Argentinian Sign Language With Human Keypoint Estimation

Duygu Bayram

Matriculation Number: 7023403, duba00001@stud.uni-saarland.de

Shibingfeng Zhang

Matriculation Number: 7022609, shzh00003@stud.uni-saarland.de

June 11, 2023

1 Introduction

Use of machines to translate sign language holds great social impact potential for hard of hearing communities as the reliance on human translators comes with some disadvantages around privacy, time, and lack of resources, due to the deficiency in the people that are equipped to translate between sign and spoken languages. Machine translation could potentially render this need fast, cheap, secure, and accessible.

However, sign language translation comes with many engineering difficulties by the nature of it consisting of embedded complex visual and linguistic information, as well as having access to limited and often low-quality data. Overcoming this challenge lies in splitting the task into subgroups of visual computing and machine translation.

We find it helpful to introduce some common concepts relevant to our implementation. The feature extraction step is often handled through the use of Convolutional Neural Networks (CNNs) [CCHKB17] [CHK+18]. For the translation step, various types of Recurrent Neural Networks (RNNs) and Transformers are used [CLZ17] [KKJC19] to handle sequential learning. As a novelty, the Ko et al. (2019) [KKJC19] paper that inspired this project employs a human keypoint estimation method to extract features from sign language video data. In this project, we adopt a human keypoint approach for feature extraction, and use a Long Short-Term Memory (LSTM) RNN architecture inspired by Cui et al. (2017) [CLZ17]. The selected task is a word-level Sign Language Translation task.

2 Background

The present implementation is motivated by the Ko et al. (2019) [KKJC19]. The authors build a sign language translation model using human keypoint estimation, motivated by the idea that it would reduce variation in the visual input, and therefore generalize better. They compare their method with various CNN models, along with comparing several normalization techniques and different translation models. The dataset used in this study is KETI, a Korean sign language dataset for emergency situations consisting of 14 signers and 14,672 videos. Unfortunately, this dataset is not opensourced. In our implementation, we use the LSA64 dataset, as detailed in Section 3.1.1.

In Ko et al. (2019) [KKJC19], researchers leverage the OpenPose library to extract 137 keypoints from body, hands, and face (25, 42, and 70 respectively) which gives an x and y coordinate information for each keypoint. For their input they opt to only use the keypoints from the upper body as those are the relevant keypoints for a sign language translation task. In contrast, we used Mediapipe to extract our keypoints, because it allowed us to also have depth information along with the x-y coordinates, and we excluded face keypoints as the paper presents that the face keypoints have a negative effect on

model performance.

Furthermore, they propose a feature normalization technique in which they separate the integer values of the keypoints in terms of x-y coordinates and objects (body, hand, or face), normalize these separately, and then concatenate again as a vector to serve as the input. Different normalization techniques are experimented, which includes separate only by coordinates (2D Normalization), only by objects (Object Normalization), and by both coordinates and objects (2D Object Normalization). Inspired by this, we have performed feature-wise Normalization, whereby we normalized the feature with the mean and standard deviation of the whole feature.

For data augmentation, they use the frame skip sampling method in which they randomly select a fixed number of frames from the available frames in a video, find the average gap length between the selected frames, form a baseline sequence, generate a random sequence, and sum this sequence with the baseline. We did not use this method of augmentation. Videos in our dataset are relatively shorter and have less frames. Therefore we adopted a simple but rather efficient data augmentation method by sampling frames. Details of data augmentation approaches are explained in Section 3.2.2

For the translation model, they compare a vanilla sequence to sequence model with a transformer and two RNN models with Bahdanau et al. [BCB14] and Luong et al. [LPM15] attention types. Here they find that Luong performs better on their validation set, and the transformer performs better on their test set. Inspired by this, and because our input length was different, we decided to use an RNN architecture consisting of 2 bidirectional LSTM layers, a linear layer, and a 64-way softmax.

3 Project Overview

This section introduces the details of our project. Section 3.1 presents the sign language corpus and Python libraries employed in the project. Section 3.2 presents the approaches adopted for feature extraction from the LSA64 dataset, data normalization, data augmentation, and the construction of the sign language translation model.

3.1 Resources Used

In this section, we present the sign language dataset and Python libraries adopted for this project.

3.1.1 LSA64: A Dataset for Argentinian Sign Language

The LSA64 dataset [RQE+16] consists of 3200 videos obtained from 10 non-expert signers, recorded in RGB cameras. Each signer produces 320 videos by repeating each of the 64 signs 5 times. 23 one-handed signs were recorded outdoors in natural lighting, and the remaining 41 signs (19 one-handed and 22 two-handed) were recorded indoors with artificial lighting. Subject 10 was changed between the two recording sets. The selected signs for the dataset are commonly used signs in LSA including nouns and verbs. The dataset was developed specifically for sign language recognition, as opposed to handshape recognition or sentence recognition as categorized by the authors. Figure 1 shows an example from the dataset as taken from Ronchetti et al. (2016) [RQE+16].

Ronchetti et al. (2016) [RQE+16] presents several other prominent sentence-level sign language recognition datasets and explain that these have difficulties associated with feature extraction due to their reliance on skin color tracking. Based on the observed limitation, the LSA64 dataset is developed with the use of single-colored gloves, with different colors on each hand. This consideration of feature extraction was our reason for choosing to use this dataset in our implementation.

3.1.2 Libraries

Mediapipe [1] is a Google-developed open-source media processing tool available to be used in multiple languages. It's shown to be useful for tasks relevant to our project such as face, hand, and pose detection, as well as other tasks such as object detection, box and motion tracking among several

[1] https://mediapipe.dev/

Figure 1: Example images of the LSA64 dataset.

others uses. We chose to use this library over OpenPose because it allowed us to get additional depth information of each keypoint.

Besides Mediapipe, we also used some other common machine learning libraries such as Pandas, Sklearn [2] and PyTorch.

3.2 Implementation

Inspired by Ko et al (2019) [KKJC19], we would like to implement a sign language translation project based on human keypoint estimation. The implementation is organized in three steps:

1. Keypoint Information Extraction
2. Data Normalization and Augmentation
3. Model Training and Testing

3.2.1 Keypoint Information Extraction

As mentioned earlier in Section 3.1.1, the LSA64 dataset is composed of 3200 videos. In order to obtain the keypoint information from the dataset, videos are cut into frames on a basis of a 0.1 second time interval. This results in 69,900 frames.

Mediapipe package is leveraged to extract keypoint information from frames. For the body keypoint, we keep only the keypoint information of upper body without face. For the hand keypoint, we keep the features of the hand that is used for signing. Body keypoint is extracted successfully from all frames. However, Mediapipe was unable to extract hand keypoint information from about one third of all frames. In case that a frame is without hand keypoint information, the averaged hand keypoint information of all frames is saved as the hand keypoint information of this frame.

3.2.2 Data Normalization and Augmentation

The hand keypoint information consists in 64 features, while the body keypoint information consists in only 24 features. We reduced the dimension of hand keypoint information from 64 to 24 using the PCA function provided by Sklearn. The features are then normalized feature wisely using z-score normalization.

Each video in the dataset is represented as a concatenation of frames (i.e., [features of frame 1, features of frame 2, features of frame 3, ...]). Each video is augmented to generate three instances: the instance that contains only frames at odd positions (i.e., [features of frame 1, features of frame 3, features of

[2] https://scikit-learn.org/

	Train set		Dev set		Test set	
	loss	**accuracy**	**loss**	**accuracy**	**loss**	**accuracy**
all aug	3.1	0.98	3.3	0.93	3.3	0.90
odd & even	3.2	0.98	3.2	0.90	3.3	0.89
all aug + orig	3.2	0.99	3.3	0.93	3.3	**0.93**
no aug + orig	3.2	0.96	3.3	0.91	3.3	0.89

Table 1: Experiments results of different data augmentation approaches. "All aug" means all three data augmentation methods. "Odd & even" uses only odd and even augmentation method. "orig" means include original data in the train set.

frame 5, ...]), the instance that contains only frames at even positions (i.e., [features of frame 2, features of frame 4, features of frame 6, ...]), and the instance that dropped first three frames and last two frames (i.e., [features of frame 4, features of frame 5, features of frame 6, ...features of frame n-2], given that n is the number of frames in the video).

3.2.3 Model Training and Testing

We opted to a LSTM neural network as the sign language translator. The model is composed of bidirectional 2-layer LSTM, a linear layer and a 64-way softmax layer.

4 Experiments

As mentioned in Section 3.1.1, there are 10 different signers in LSA64 dataset. In order to guarantee the efficiency of model, videos of signer 1 to 8 is used as the train set, videos of signer 9 is used as development set, and videos of signer 10 is used as test set. In other words, there is no overlap of signer between train, dev, and test sets.

Different data augmentation strategies are experimented on the model. Results are shown in Table 1. The best performance model has a learning rate of 0.001, 64 hidden dimension size in LSTM layer, and is trained using all three data augmentation strategies. As can be seen, the model benefits from data augmentation. This strategy avoids overfitting issue by adding more variety into the train set.

5 Conclusion

In the study, we proposed to translate Argentinian sign language based on human keypoint information estimation. Each video is represented as a sequence of frame features. The classifier used for translation is a bidirectional LSTM. We also experimented with several augmentation approaches. The results show that by adding variety to training data, the model achieves better result.

References

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[CCHKB17] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. Subunets: End-to-end hand shape and continuous sign language recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3056–3065, 2017.

[CHK+18] 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*, pages 7784–7793, 2018.

[CLZ17] Runpeng Cui, Hu Liu, and Changshui Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7361–7369, 2017.

[KKJC19] 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, 2019.

[LPM15] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[RQE+16] Franco Ronchetti, Facundo Quiroga, César Armando Estrebou, Laura Cristina Lanzarini, and Alejandro Rosete. Lsa64: an argentinian sign language dataset. In *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*., 2016.