suitable for the automatic VSL translation task. The initial steps in applying this model to the task include data encoding and decoding, applying the translation model, and evaluating the effectiveness of the translations.

A. Encoding and Decoding

First, the data needs to be transformed into a numerical representation. Typically, the text is converted into an encoded sequence, which is used as input to create an embedding. The training data consists of two tokenized forms of text, one for regular Vietnamese and one for VSL. Both employ similar methods. The encoding process converts a series of sentences into tokens. The decoding process converts these tokens back into human-readable text.

• Setting up the input pipeline: To construct a suitable input pipeline for training, some transformations need to be applied to the dataset. The following function will be used to encode batches of raw text.

```
def tokenize_pairs(vsl, vi):
    vi = tokenizers.pt.tokenize(vi)
    # Convert from ragged to dense, padding with zeros.
    vsl = vsl.to_tensor()
    vi = tokenizers.vi.tokenize(vi)
    # Convert from ragged to dense, padding with zeros.
    vi = vi.to_tensor()
    return vsl, vi
```

Positional Encoding: Attention layers treat the input as a set of unordered vectors. This model does not contain any recurrent layers. Therefore, a "positional encoding" is added to provide the model with information about the relative positions of tokens within a sentence. The positional encoding vector is added to the embedding vector. The embedding vector represents a token in a d-dimensional space, where tokens with similar meanings are closer to each other. However, the embedding does not encode the relative positions of tokens within a sentence. Hence, after adding positional encoding, the tokens will be closer based on both their semantic similarity and their positions within the sentence, in the d-dimensional space. The formula for calculating the positional encoding is as follows

$$PE_{(pos,2i)} = \sin(pos/1000^{2i/d_{model}}),$$
$$PE_{(pos,2i+1)} = \cos(pos/1000^{2i/d_{model}}).$$

• Look-ahead mask is used to hide future tokens in a sequence. In other words, the mask indicates which entries should not be used. This means that to predict the third token, only

the first and second tokens will be used. Similarly, to predict the fourth token, only the first, second, and third tokens will be used, and so on.

• The attention function used by the Transformer has three inputs: Q (query), K (key), and V (value). The equation used for computation is as follows

$$Attention(Q, K, V) = softmax_k(\frac{QK^T}{\sqrt{d_k}})V.$$

During the softmax normalization process applied to K, its values determine the level of importance for Q. The output represents the weighted sum of attention weights and the V (value) vector. This ensures that tokens of interest are preserved while irrelevant tokens are discarded.

B. Initializing the transformer model

The transformer consists of an encoder, a decoder, and a final linear layer. The output of the decoder serves as the input to the linear layer, and its output is returned.

• Setting hyperparameters.

• Optimization algorithm: Using the Adam optimization algorithm with customized learning rate scheduling (Adam is an extension of stochastic gradient descent that has been widely adopted for deep learning applications in computer vision and natural language processing) [15].

• Training and testing

```
Transformer = Transformer(
    num_layers=num_layers,
    d_model=d_model,
    num_heads=num_heads,
    dff=dff,
    input_vocab_size=tokenizers.pt.get_vocab_size().numpy(),
    target_vocab_size=tokenizers.en.get_vocab_size().numpy(),
    pe_input=1000,
    pe_target=1000,
    rate=dropout_rate)
```

Next, create a checkpoint path and a checkpoint manager to save checkpoints after every n epochs. The regular Vietnamese sentence is used as the input language, and VSL is the target language.

• The following steps are used for inference:

- Encode the input sentence using the Vietnamese tokenizer (tokenizers.vie). This serves as the input to the encoder.

- Initialize the input to the decoder as a token (Start).

- Compute the padding masks and look-ahead masks.

- The decoder then makes predictions by looking at the output of the encoder and its own output (self-attention).

- Concatenate the predicted tokens with the input to the decoder and pass it through the decoder. In this approach, the decoder predicts the next token based on the previously predicted tokens.

• Display attention: The translator class returns a dictionary mapping that can be used to visualize the inner workings of the model.

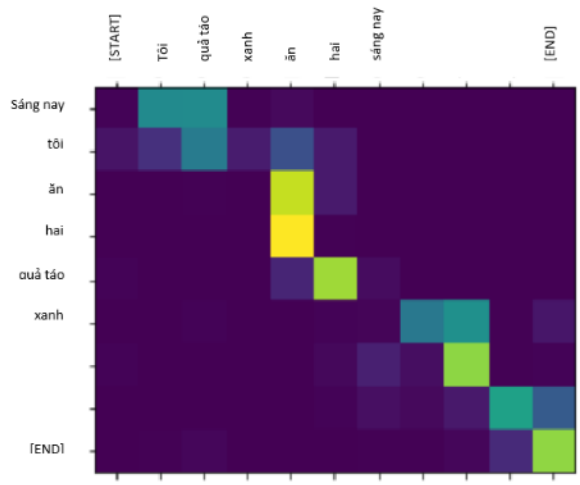


Figure 7: Attention map

Training time and environment:
- Training time: Approximately 8 hours with 30 epochs.
- Training environment: Configured with a Tesla T4 GPU and 16GB RAM.
- Batch size: 64.
- Number of layers in the model: 6.
- Number of heads in multi-head attention: 8.
- Embedding size: 512.
- Dimensionality of the Encoder and Decoder: 512.

4. EVALUATION OF RESULTS

With the parameters applied to the transformer translation model presented above, it is considered quite good and suitable for handling translation data with around 60,000 bilingual sentence pairs. The training time of 8 hours with 30 epochs is notably reasonable. The training environment on a Google Colab virtual machine with a Tesla T4 GPU and 16GB RAM is powerful and suitable for model training. A batch size of 64 is a suitable choice given the amount of data and other model parameters. The number of layers in the model of 6 and the number of heads in the multi-head attention of 8 are also appropriate and noteworthy parameters. The embedding size of 512 and the dimensionality of the encoder and decoder of 512 are common and suitable choices to achieve good results for the Transformer translation model. The Seq2seq model with the given parameters, including batch size ranging from 64 to 128, the number of epochs from 30 to 50, and learning rate from 0.001 to 0.01, along with the LSTM architecture consisting of 3 hidden layers with a hidden dimension of 256, has achieved good performance in the Vietnamese-VSL machine translation task. Due to the relatively low complexity of the input data and the high similarity between the two languages, the training time is better compared to other language pairs. Furthermore, to evaluate the experimental performance, we rely on the BLEU score is used to assess the data enrichment on a new dataset compared to the original dataset using various machine translation models. BLEU is a method for evaluating the quality of automatically generated

machine translations, originally proposed by IBM and widely used as a primary evaluation metric in machine translation research [14].

Table 5: Comparison of BLEU scores on models training with the original data and augmented data

No	Translation model	Original data	Augmented data
1	Rule-based translation	68.02	68.02
2	Seq2Seq	58.5	81.44
3	Transformer	65.2	89.23

Note: BLEU scores range from 0 to 100, with higher scores indicating better translation quality. The augmented data shows improved BLEU scores across all models, indicating better translation performance compared to the original data. Through the experimentatal process with the mentioned models, we can observe that with a training dataset of 10,000 sentence pairs, rule-based translation yields higher BLEU scores compared to statistical models. However, as the dataset size increases, the performance of statistical models gradually improves. Among the statistical models used in our research, the Transformer model consistently provides better results. However, it is worth noting that the BLEU score is appropriate for evaluation, but may not hold significant value when it comes to sign language translation or other specific language translation tasks. For example, the German sign language translation achieves an 82.87 BLEU score [25], and the Thai sign language translation [22], indicates the need for domain-specific evaluation metrics in such cases.

5. CONCLUSION

In this paper, we have addressed the challenges of the Vietnamese sign language translation problem. We proposed a simple and effective method for data augmentation based on Wordnet. The results showed that the augmented data increased sixfold while the perplexity score only increased by up to 1.5 times. This indicates that the language model with a 3-gram approach performs well in capturing semantic similarity. With the available data, we applied modern translation models such as Seq2Seq with attention and the transformer model to experiment with this data. The best achieved BLEU score is 89.23, which is for the transformer model using 60,000 bilingual sentence pairs for training data, outperforming other baseline methods. We observed that the transformer model with a pretrained model can be used effectively even with a small amount of training data, allowing us to apply various techniques designed for the transformer. The higher BLEU score compared to other language translation models is due to the unique characteristics of sign language translation. However, this score is not surprisingly high compared to other sign language translation tasks.

REFERENCES

[1] G. Angelova, E. Avramidis, and S. Möller, "Using neural machine translation methods for sign language translation," in *60th Annual Meeting of the Association for Computational Linguistics Student Research Workshop*, 2022, pp. 273–284.

[2] J. A. Bangham, S. J. Cox, R. Elliot, J. R. W. Glauert, I. Marshall, S. Rankov, , and M. Wells, "Virtual signing: Capture, animation, storage and transmission – an overview of the visicast

project," *IEEE Seminar on Speech and Language Processing for Disabled and Elderly People*, 2000.

[3] A. Chintha, I. Nwogu, and S. Sah, "Deep learning methods for sign language translation," *ACM Transactions on Accessible Computing*, vol. 14, pp. 1–30, 2021.

[4] S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, and M. Tutt, "Tessa - a system to aid communication with deaf people," 2002.

[5] B. D, K. H, and et al, "Sign language recognition, generation, and translation: An interdisciplinary perspective," in *The 21st International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS'19.* ACM Press, 2019, pp. 16–31.

[6] Q. L. Da and et al, "Converting the Vietnamese television news into 3D sign language animations for the deaf," *Lecture Notes of the Institute for Computer Sciences - Social Informatics and Telecommunications Engineering*, vol. 257, 2019.

[7] Q. L. Da and N. C. N, "Conversion of the vietnamese grammar into sign language structure using the example-based machine translation algorithm," in *International Conference on Advanced Technologies for Communications*, 2018, pp. 27–31.

[8] J. F and M. R. L, "Interpolated estimation of Markov source parameters from sparse data," in *Proceedings of The Workshop on Speech and Natural Language, Association for Computational Linguistics*, 1980, pp. 357–366.

[9] A. Grieve-Smith, "SignSynth: A sign language synthesis application using Web3D and perl," in *Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*, 2002.

[10] K. H., H. M., E. S., P. K., M. K., and W. M., "Regression analysis of demographic and technology-experience factors influencing acceptance of sign language animation," *ACM Transactions on Accessible Computing*, vol. 10, no. 1, pp. 1–33, 2017.

[11] T.-B. Ho, P.-T. Nguyen, and et al, "VLSP research topic," in *https://vlsp.hpda.vn/*, 2022.

[12] S. I., V. O., and L. Q. V, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[13] L. Jones, A. N. Gomez, and Łukasz Kaiser, "Attention is all you need," in *31st Conference on Neural Information Processing Systems USA*, 2017.

[14] P. K., R. S., W. T., and Z. J, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2001, pp. 311–318.

[15] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[16] B. Krieg-Brückner, J. Peleska, E.-R. Olderog, and A. Baer, "The uniform workbench - a universal development environment for formal methods," *Lecture Notes in Computer Science 1709*, 1999.

[17] Q. LD, Duong-Trung, N. Vu, and A. Nguyen, "Recommending the workflow of vietnamese sign language translation via a comparison of several classification algorithms," *Computational Linguistics, Communications in Computer and Information Science*, vol. 1215, 2020.

[18] G. S. Mishra, A. K. Sahoo, and K. K. Ravulakollu, "Word based statistical machine translation from English text to indian sign language," *ARPN Journal of Engineering and Applied Sciences*, vol. 12, no. 2, pp. 481–488, 2017.

[19] T.-B.-D. Nguyen, T.-N. Phung, and T.-T. Vu, "Some issues on syntax transformation in Vietnamese sign language translation," *Sign Language Studies. IJCSNS International Journal of Computer Science and Network Security*, vol. 17, no. 5, pp. 292–297, 2017.

[20] ——, "A rule-based method for text shortening in Vietnamese sign language translation," in *International Conference on Advanced Technologies for Communications*, 2018, pp. 655–662.

[21] A. Othman and M. Jemni, "Statistical sign language machine translation from Englishwritten text to American sign language gloss," *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 3, 2021.

[22] D. S, N. K, Cercone, and S. B, "Intelligent Thai text – Thai sign translation for language learning," *Computers Education*, vol. 51, no. 1, pp. 1125–1141, 2008.

[23] Soergel and Dagobert, "Wordnet. an electronic lexical database," 10 1998.

[24] S. W and Lillo-Martin, "Sign language and linguistic universals," *Linguist*, pp. 738–742, 2006.

[25] K. Yin and J. Read, "Better sign language translation with STMC-transformer," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

[26] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler, and M. Palmer, "A machine translation system from English to American sign language," *Envisioning Machine Translation in the Information Future*, vol. 1934, pp. 191–193, 2000.

Received on March 25, 2023
Accepted on May 11, 2023