

Neural Sign Language Translation based on Human Keypoint Estimation

Sang-Ki Ko

Chang Jo Kim

Hyedong Jung

Choongsang Cho

Korea Electronics Technology Institute

22 Dawangpangyo-ro 712 beon-gil, Seongnam-Si, Gyeonggi-do 13488, South Korea

{naram7, wowchangjo, hudson.keti, ideafisher.cho}@gmail.com

Abstract

We propose a sign language translation system based on human keypoint estimation. It is well-known that many problems in the field of computer vision require a massive amount of dataset to train deep neural network models. The situation is even worse when it comes to the sign language translation problem as it is far more difficult to collect high-quality training data. In this paper, we introduce the KETI sign language dataset which consists of 11,578 videos of high resolution and quality. Considering the fact that each country has a different and unique sign language, the KETI sign language dataset can be the starting line for further research on the Korean sign language translation.

Using the KETI sign language dataset, we develop a neural network model for translating sign videos into natural language sentences by utilizing the human keypoints extracted from a face, hands, and body parts. The obtained human keypoint vector is normalized by the mean and standard deviation of the keypoints and used as input to our translation model based on the sequence-to-sequence architecture. As a result, we show that our approach is robust even when the size of the training data is not sufficient. Our translation model achieves 94.6% (60.6%, respectively) translation accuracy on the validation set (test set, respectively) for 105 sentences that can be used in emergency situations. We compare several types of our neural sign translation models based on different attention mechanisms in terms of classical metrics for measuring the translation performance.

1. Introduction

Sign language recognition or translation is a study that interprets a visual language that has its independent grammar into a spoken language. The visual language combines various information on the hands and facial expression according to this grammar to present the exact meaning [12, 51]. The issue is a challenging subject in computer vision and a significant topic for hearing-impaired people.

In recent years, Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) architecture [17], and Gated Recurrent Units (GRUs) [6] in particular, have been primarily employed as essential approaches to model a sequence and solve the sequence to sequence problems such as machine translation and image captioning [9, 31, 47, 53]. Convolutional neural networks (CNNs) are powerful models that have archived excellent performance in various visual tasks such as image classification [19, 20], object detection [14, 42], semantic segmentation [32, 54], and action recognition [10, 34].

Sign language with a unique grammar express the linguistic meaning through the shape and movement of hands, moreover, the facial expression that present emotion and specific intentions [51]. Understanding sign languages that it requires a high level of spatial and temporal knowledge is difficult with the current level of computer vision techniques based on neural networks [11, 12, 15, 25, 27, 28, 46]. More importantly, the main difficulty comes from the lack of dataset for training neural networks. Many sign languages represent different words and sentences of spoken languages with gestures sequences comprising continuous pose of hands and facial expressions while ‘hand (finger) languages’ only represent each letter in an alphabet with the shape of a single hand [7]. This implies that there are uncountably many combinations of the cases even to describe a single human intention with the sign language.

Hence, we restrict ourselves to a specific domain which is related to various emergencies. We build the Korean sign language dataset collected from eleven Korean professional signers who are hearing-impaired people. The dataset consists of high-resolution videos that recorded Korean sign languages corresponding to 419 words and 105 sentences related to various emergency situations. We, then, present our sign language translation system based on human keypoints of hands, pose, and face. Our system is trained and tested with the sign language dataset built by our corpus, and we show a robust performance considering that the scale of dataset is not large enough.

2. Related Work

There have been many approaches to recognize ‘hand languages’ that are used to describe letters of the alphabet with a single hand. It is relatively easier than recognizing sign languages as each letter of the alphabet simply corresponds to a unique hand shape. In [11], an English alphabet is recognized using a Random Forest (RF) method to classify hand poses expressed by depth images and it shown 92% recognition accuracy. A pose estimation method of the upper body represented by seven key points was proposed for American Sign Language (ASL) alphabet recognition. [15]. We also note that there has been an approach by Kim et al. [23] to recognize the Korean hand language by analyzing latent features of hand images.

In general, researchers rely on the movements and shapes of both hands to recognize sign languages. Starner et al. [46] have developed a real-time system based on Hidden Markov model (HMM) to recognize sentence-level ASL. They have demonstrated two experimental results: they have used solidly colored gloves to make tracking of hands easier in the first experiment and the second experiment have been conducted without gloves. They have claimed that the word accuracy of glove-based system is 99.2% but the accuracy drops to 84.7% if they do not use gloves. It should be noted that those accuracy can be reached because they have exploited the grammar to reviewing the errors of the recognition. The word accuracy without grammar and gloves is 74.5%.

On the other hand, there have been approach to automatically learning signs from weakly annotated data such as TV broadcasts by using subtitles provided simultaneously with the signs [4, 8, 41]. Following this direction, Forster et al. released the RWTH-PHOENIX-Weather 2012 [12] and its extended version RWTH-PHOENIX-Weather 2014 [13] that consist of weather forecasts recorded from German public TV and manually annotated using glosses and natural language sentences where time boundaries have been marked on the gloss and the sentence level. Based on the RWTH-PHOENIX-Weather corpus, Koller et al. [27] have presented a statistical approach performing large vocabulary continuous sign language recognition across different signers. They have developed a continuous sign language recognition system that utilizes multiple information streams including the hand shape, orientation and position, the upper body pose, and face expression such as mouthing, eye brows and eye gaze.

Until recently, there have been many attempts to recognize and translate sign language using deep learning (DL). Oberweiger et al. [35] have introduced and evaluated several architectures for CNNs to predict the 3D joint locations of a hand given a depth map. Kishore et al. [25] have developed a sign language recognition system that is robust in different video backgrounds by extracting signers using boundary

and prior shape information. Then, the feature vector is constructed from the segmented signer and used as input to artificial neural network. An end-to-end sequence modelling using CNN-BLSTM architecture usually used for gesture recognition was proposed for large vocabulary sign language recognition with RWTH-PHOENIX-Weather 2014 [28].

At the same time, one of the most interesting breakthroughs in neural machine translation or even in the entire DL was introduced under the name of ‘sequence-to-sequence (seq2seq)’ [47]. The seq2seq model relies on a common framework called an encoder-decoder model with RNN cells such as LSTMs or GRUs. The seq2seq model proved its effectiveness in many sequence generation tasks by achieving almost the human-level performance [47]. Despite its effectiveness, the seq2seq model still has some drawbacks such as the input sequences of varying lengths being represented in fixed-size vectors and the vanishing gradient due to the long-term dependency between distant parts.

Camgoz et al. [7] formalized a sign language translation based on the pre-existing framework of Neural Machine Translation (NMT) with word and spatial embeddings for target sequences and sign videos, respectively. The extracted non-linear frame from a sign video is converted into the spatial representation through 2D CNN, and then it is tokenized. The sequence-to-sequence (seq2seq) based deep learning methods learns how to translate the spatio-temporal representation of signs into the spoken or written language. Recently, researchers developed a simple sign language recognition system based on bidirectional GRUs which just classifies a given sign language video into one of the classes that are predetermined [26]

3. KETI Sign Language Dataset

The KETI dataset is constructed to understand the Korean sign language of hearing-impaired people in various emergencies because they are challenging to cope with the situations and sometimes are in severe conditions. In that cases, even when they are aware of that situations, it is very hard to report the situations and receive help from government agencies due to the communication problem. Therefore, we have carefully examined the relatively general conversation of emergency cases and chosen useful 105 sentences and 419 words used in such situations.

The KETI sign language dataset consists of 11,578 full high definition (HD) videos, that are recorded at 30 frames per second and from two camera angles; front and side. The dataset is recorded by the designed corpus and contains sentences and words performed by eleven different hearing-impaired signers to eliminate the expression error by signers who are non-disabled people. Moreover, the meanings of the sentences and words are delivered to hearing-impaired signers through the expert’s sign languages in order to induce the correct expression. Each signer records a total of 1,048

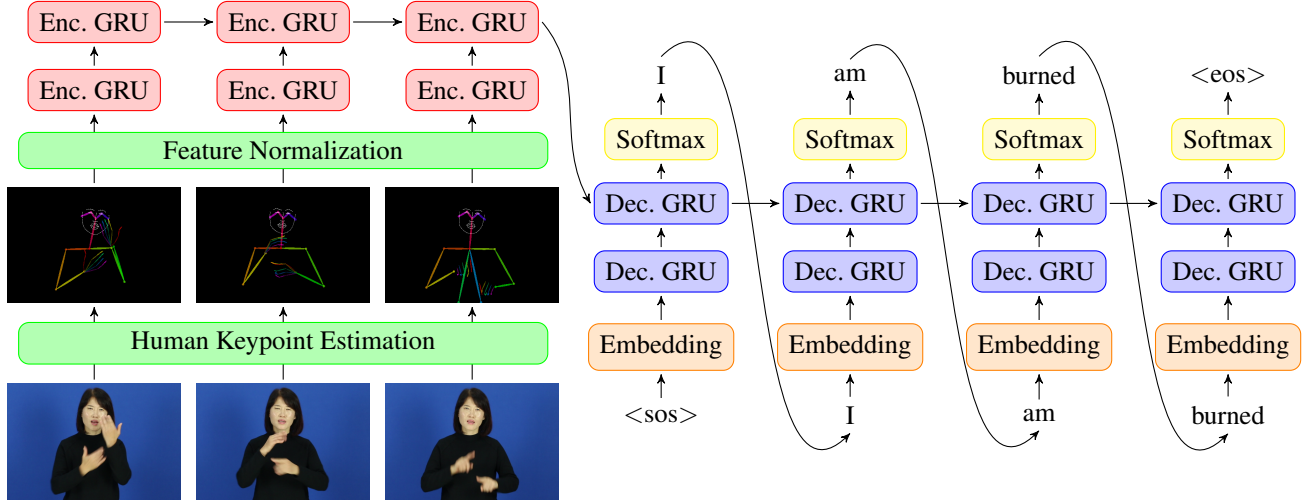


Figure 1. An overall architecture of our approach that translates a sign language video into a natural language sentence using sequence to sequence model based on GRU cells.

videos for the dataset. For the training and validation sets, we have chosen ten signers from eleven signers and chosen nine sign videos for each sign for the training set. The remaining sign videos are assigned to the validation set. The Test Set consists of a single signer whose sign video does not exist in the training set or the validation set. Several statistics of the dataset are given in Table 3 and an example frame from the dataset is presented in Figure 3.

In particular, we have annotated each of the 105 signs that correspond to the useful sentences in emergencies mentioned above with five different natural language sentences in Korean. Moreover, we have annotated all sign videos with the corresponding sequences of *glosses* [29], where a gloss is a unique word that corresponds to a unit sign and used to transcribe sign language. For instance, a sign implying ‘I am burned.’ can be annotated with the following sequence of glosses: (‘FIRE’, ‘SCAR’). Similarly, a sentence ‘A house is on fire.’ is annotated by (‘HOUSE’, ‘FIRE’). Apparently, glosses are more appropriate to annotate a sign because it is possible to be expressed in various natural sentences or words with the same meaning. For this reason, we have annotated all signs with the glosses with the help of Korean sign language experts.

For the communication with hearing-impaired people in the situations, the KETI dataset is used to develop an artificial intelligence-based sign language recognizer or translator. All videos are recorded in a blue screen studio to minimize any undesired influence and learn how to recognize or translate the signs with insufficient data.

Metric	Training	Dev	Test
# of sign videos	9,432	1,048	1,048
Duration [hours]	20.05	2.24	1.82
# of frames	2,165,682	241,432	153,350
# of signers	10	10	1
# of camera angles		2	

Table 1. Statistics of KETI sign language dataset



Figure 2. Example frame from our sign language dataset. The frame is from the video for the sentence ‘I am burned’.

4. Our Approach

We propose a sign recognition system based on the human keypoints that are estimated by pre-existing libraries such as OpenPose [5, 43, 52]. Here we develop our system based on OpenPose, an open source toolkit for real-time multi-person keypoint detection. OpenPose can estimate in total 130 keypoints where 18 keypoints are from body pose, 21

keypoints are from each hand, and 70 keypoints from a face. The primary reason of choosing OpenPose as a feature extractor for sign language recognition is that it is robust to many types of variations.

4.1. Human Keypoint Detection by OpenPose

First, our recognition system is robust in different cluttered backgrounds as it only detects the human body. Second, the system based on the human keypoint detection works well regardless of signer since the variance of extracted keypoints is negligible. Moreover, we apply the vector standardization technique to further reduce the variance which is dependent on signer. Third, our system can enjoy the benefits of the improvement on the keypoint detection system which has a great potential in the future because of its versatility. For instance, the human keypoint detection system can be used for recognizing different human behaviors and actions given that the relevant dataset is secured. Lastly, the use of high level features is necessary when the scale of the dataset is not large enough. In the case of sign language dataset, it is more difficult to collect than the other dataset as many professional signers should be utilized for recording sign language videos of high quality.

4.2. Feature Vector Normalization

There have been many successful attempts to employ various types of normalization methods in order to achieve the stability and speed-up of the training process [1, 21, 48]. One of the main difficulty in sign language translation with the small dataset is the large visual variance as the same sign can look very different depending on the signer. Even if we utilize the feature vector which is obtained by estimating the keypoints of human body, the absolute positions of the keypoints or the scale of the body parts in the frame can be very different. For this reason, we apply a special normalization method called the *object 2D normalization* that suits well in our purpose.

After extracting high-level human keypoints, we normalize the feature vector using the mean and standard deviation of the vector to reduce the variance of the data. Let us denote a 2D feature vector by $V = (v_1, v_2, \dots, v_n) \in \mathbb{N}^{n \times 2}$ that consists of n elements where each element $v_i \in \mathbb{N}^2$, $1 \leq i \leq n$ stands for a single keypoint of human part. Each element $v_i = (v_i^x, v_i^y)$ consists of two integers v_i^x and v_i^y that imply the x - and the y -coordinates of the keypoint v_i in the video frame, respectively. From the given feature vector V , we can extract the two feature vectors as follows:

$$\begin{aligned} V_x &= (v_1^x, v_2^x, \dots, v_n^x) \text{ and} \\ V_y &= (v_1^y, v_2^y, \dots, v_n^y). \end{aligned}$$

Simply speaking, we collect the x and y -coordinates of keypoints separately while keeping the order. Then, we normal-

ize the x -coordinate vector V_x as follows:

$$V_x^* = \frac{V_x - \bar{V}_x}{\sigma(V_x)},$$

where \bar{V}_x is the mean of V_x and $\sigma(V_x)$ is the standard deviation of V_x . Note that V_y^* is calculated analogously. Finally, it remains to concatenate the two normalized vectors to form the final feature vector $V^* = [V_x^*; V_y^*] \in \mathbb{N}^{2n}$ which will be used as the input vector of our neural network.

It should be noted that we assume that the keypoints of lower body parts are not necessary for sign language recognition. Therefore, we only use 124 keypoints from 137 keypoints detected by OpenPose since six keypoints of human pose correspond to lower body parts such as both feet, knees and pelvises as you can see in Figure 2. We randomly sample 10 to 50 keyframes from each sign video. Hence, the dimension of input feature vector is $248 \times |V|$, where $|V| \in \{10, 20, 30, 40, 50\}$.

4.3. Frame Skip Sampling for Data Augmentation

The main difficulty of training neural networks with small datasets is that the trained models do not generalize well with data from the validation and the Test Sets. As the size of dataset is even smaller than the usual cases in our problem, we utilize the *random frame skip sampling* that is commonly used to process video data such as video classification [22] for augmenting training data. The effectiveness of data augmentation has been proved in many tasks including image classification [40]. Here, we randomly extract multiple representative features of a video.

Given a sign video $S = (f_1, f_2, \dots, f_l)$ that contains l frames from f_1 to f_l , we randomly select a fixed number of frames, say n . Then, we first compute the average length of gaps between frames as follows:

$$z = \left\lfloor \frac{l}{n-1} \right\rfloor.$$

We first extract a sequence of frames with indices from the following sequence $Y = (y, y+Z, y+2z \dots, y+(n-1)z) \in \mathbb{N}^n$, where $y = \lfloor \frac{l-z(n-1)}{2} \rfloor$ and call it a *baseline sequence*. Then, we generate a random integer sequence $R = (r_1, r_2, \dots, r_n) \in [1, z]^n$ and compute the sum of the random sequence and the baseline sequence. Note that the value of the last index is clipped to the value in the range of $[1, l]$. We start from the baseline sequence instead of choosing any random sequence of length l to avoid generating random sequences of frames that are possibly not containing ‘key’ moments of signs.

4.4. Attention-based Encoder-Decoder Network

The encoder-decoder framework based on RNN architectures such as LSTMs or GRUs is gaining its popularity for

neural machine translation [2, 33, 47, 49] as it successfully replaces the statistical machine translation methods.

Given an input sentence $\mathbf{x} = (x_1, x_2, \dots, x_{T_x})$, an encoder RNN plays its role as follows:

$$h_t = \text{RNN}(x_t, h_{t-1})$$

where $h_t \in \mathbb{R}^n$ is a hidden state at time t . After processing the whole input sentence, the encoder generates a fixed-size context vector that represents the sequence as follows:

$$c = q(h_1, h_2, \dots, h_{T_x}),$$

For instance, the RNN is an LSTM cell and q simply returns the last hidden state h_{T_x} in one of the original sequence to sequence paper by Sutskever et al. [47].

Now suppose that $\mathbf{y} = (y_1, y_2, \dots, y_{T_y})$ is an output sentence that corresponds to the input sentence \mathbf{x} in training set. Then, the decoder RNN is trained to predict the next word conditioned on all the previously predicted words and the context vector from the encoder RNN. In other words, the decoder computes a probability of the translation \mathbf{y} by decomposing the joint probability into the ordered conditional probabilities as follows:

$$p(\mathbf{y}) = \prod_{i=1}^{T_y} p(y_i | \{y_1, y_2, \dots, y_{i-1}\}, c).$$

Now our RNN decoder computes each conditional probability as follows:

$$p(y_i | y_1, y_2, \dots, y_{i-1}, c) = \text{softmax}(g(s_i)),$$

where s_i is the hidden state of decoder RNN at time i and g is a linear transformation that outputs a vocabulary-sized vector. Note that the hidden state s_i is computed by

$$s_i = \text{RNN}(y_{i-1}, s_{i-1}, c),$$

where y_{i-1} is the previously predicted word, s_{i-1} is the last hidden state of decoder RNN, and c is the context vector computed from encoder RNN.

Bahdanau attention. Bahdanau et al. [2] conjectured that the fixed-length context vector c is a bottleneck in improving the performance of the translation model and proposed to compute the context vector by automatically searching for relevant parts from the hidden states of encoder. Indeed, this ‘attention’ mechanism has proven really useful in various tasks including but not limited to machine translation. They proposed a new model that defines each conditional probability at time i depending on a dynamically computed context vector c_i as follows:

$$p(y_i | y_1, y_2, \dots, y_{i-1}, \mathbf{x}) = \text{softmax}(g(s_i)),$$

where s_i is the hidden state of the decoder RNN at time i which is computed by

$$s_i = \text{RNN}(y_{i-1}, s_{i-1}, c_i).$$

The context vector c_i is computed as a weighted sum of the hidden states from encoder:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j,$$

where

$$\alpha_{ij} = \frac{\exp(\text{score}(s_{i-1}, h_j))}{\sum_{k=1}^{T_x} \exp(\text{score}(s_{i-1}, h_k))}.$$

Here the function ‘score’ is called an *alignment function* that computes how well the two hidden states from the encoder and the decoder, respectively, match. For example, $\text{score}(s_i, h_j)$, where s_i is the hidden state of the encoder at time i and h_j is the hidden state of the decoder at time j implies the probability of aligning the part of the input sentence around position i and the part of the output sentence around position j .

Luong attention. Later, Luong et al. [33] examined a novel attention mechanism which is very similar to the attention mechanism by Bahdanau et al. but different in some details. First, only the hidden states of the top RNN layers in both the encoder and decoder are used instead of using the concatenation of the forward and backward hidden states of the bi-directional encoder and the hidden states of the uni-directional non-stacking decoder. Second, the computation path is simplified by computing the attention matrix after computing the hidden state of the decoder at current time step. They also proposed the following three scoring functions to compute the degree of alignment between the hidden states as follows:

$$\text{score}(h_t, h_s) = \begin{cases} h_t^\top h_s, & \text{(Dot)} \\ h_t^\top W h_s, & \text{(General)} \\ V^\top \tanh(W[h_t; h_s]), & \text{(Concat.)} \end{cases}$$

where V and W are learned weights. Note that the third one based on the concatenation is originally proposed by Bahdanau et al. [2].

Multi-head attention (Transformer). While the previous encoder-decoder architectures are based on RNN cells, Vaswani et al. [49] proposed a completely new network architecture which is based solely on attention mechanisms without any recurrence and convolutions. The most important characteristic of the Transformer is the *multi-head attention* which is used in three different ways as follows:

1. Encoder-decoder attention: each position in the decoder can attend over all positions in the input sequence.

2. Encoder self-attention: each position in the encoder can attend over all positions in the previous layer of the encoder.
3. Decoder self-attention: each position in the decoder can attend over all positions in the decoder up to and that position.

Moreover, as the Transformer uses neither recurrence nor convolution, the model requires some information about the order of the sequence. To cope with this problem, the Transformer uses *positional encoding* which contains the information about the relative or absolute position of the words in the sequence using sine and cosine functions.

5. Experimental Results

We implemented our networks using PyTorch [39], which is an open source machine learning library for Python. The Adam optimizer [24] was used to train the network weights and biases for 50 epochs with an initial learning rate 0.001. During the training, we changed the learning rate every 20 epochs by the exponential decay scheduler with discount factor 0.5. We also used the dropout regularization with a probability of 0.8 and the gradient clipping with a threshold 5. Note that the dropout regularization is necessarily high as the size and the variation of the dataset is small compared to other datasets specialized for deep learning training. For the sequence-to-sequence models including the vanilla seq2seq model and two attention-based models, the dimension of hidden states is 256. For the Transformer model, we use the dimension for input and output (d_{model} in [49]) of 256. The other hyper-parameters used for the Transformer are the same as in the original model including the scheduled Adam optimizer in their own setting. Moreover, the batch size is 128, the augmentation factor is 100, the number of chosen frames is 50, and the object 2D normalization is used unless otherwise specified.

As our dataset is annotated in Korean which is an agglutinative language, the morphological analysis on the annotated sentences should be performed because the size of dictionary can be arbitrarily large if we split sentences into words simply by white-spaces in such languages. For this reason, we used the Kkma part-of-speech (POS) tagger in the KoNLPy package which is a Python package developed for natural language processing of the Korean language to tokenize the sentences into the POS level [37].

In order to evaluate the performance of our translation model, we basically calculate ‘accuracy’ which means the ratio of correctly translated words and sentences. Besides, we also utilized three types of metrics that are commonly used for measuring the performance of machine translation models such as BLEU [36], ROUGE-L [30], METEOR [3], and CIDEr [50] scores.

Sentence-level vs Gloss-level training. As in [7], we conduct an experiment to compare the translation performance depending on the type of annotations. Because each sign corresponds to a unique sequence of glosses while it corresponds to multiple natural language sentences, it is easily predictable that the gloss-level translation shows better performance. Indeed, we can confirm the anticipation from the summary of results provided in Table 10.

This also leads us to the future work for translating sequences of glosses into natural language sentences. We expect that the sign language translation can be a more feasible task by separating the task of annotating sign videos with natural language sentences by two sub-tasks where we annotate sign videos with glosses and annotate each sequence of glosses with natural language sentences.

Effect of feature normalization methods. In order to evaluate the effect of the feature normalization method on the keypoints estimated by OpenPose, we compare the following five cases: 1) no normalization, 2) feature normalization, 3) object normalization, 4) 2-dimensional (2D) normalization, and 5) object 2D normalization. In the first case, we do not perform any normalization step on the keypoint feature generated by concatenating the coordinate values of all keypoints. In the feature normalization, we create a keypoint feature as in 1) and normalize the feature with the mean and standard deviation of the whole feature. In the object normalization, we normalize the keypoint features obtained from two hands, body, and face, respectively, and concatenate them to generate a feature that represents the frame. We also consider the case of 2D normalization in which we normalize the x - and y -coordinates separately. Lastly, the object 2D normalization is the normalization method that we propose in the paper.

Table 4 summarizes the result of our experiments. The table does not contain the results of the case without any normalization as it turns out that the proposed object 2D normalization method is superior to the other normalization methods we considered. Especially, when we train our neural network with the keypoint feature vector which is obtained by simply concatenating the x and y coordinates of keypoints without any normalization, the validation loss never decreases. While any kind of normalization seems working positively, it is quite interesting to see that there is an additional boost in translation performance when the object-wise normalization and the 2D normalization are used together.

Effect of augmentation factor. We examine the effect of data augmentation by random frame skip sampling and summarize the experimental results in Table 5. We call the number of training samples randomly sampled from a single sign video the *augmentation factor*.

Attention type	Validation Set				Test Set			
	ROUGE-L	METEOR	BLEU	CIDEr	ROUGE-L	METEOR	BLEU	CIDEr
Vanilla seq2seq [47]	88.21	65.80	88.14	3.904	66.29	43.36	59.57	2.537
Bahdanau et al. [2]	87.75	62.53	88.35	3.898	65.80	43.32	61.93	2.537
Luong et al. [33]	90.90	68.90	91.06	4.022	64.93	43.73	59.58	2.485
Transformer [49]	88.29	56.59	87.75	3.796	73.93	44.03	69.48	2.928

Table 2. Performance comparison of sign language translation on different types of attention mechanisms.

Annotation	Validation Set				Test Set			
	Accuracy	ROUGE-L	METEOR	BLEU	Accuracy	ROUGE-L	METEOR	BLEU
Sentence-level	80.93	92.93	65.11	88.08	44.76	64.89	39.89	55.61
Gloss-level	94.59	96.95	72.60	95.03	60.57	68.34	41.32	57.94

Table 3. Comparison of sign language translation performance on different types of annotations.

Method	ROUGE-L	METEOR	BLEU	CIDEr
Feature Norm.	63.56	41.02	55.39	2.263
2D Norm.	64.22	41.62	56.79	2.380
Object Norm.	64.41	41.14	55.82	2.415
Object 2D Norm.	64.93	43.73	59.58	2.485

Table 4. Effect of different feature normalization methods on the translation performance. The results are obtained on the test set.

It should be noted that we do not include the result when we do not augment data by random frame sampling because the validation loss does not decrease at all due to overfitting. The result shows that the optimal augmentation factor is indeed 50. Considering the fact that the average number of frames in a sign video is larger than 200, the average length of gaps between frames is larger than 4. Then, there are 4^{50} possible random sequences on average and consequently the probability of having exactly same training sample is really low. However, the result implies that increasing the augmentation factor has a limit at some point.

Effect of attention mechanisms. Here we compare four types of encoder-decoder architectures that are specialized in various machine translation tasks. Table 2 demonstrates the clear contrast between the attention-based model by Luong et al. [33] and the Transformer [49]. While the model of Luong et al. shows better performance than the Transformer on the validation set that contains more similar data to the training set, the Transformer generalizes much better to the test set which consists of sign videos of an independent signer.

Effect of the number of sampled frames. It is useful to know the optimal number of frames if we plan to develop a real-time sign language translation system because we can reduce the computational cost of the inference engine

by efficiently skipping unnecessary frames. Table 6 shows how the number of sampled frames affects the translation performance. As the sequence-to-sequence model works for any variable-length input sequences, we do not necessarily fix the number of sampled frames. However, it is useful to know the optimal number of frames as the translation performance of the sequence-to-sequence models tends to decline due to the vanishing gradient problem [38].

Interestingly, our experimental result shows that the optimal number of frames for the best translation performance is 30 for the validation set and 50 for the test set.

Effect of batch size. Recently, it is increasingly accepted that training with small batch often generalizes better to the test set than training with large batch [18, 45]. However, our experimental results provided in Table 7 shows the opposite phenomenon. We suspect that this is due to the scale of the original dataset because large batch is known to be useful to prevent overfitting to some extent.

5.1. Ablation study

We also study the effect of the use of keypoint information from two hands, body, and face. The experimental results summarized in Table 8 imply that the keypoint information from both hands is the most important among all the keypoint information from hands, face, and body.

Method	ROUGE-L	METEOR	BLEU	CIDEr
Body	57.14	36.41	49.29	2.080
Hand	65.49	42.94	58.99	2.433
Body, Face	48.10	30.20	38.56	1.528
Hand, Face	62.85	40.48	55.03	2.350
Hand, Body	68.13	44.37	60.61	2.621
Hand, Body, Face	64.93	43.73	59.58	2.485

Table 8. Ablation study on the contributions of keypoints from body, face, and hands. The results are obtained on the test set.

Augmentation factor	Validation Set				Test Set			
	ROUGE-L	METEOR	BLEU	CIDEr	ROUGE-L	METEOR	BLEU	CIDEr
100	90.90	68.90	91.06	4.022	64.93	43.73	59.58	2.485
50	90.31	69.01	91.04	4.052	67.16	45.37	63.12	2.680
10	89.01	65.19	89.25	3.935	62.62	40.81	57.05	2.365

Table 5. Effects of data augmentation by random frame sampling on sign language translation performance.

Number of frames	Validation Set				Test Set			
	ROUGE-L	METEOR	BLEU	CIDEr	ROUGE-L	METEOR	BLEU	CIDEr
50	90.90	68.90	91.06	4.022	64.93	43.73	59.58	2.485
40	90.66	68.98	91.33	4.094	67.51	44.78	62.50	2.633
30	91.19	69.30	91.44	4.083	65.98	43.37	59.23	2.528
20	89.07	65.76	89.46	3.995	67.20	45.08	62.29	2.651
10	75.99	51.58	72.98	3.104	61.69	40.28	55.02	2.322

Table 6. Effects of the number of sampled frames on sign language translation performance.

Batch size	Validation Set				Test Set			
	ROUGE-L	METEOR	BLEU	CIDEr	ROUGE-L	METEOR	BLEU	CIDEr
128	90.90	68.90	91.06	4.022	64.93	43.73	59.58	2.485
64	90.16	68.19	91.08	3.956	63.45	41.62	57.34	2.398
32	86.73	64.79	87.49	3.748	64.34	42.34	57.72	2.394
16	86.84	62.54	86.56	3.733	63.68	41.38	57.07	2.355

Table 7. Effects of the batch size on sign language translation performance.

Interestingly, the experimental result tells us that the key-point information from face does not help to improve the performance in general. The performance even drops when we add face keypoints in all cases. We suspect that the reason is partly due to the imbalanced number of keypoints from different parts. Recall that the number of keypoints from face is 70 and this is much larger than the number of the other keypoints.

While the keypoints from both hands are definitely the most important features to understand signs, it is worth noting that the 12 keypoints from body are boosting up the performance. Actually, we lose the information about relative positions of parts from each other as we normalize the coordinates of each part separately. For instance, there is no way to infer the relative positions of two hands with the normalized feature vectors from both hands. However, it is possible to know the relative position from the keypoints of body as there also exist keypoints corresponding to the hands.

6. Conclusions

In this work, we have introduced a new sign language dataset which is manually annotated in Korean spoken language sentences and proposed a neural sign language translation model based on the sequence-to-sequence translation

models. It is well-known that the lack of large sign language dataset significantly hinders the full utilization of neural network based algorithms for the task of sign language translation that are already proven very useful in many tasks. Moreover, it is really challenging to collect a sufficient amount of sign language data as we need helps from sign language experts.

For this reason, we claim that it is inevitable to extract high-level features from sign language video with a sufficiently lower dimension. We are able to successfully train a novel sign language translation system based on the human keypoints that are estimated by a famous open source project called OpenPose developed by Hidalgo et al.

In the future, we aim at improving our sign language translation system by exploiting various data augmentation techniques using the spatial properties of videos. It is also important to expand the KETI sign language dataset to sufficiently larger scale by recording videos of more signers in different environments.

References

- [1] L. J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. [4](#)
- [2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. [5](#), [7](#)
- [3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, 2005. [6](#)
- [4] P. Buehler, A. Zisserman, and M. Everingham. Learning sign language by watching TV (using weakly aligned subtitles). In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2961–2968, 2009. [2](#)
- [5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. [3](#), [11](#), [12](#)
- [6] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014. [1](#)
- [7] N. Cihan Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. Neural sign language translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [1](#), [2](#), [6](#)
- [8] H. Cooper and R. Bowden. Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2568–2574, 2009. [2](#)
- [9] B. Dai and D. Lin. Contrastive learning for image captioning. In *Advances in Neural Information Processing Systems*, pages 898–907, 2017. [1](#)
- [10] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, 2015. [1](#)
- [11] C. Dong, M. C. Leu, and Z. Yin. American sign language alphabet recognition using Microsoft Kinect. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 44–52, June 2015. [1](#), [2](#)
- [12] J. Forster, C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. Piater, and H. Ney. Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, 2012. [1](#), [2](#)
- [13] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney. Extensions of the sign language recognition and translation corpus rwth-phoenix-weather. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 1911–1916, 2014. [2](#)
- [14] M. Gao, R. Yu, A. Li, V. I. Morariu, and L. S. Davis. Dynamic zoom-in network for fast object detection in large images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–26, 2017. [1](#)
- [15] S. Gattupalli, A. Ghaderi, and V. Athitsos. Evaluation of deep learning based pose estimation for sign language recognition. In *Proceedings of the 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, page 12. ACM, 2016. [1](#), [2](#)
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. [12](#)
- [17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. [1](#)
- [18] E. Hoffer, I. Hubara, and D. Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 1729–1739, 2017. [7](#)
- [19] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018. [1](#)
- [20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. [1](#)
- [21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015. [4](#)
- [22] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. [4](#)
- [23] T. Kim and S. Kim. Sign language translation system using latent feature values of sign language images. In *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 228–233, 2016. [2](#)
- [24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. [6](#)
- [25] P. V. V. Kishore, A. S. C. S. Sastry, and A. Kartheek. Visual-verbal machine interpreter for sign language recognition under versatile video backgrounds. In *2014 First International Conference on Networks Soft Computing (ICNSC2014)*, pages 135–140, 2014. [1](#), [2](#)
- [26] S. Ko, J. G. Son, and H. D. Jung. Sign language recognition with recurrent neural network using human keypoint detection. In *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems, RACS 2018, Honolulu, HI, USA, October 09-12, 2018*, pages 326–328, 2018. [2](#)
- [27] O. Koller, J. Forster, and H. Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, 2015. [1](#), [2](#)

- [28] O. Koller, S. Zargaran, and H. Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3416–3424, 2017. 1, 2
- [29] S. K. Liddell. *Grammar, Gesture, and Meaning in American Sign Language*. Cambridge University Press, 2003. 3
- [30] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of ACL workshop on Text Summarization Branches Out*, 2004. 6
- [31] C. Liu, J. Mao, F. Sha, and A. L. Yuille. Attention correctness in neural image captioning. In *AAAI*, pages 4176–4182, 2017. 1
- [32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1
- [33] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015. 5, 7
- [34] D. C. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5137–5146, 2018. 1
- [35] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *CoRR*, abs/1502.06807, 2015. 2
- [36] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 6
- [37] E. L. Park and S. Cho. Konlpy: Korean natural language processing in python. In *Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology*, 2014. 6
- [38] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, pages 1310–1318, 2013. 7
- [39] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 6
- [40] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621, 2017. 4
- [41] T. Pfister, J. Charles, and A. Zisserman. Large-scale learning of sign language by watching TV (using co-occurrences). In *British Machine Vision Conference*, 2013. 2
- [42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 1
- [43] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 3, 11, 12
- [44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 12
- [45] S. L. Smith, P. Kindermans, and Q. V. Le. Don’t decay the learning rate, increase the batch size. *CoRR*, abs/1711.00489, 2017. 7
- [46] T. Starner and A. Pentland. Real-time American sign language recognition from video using hidden markov models. In *Proceedings of International Symposium on Computer Vision - ISCV*, pages 265–270, 1995. 1, 2
- [47] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112, 2014. 1, 2, 5, 7
- [48] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. 4
- [49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 6000–6010, 2017. 5, 6, 7
- [50] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575, 2015. 6
- [51] U. Von Agris, M. Knorr, and K.-F. Kraiss. The significance of facial features for automatic sign language recognition. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–6. IEEE, 2008. 1
- [52] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016. 3, 11, 12
- [53] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. 1
- [54] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7151–7160, 2018. 1

7. Supplemental Material

Keypoint information used in sign language translation. We use the estimated coordinates of 124 keypoints of a signer to understand the sign language of the signer, where 12 keypoints are from human body, 21 keypoints are from each hand, and 70 keypoints are from face. See Figure 3 for example.¹ Note that the number of keypoints from human body is 25 but we select 12 keypoints that correspond to upper body parts. The chosen indices and the names of the parts are as follows:

- 0 (nose),
- 1 (neck),
- 2 (right shoulder),
- 3 (right elbow),
- 4 (right wrist),
- 5 (left shoulder),
- 6 (left elbow),
- 7 (left wrist),
- 15 (right eye),
- 16 (left eye),
- 17 (right ear), and
- 18 (left ear).

In future, we plan to plug in an additional attention module to learn which keypoint contributes more to understand the sign video

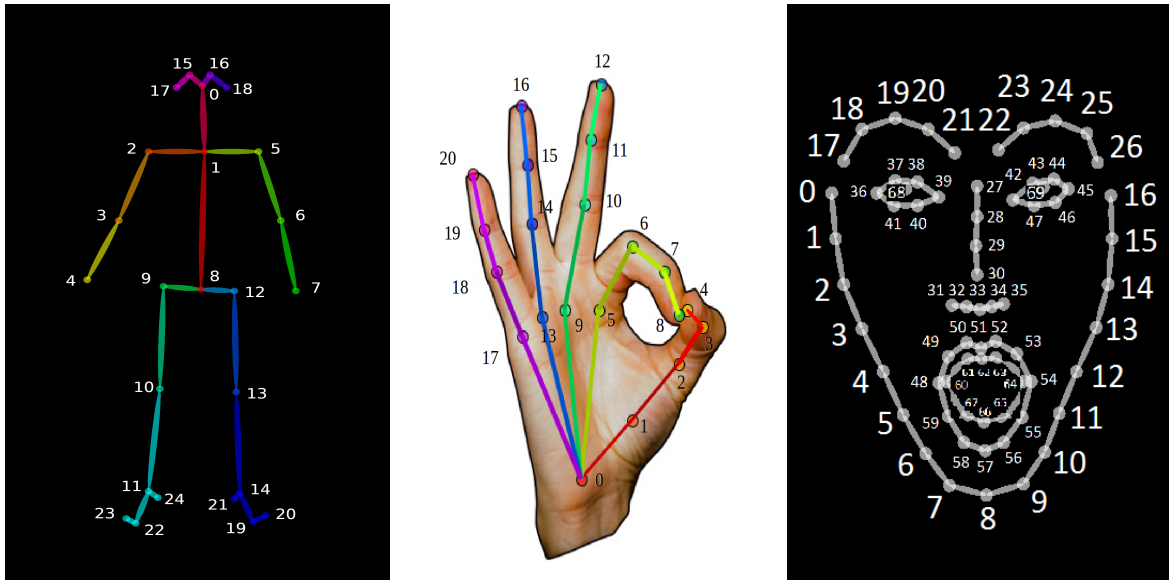


Figure 3. The human keypoints used for sign language recognition. Note that the figures are borrowed from the public web page of the OpenPose project [5, 43, 52].

¹<https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/doc/output.md>

Comparison with CNN-based approaches. In Table 9, we compare our approach to the classical methods based on CNN features extracted from well-known architectures such as ResNet [16] and VGGNet [44].

Since the size of sign video frames ($1,920 \times 1,080$) is different to the size of input of CNN models (224×224), we first crop the central area of frames in $1,080 \times 1,080$ and resize the frames to 224×224 .

The experimental results show that ResNet-101 exhibits the best translation performance on the validation set and the VGGNet-19 demonstrates the best performance on the test set. In general, the performance difference on the validation set is not large but it is apparent that the VGGNet models are much better in generalizing to the test set compared to the ResNet models.

Expectably, the translation models using the CNN extracted features show significantly worse translation performance than the models using the human keypoint features. It is still interesting to know whether the combination of any CNN-based features and human keypoint features works better than when we solely rely on the human keypoint features. As the size of sign language dataset grows, we expect that the CNN-based models improve their performances and generalize much better.

Feature type	Validation Set				Test Set			
	ROUGE-L	METEOR	BLEU	CIDEr	ROUGE-L	METEOR	BLEU	CIDEr
VGGNet-16 [44]	69.84	44.15	60.39	2.611	45.63	26.58	29.86	1.170
VGGNet-19 [44]	68.38	42.87	57.71	2.425	49.16	28.91	32.51	1.225
ResNet-50 [16]	65.30	40.62	56.70	2.305	36.35	21.22	17.92	0.642
ResNet-101 [16]	70.66	44.61	61.88	2.627	40.91	22.76	23.88	0.807
ResNet-152 [16]	64.54	40.09	54.32	2.241	37.52	20.91	18.08	0.592
OpenPose [5, 43, 52]	90.31	69.01	91.04	4.052	67.16	45.37	63.12	2.680

Table 9. Performance comparison with translation models based on CNN-based feature extraction techniques. Note that the augmentation factor in this experiment is all set to 50.

Attention maps. In Figure 4, we depict attention maps of the sentence-level translation model on several successful and unsuccessful cases. We can see that the attention weights are more well-distributed on the important frames of the video in the successful case when generating the natural language sentence compared to the failure case. However, the order of the attentions is quite irregular in Figure 4 as there is no direct mapping between sign video frames and tokens of the output sentence.

We also describe the attention maps of the gloss-level translation model in Figure 5. In the attention map of the successful case, we can see that the order of the attentions are more regular than the successful one on the sentence-level. This is because there is a more clear mapping between the continuous frames in the video and the sign gloss on the gloss-level translation.

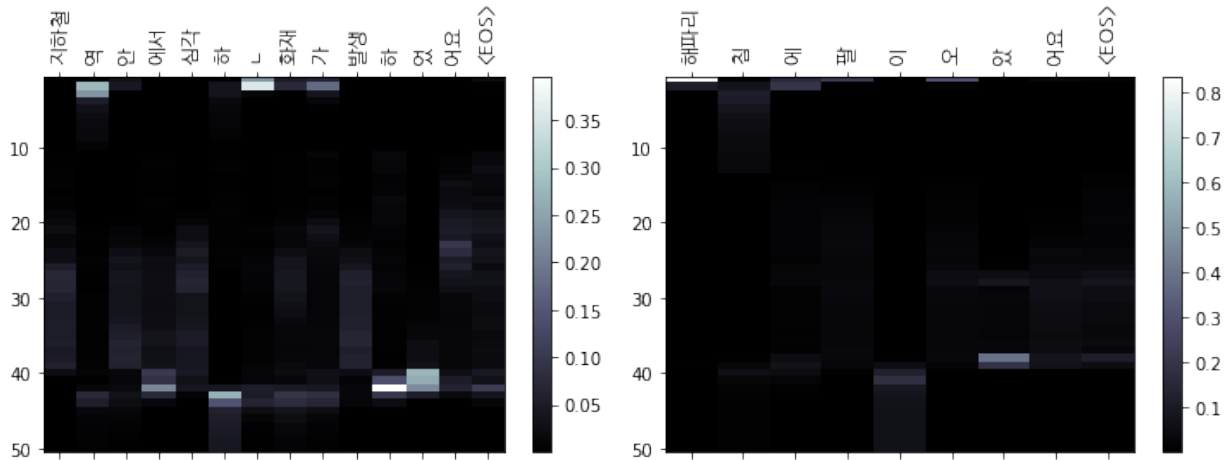


Figure 4. Attention maps of the sentence-level translation model. The first one is a map of a successful case and the second one is a failure case.

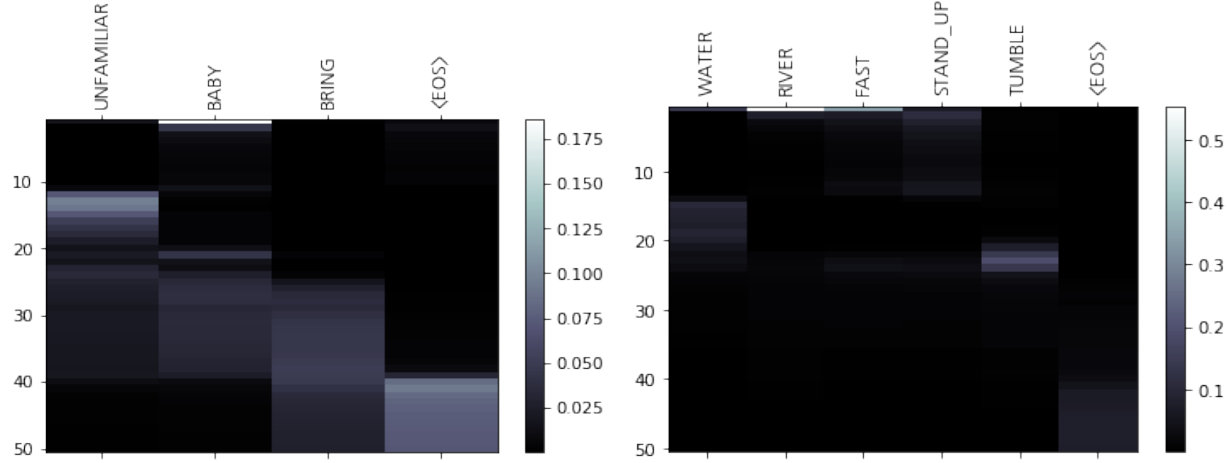


Figure 5. Attention maps of the gloss-level translation model. The first one is a map of a successful case and the second one is a failure case.

Sign language annotation. We annotate each sign video with five different natural language sentences in Korean. Table 10 contains ten examples from 105 examples in total.

Moreover, we annotate a sign video with a unique sign gloss as presented in Table 10.

ID	Sentence 1 (in Korean)	Sentence 2 (in Korean)	English sentence	Sign gloss
1	화상을 입었어요.	불에 데었어요.	I got burned.	FIRE SCAR
2	폭탄이 터졌어요.	폭발물이 터졌어요.	The bomb went off.	BOMB
3	친구가 숨을 쉬지 않아요.	친구가 호흡을 하지 않아요.	My friend is not breathing.	FRIEND BREATHE CANT
4	집이 흔들려요.	집에 지진이 난 것 같아요.	The house is shaking.	HOUSE SHAKE
5	집에 불이 났어요.	집에 화재가 발생했어요.	The house is on fire.	HOUSE FIRE
6	가스가 새고 있어요.	가스가 누출됐어요.	Gas is leaking.	GAS BROKEN FLOW
7	112에 신고해주세요.	112에 연락해주세요.	Please call 112.	112 REPORT PLEASE
8	도와주세요.	도와주실 수 있나요.	Help me.	HELP PLEASE
9	너무 아파요.	심하게 아파요.	It hurts so much.	SICK
10	무릎 인대를 다친 것 같아요.	무릎 인대가 끊어진 것 같아요.	I hurt my knee ligament.	KNEE LIGAMENT SCAR

Table 10. Ten examples of our sign language annotations. We annotate each sign with five natural language sentences in Korean and a unique sign gloss. We only provide two sentences in the table due to space limitations.