Examination of Translation Performance with Various Input Types to a Continuous Sign Language Translation Model

Brian Landy
Rochester Institute of Technology
Rochester, NY, USA
bxl1703@rit.edu

Abstract

Sign language is a method of communication used by those who are deaf or hard of hearing and like all languages, there is contextual information in an overall sign language expression that isn't present in just a small subset of that expression. Automatic translation would have to take this into account to convey meaning properly. This paper contains a set of experiments where the goal is to work with an existing Continuous Sign Language Translation model and apply different pre-extracted input features from AlexNet, OpenPose, ResNet as well as ResNet3D and evaluate translation performance using BLEU scores. This project is based on a CVPR paper from 2018. The given name for the translation model is Neural Sign Language Translation and the dataset used is a continuous German sign language dataset crafted for these experiments from an earlier set. The work detailed in this paper stems from an ongoing set of experiments by the Machine Intelligence Laboratory.

1. Introduction

1.1. Why Translate Sign Language

Sign language translation opens many new doors and can aid in learning tools, as well as improve communication efficiency for signers and non signers. Bragg et al. [2] discuss how translation techniques that don't require an interpreter will improve interactions in, "any point of sale, restaurant service, and daily spontaneous interactions." See the appendix for a small exploration of earlier sign language translation techniques.

2. Background

2.1. NSLT and Similar Works

Many translation attempts focused on word-level recognition. However, in 2018, Neural Sign Language Translation (NSLT) was proposed as a solution to solve the problem of Continuous Sign Language Translation. NSLT starts with the goal of translating sequences of sign language video and aims to translate that into text output. Refer to the appendix to read about important foundational papers that are used in NSLT. Many methods of translation are tried by Camgoz et al. [3] whether it is end to end sign to text or sign to gloss/gloss to text. The most relevant sections of the paper by Camgoz et al.[3] to this paper, however, are those that discuss Sign to Text (S2T) translation. This is a full end to end method that considers CNN-extracted features of each frame in a video of sign language and uses sequence to sequence learning to generate the text-based representation of the continuous sign language expression.

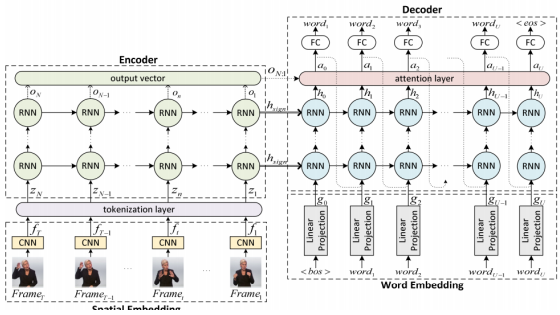

Figure 1. This is the general architecture for continuous Sign Language Translation proposed by Camgoz et al. [3]

2.2. NSLT Architecture

The NSLT Model is shown in Figure 1. It includes many components found in Neural Machine Translation and Sequence to Sequence models.This model utilizes image feature extraction with CNNs, tokenization, input and output embedding layers, attention modeling and deep RNNs. Many aspects of this model can be optionally swapped out. For example, Luong or Bahdanau attention can be employed. The type of RNN cell can be either the regular an

LSTM or GRU unit and the encoder can be made to be unidirectional or bidirectional. The number of layers and units in the stacked RNNs are variable. Tokenization is the process of breaking up a sequence into smaller parts and in this case, the output sequences are tokenized into words, while the input sequences are split up frame by frame. In order to pull the necessary information needed in a frame and resize appropriately, CNNs ared. Embedding in this model is done via word embeddings for the output sequences, but since the input is video frames, it is embedded spatially with the CNNs. In the same way a word embedding is a denser representation of a word, a CNN output is a denser representation of an image.

With so much variability in the model, there is a lot that can be tested and analyzed. BLEU evaluation is used to calculate performance of the model. See the appendix for more information on scoring. The sign to text model reported from Camgoz et al. [3] with the best BLEU performance was one with: GRU RNN Cells, Luong Attention, AlexNet for feature extraction, and 2-layer, bidirectional RNNs.

2.3. Experiments Performed

For our first experiments with varying input features for NSLT, we use LSTM cells, bidirectional encoding, and Luong attention. The RNN is made up of 1000 hidden units and is 4 layers deep. Input tokenization occurs at the frame level as it does in [3]. The features experimented with are AlexNet, for a reproduced baseline from [3], ResNet, OpenPose, OpenPose with smoothing applied to body movements as well as ResNet 3D for extracted features over a video sequence. ResNet 3D is much like ResNet but with 3D convolutions to take in multiple frames at one time. OpenPose is human keypoint estimation which reduces the depth of the extracted information. The smoothed OpenPose features use a Savitzky-Golay filter seen in [6]. This is a type of low pass filtering to prevent the keypoints from jumping around too much frame to frame. There are similar papers out that use human keypoint estimation as an input feature for sign language translation. One such paper is Ko et al. [4] which uses OpenPose features as input to the encoder network of a NSLT model and achieves great BLEU scores with this approach and a Korean Sign Language dataset. BLEU results are to be reported using all of these input feature types.

2.4. Results

2.4.1 Datasets

At the time of developing this, there were not many appropriate continuous sign language datasets to use. So, Camgoz et al. [3] started with the PHOENIX14 dataset and made some improvements and additions that led to the creation of RWTH-PHOENIX-Weather 2014T. This data set is much more suited for continuous sign language research. It is based on the PHOENIX Weather dataset from a german broadcasting station. One of the largest issues with continuous translation is the lack of data to train with. Bragg et al. [2], in their panel on Sign Language Recognition, mention that a goal for the future of this area of research should be to create a large composition of public SLT data to work with.

2.5. Methods

The BLEU scores were recorded after training with several different input feature types. The best performing model so far is one that uses ResNet for feature extraction and input to the RNNs. It performed better than AlexNet which is used in the original paper [3]. For this set up, videos over 300 frames are cut off, and videos under 300 frames are padded to fit 300. Running the training for one of these models takes anywhere from a couple of days to around a week for 150,000 training steps. The model is coded in Tensorflow. The table of results for BLEU scores is shown in table 1.

Table 1. BLEU Scroes on Test Data for NSLT

	BLEU 1	BLEU 2	BLEU 3	BLEU 4
OpenPose	18.50	11.92	8.73	6.95
OpenPose Smooth	19.65	12.36	8.97	7.00
ResNet	21.71	13.89	10.07	7.87
ResNet3D	13.010	8.01	5.95	4.80
AlexNet	20.72	12.82	8.85	6.7

Validation set results can be found in the appendix. Shown in table 1, ResNet for feature extraction provided the best BLEU results, and smoothing had a large impact on translation performance when using human keypoint extraction with OpenPose, but it still did not perform as well as AlexNet.

3. Conclusions

Sign language is a complex representation of information and converting meaning from sign language video to text representation is not a trivial challenge. And yet, Camgoz et al. [3] set out to try and make it happen at a continuous level. The training experiments done here are only a small subset of training experiments that could potentially be tried. There is still much room for improvement and many things to do with these translation models including things like padding adjustments. The difficulty presented here is the duration of training time and the lack of large continuous sign language datasets. Here, the NSLT model was used and the input was swapped with those from other CNNs and OpenPose. Compared to a baseline with AlexNet, ResNet performed slightly better in terms of BLEU scores, and keypoint estimation with OpenPose did worse, but improved with smoothing.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] D. Bragg, T. Verhoef, C. Vogler, M. Ringel Morris, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, and et al. Sign language recognition, generation, and translation. *The 21st International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '19*, 2019.

[3] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. Neural sign language translation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, 18 – 22 June 2018. IEEE.

[4] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho. Neural sign language translation based on human keypoint estimation. *Applied Sciences*, 9(13):2683, Jul 2019.

[5] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[6] S. Meschke. Reducing jitter using savgol filter, Sep 2018.

[7] 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*, pages 311–318. Association for Computational Linguistics, 2002.

[8] B. S. Parton. Sign Language Recognition and Translation: A Multidisciplined Approach From the Field of Artificial Intelligence. *The Journal of Deaf Studies and Deaf Education*, 11(1):94–101, 09 2005.

[9] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

4. Appendix

4.1. Earlier Methods of Translation

Methods of sign language translation are much broader than deep learning and computer vision, of course. A lot of translation was much shallower than continuous translation and only focused on the word level or character level signs. Some approaches to translation also required gloves or expensive depth camera rigs which can be cumbersome. As pointed out by Parton et al. [8], techniques like this are similar to "a mouse or keyboard" and the real solution for convenient translation should be found in software. In the past, much work has been put towards translating sign language. Many technologies in this area have been laid out by Parton et al. [8] in their multidisciplinary exploration of recognition and translation techniques. Some earlier methods for computer vision-based translation were digitizing of sign language gestures in 1992 as well as translation at the finger spelling level in 2005. Computer vision based continuous translation is now closer than ever with the introduction of NSLT.

4.2. Background Papers on building blocks of NSLT

Similar sequence to sequence models have been employed to convert video content to text content. This is similar to the nature of image captioning. In 2015, Venugopalan et al. [9] proposed a sequence to sequence method to provide text captions for videos by using a mix of Convolutional Neural Networks and LSTMs. This NSLT framework has a similar architecture to [9] but there are more layers to this like the attention models which can be found in neural machine translation models. In the domain of Neural machine translation, attention is something that is needed to boost translation performance over long sequences of information. It works by informing the model of what information is relevant and what can be lost. Luong et al. [5] introduced an attention model to help prevent the vanishing gradient problem spawned by long input sequences to neural translation models which was an improvement on the attention model created by Bahdanau et al. [1] in the same year. Both of these attention models can be used in NSLT.

4.3. BLEU Evaluation

Evaluation of performance is always important with a new model and in the domain of language translation, the BLEU score has been a great way to express translation performance. Papineni et al [7] introduce the BLEU metric. This scoring system is used to assess translation performance. It relies on calculating precision in a sub-sequence of words compared to the ground truth. After this is done, a penalty is applied called "brevity" to even out the precision values as long and short sentences will achieve different results for the same quality of translation. The reasoning is that when a generated sentence is longer than the ground truth, there will be words that are in the generated sentence that are not in the ground truth and thus the precision score will be lower. Papimeni et al. [7] take the geometric mean and multiply by the brevity metric to get the final BLEU score used in evaluations. This is the primary metric in these experiments.

4.4. Validation and Test Set Results for NSLT with Different Input Types

Table 2. BLEU Scroes on Test and Dev Data for NSLT

TEST	BLEU 1	BLEU 2	BLEU 3	BLEU 4
OpenPose	18.50	11.92	8.73	6.95
OpenPose Smooth	19.65	12.36	8.97	7.00
ResNet	21.71	13.89	10.07	7.87
ResNet3D	13.010	8.01	5.95	4.80
AlexNet	21.40	13.29	9.46	7.33
DEVELOPMENT	BLEU 1	BLEU 2	BLEU 3	BLEU 4
OpenPose	17.64	11.42	8.52	6.78
OpenPose Smooth	20.65	13.39	9.80	7.68
ResNet	23.01	14.48	10.20	7.79
ResNet3D	12.51	7.26	5.22	4.14
AlexNet	23.40	14.89	10.76	8.327

4.5. Characteristics of German Sign Dataset

Table 3. RWTH-PHOENIX-Weather 2014 T Information from [3]

	TRAIN	DEV	TEST
Segments	7,096	519	642
Frames	827,354	55,775	64,627
vocab. size	2,887	951	1,001
Total Words	99,081	6,820	7,816
Total OOVs (out of vocab)	-	57	60
Singletons (only once)	1,077	-	-