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.

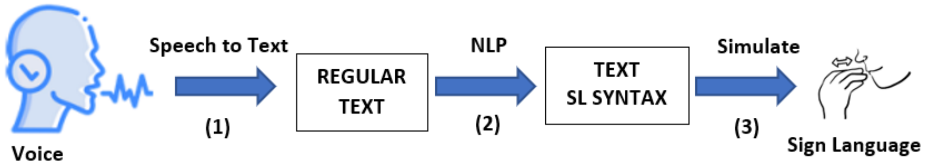

Figure 1: The process of translating speech into sign language

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].

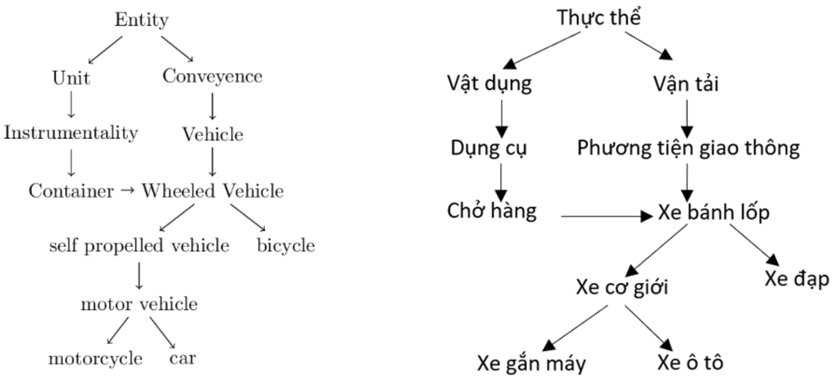

Figure 2: Hierarchical structure of Wordnet

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.

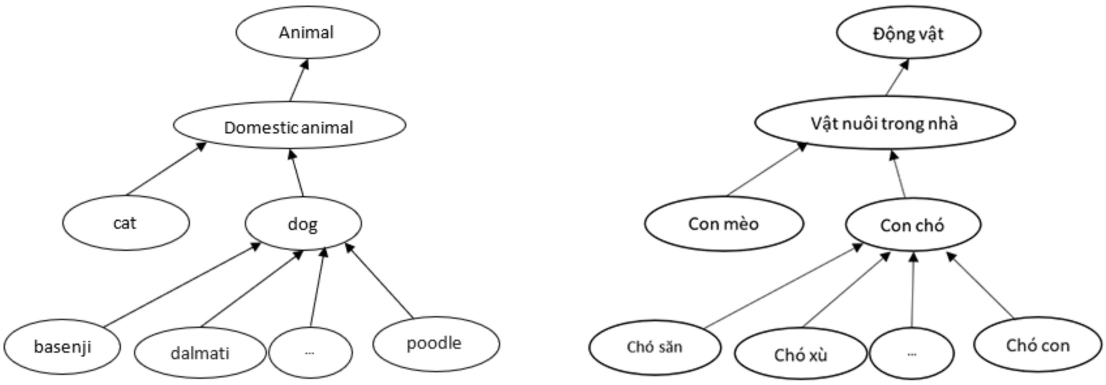

Figure 3: The structure of the hypernyms - hyponyms for the keyword “con chó”

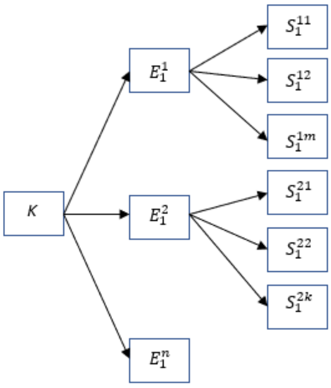

Figure 4: Illustration of criteria using the Synset E_i^j

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 $\{E_1^1, E_1^2, \ldots\}$ 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 $\{E_1^1, E_1^2, \ldots\}$ 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 $\{E_1^1, E_1^2, \ldots\}$ 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.

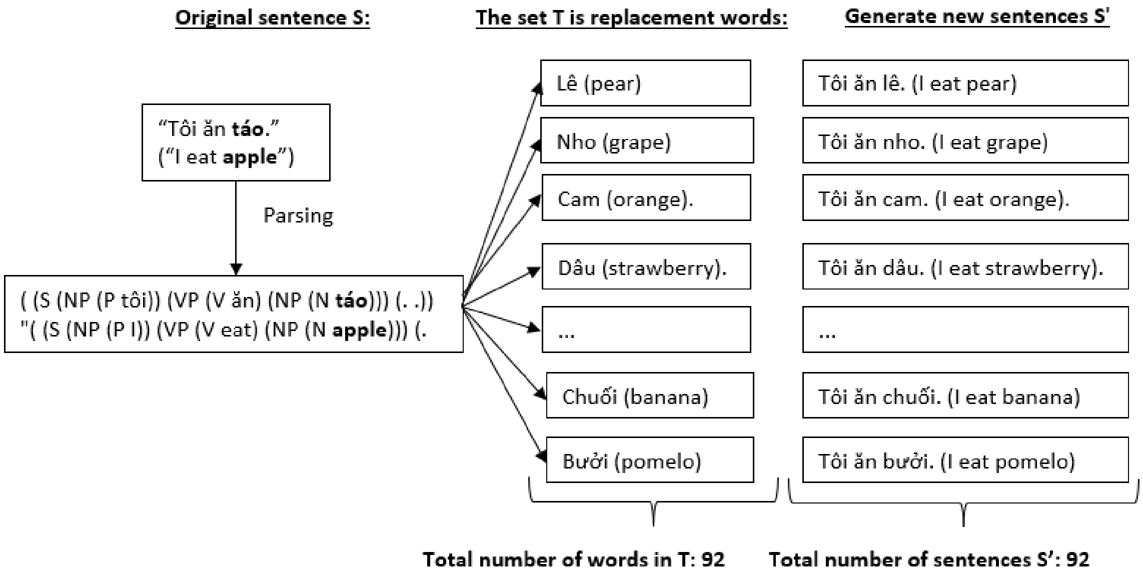
Original sentence S:
The set T is replacement words:
Generate new sentences S'
"Tôi ăn **táo**."
("I eat **apple**")
Parsing
((S (NP (P tôi)) (VP (V ăn) (NP (N **táo**))) (. .))
"((S (NP (P I)) (VP (V eat) (NP (N **apple**))) (.
Lê (pear)
Nho (grape)
Cam (orange).
Dâu (strawberry).
...
Chuối (banana)
Bưởi (pomelo)
Tôi ăn lê. (I eat pear)
Tôi ăn nho. (I eat grape)
Tôi ăn cam. (I eat orange).
Tôi ăn dâu. (I eat strawberry).
...
Tôi ăn chuối. (I eat banana)
Tôi ăn bưởi. (I eat pomelo)
Total number of words in T: 92
Total number of sentences S': 92

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 applicability 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.

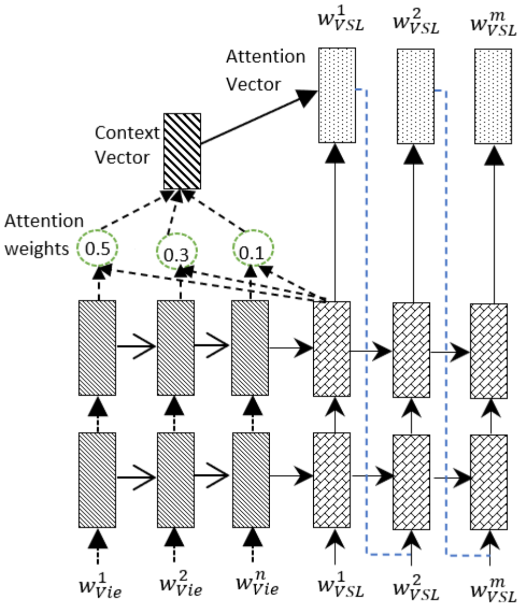

Figure 6: The encoder-decoder architecture of the Seq2Seq model in the Vietnamese-VSL translation task.

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.

```python
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.

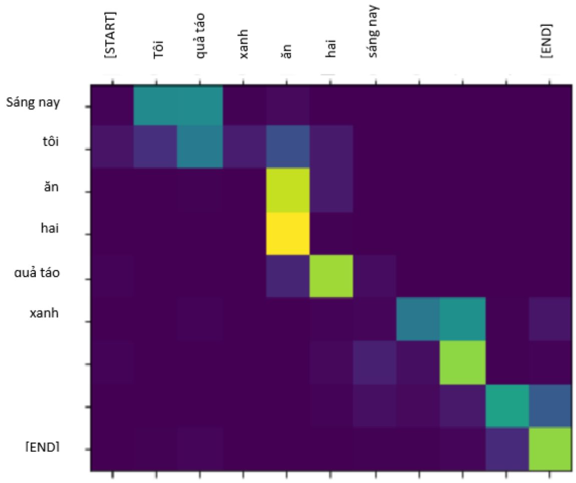

Figure 7: Attention map

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