Neural Machine Translation for Indian Sign Language

Dissertation Submitted In Partial Fulfillment Of The Requirements For The Degree Of

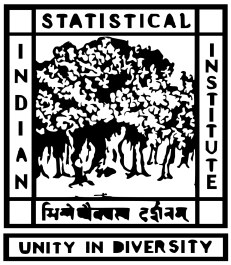
Master of Technology
in
Computer Science

by

Sushant Sharad Moon
[Roll No: CS1807]

Under the Guidance of

Dr. Utpal Garain
Professor
Computer Vision and Pattern Recognition Unit(CVPR)



Indian Statistical Institute
Kolkata-700108, India

CERTIFICATE

This is to certify that the dissertation entitled **"Neural Machine Translation for Indian Sign Language"** submitted by **Sushant Sharad Moon** to Indian Statistical Institute, Kolkata, in partial fulfilment for the award of the degree of **Master of Technology in Computer Science** is a *bona fide* record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Dr. Utpal Garain
Professor,
Computer Vision and Pattern Recognition Unit,
Indian Statistical Institute,
Kolkata-700108, India.

Acknowledgment

I would like to thank Dr. Utpal Garain, it was an absolutely great privilege and learning experience to work with him. He has been a constant source of support, starting from my first year. I wish to express my sincere gratitude to all the research scholars in NLP Lab. Discussions with Akshay Chaturvedi and Joey Mohapatra enlightened me. Thanks to all the friends who have been there with me for the past two years. Lastly, I would like to thank my parents who have supported me and ensured my well being.

Contents

1 Introduction **1**
- 1.1 Background . . . 2
  - 1.1.1 Recurrent Neural Networks . . . 2
  - 1.1.2 Long Short Term Memory Networks . . . 2
  - 1.1.3 Gated Recurrent Unit . . . 3
  - 1.1.4 Encoder-Decoder Architecture . . . 4
  - 1.1.5 Sign Language Processing . . . 5
- 1.2 Previous works . . . 6
  - 1.2.1 Datasets available for different Sign Languages . . . 6
  - 1.2.2 Sign Language Recognition Systems . . . 7
  - 1.2.3 Attempts at Sign Language Translation . . . 7

2 Dataset Creation **9**
- 2.1 Collection Procedure . . . 9
- 2.2 Transcribing Videos . . . 10
- 2.3 Preliminary Analysis . . . 11

3 Methodology And Results **12**
- 3.1 Frame and Word Embedding . . . 13
- 3.2 Attention-based Encoder-Decoder Network . . . 13
  - 3.2.1 Encoder-Decoder Network . . . 13
  - 3.2.2 Attention in Encoder Decoder Model . . . 14
- 3.3 Implementation and Evaluation details . . . 15
- 3.4 Results . . . 16

4 Conclusion and Future Works **19**

5 Bibliography **20**

Abstract

Sign languages being the primary language of the deaf community, researchers from many fields have been working in this domain from the past two decades. Until now, the majority of the work was in Sign Language Recognition. And only recently, few methods on Sign Language Translation have been developed, but even today, there does not exist any work on Indian Sign Language Translation. This work aims to translate Indian sign language videos to their corresponding spoken Indian English sentences.

In this work, we are publicly releasing the first of its kind Indian Sign Language Translation dataset, namely, the *ISI-ISL-DDNEWS-2020T* that we collected and annotated. Our dataset has *>3 Million* sign language frames, which translate to *>93 Thousand* words made out of *>6 Thousand* vocabulary words in spoken Indian English language.

We also formalize a neural machine translation system trainable end-to-end for Indian Sign Language and benchmark on the said dataset. The model jointly learns the spatial & temporal relationship, underlying language model, and the sign & spoken language alignment. This baseline model gives the translation a BLEU-4 score of 4.02.

Introduction

Worldwide more than 360 million people, which is 5.3% of the world's population, suffer from disabling hearing loss. In India, it is estimated that there are 63 million people suffering from this condition. 4 in every 1000 children suffer from severe to profound hearing loss and over 100,000 babies are born with hearing deficiency every year [1] and many cases are not reported due to social stigma faced by the family. Another article [2] states that, 99% of the deaf population is not able to matriculate. These staggering numbers, inspires this work which is an effort to help our community to understand and popularize Indian Sign Language (ISL).

Sign Language being a primary language for deaf has been studied by linguists for many years world wide. Like any other language, sign language has its own unique linguistic and grammatical structures [3], and it does not translate to spoken language word by word with one-to-one mapping as shown in Fig. (1.1).

In the past, most of the research was done in Sign Language Recognition [4] and they approached the task as a basic gesture recognition problem, ignoring underlying linguistic properties. In recent years, due to advances in computer vision, Continuous Sign Language Recognition became possible. Sign Language Translation is relatively unexplored area and is currently a hot topic among researchers [5] [6] [7].

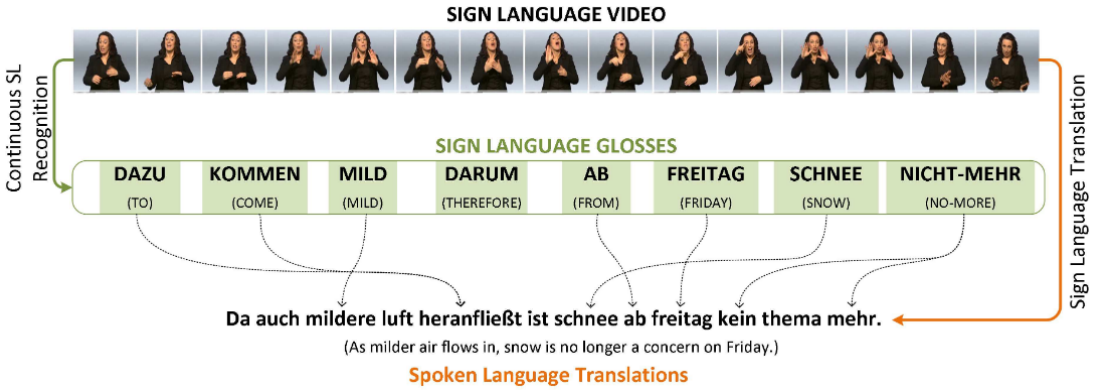


Figure 1.1: Difference between SLR and SLT

When considering Indian Sign Language, to the best of our knowledge, there does not exist any work aimed at translation. This work is the first of its kind for ISL and following are our contributions,

1. First publicly available translation dataset for ISL, which has sign video and associated spoken language annotation.
2. Baseline model for future research on ISL, we also share the parameters scheme for the model.

In this chapter, next is the brief of the terminology used often in this thesis, after which we discuss previous works in the field of sign language in general. Subsequent chapters are organised as follows : In Dataset Creation we talk of how the dataset was collected, processed and consolidated. In Methodology And Results we discuss our baseline model, its implementation, training and final results.

1.1 Background

Before we go further, following is a brief of important terms that we will be referring to later in this work.

1.1.1 Recurrent Neural Networks

Standard neural network type is feed forward neural network, where sets of neurons are organised in layer like fashion namely one input layer, one output layer and at least one intermediate hidden layer. This type of neural network is typically limited to static classification tasks and is hence limited to provide a static mapping between the input and the output. This network would fail in the task of modelling the temporal changes in a sequence while keeping the number of parameters constant. To overcome this shortcoming, Williams et al. [8] introduced Recurrent Neural Networks(RNN) in 1989, where they circularly fed back the signal from the previous time step to the hidden neuron along with the originating signal. This new architecture enabled to learn the sequential information, while keeping the number of parameters independent of the sequence length by sharing the parameters at each time step.

For an input sequence $X = (x_1, x_2, ..., x_{n-1}, x_n)$, at each time step t RNN calculates the hidden state h_t as shown in Eq. (1.1) and from h_t, the corresponding output o_t for that time step is calculated as shown in Eq. (1.2) where, $R()$ is a RNN Cell and $F()$ is the Feed Forward Neural Network.

$$h_t = R(x_t, h_{t-1}) \tag{1.1}$$

$$o_t = F(h_t) \tag{1.2}$$

1.1.2 Long Short Term Memory Networks

After RNN was introduced it was soon noticed that over time the gradient of the feed back signal would either vanish or explode. Schmidhuber et al. introduced Long Short Term Memory - Recurrent Neural Networks (LSTM-RNN) in 1997 [9] and improved it over time [10] [11] [12] to address the shortcomings of the regular RNN.

LSTM consists of special modules namely *Input Gate* $I()$, *Forget Gate* $F()$ and *Output Gate* $O()$. Also, apart from hidden state, LSTM internally maintains the *Cell State* which helps it to keep track of the long term dependencies. Information can flow along the cell state unchanged, and if needed LSTM can easily add or remove information (gradient) with the help of input gate and forget gate respectively. And output gate helps the LSTM to generate the hidden state with the help of cell state.

For an input sequence $X = (x_1, x_2, ..., x_{n-1}, x_n)$, at time step t, the hidden state h_t and cell state c_t is calculated as shown in Eq. (1.3), where f_t, i_t and o_t are outputs from forget, input and output gates respectively, and $C()$ is an intermediary function with tanh activation.

$$\begin{aligned} f_t &= F(x_t, h_{t-1}) \\ i_t &= I(x_t, h_{t-1}) \\ o_t &= O(x_t, h_{t-1}) \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot C(x_t, h_{t-1}) \\ h_t &= o_t \cdot H(c_t) \end{aligned} \tag{1.3}$$

1.1.3 Gated Recurrent Unit

Motivated by the LSTM unit, in 2014 Cho et al. introduced Gated Recurrent Unit (GRU) which were much simpler to compute and implement [13]. Similar to what we saw in a LSTM unit, GRU also has gating units that modulate the flow of information inside the unit but with just the hidden state[13] [14]. No additional cell/memory state is present in GRU.

Apart from this, it also has fewer gates compared to LSTM, namely, *Reset Gate* $R()$ and *Update Gate* $U()$. When reset gate is close to 0, the hidden state is forced to ignore the previous hidden state and reset with the current input only. This effectively allows the hidden state to drop any information that is found to be irrelevant later in the future, thus, allowing a more compact representation. On the other hand, the update gate controls how much information from the previous hidden state will carry over to the current hidden state [13].

Eq.(1.4) explains the calculation for the hidden unit h_t at time t for input sequence $X = (x_1, x_2, ..., x_{n-1}, x_n)$. r_t, u_t are output of reset gate and update gate respectively and $H()$ is an intermediary function

with tanh activation.

$$r_t = R(x_t, h_{t-1})$$

$$u_t = U(x_t, h_{t-1}) \tag{1.4}$$

$$h_t = u_t \cdot h_{t-1} + (1 - u_t) \cdot H(r_t, h_{t-1}, x_t)$$

1.1.4 Encoder-Decoder Architecture

This architecture was introduced by Kalchbrenner et al. [15], Sutskever et al. [16], Cho et al. [17] [13]. Encoder-Decoder (ED) Architecture has gained lot of popularity in recent years and many of the state of the art models in Neural Machine Translation (NMT) [18] [19] [20] [21] [22] [23], Text Summarization [24] [25] [26] [27], Image Captioning [28], etc. have ED Architecture at its core. Hereafter, we shall be referring to ED Architecture in context of the NMT.

Fundamental idea behind this architecture is, the encoder part reads the input (or a sequence of input) and condenses its meaning down to a fixed sized vector referred as context vector. This context vector is then fed to the decoder part which generates the desired results. The encoder part and the decoder part are generally recurrent neural networks and are both jointly trained in-order to maximize the probability of translation given a source sentence.

To understand this architecture more clearly, consider the following example. Let sample sentence X be the source of length n where x_i is the word at i^{th} position. Similarly, let Y be the corresponding target (translation of X) sentence of length m and y_j be the j^{th} word in y. Now, the encoder generates the hidden state $he_i = Encoder(x_i, he_{i-1})$ using i^{th} word in x, i.e. x_i, and the hidden state from $i - 1^{th}$ step, i.e. he_{i-1}. This process is continued till he_n is obtained, which is then used to initialize the hidden state decoder, i.e. $hd_0 = he_n$. It should be noted that he_n or hd_0 is also called the context vector. Now the decoder takes in hidden state as hd_0 and a *special token* $< eos >$ as input and predicts $\hat{y}_1$ and outputs the next hidden state hd_1. This process continues in auto-regressive fashion, which can be summarized by the equation $\hat{y}_j, hd_j = Decoder(\hat{y}_{j-1}, hd_{j-1})$. The predictions by decoder continues and once $< eos >$ is obtained in the prediction, the decoder stops and it symbolizes the end of prediction for the source sentence x. Figure (1.2) shows the high level overview of the architecture.

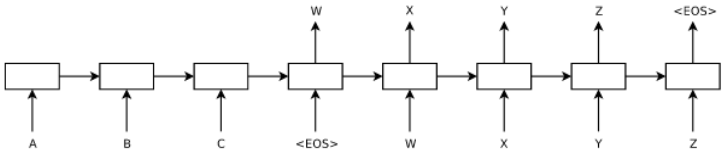


Figure 1.2: Model based on Encoder-Decoder Architecture for translating input sentence "ABC" and producing "WXYZ" as the output [16].

As one would notice, however long the input sample is, the context vector summaries all the information from it and is solely responsible for passing that information to the decoder for the final translation, due to which context vector itself became a bottleneck along with the long term dependencies between the source and the target sequence.

A variation of the novel differentiable attention mechanism given by Graves [29] was successfully applied to machine translation by Bahdanau et al. [30] to solve the bottleneck and improved by Luong et al. [31]. It allowed the ED models to jointly learn the alignment between the of source and target sentences along with the translation. The idea was, instead of relying solely on the single context vector hd_{j-1}, for each step j of prediction the decoder will takes c_{j-1} along with $\hat{y}_{j-1}$ to output $\hat{y}_j$ and hd_j, where c_{j-1} is a weighted average of all the hidden states vectors of the encoder, i.e. $he_0, he_1, ..., he_n$, along with hd_{j-1}. The weights associated are learned parameters and are called soft attention over the input words, they also give the alignment of the source word with the target word.

Apart from this, Connectionist Temporal Classification given by Graves et al. [32] has also gained popularity. For a given X it gives us an output distribution over all possible Y's. This distribution then can be used to either infer a likely output or assess the probability of a given output. It has to be noted that it assumes a monotonic alignment between inputs and outputs [16].

1.1.5 Sign Language Processing

Though machine translation between written languages has advanced, the field of sign language processing still lags behind. Following are the major reasons for it [33] [34] [35] [36],

1. Sign language is a multi dimensional form of communication, it not only involves manual cues (like hand gestures and pose) but also non-manual cues (like subtle facial features specifically eye gaze, head pose and facial expression) which presents complex computer vision challenge.
2. Depending on the speed or the magnitude with which a gesture is performed, the two sequences of the same sign may differ.
3. Camera position affects the recognition.
4. No universal transcribing method for converting sign language in video form to sign glosses which are in written form.
5. No fixed number of frames per gloss.
6. Sparse annotated sign language data sets, with limited size and/or vocabulary.

Sign Language Glossing

A sequence of gesture that forms a single sign when transcribed in written form is called a sign gloss. With series of such signs the sign language speaker conveys a concept that has to be understood by the interpreter. A series of glosses do not directly translate to the appropriate sentence in written language that convey the same meaning, as they are mere representation of what is being said in the sign language sequence. Many times, glosses would also include the notations for facial expressions and body language that comes with the sign language. Though for various sign language corpus

projects have gloss annotation guidelines [37] [38] [39] [40], there exists no single standard which they all follow, hence data is not consistent and interchangeable across the projects [41] [42].

Sign Language Recognition

Recognizing independent sign gloss from isolated sign language videos forms the Sign Language Recognition (SLR) task. And detecting glosses from a given video form the Continuous Sign Language Recognition (CSLR) task. This represents a considerably more challenging task from the computer vision prospective when considering that the input is the high dimensional spatio-temporal data, i.e. sign videos, and model is required to understand what signer looks like, how they move and interact within their 3D signing space. Moreover, model needs to comprehend what these aspects mean in combination. Despite all that, with availability of annotated data sets, few CSLR systems have been developed and are showing promising results [43] [44] [45] [46] [47], however they can only recognize sign glosses and operate within a limited domain [40].

Sign Language Translation

As with any other natural language, sign language has it own unique linguistic and grammatical structures, which often does not match one-to-one with its spoken language counterparts. Sign language translation (SLT) is harder problem than SLR or CSLR as the later gives the sign gloss which are simplified representation of the sign language and much of the linguistic and grammatical structures are lost. SLT, inherently is a novel problem and more difficult compared to normal spoken language translation tasks because it involves extracting meaningful features from a video of a multi-cue language. Recently in-depth study on German sign language was conducted [5], [7], [6], which translated from sign to text by using gloss as intermediate representation to obtain state of the art performance on [40].

1.2 Previous works

Computer vision community has studied sign languages since 1980s [48] [49] with an end goal to build translation systems [50] capable of translating sign language to spoken language sentence and vice versa to help daily life of the Deaf community [4] [51]. We have divided the previous works in corresponding relevant subsections for better understanding. They are as follows,

1.2.1 Datasets available for different Sign Languages

Data collection and annotation by human experts is a very laborious and costly process. Though datasets from linguistic sources [52] [53] and sign language interpretation broadcasts [54] were available, they were weakly annotated (subtitles) or the size was limited so as to build a generalized model. It has to be noted that they lacked human pose information which the legacy SLR systems heavily were dependent on. In those circumstances, researchers created isolated sign language recognition datasets in controlled environment with very limited vocabulary, these datasets were essentially application specific [55] [56] [57] [58]. This hindered development of SLR and SLT tasks.

As deep learning methods and algorithms capable of learning from weakly annotated datasets were developed [59] [54] [60] and the field of human pose estimation progressed [61] [62] [63], datasets

from linguistic sources and sign language broadcasts started being used for SLR tasks. Nonetheless as sign sentence and the spoken sentence are non-monotonic, i.e. they have different ordering and coupled with fact that sign glosses and linguistic constructs neither always have a one-to-one mapping with their spoken language counterpart nor they share the same temporal order, hence these SLR datasets could not serve the end goal SLT. This lead to the development of RWTH-PHOENIX-Weather dataset for recognition and translation tasks [64] [40]. RWTH-PHEONIX dataset has gloss level and sentence level annotation for each of the sign language video and it soon became the benchmark dataset for sign language translation. To our knowledge, there does not exists any Indian Sign Language Datasets publicly available/suitable for SLT tasks.

1.2.2 Sign Language Recognition Systems

Initially majority of the research was done to recognize isolated signs [65] [66] [55] [67] [68] [69] [70]. Till recently many SLR methods used hand crafted intermediate representations [43] [71] and temporal changes were modelled with classical graph based methods like template/pattern matching [59] [72], Hidden Markov Models (HMMs) [73] or Conditional Random Fields [74].

With Deep Learning [75] gaining popularity in computer vision [76] and speech recognition [77], SLR community quickly adopted it. For extracting manual [78] [79] and automatic [80] feature representations Convolutional Neural Networks (CNNs) [81] were used, and Recurrent Neural Networks (RNNs) were used to model the temporal changes [82] [83] [84] [5].

In 2006, Graves proposed Connectionist Temporal Classification (CTC) Loss [32]. It considers all possible alignments between the source and target sequences when calculating the loss and this enabled to train, the models, end-to-end on weakly annotated datasets. It quickly became popular for many seq-to-seq tasks, models with CTC loss layer achieved state of the art performances for speech recognition [77] [85] and hand writing recognition [86]. Computer vision community applied it to weakly labelled visual problems like action recognition [87], lip reading [88], hand shape recognition [83] and CSLR [79] [83] [84] [89] [90] on continuous/video data.

1.2.3 Attempts at Sign Language Translation

In early 2000s, Ney et al. [91] introduced conceptual video based SLT systems. Initially SLT was mostly being done as text to text translation, but with limited dataset size (averaging about 3000 words) [36] [35] [34]. Chai et al. [65] proposed method which recognized signs in isolation and then constructed sentences using language model. Stein et al. [35] introduced weather broadcast translation system from German Sign Language, i.e. Deutsche Gebärdensprache (DGS) to spoken German Language and vice versa using RWTH-PHEONIX [64] dataset. Morrissey [92] gave method that translated spoken German to Irish Sign Language and DGS, her model also translated spoken English to Irish Sign Language and DGS. Ebling [93] developed an approach to translate written German to Swiss-German Sign Language, i.e. Deutschschweizer Gebärdensprache (DSGS).

Before Deep Learning was applied to machine translation [94], and inturn to SLT, it was not possible to train a model end-to-end from videos. It was observed that CTC loss function assumes source and target sequences to share the same temporal order and conditional independence within the target sequence, this inhibits network to learn the language model and hence was deemed unsuitable for machine translation [94] [15]. This led to the the development of Encoder Decoder (ED) Models, it uses intermediary latent space to map source and target sequences, similar to auto-encoders[75],

but for temporal sequences. ED models were first proposed by Kalchbrenner et al. [15] but with single RNN for both encoding and decoding. Cho et al. [17] and Sutskever et al. [16] introduced independent encoder and decoder modules, each based on RNNs. The bottle neck of the ED model, namely, fixed sized context vector and long term dependencies were resolved by Bahdanau et al. [30] when he proposed attention mechanism. This attention function, at its core calculates the alignment between source and target sequence. Attention Mechanism was further improved by Luong et al. [31] who introduced additional type of attention score calculation and input feeding approach. ED models led the emergence of Neural Machine Translation (NMT)[94], and since then many attention based models for NMT have been proposed, such as GNMT introduced by Wu t al. [95] which combines bi-directional and uni-directional encoders and Gehring et al.'s [96] convolution based seq-to-seq learning method. Attention based models have also been proposed for tasks like image captioning [97], lip reading [98] and action recognition[99].

With the developments in ED Models, Camgoz et al. [5] proposed an attention based encoder decoder method for SLT on German Sign Language. Ko et al. [100] proposed a similar method for Korean Sign Language but used body key point coordinates as input. Transformers were introduced by Vaswani et al. [23] and it achieved astounding success in various challenges like multi-modal language understanding, learning sentence representation, language modelling, activity and speech recognition [101] [102] [103] [104] [105] [106]. Transformer when applied to SLT also gave good results [7] [6].

Dataset Creation

There does not exists any Indian Sign Language Datasets, as we have seen in Section (1.2.1), and with this work for the very first time we are introducing *ISI-ISL-DDNEWS-2020T*, an Indian Sign Language (ISL) Translation Dataset, which we plan to make publicly available for facilitating the future growth of SLT research. Sign language videos in ISI-ISL-DDNEWS-2020T are the captures of DDNews Broadcast [107] aired between 13^{th} *August* 2018 and 5^{th} *May* 2019. For each news video we transcribed it in text format to get the corresponding annotation. The details are discussed below,

2.1 Collection Procedure

The news videos were downloaded with the help of *Youtube Downloader* [108] at 25 frame rate and the processed with *FFmpeg* [109] to crop and rescale the video down to the region were only the Sign Language Anchor is speaking the Sign Language sentences. A Sample complete frame and its processed version from the one of the DDNews video is shown in Figure (2.1).

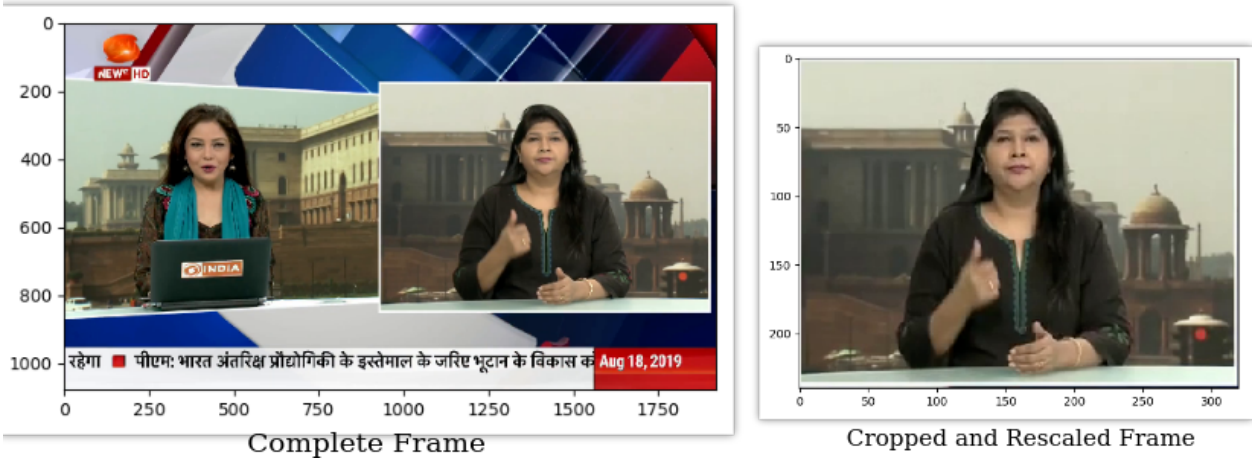


Figure 2.1: Sample Original & Processed Frame of DDNews video

These videos also contained noise, namely, buffer frames, animation frames and frames where the signer is fading in & out. We mark these frames and the video is clipped into sub-videos at any

Figure 2.2: Noise Frames

such noisy frames. Sample of noise frames are shown in Fig.(2.2)

It was noticed, the news sentences were spoken in intervals with silence regions in between, we used this observations to manually clip the videos to smaller lengths bringing down the number of frames per new video and increasing the total number of videos. Originally each video had video (RGB frames) and audio components to them, we extracted audio component for transcribing which discuss next.

2.2 Transcribing Videos

After separating audio for each of the video, we test several best transcribing services that are available in the market, namely, YouTube [110], Google [111] and Amazon Web Service (AWS) [112]. All of them being commercial services, have been trained heavily for identifying spoken Indian English Accent. Table (2.1) shows the text sample transcribed by different services.

Empirically it was observed that AWS Transcribe outperformed all of the other transcribing services, leading us to train our models with the transcribes generated by the AWS.

For each video, the transcribed sentence represents the annotation. We parse, tokenize and lowercase the sentence, then clean all the punctuation, stop words and stem the words to their root using *PorterStemmer* from popular *NLTK* Library to obtain the final annotation.

Original Sentence	Good morning and welcome to news for Hearing Impaired and I Subhedu with my colleague Meera Bhatia.
AWS Transcribe	Good morning and welcome to news for Hearing Impaired and Subin do with my colleague Meera Bhatia.
Google Transcribe	good morning and welcome to news for hearing-impaired I'm Shivan do with my colleague Mira partyi

Table 2.1: Comparison between off-the-shelf top-of-the-line Transcribing Services available in the market.

2.3 Preliminary Analysis

Our final sign language video and its corresponding annotation looks like as shown in Figure (2.3) (without the down-sampling) , which shows frames of a sample sign language video down-sampled for illustration and its annotation.

Figure 2.3: Frames of sample sign language video after down-sampling, and its corresponding annotation.

Overall, ISI-ISL-DDNEWS-2020T has 7 ISL anchors (signers), who together generate 1465 sign language videos, totalling to 11 hours 9 minutes and 48 seconds ISL data. The maximum-duration video is 1 minute, 51 seconds long and minimum-duration video is 2 seconds long. Among the signers, the data split is shown in the Table (2.2). The word count of the annotations is 93103 and the vocabulary size is 6972 with 621 OOV[1] words. Details of the analysis are shown in Table (2.3).

Signer	Train Set	Test Set	Validation Set	Total Videos
Signer 1	179	22	22	223
Signer 2	412	51	51	514
Signer 3	121	15	15	151
Signer 4	165	20	20	205
Signer 5	222	27	27	276
Signer 6	60	7	7	74
Signer 7	18	2	2	22
Total	**1177**	**144**	**144**	**1465**

Table 2.2: Total number of videos per signer.

	Train Set	Test Set	Validation Set	Total
Words	73895	9374	9834	93103
Vocabulary Size	6351	2096	2121	6972
Number of OOV Words	-	304	342	621

Table 2.3: Annotation Analysis

[1]OOV (Out of Vocabulary) words are never encountered by the model during training

Methodology And Results

In this chapter, we introduce our Indian Sign Language Translation system. It translates the ISL videos to Spoken Indian English language sentences in end-to-end manner. The objective at its core, is to learn the conditional probability $p(Y|X)$ of generating a spoken Indian English language sentence $Y = (y_1, y_2, ..., y_m)$ with m words from a sign video $X = (x_1, x_2, ..., x_n)$ with n number of frames. By nature, modeling conditional probability $p(Y|X)$ is a seq-to-seq learning problem. And this is not a straight forward task, as videos generally have a higher frame rate, which leads to very high number of frames when compared to the words spoken in the video, i.e. $n >> m$. Furthermore, as ISL video Y and spoken English Language sentence X each have different vocabularies, grammatical rules and ordering, hence their alignment is unknown and non-monotonic. To tackle this problem, we use an attention based encoder decoder model architecture equipped with CNNs input layer [5] [113] trainable end-to-end. Figure (3.1) gives overview of the architecture used for ISL video to spoken Indian English language sentence translation.

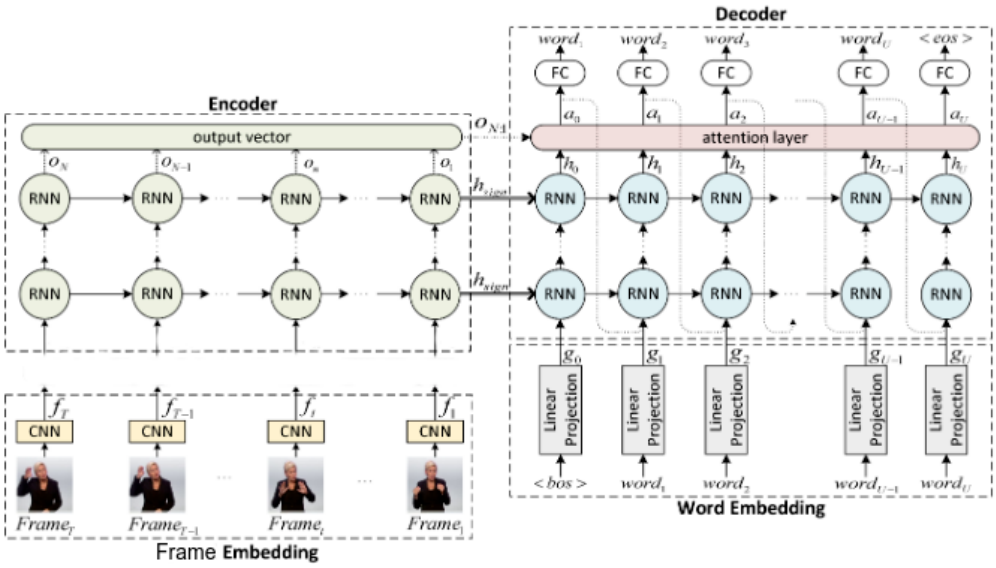


Figure 3.1: High level overview of attention based encoder-decoder model with modified CNN input layer used for SLT

3.1 Frame and Word Embedding

Following the NMT approach, each video X is tokenized as frames, i.e frame x_i is the i^{th} frame in the video, and embedded with $FrameEmbedding()$.

The idea behind word embedding is to transform the initial sparse one-hot vector representation, where words are equidistant from each other, to a denser form, where similar meaning words are closer. In this case, instead of words we have frame/images, so we use 2D CNN model pretrained on larger dataset and fine-tuned during training for generating the frame/image embedding.

Given an image x_i, $FrameEmbedding()$ gives out feature vector f_i as shown in Eq. (3.1) which is then passed to the Encoder model.

$$f_i = FrameEmbedding(x_i) \tag{3.1}$$

Each annotation Y is tokenized as words, y_j being the j^{th} word, and embedded with $WordEmbedding()$. For $WordEmbedding()$, we use fully connected layer that learns a denser representation g_j from one-hot vector representation of the word y_j as shown in the Eq.(3.2).

$$g_j = WordEmbedding(y_j) \tag{3.2}$$

3.2 Attention-based Encoder-Decoder Network

We shall now discuss each of the part of our model in details.

3.2.1 Encoder-Decoder Network

As explained before in (1.1.4), the RNNs in encoder-decoder models work in two modules, namely encoder and decoder. In the encoder module, embedded sign language video frames are summarized and projected to a fixed sized vector representation which are later used by the decoder module for generating the spoken language sentence.

In the encoding phase, given a sequence of frame level embedding for each frames of the video $f_{1:n}$, the order of the sequence is reversed to shorten the long term dependency of the start of the video to the start of the spoken sentence as explained by Sutskever et al. [16]. The encoder RNNs then reads the reversed feature representations $f_{n:1}$ one step at a time along with the hidden state of the encoder from the last time step to model the temporal changes in the hidden state of the current time step. As shown in Eq. (3.3) as $i = n..1$, the hidden states $o_n, o_{n-1}, ..., o_1$ are generated, where o_{n+1} is the zero vector and o_1 corresponds to the h_{sign} which has the condensed representation of all the frames $f_{n..1}$. Now, h_{sign} is passed to the decoder to initialize its hidden state for the next phase.

$$o_i = Encoder(f_i, o_{i+1}) \tag{3.3}$$

In decoding phase, at each time step $j = 1..n$, the decoder as shown in the Eq. (3.4) takes in the previous hidden state of the decoder h_{j-1} and word embedding g_{j-1} of the previously predicted

word y_{j-1} as input and gives out the prediction y_j and updated hidden state h_j for the j^{th} time step, where the decoder is initialized with h_{sign} (sole input form the encoder, called context vector), i.e. $h_0 = h_{sign}$ and first input, i.e. g_0, is the word embedding for the special start token $< bos >$ which represents beginning of the sentence. This triggers the start of the translation and decoder generates the translation word by word in auto regressive fashion till the special token $< eos >$ signalling the end of the sentence is not obtained from the network. With the network behaving in this manner, the decoder at its core decomposes the conditional probability $p(Y|X)$ to ordered conditional probabilities as shown in the Eq. (3.5), which is then used to calculate the errors by applying cross entropy loss for each word. The errors are then back propagated through out the encoder-decoder network along with the CNN and word embedding layers.

$$y_j, h_j = Decoder(g_{j-1}, h_{j-1}) \tag{3.4}$$

$$p(Y|X) = \prod_{j=1}^{M} p(y_j|y_{j-1}, h_{j-1}) \tag{3.5}$$

3.2.2 Attention in Encoder Decoder Model

With classical encoder-decoder models, there are following two major drawbacks,

1. Bottleneck for flow information from encoder using a fixed sized context vector to the decoder.
2. Long term dependencies, and it is a big hurdle especially in our case, as number of frames is vastly larger than the number of words in the spoken language sentence, i.e. $n >> m$. This also leads to the problem of vanishing gradient problem.

To overcome these problems, we use attention mechanism as suggested by Bahdanau et al. [30] and later improved by Luong et al. [31]. The idea behind attention is to pass the information in the hidden layer from all the time steps of the encoding phase, i.e the information from $o_{1:n}$ is passed to the decoder at every time step of the decoding phase. So for time step j of the decoding phase, the context vector c_j for is the weighted sum of hidden state, $o_{1:n}$, of all the time steps of the encoding phase as stated in Eq. (3.6), where γ_i^j represents the *attention* the decoder is giving to the input f_i at j^{th} time step of the decoding phase when predicting the word y_j. The weight vector $[\gamma_i]$ also help in visualizing the alignment of the i^{th} frame of the sign language video to all the predicted words of the spoken language sentence, and this is similarly extended to all the remaining frames and vice versa.

$$c_j = \Sigma_{i=1}^{n} \gamma_i^j o_i \tag{3.6}$$

$$\gamma_i^j = \frac{e^{score(h_j, o_i)}}{\Sigma_{i'=1}^{n} e^{score(h_j, o_{i'})}} \tag{3.7}$$

The attention weight γ_i^j is calculated as shown in the Eq. (3.7), where the scoring function, $score()$ depends on the attention mechanism. There are two popular attention mechanism, namely,

concatenation based as shown in Eq. (3.9) given by Bahdanau et al. [30] and other is multiplication based as shown in Eq. (3.8) given by Luong et al. [31], where W_l, W_b and V_b are trainable parameters.

$$score(h_j, o_i) = h_j^T W_l o_i \tag{3.8}$$

$$score(h_j, o_i) = V_b^T \tanh(W_b[h_j; o_i]) \tag{3.9}$$

$$a_j = \tanh(W_c[c_j; hj]) \tag{3.10}$$

Once c_j is calculated it is combined with the current hidden state h_j using the trainable parameter W_c as shown in Eq. (3.10) to get the attention vector a_j. This attention vector is passed to the final fully connected layer to model the ordered conditional probability mentioned in Eq. (3.5), the attention vector is also passed to the next decoding stage $j+1$, thus the decoder Eq. (3.4) changes to (3.11). This completes the attention based encoder-decoder model. In this work, we shall be using the improved attention mechanism (3.8) given by Luong et al. [31].

$$y_j, h_j = Decoder(g_{j-1}, h_{j-1}, a_{j-1}) \tag{3.11}$$

3.3 Implementation and Evaluation details

Framework We modified Neural Sign Language Translation model framework [1] given by Camgoz et al. [5] and based out of Luong et al.'s NMT seq2seq library [113]. All of our implementation use Tensorflow as base [114].

Network Details The RNNs in our Encoder-Decoder network are made up of two stacked GRU [13], each with hidden layer of size 1000. For $FrameEmbedding()$ we use AlexNet, proposed by Krizhevsky et al. [76], without its last layer. It is initialized with the model weights obtained from the training on ImageNet dataset [115]. All the remaining parts of the network are initialized using Xavier's initialization [116].

Performance Metrics To measure the translation performance of our network, we utilize BLEU [117] score, which is the most common and popular metric for measuring machine translation. We report BLEU-(1,2,3,4) score for better perspective of the translation performance.

Training We use Adam Optimizer with default parameter, except learning rate which is set to 10^{-5}. We also employ dropout connections with a drop probability of 0.2 and gradient clipping with a threshold of 5.

[1] https://github.com/neccam/nslt

The Training process is two staged, first we train our model on RWTH-PHEONIX Translation [40]. After that we fine-tune on ISI-ISL-DDNEWS-2020T. We follow the train, test and validation split as mentioned in (2.2).

In both the cases, training and fine-tuning, the networks learns till perplexity converges which happens at 20 epochs on average for both the stages. The model is evaluated on validation and testing sets at every half-epoch and report results for each step using the model that performed best on the test set. In total it took 10 days to train the complete model.

System Level Information All the data processing took place on a small machine with *8GB Ram* and *Intel(R) Core(TM) i5-4200M CPU @ 2.50GHz*. The training took place on server machine with *1TB Ram*, *Intel(R) Xeon(R) Platinum 8164 CPU @ 2.00GHz* and *3 Nvidia Tesla P6 GPU*. Due to the resource sharing of the server machine we used roughly 7% of the total system RAM, CPU and 1 GPU was used at 60% utilization during the training period. Resource sharing also limited the size of input which could be fed to the network, currently only the inputs with upto 800 frames could be used for training.

3.4 Results

In our experiment of translating Indian Sign videos to spoken Indian Engish, we obtain BLEU-4 score of 4.02 on test-set. The details of the model performance on validation and test set are mentioned in the table (3.1).

	Validation Set	Test Set
BLEU-1	10.79	10.70
BLEU-2	8.47	9.02
BLEU-3	6.92	5.64
BLEU-4	4.94	4.02

Table 3.1: Indian Sign to Text Translation Evaluation scores.

It is observed, this baseline sign2text model performed better for small videos when compared to longer videos, which can be attributed to poor resolution of long term dependencies by GRU and less number of stacked layers. Table (3.2) lists out few of the samples where model performance was extremely poor.

We believe, these preliminary results can be improved. More rigorous training/testing is required. Possible options for better results would be,

1. Training/Testing with better set of hyper-parameters.
2. Instead of working with just 2 layers of GRU, training/testing needs to be done with 3 and 4 layers of GRU.
3. Other popular Neural Networks popular with ED architecture like LSTM, Transformers should boost the results
4. We need to experiment with different Attention Mechanisms.

Original Annotation	well storey presid ramnath present gandhi peac prize year 2015 16 17 today rashtrawadi bhavan new delhi prime minist narendra modi also attend award ceremoni fill state awarde .
Model Prediction	prime minist narendra modi address public ralli suru assam prime minist narendra modi address public ralli look sonia prime minist narendra gandhi address public ralli strengthen .
Original Annotation	prime minist narendra modi confer soul peac prize award ceremoni organis soul peac prize foundat pm dedic price peopl india donat cash award na mammi gaia initi .
Model Prediction	prime minist narendra modi address public meet tour korea today today today public public public ralli look stone cooper cooper gather gather strengthen strengthen strengthen strengthen strengthen .
Original Annotation	congress presid rahul gandhi visit riberi lee today address public meet voter congress presid also visit amethi attend variou polit ralli congress gener secretari priyanka gandhi wardrob also visit uttar pradesh address elect ralli riberi lee district .
Model Prediction	bjp presid amit shah hold andhra pradesh andhra pradesh arunach pradesh bihar jd ljm bihar contest contest contest contest contest contest contest contest contest .
Original Annotation	prime minist narendra modi congratul isra counterpart benjamin netanyahu appear head victori parliamentari poll tweet pm modi describ netanyahu great friend india say new delhi look forward continu work take bilater partnership new height .
Model Prediction	storey detail sabha elect poll phase lok sabha elect spread spread 18 union union constitu full full full arrang poor lok sabha elect .

Table 3.2: Instances where model performed poorly.

5. With better Spatial Embeddings, the results should improve.
6. With sign articulators, namely, facial features, hands, etc, being so subtle, specifically modelling them and feeding it as an input along with the base image should also improve the performance.
7. More training dataset would definitely help in improving the performance.

Conclusion and Future Works

We saw researches that are going on around the world on different Sign Languages, but due to lack of proper dataset on Indian Sign Language, Machine Translation of ISL is lagging behind in every aspect. With more than 63 Million people with disabling hearing condition, it is of utmost importance that we help the deaf community who are facing difficulties even for obtaining basic education and amenities.

In this work, we introduced ISI-ISL-DDNEWS-2020T, a first of its kind Indian Sign Language Translation dataset which we plan to publicly release for research purposes in coming months. We have described the complete process that we used to generate the dataset making it easily reproducible and extendable. This should allow everyone interested to contribute to the dataset. Resulting in a better benchmark dataset which could stand among the leading SLT datasets.

We also have introduced an attention based encoder-decoder model along with the parameters for its training. This translation model achieved a baseline performance to guide future research work in ISL to text translation. We have listed out possible improvements that could be made in the model in future. Our results show that it is possible to train a model in an end to end fashion to translate Indian sign language videos without using the intermediate gloss representation. This work differs from all the previous works as it achieves translation directly from the videos instead of relying on gloss or performing any CSLR. Our model never uses any human pose information and is hence superior to previous works which relied heavily on human poses for CSLR.

In future, we would like to perform more experiments with various tweaks suggested in Section (3.4) in the model architecture. We would expand the ISL dataset and improve on its annotations.

Lastly, the domain of SLT is nowhere near the performances achieved by Machine Translation models in spoken language translations and there exists a huge potential for future research.

Bibliography

[1] Saurabh. Varshney. Deafness in India. *Indian Journal of Otology*, 22(2):73–76, 2016.

[2] Javed Abidi. 99pc of people with hearing disabilities in india are not matriculates, December 2016.

[3] William C Stokoe. Sign language structure. *Annual Review of Anthropology*, pages 365–390, 1980.

[4] Helen Cooper, Brian Holt, and Richard Bowden. Sign language recognition. In *Visual Analysis of Humans*, pages 539–562. Springer, 2011.

[5] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. Neural sign language translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, 2018.

[6] Kayo Yin. Sign language translation with transformers. *arXiv preprint arXiv:2004.00588*, 2020.

[7] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10023–10033, 2020.

[8] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280, June 1989.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[10] Juan Antonio Pérez-Ortiz, Felix A Gers, Douglas Eck, and Jürgen Schmidhuber. Kalman filters improve lstm network performance in problems unsolvable by traditional recurrent nets. *Neural Networks*, 16(2):241–250, 2003.

[11] Felix A. Gers, Jürgen A. Schmidhuber, and Fred A. Cummins. Learning to forget: Continual prediction with lstm. *Neural Comput.*, 12(10):2451–2471, October 2000.

[12] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[13] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[15] Nal Kalchbrenner and P Blunsom. Recurrent continuous translation models. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 3:1700–1709, 01 2013.

[16] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[17] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics.

[18] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[19] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation, 2019.

[20] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. An effective approach to unsupervised machine translation, 2019.

[21] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale, 2018.

[22] Guangxiang Zhao, Xu Sun, Jingjing Xu, Zhiyuan Zhang, and Liangchen Luo. Muse: Parallel multi-scale attention for sequence to sequence learning. *arXiv preprint arXiv:1911.09483*, 2019.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, USA, 2017. Curran Associates Inc.

[24] Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*, 2020.

[25] Sho Takase and Naoaki Okazaki. Positional encoding to control output sequence length. *arXiv preprint arXiv:1904.07418*, 2019.

[26] Alexios Gidiotis and Grigorios Tsoumakas. A divide-and-conquer approach to the summarization of long documents, 2020.

[27] Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher Pal. On extractive and abstractive neural document summarization with transformer language models, 2019.

[28] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *AAAI*, pages 13041–13049, 2020.

[29] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[30] Dzmitry Bahdanau, Kyunghyun Cho, and Y Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09 2014.

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

[32] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA, 2006. Association for Computing Machinery.

[33] George Caridakis, Stylianos Asteriadis, and Kostas Karpouzis. Non-manual cues in automatic sign language recognition. *Personal and ubiquitous computing*, 18(1):37–46, 2014.

[34] Christoph Schmidt, Oscar Koller, Hermann Ney, Thomas Hoyoux, and Justus Piater. Using viseme recognition to improve a sign language translation system. In *International Workshop on Spoken Language Translation*, pages 197–203, 2013.

[35] Daniel Stein, Christoph Schmidt, and Hermann Ney. Analysis, preparation, and optimization of statistical sign language machine translation. *Machine Translation*, 26(4):325–357, 2012.

[36] Sara Morrissey, Harold Somers, Robert Smith, Shane Gilchrist, and Sandipan Dandapat. Building a sign language corpus for use in machine translation. *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (CSLT 2010)*, 01 2010.

[37] O Crasborn, R Bank, I Zwitserlood, E Van Der Kooij, A De Meijer, A Sáfár, and E Ormel. Annotation conventions for the corpus ngt, version 3. *Centre for Language Studies & Department of Linguistics, Radboud University Nijmegen*, 2015.

[38] Trevor Johnston and L De Beuzeville. Auslan corpus annotation guidelines. *Centre for Language Sciences, Department of Linguistics, Macquarie University*, 2014.

[39] Onno A Crasborn, Johanna Mesch, Dafydd Waters, Annika Nonhebel, Els Van der Kooij, Bencie Woll, and Brita Bergman. Sharing sign language data online: Experiences from the echo project. *International journal of corpus linguistics*, 12(4):535–562, 2007.

[40] Jens Forster, Christoph Schmidt, Oscar Koller, Martin Bellgardt, and Hermann Ney. Extensions of the sign language recognition and translation corpus rwth-phoenix-weather. In *LREC*, pages 1911–1916, 2014.

[41] Trevor Johnston et al. Corpus linguistics and signed languages: no lemmata, no corpus. In *3rd Workshop on the Representation and Processing of Sign Languages*, 2008.

[42] Adam Schembri and OA Crasborn. Issues in creating annotation standards for sign language description. *Proceedings of LREC 2010*, pages 212–216, 2010.

[43] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, 2015.

[44] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. Subunets: End-to-end hand shape and continuous sign language recognition. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3075–3084. IEEE, 2017.

[45] Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, Ashwin Thangali, Haijing Wang, and Quan Yuan. Large lexicon project: American sign language video corpus and sign language indexing/retrieval algorithms. In *Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (CSLT)*, volume 2, 2010.

[46] Eleni Efthimiou, Stavroula-Evita Fotinea, Thomas Hanke, John Glauert, Richard Bowden, Annelies Braffort, Christophe Collet, Petros Maragos, and François Lefebvre-Albaret. Sign language technologies and resources of the dicta-sign project. In *Proc. of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon. LREC*, pages 23–27, 2012.

[47] Oscar Koller, Cihan 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*, 2019.

[48] Shinichi Tamura and Shingo Kawasaki. Recognition of sign language motion images. *Pattern Recognition*, 21(4):343–353, 1988.

[49] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on pattern analysis and machine intelligence*, 20(12):1371–1375, 1998.

[50] Kearsy Cormier, Neil Fox, Bencie Woll, Andrew Zisserman, Necati Cihan Camgöz, and Richard Bowden. Extol: Automatic recognition of british sign language using the bsl corpus. In *Proceedings of 6th Workshop on Sign Language Translation and Avatar Technology (SLTAT) 2019*. Universitat Hamburg, 2019.

[51] Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, et al. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pages 16–31, 2019.

[52] Adam Schembri, Jordan Fenlon, Ramas Rentelis, Sally Reynolds, and Kearsy Cormier. Building the british sign language corpus. *Language Documentation & Conservation*, 7:136–154, 2013.

[53] Thomas Hanke, Lutz König, Sven Wagner, and Silke Matthes. Dgs corpus & dicta-sign: The hamburg studio setup. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (CSLT 2010), Valletta, Malta*, pages 106–110, 2010.

[54] Helen Cooper and Richard 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. IEEE, 2009.

[55] Fang Yin, Xiujuan Chai, and Xilin Chen. Iterative reference driven metric learning for signer independent isolated sign language recognition. In *European Conference on Computer Vision*, pages 434–450. Springer, 2016.

[56] Necati Cihan Camgöz, Ahmet Alp Kındıroğlu, Serpil Karabüklü, Meltem Kelepir, Ayşe Sumru Özsoy, and Lale Akarun. Bosphorussign: A turkish sign language recognition corpus in health and finance domains. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1383–1388, 2016.

[57] Hanjie Wang, Xiujuan Chai, and Xilin Chen. Sparse observation (so) alignment for sign language recognition. *Neurocomputing*, 175:674–685, 2016.

[58] Sarah Ebling, Necati Cihan Camgöz, Penny Boyes Braem, Katja Tissi, Sandra Sidler-Miserez, Stephanie Stoll, Simon Hadfield, Tobias Haug, Richard Bowden, Sandrine Tornay, et al. Smile swiss german sign language dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[59] Patrick Buehler, Andrew Zisserman, and Mark Everingham. Learning sign language by watching tv (using weakly aligned subtitles). In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2961–2968. IEEE, 2009.

[60] Tomas Pfister, James Charles, and Andrew Zisserman. Large-scale learning of sign language by watching tv (using co-occurrences). In *BMVC*, 2013.

[61] James Charles, Tomas Pfister, Mark Everingham, and Andrew Zisserman. Automatic and efficient human pose estimation for sign language videos. *International Journal of Computer Vision*, 110(1):70–90, 2014.

[62] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[63] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[64] Jens Forster, Christoph Schmidt, Thomas Hoyoux, Oscar Koller, Uwe Zelle, Justus H Piater, and Hermann Ney. Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus. In *LREC*, volume 9, pages 3785–3789, 2012.

[65] Xiujuan Chai, Guang Li, Yushun Lin, Zhihao Xu, Yili Tang, Xilin Chen, and Ming Zhou. Sign language recognition and translation with kinect. In *IEEE Conf. on AFGR*, volume 655, page 4, 2013.

[66] Hamid Vaezi Joze and Oscar Koller. Ms-asl: A large-scale data set and benchmark for understanding american sign language. In *The British Machine Vision Conference (BMVC)*, September 2019.

[67] Hanjie Wang, Xiujuan Chai, Xiaopeng Hong, Guoying Zhao, and Xilin Chen. Isolated sign language recognition with grassmann covariance matrices. *ACM Transactions on Accessible Computing (TACCESS)*, 8(4):1–21, 2016.

[68] Necati Cihan Camgöz, Ahmet Alp Kındıroğlu, and Lale Akarun. Sign language recognition for assisting the deaf in hospitals. In *International Workshop on Human Behavior Understanding*, pages 89–101. Springer, 2016.

[69] Muhammed Süzgün, Hilal Özdemir, Necati Camgöz, Ahmet Kındıroğlu, Doğaç Başaran, Cengiz Togay, and Lale Akarun. Hospisign: an interactive sign language platform for hearing impaired. *Journal of Naval Sciences and Engineering*, 11(3):75–92, 2015.

[70] Sandrine Tornay, Marzieh Razavi, Necati Cihan Camgoz, Richard Bowden, and Mathew Magimai Doss. Hmm-based approaches to model multichannel information in sign language inspired from articulatory features-based speech processing. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2817–2821. IEEE, 2019.

[71] Helen Cooper, Eng-Jon Ong, Nicolas Pugeault, and Richard Bowden. Sign language recognition using sub-units. *The Journal of Machine Learning Research*, 13(1):2205–2231, 2012.

[72] Eng-Jon Ong, Helen Cooper, Nicolas Pugeault, and Richard Bowden. Sign language recognition using sequential pattern trees. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2200–2207. IEEE, 2012.

[73] Christian Vogler and Dimitris Metaxas. Parallel hidden markov models for american sign language recognition. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 116–122. IEEE, 1999.

[74] Hee-Deok Yang, Stan Sclaroff, and Seong-Whan Lee. Sign language spotting with a threshold model based on conditional random fields. *IEEE transactions on pattern analysis and machine intelligence*, 31(7):1264–1277, 2008.

[75] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[76] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[77] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

[78] Oscar Koller, Hermann Ney, and Richard Bowden. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3793–3802, 2016.

[79] Oscar Koller, O Zargaran, Hermann Ney, and Richard Bowden. Deep sign: hybrid cnn-hmm for continuous sign language recognition. In *Proceedings of the British Machine Vision Conference 2016*, 2016.

[80] Oscar Koller, Hermann Ney, and Richard Bowden. Deep learning of mouth shapes for sign language. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 85–91, 2015.

[81] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[82] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4297–4305, 2017.

[83] 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.

[84] 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.

[85] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[86] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008.

[87] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling, 2016.

[88] Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. Lipnet: End-to-end sentence-level lipreading, 2016.

[89] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[90] Runpeng Cui, Hu Liu, and Changshui Zhang. A deep neural framework for continuous sign language recognition by iterative training. *IEEE Transactions on Multimedia*, 21(7):1880–1891, 2019.

[91] Jan Bungeroth and Hermann Ney. Statistical sign language translation. In *Workshop on representation and processing of sign languages, LREC*, volume 4, pages 105–108. Citeseer, 2004.

[92] Sara Morrissey. *Data-driven machine translation for sign languages*. PhD thesis, Dublin City University, 2008.

[93] Sarah Ebling. *Automatic Translation from German to Synthesized Swiss German Sign Language*. PhD thesis, University of Zurich, 2016.

[94] Graham Neubig. Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv preprint arXiv:1703.01619*, 2017.

[95] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[96] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*, 2017.

[97] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[98] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6447–6456, 2017.

[99] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[100] 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.

[101] Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney. Language modeling with deep transformers. *arXiv preprint arXiv:1905.04226*, 2019.

[102] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[103] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language se-

quences. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2019, page 6558. NIH Public Access, 2019.

[104] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[105] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019.

[106] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[107] Doordarshan. DD News for Hearing Impaired. https://www.youtube.com/playlist?list=PLxx0m3vtiqMYFSgSnR9IL3GyoYCTwNBy4, 2018.

[108] Open Source Community. Youtube downloader. https://ytdl-org.github.io/youtube-dl/, 2019.

[109] Open Source Community. Ffmpeg. https://ffmpeg.org/, 2020.

[110] YouTube A subsidiary of parent company Google. Youtube Auto-Subtitles. https://youtube.org/.

[111] Google. Google Transcribe. https://google.com/.

[112] Amazon Web Services. AWS Transcribe. https://amazon.com/.

[113] D. Britz, A. Goldie, T. Luong, and Q. Le. Massive Exploration of Neural Machine Translation Architectures. *ArXiv e-prints*, March 2017.

[114] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[115] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[116] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[117] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.