Video ConvNet-aided Sign Language Translation

Seung-Hoon Yi, Soh-Hyung Park, Eunji Lee
Seoul National Uiversity
MLVU 21-1,
jaguar6182@snu.ac.kr,sohhyung@dm.snu.ac.kr,leee4321@snu.ac.kr

1. Introduction

Sign Languages are the primary communication medium of the Deaf. Sign Languages are distinct language systems which convey information through hand shape, facial expression, upper body posture, etc. Generally, sign languages are developed independently of spoken languages and have different linguistic rules compared to those of spoken languages. Hence, converting sign language and natural language one to another is an important task to bridge communication gaps with the deaf people.There have been various approaches to interpret sign video sequences into natural language text. This, especially sign Language Recognition (SLR) or Sign Language Translation (SLT) is a challenging task in the field of computer vision since it involves interpreting several visual information such as body movements, facial expression into linguistic information.

Early works had focused on SLR approaches to interpret sign language into natural language. SLR methods mainly regards this task as a gesture recognition problem with the assumption that there exists one-to-one mapping between sign language and spoken language. However, as we can see in Figure 1, when interpreting sign language glosses into spoken language, linguistic and grammar characteristics such as sentence length and word order are significantly different. So, it is challenging to precisely align sign language into spoken language with existing SLR methods.

As such, there have been Sign Language Translation approaches aiming at full translation dealing this task with an aspect of machine translation because one-to-one mapping between sign language and spoken language does not exist. Conceptual video-based methods were introduced in early SLT works. Recently, end-to-end approaches were introduced using attention-based Neural Machine Translation (NMT) models [2].

The biggest obstacle of video based continuous SLT research has been lack of suitable datasets to train models. Recently, Camgoz et al. [2] released the first continuous SLT dataset containing video segments, gloss annotations and spoken language translation, RWTH-PHOENIX-Weather-2014T (PHOENIX14T), which comprises glosses of popular SLR dataset RWTH-PHOENIX-Weather-2014 (PHOENIX14). The authors also approached translation task as a NMT problem, namely Sign2Text approaching the end goal of SLT without going via gloss annotation, Sign2Gloss2Text extracting gloss sequence (Sign2Gloss) and then approaching the task as a text-text problem (Gloss2Text).

More recently, Camgoz et al. [5] proposed an end-to-end transformer based architecture jointly learning sign language recognition and translation which is the current state-of-art on PHOENIX14T, namely Sign2(Gloss+Text). The authors used gloss annotations to train transformer encoders to learn spatial representation for SLT named Sign Language Recognition Transformers (SLRT) and then autoregressive transformer decoder named Sign Language Translation Transformers (SLTT) exploits this representation learned by SLRT. They use Spatial Embedding approach to embed video frames during which each image passes through CNNs and employ Positional Encoding afterwards to add temporal ordering information.

However, Camgoz et al. [5] has limitations in two aspects. 1. When input data passes through CNN stage in frame to extract features, sequential information loss is unavoidable. 2.Since it involves Positional Encoding, there are some limitations in the aspect of both time and continuous SLT. In this project, we employ an encoder-decoder based architecture of a transformer network which performs well for the end goal of sign language translation on PHOENIX14T. The limitation of Camgoz et al. [5] is that it loses temporal information. So, to avoid this, we tried changing the input in the model to features extracted from 3D ConvNet. Also, since our goal is to convert video features directly into plain text, we utilize human keypoint information for better performance. In this process, we propose a novel approach for a machine to learn to focus on keypoints itself.

2. Related works

The researches in SLR have been developed for decades, from sign segmentation to end-to-end sign language transla-

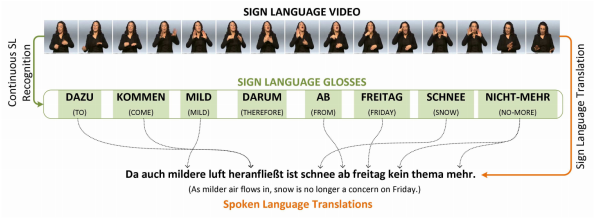

Figure 1. Difference between CSLR and SLT [2]

tion. They are divided by two main goals: Recognition and translation. The goal of sign language recognition is to detect and locate the signs so that the systems understand the information that sign language videos deliver. The goal of the latter is to translate sign language into natural language.

Initial works in SLR had focused on recognizing isolated sign gestures [10, 7, 11, 14]. Beyond recognizing only isolated signs, continuous SLR (CSLR) has emerged in order to apply to real-life signs. The studies have focused on recognizing sign glosses. However, such SLR using glosses have limitations since glosses, even manually provided by experts, can represent only a small number of frames in sign videos compared to actual actions involved in sign languages.With such failure of utilizing only glosses, other feature extraction methods have developed in SLR to extract visual information from sign videos. Recently, convolutional neural networks (CNNs) [12, 13] or pose estimation technique [8, 6] have been widely used as feature extractors.

The research in sign language translation (SLT) is challenging because sign languages have their own grammatical and semantic structures. Sign language and natural language cannot be converted to each other as one-to-one mapping. Glosses could also be used in SLT research; therefore, to reach the end goal of SLT, one may go through two consecutive steps, Sign2Gloss and Gloss2Text. The CSLR models are used for the first step, Sign2Gloss, and the output of the CSLR models are used for text-to-text translation, which is the second step Gloss2Text.

Camgoz et al. [2] achieved the translation process without glosses, using attention-based NMT. Their work was the first end-to-end learning which enabled deriving text from the sign videos. Subsequently, Camgoz et al. [5] used similar attention-based encoder-decoder architecture and supplemented it by adding Positional Encoding to Word and Spatial Embeddings. Despite the high performances of [2] and [5], they lose temporal information among the frames and still use gloss representation. In that context, Yin and Read [15] achieved state-of-the-art performance using Spatial-Temporal Multi-Cue (STMC) Network. We apply this encoder-decoder architecture substituting the spatial embeddings for 3D-CNN features.

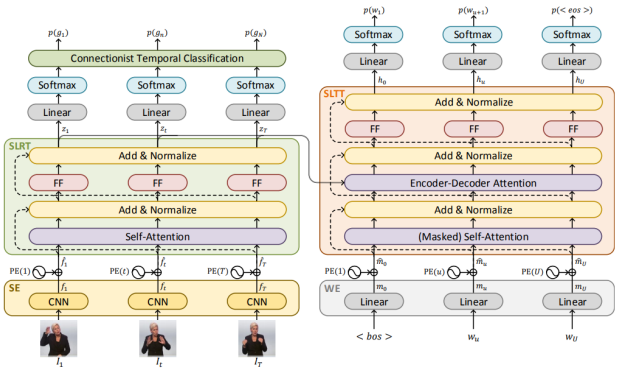

Figure 2. The SOTA model structure from Camgoz et al [5]

3. Method and Results

As Camgoz et al. [5] still demonstrates the State-of-the art record for BLEU-4 score in SLT, We conclude that transformer encoder-decoder models would be the best choice to perform reasonable scores compared to existing SOTA models. However, the suggested model has a defect that loses sequential information while passing the frames through the CNN stage (Figure 2). This forces us to implement positional encodings which cannot apply in CSLR.

Currently, the standard solution is to extract framewise features, interpret them into separate glosses, store the sequence until the last video frame, and translate them into plain text as a whole. Here, glosses are used as a 'bridge' representations between sign language and plain text, so as to boost the performance even when various noises are included in sign videos. Though this guarantees robustness in variation in actions, there is a following issue about a large number of parameters that makes the training and inferring process slow. Thus we suggest a translation model which receives video feature and as inputs, and directly returns plain text corresponding to the sign video. As features are passed through a pre-trained 3D CNN model, we expect to remove extra positional encoding layers in both encoder-decoder stages, reducing time needed in inference.

3.1. Word embeddings

We can either manually employ one-hot encoding for obtaining word vectors, or use pre-trained language models (LMs). As we target direct translation from feature vectors, we assume each word vector is Word2Vec-like. i.e. related to each other, while the magnitude of the inner product between two vectors represents similarity. Here we used the deutsch model of fasttext [1] in every translated text of the PHOENIX14T dataset to gain word-wise vector representation. Currently, embeddings are carried out with the following steps : 1)Embedding each word separately, 2)Pad every sentence into the max length of our dataset. The resulting

```
Example string :
 und nun die wettervorhersage für morgen donnerstag den zwölften august
[3, 234, 6, 1191123, 15, 2505, 127864, 12, 25157, 102100]
Padded string
[3, 234, 6, 1191123, 15, 2505, 127864, 12, 25157, 102100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Figure 3. A sample embedding and padding for our dataset. the list in the string below is the encoded word IDs from fasttext

output is a 300x56 matrix, where 300 and 56 are both the embedding dimension and maximum length of our dataset text. A sample of embedding words into IDs is depicted in Figure 3. These word vectors will be the ground truth of the decoder output in our training step under the loss defined as

$$\mathcal{L}(g, \hat{g}) = -\frac{1}{N}\sum_{i=1}^{N} log(1 + g_i \cdot \hat{g}_i) \tag{1}$$

Where g_i and $\hat{g}_i$ represents each word vector in a certain sentence. The total loss have to be summed-up through the whole batches. and Making the model to predict the correct word vector sequence of a given input will be the final goal of this project.

3.2. Video embeddings

We preprocess all video frames into 1024-dimensional features using I3D [4]. Then these features are passed through a transformer-based encoder. We also use open sources from past trials that employed I3D as a feature extractor from videos [9]. Currently, we have features from I3D with span size of 8, 12 and 16 with no padding, equal strides of 2.Our first objective is to utilize features from a single span size in training. So we planned three experiments using features from different span sizes to compare performance. Finally, we are going to incorporate concatenated features from all three span sizes or using a single span size to compare the performance when using different temporal information. Using human keypoints extracted from openpose [3] is also a one considerable method for training a machine where to focus. However, concatenating normalized keypoint in each feature in the image is impossible as we incorporate temporal pooling when preprocessing frames. Hence we devised a novel method for the machine to learn how to focus on keypoints. First, extract the pixel coordinates of 15 keypoints from each frame, and reverse normalize RGB values in the corresponding pixel of the given frame. When the frames are given as inputs for the 3D CNN, the output features will reflect the keypoint information from the given dataset.

3.3. Model architecture

The temporal layout of our models is depicted in Figure 4. In this project, first we are going to employ a basic encoder-decoder architecture to confirm whether the model

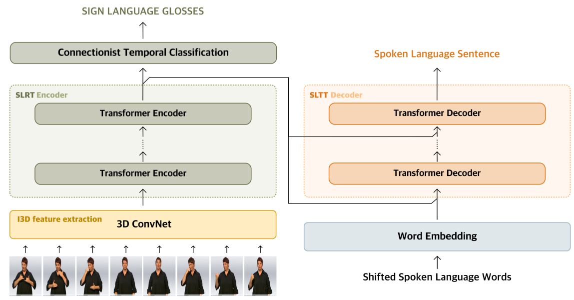

Figure 4. Currently we are implementing without gloss representation. The next goal of this project is to successfully implement gloss embedding layers and jointly training with the encoder-decoder transformer.

returns expected outputs. Then, stack both stages in various depths while assessing the performance with models listed in section 2 to figure out optimal design specifications for our dataset. Each encoder layer is designed with basic transformer architecture employing self-attention. At every end of one layer, outputs of encoded features are passed for encoder-decoder attention in the decoder stage. Decoder stages are expected to return a full translated sentence of a given sign video. It will also be stacked multiple times depending on the output accuracy when training through a fixed epoch.

4. Future Works

As we aim both improving accuracy with augmenting visual features for sign language videos, we planned to employ gloss representation and keypoint features in the training step. Hence in the perspective of architecture, we are going to add fully connected layers after the encoder to predict glosses with the whole video features from the encoder representation. Expectation for the final model will be similar to Figure 4. Methods to utilize human keypoints are introduced in section 3.

For further research, our task is focused on finding the best hyperparameters for training, and to mix or concatenate visual features obtained from 3D ConvNet layers to gain speed and accuracy in both training and inference steps. This is expected to boost accuracy in both the training and inferring process. Furthermore, since the convolution steps need huge amounts of computing power, we are aiming to find a more efficient way to extract spatio-temporal information from the given video. Current candidates are S3D or Vision Transformer (ViT) which can speed-up the entire process and to achieve competitive results with the presenting models.

References

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. 2

[2] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 3

[4] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 3

[5] Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10020–10030, 2020. 1, 2

[6] Mathieu De Coster, Mieke Van Herreweghe, and Joni Dambre. Sign language recognition with transformer networks. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6018–6024, Marseille, France, May 2020. European Language Resources Association. 2

[7] K. Grobel and M. Assan. Isolated sign language recognition using hidden markov models. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 1, pages 162–167 vol.1, 1997. 2

[8] Sang-Ki Ko, Jae Gi Son, and Hyedong 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 '18, page 326–328, New York, NY, USA, 2018. Association for Computing Machinery. 2

[9] Dongxu Li, Chenchen Xu, Xin Yu, Kaihao Zhang, Benjamin Swift, Hanna Suominen, and Hongdong Li. Tspnet: Hierarchical feature learning via temporal semantic pyramid for sign language translation. In *Advances in Neural Information Processing Systems*, volume 33, 2020. 3

[10] Rung-Huei Liang and Ming Ouhyoung. A sign language recognition system using hidden markov model and context sensitive search. page 59–66, New York, NY, USA, 1996. Association for Computing Machinery. 2

[11] Kouichi Murakami and Hitomi Taguchi. Gesture recognition using recurrent neural networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, page 237–242, New York, NY, USA, 1991. Association for Computing Machinery. 2

[12] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen. Sign language recognition using convolutional neural networks. In Lourdes Agapito, Michael M. Bronstein, and Carsten Rother, editors, *Computer Vision - ECCV 2014 Workshops*, pages 572–578, Cham, 2015. Springer International Publishing. 2

[13] Junfu Pu, Wengang Zhou, and Houqiang Li. Iterative alignment network for continuous sign language recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4160–4169, 2019. 2

[14] Fang Yin, Xiujuan Chai, and Xilin Chen. Iterative reference driven metric learning for signer independent isolated sign language recognition. volume 9911, pages 434–450, 10 2016. 2

[15] Kayo Yin and Jesse Read. Better sign language translation with STMC-transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics. 2