Sign Language Transformers Based on Human Keypoint Estimation

Stanford CS231N Project

Erdenebold Battulga
Department of Computer Science
Stanford University
ebo@2020stanford.edu

Abstract

The current state-of-the-art Sign Language Translation systems use gloss-level annotations. The transformer based architecture introduced in this paper does away with glosses and directly translates a video input to a spoken language sentence. It uses OpenPose to extract gesture information from images, which is essential for Sign Language Understanding. The model is trained and evaluated on the RWTH-PHOENIX-Weather-2014T (PHOENIX14T) dataset and achieves promising results.

1 Introduction

Sign Languages are the main mode of communication for the Deaf Community. To convey information, they utilize multiple complementary channels, such as manual and non-manual movements, facial expression and use of direction and space to express relationship between objects. Sign Language Translation (SLT) aims to convert spoken or written language sentences into a video of sign sequences or vice versa. The latter is inarguably a harder task and much of the work in this field is focused on recognizing sign glosses (Continuous Sign Language Translation (CSLR)) rather than the full translation to a non-visual language (SLT). Unlike text-to-text machine translation, in which words and sentences are separated by punctuation marks, SLT system needs to segment signs and sign sentences. Following segmentation, the model tries to understand the sign sentences and a common approach is to recognize individual glosses from a video input. However, this creates information bottleneck in the translation pipeline as glosses are simplified representations of sign languages and linguists are yet to come to a consensus on how sign languages should be annotated. Lastly, once the sign sentence is understood by the system, it has to generate written or spoken language sentences.

2 Dataset

The biggest obstacle to vision based SLT research is the availability of suitable datasets. Curating and annotating continuous sign language videos with spoken language translations is a laborious task and available datasets are often weakly annotated or too small to build models that would work on broad domain of discourse. To address these issues, Camgoz et al. released the first publicly available SLT dataset, PHOENIX14T, which is a translation focused extension of the popular RWTH-PHOENIXWeather-2014 (PHOENIX14) CSLR dataset [1]. PHOENIX14T contains parallel German sign language videos, gloss annotations and their translations, which makes it the only available dataset suitable for training and evaluating joint CSLR and SLT techniques [2]. The corpus includes unconstrained continuous sign language from 9 different signers with a vocabulary of 1066 different signs. Translations for these videos are provided in German spoken language with a vocabulary of 2887 different words. It's divided into train, dev and test sets which contain 7096, 519 and 642 examples respectively and in total has 947,757 210x260 px. image frames. As a proof of concept, only 1541 images were used for this paper.

3 Related Work

In 2018, Camgoz et al. approached LST as as a spatio-temporal neural machine translation problem, which they term Neural Sign Language Translation (NSLT) [1]. They proposed a system using AlexNet [3], a Convolutional Neural Network (CNN), to extract spatial features in combination with attention-based recurrent neural network architecture to realize the first end-to-end SLT models. Following this, Ko et al. proposed a similar approach but used OpenPose to extract body keypoint coordinates from images as input for their translation networks, and evaluated their method on a Korean Sign Language dataset [4]. Earlier this year, Camgoz et al. used Inception Network [5], (which was pretrained for sign language recognition in a CNN+LSTM+HMM setup [6]), as sign video feature extractor with Transformers to jointly train for CSLR and SLT. They injected gloss-level intermediate supervision through Connectionist Temporal Classification (CTC) loss at the end of their Encoder Network and their code is set to be released for CVPR 2020 [2].

4 Methods

Our proposed SLT system uses OpenPose to extract spatial features from a video input and uses Transformer Architecture to translate sign sequences into a written language counterpart without any intermediate supervision (S2T (Sign to Text)). Theoretically, such translation without gloss-level recognition can outperform systems that first translate image sequences to glosses, as they cause information bottleneck. However, currently no vision-based SLT system is able to outperform text-to-text machine translation from gloss annotations to a written language sequence(G2T(Gloss to Text)). Thus, this empirical upper bound is used as a baseline.

4.1 Baseline: G2T Transformer

We start by adding start and end tokens to the source and target tokens, namely, gloss annotations and spoken German sentence. Then, as embedding, we use linear layer to project the one-hot-encoded sequences to a continuous space of dim-512:

$$b_t = GlossEmbedding(g_t), m_t = WordEmbedding(w_t)$$

where i denotes position of a token in a sentence. Temporal information is added through positional encoding, which produces unique vector in the form phase shifted sine wave for each t:

$$\hat{b}_t = GlossEmbedding(g_t), \hat{m}_t = WordEmbedding(w_t)$$

Source and target sequences are then fed into the cannon Transformer model from Vaswani et al.'s paper [7], which has 6 layers, 8 attention heads and dim-2048 position-wise feed forward network. The Encoder of Transformer learns meaningful representations from glosses and can be formulated as:

$$c_t = Encoder(\hat{b}_t|\hat{b}_{1:T})$$

where T is length of the sequence. The encoded sequence is then fed into the Decoder along with embedded target sequence .The output from the decoder is then fed into a linear layer to find score (probability proxy) over the target vocabulary:

$$h_{u+1} = Decoder(\hat{m}_u|\hat{m}_{1:u-1}, c_{0:T}), w_u = \underset{w_i, i \in D}{\arg\max}\, p(w_i|h_u)$$

where D is the size of the target vocabulary. Decoder is autoregressive with regards to the target sequence as each output token is produced from its predecessors. Also, it uses teacher-forcing during training as true output is passed to the next time step regardless of what the model predicted at the current time step. Lastly, cross-entropy loss for a sentence is computed as:

$$\mathcal{L} = 1 - \Pi_{u=1}^{U} \Sigma_{d=1}^{D} p(\hat{w}_u^d) p(w_u^d|h_u)$$

where $p(\hat{w}_u^d)$ represents the ground truth probability of word w_d at decoding step u.

4.2 S2T Transformer

S2T Transformer architecture is exactly the same as G2T's but the source sequence is a sign video. We use OpenPose to extract human keypoint estimation from each image frame, which outputs x, y

locations of keypoints along with confidence c. OpenPose produces 25, 21 and 70 features for body, each hand and face respectively and in total we get dim-411 (=25x3+21x2x3+70x3) output for each image:

$$z_t = OpenPose(s_t)$$

We don't add start and end tokens to the feature sequence and the rest of the model follows exactly the same pipeline as G2T Transformer.

5 Results

Table 1
Baseline G2T vs. S2T Results

	BLEU-4 Score	
	Dev	Test
G2T	5.72	4.59
S2T	2.35	2.27

The model's performance didn't generalize well to the dev and test set as only a tiny fraction of the training data was used. This was partly because OpenPose was run on Colab gpu as GCE was unavailable and feature extraction combined with other preprocessing took 1sec per image frame. This means that I would need 11 days to preprocess all images and extract features. However, the model's performance wasn't too bad on the training set, which implies that the S2T transformer can work well if there is enough data.

6 Future work

Currently, I am looking into how I can accelerate the preprocessing and feature extraction steps. I'm also planning to use CNNs as feature extractor. I will further experiment with different hyperparameters and inject intermediate supervision to the model by using the gloss-level annotations of the datset.

References

[1] Necati Cihan Camgoz. Neural Sign Language Translation. *ResearchGate*, 2018.

[2] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. *arXiv:2003.13830 [cs]*, March 2020. arXiv: 2003.13830.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017.

[4] Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. Neural Sign Language Translation Based on Human Keypoint Estimation. *Applied Sciences*, 9(13):2683, January 2019.

[5] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv:1602.07261 [cs]*, August 2016. arXiv: 1602.07261.

[6] Oscar Koller, Necati Camgoz, Hermann Ney, and Richard Bowden. Weakly Supervised Learning with Multi-Stream CNN-LSTM-HMMs to Discover Sequential Parallelism in Sign Language Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, April 2019.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, December 2017. arXiv: 1706.03762.