Sign Language Translation with Multiple models

*DS605 Course project/Article review; Professor: Rajesh Kumar Mundotiya

1st Ampolu Sahithi
B.Tech DSAI
IIT BHILAI
ampolusahithi@iitbhilai.ac.in

2nd Lahari Sreeja
B.Tech EE
IIT BHILAI
tallapakalahari@iitbhilai.ac.in

***Abstract*—When signing a sentence from a spoken language, one must first determine the necessary signs and their order. This project primarily centers around translating spoken language sentences into glosses. Due to limited resources, we treat the text-to-gloss translation task as a low-resource machine translation challenge. We fine-tune the hyperparameters and add copy attention to enhance the translation quality of the current models. At the end we also propose a translation technique to generate sign language glosses for different languages.**

***Index Terms*—Gloss, Attention, Neural Machine Translation, Transformers**

I. Introduction

Sign language translation involves converting expressions between sign language and spoken language. In this context, glosses, representing words associated with signs, serve as a crucial intermediary step. This paper explores the challenge of translating text to gloss in sign language, with a focus on the scarcity of resources. Sign language, being a low-resource language, presents unique obstacles, making glosses essential for translation. Many translation tasks leverage glosses as intermediaries when translating sign images to spoken language. This project tries to increase the efficiency of existing sign language models.

Even though ISL exists the text and glosses both are in English, so given a Hindi sentence it would be harder for the signer to sign directly .We wanted to give English glosses for Hindi text and even translate them back to Hindi, so it would be helpful for the signers to get the signs on web. In the end we also introduce an approach to generate glosses for different languages.

II. Problem Statement

The selected problem statement that we aim to address involves the conversion of spoken language text into corresponding glosses, facilitating the process of signing a given sentence for individuals proficient in sign language.

Next part, includes generating glosses for Hindi text.

III. Earlier Works

The early-stage of text-to-gloss translation systems were built using Statistical Machine Translation (SMT; San-Segundo et al., 2012; López-Ludeña et al., 2014), in an attempt to translate spoken language into a signing 3D avatar using SL glosses as intermediate. Later with the advancement of NMT, more systems based of the RNN, LSTM and transformers have emerged. The paper introducing STMC Transformer (Yin, K., Read, J. 2020) makes use of a transformer architecture with different type of embedding initializations for the translating task.Another research (Xuan Zhang, Kevin Duh) focuses on showing the importance of hyperparameter search and back-translation. The paper Neural Machine Translation Methods (Dele-Zhu et al. 2023) also makes use of different techniques Data Augumentation, Semi-Supervised NMT, Transfer Learning etc. along with hyperparameter tuning to produce efficient glosses from text.

https://github.com/Ai-deen/DS605-Sign-Lanaguage-Translation-Project

IV. Novelty

There are two specific tasks we would like to address here:

- Increasing the efficiency of existing models using hyperparameter search.
- Propose an approach to produce hindi laguage glosses to hindi lagugae text which highly relies on the English-ASL model, since there is no specific sign language proposed in Hindi and we want to learn the sign translations of English Model itself.

V. Experimental Architecture

A. Models

We started experimenting on different architectures. Below three listed are the different architectures we used:

Neural machine translation with attention

This experimental architecture is derived from a tutorial on training a sequence-to-sequence (seq2seq) model for Spanish-to-English translation. The process involves Unicode normalization, standardization, and vectorization of the data before feeding it into training. The model consists of an encoder, attention layer, and decoder, each serving a specific purpose in the translation task.

Encoder:

Embedding Layer: Utilizes an embedding layer to obtain embedding vectors for each token.

Bidirectional GRU (Gated Recurrent Unit): Processes the embeddings bidirectionally, capturing contextual information

effectively.

Attention Layer:

Function: Allows the decoder to access information extracted by the encoder by computing a weighted average across the context sequence.

Weight Calculation: Calculates attention weights by considering both context and "query" vectors.

Initialization: The attention weights are initially close to 1/sequence_length, and the model learns to adjust them during training.

Decoder:

Unidirectional RNN: Processes the target sequence, keeping track of generated content.

Attention Mechanism: Uses RNN output as the "query" for the attention layer, enhancing the model's focus on relevant information from the encoder.

Token Prediction: Predicts the next token at each location in the output sequence.

Training vs. Inference: During training, the model predicts the next word at each location. During inference, the model produces one word at a time, with predictions fed back into the model.

ONMT translation model with RNN encoder/decoder

This architecture is obtained from tutorial in OpenNMT-py for Language Translation.The experimental architecture for training a sequence-to-sequence (seq2seq) model for machine translation involves three main components: Data Preparation, Model Training, and Translation.

Data Preparation:

Yaml file: Yaml file is used to specify the configuration settings for various tasks such as vocabulary,tokenisation,training, evaluation, translation,etc.

Training:

Encoder: Bidirectional LSTM with 500 hidden units and there are two layers of LSTM in the encoder.

Global Attention: Global attention is implemented with linear layers for input and output transformations.

Decoder: Input-feeding RNN decoder with similar embedding and dropout layers as the encoder.It has stacked LSTM with 500 hidden units in each layer.Global attention mechanism for capturing dependencies between source and target sequences.

Generator: Linear layer mapping the decoder's hidden states to the target vocabulary.

Translation:

Beam search: The model explores and expands a beam of top-scoring hypotheses at each decoding step, ultimately selecting the translation with the highest overall score within a specified beam size.

ONMT translation model with Transformer model

This experimental architecture is derived from a transformer model architecture discussed in the reference paper Attention is all you need. The above architecture is implemented in ONMT model and the hyper parameters are tweaked to achieve better accuracy.

Encoder:

Embeddings Layer: Utilizes embedding vectors for tokens, with positional encoding.

Transformer Layers: Two transformer layers, each containing: Multi-Headed Self-Attention Mechanism. Positionwise Feedforward Layer. Layer Normalization.

Decoder:

Embeddings Layer: Similar to the encoder, with embedding vectors and positional encoding.

Transformer Layers: Two transformer layers, each containing: Multi-Headed Self-Attention Mechanism and Multi-Headed Context Attention Mechanism. Also we have positionwise Feedforward Layer and Layer Normalization.

Generator:

Linear and Cast Layer: Transforms the decoder's hidden states to match the target vocabulary size and performs casting operations.

LogSoftmax Layer: Applies log softmax activation to produce probability distribution.

Mutli-head attention enabled with shared vocabulary and copy attention. Two layers, compared to six in most spoken language translation, is empirically shown to be optimal in (Kayo Yin,Jesse Read), likely because our datasets are limited in size.

Translation: For translation, we train a two-layered Transformer to maximize the log-likelihood

$$\sum_{(x_i,y_i)\in D} \log P(y_i|x_i,\theta)$$

where D contains text-gloss pairs.

We refer to the original Transformer paper (Vaswani et al., 2017) for more architecture details.

VI. Experimental Settings

A. Datasets

PHOENIX-Weather 2014T (Camgoz et al., 2018)

This dataset is extracted from weather forecast airings of the German TV station PHOENIX. This dataset consists of a parallel corpus of German sign language videos from 9 different signers, gloss-level annotations with a vocabulary of 1,066 different signs and translations into German spoken language with a vocabulary of 2,887 different words. It contains 7,096 training pairs, 519 development, and 642 test pairs.

ASLG-PC12 (Othman and Jemni, 2012)

This dataset is constructed from English data of Project Gutenberg that has been transformed into ASL glosses following a rule-based approach. This corpus with 87,709 training pairs allows us to evaluate Transformers on a larger dataset, where deep learning models usually require lots of data. It also allows us to compare performance across different sign languages. However, the data is limited since it does not contain sign language videos and is less complex due to being

created semi-automatically. We use this dataset to train En-ASL translation model.

HindiEnCorp

A parallel corpus of Hindi and English, and HindMonoCorp, a monolingual corpus of Hindi in their release version 0.5. Both corpora were collected from web sources and preprocessed primarily for the training of statistical machine translation systems. HindEnCorp consists of 274k parallel sentences (3.9 million Hindi and 3.8 million English tokens). HindMonoCorp amounts to 787 million tokens in 44 million sentences.This is used for training Hindi-English translation model.

B. Framework

The ONMT framework which is written in PyTorch is used to preprocess and train the translation model.

OpenNMT

The OpenNMT (ONMT) framework is a powerful and versatile open-source toolkit designed for neural machine translation (NMT) tasks. Developed to facilitate the implementation and experimentation of state-of-the-art NMT models, ONMT provides a flexible and modular architecture that supports various model configurations and training strategies. With a focus on usability, the framework offers tools for preprocessing, training, and evaluation, streamlining the end-to-end process of building and deploying NMT models.

C. Methods

1) Increasing the efficiency: The OpenNMT (ONMT) framework builds a transformer model with multi-head attention, utilizing 8 heads. Initially, the input corpus undergoes tokenization, and subsequently, the vocabulary is constructed for further training.

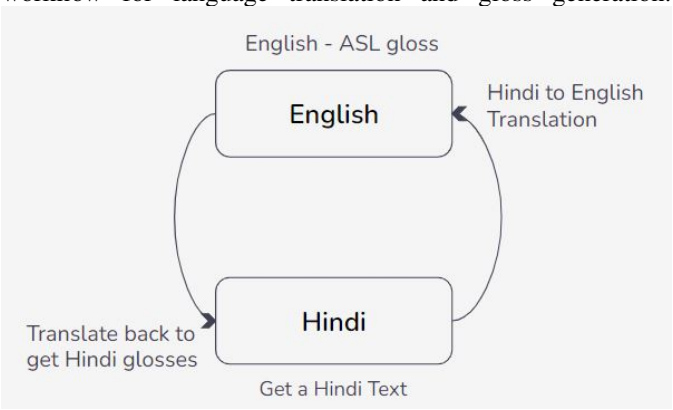
In the context of sign language, the source and target languages are identical, differing only in arrangement with minimal replacements. To address this, we experimented with enabling a shared vocabulary in the transformer equipped with a copy mechanism. This was done while varying the batch size and learning rate.

Additionally, modifications to the beam size during text production had a noticeable impact. Specifically, a beam size of 3 resulted in improved translations.

As we explore these adjustments, it becomes evident that the beam size plays a significant role in influencing translation quality, with a beam size of 3 demonstrating superior results.

2) Producing glosses for a Hindi Sentence: Initially, a Hindi-English (Hin-En) model is trained to translate Hindi text into English. Following this training, the tuned model for English ASL is employed to obtain glosses in English.

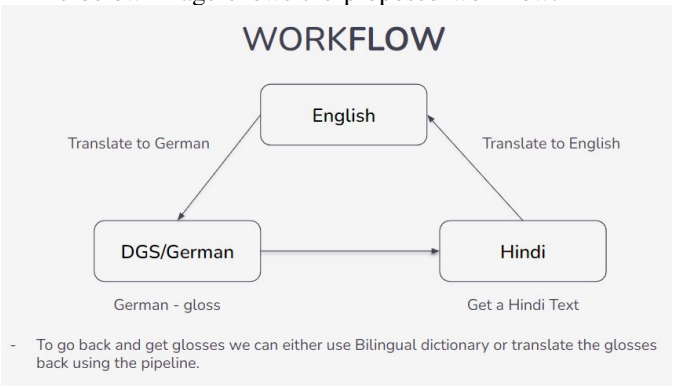
This sequential approach involves the two distinct phases: the Hin-En model training and the subsequent application of the English to ASL model using a dataset of 1000 senetences for each language due to computational constraints. Both the translation model and the English to ASL model utilise tuned ONMT Encoder/Decoder model architecture to train, offering a comprehensive workflow for language translation and gloss generation.



D. Proposed Approach

To improve the generation of sign language glosses, we suggest a unique approach that employs English as an intermediary language. While a large English corpus exists, it is synthetic, lacking the natural nuances found in the German Phoenix dataset. Unfortunately, aligning a Hindi to German dataset proves challenging due to domain differences. We don't have a proper dataset to directly translate from Hindi to German. Therefore, we propose using English as a bridge, leveraging available datasets for Hindi to English translation. This strategy aims to preserve cultural intricacies, offering a promising solution for more authentic and contextually rich sign language gloss generation. We tried to implement this model but due to domain differences in datasets we were only able to partially implement this approach

The below image shows the proposed workflow:



VII. Results and Analysis

The BLEU score is employed as the accuracy metric throughout this process. We explored NMT with Attention model to understand how Neural Machine Translation works. While yielding average results in comparison to other models, it nonetheless provided valuable insights, guiding the development of subsequent models.

The subsequent ONMT Encoder/Decoder model represents an improvement, even with its basic architecture. Featuring LSTM blocks and Global Attention, it outperformed the NMT with Attention model, aligning with results from various

referenced papers.

The achievement lies in the ONMT Transformer model, surpassing the models cited in reference papers.The English translations had an increment of 0.1 in the BLEU-3 score compared to the results in the paper. This increment is due to employment of techniques like shared vocabulary and copy attention, tailored to Sign language glosses, where both glosses and text share similar vocabulary and tokens. These techniques were dispersed across various papers we considered, and a comprehensive attempt to combine all these strategies within was undertaken by us.

Model 1:

BLEU SCORES - NMT with Attention model				
Translation	BLEU-1	BLEU-2	BLEU-3	BLEU-4
German - German Gloss	0.536	0.345	0.233	0.159
English - English Gloss	0.909	0.867	0.826	0.790

Model 2:

BLEU SCORES - ONMT Encoder/Decoder model				
Translation	BLEU-1	BLEU-2	BLEU-3	BLEU-4
German - German Gloss	0.504	0.322	0.219	0.155
English - English Gloss	0.941	0.922	0.905	0.887
Hindi - English Translation	0.384	0.266	0.197	0.151

Model 3:

BLEU SCORES - ONMT Transformer model				
Translation	BLEU-1	BLEU-2	BLEU-3	BLEU-4
German - German Gloss	0.561	0.377	0.267	0.193
English - English Gloss	0.985	0.978	0.972	0.966

VIII. Conclusion

In conclusion, our article review addresses the challenge of translating spoken language into sign language glosses, treating it as a low-resource machine translation task. Through iterative model development, we enhanced translation quality by fine-tuning hyperparameters and incorporating copy attention. Our journey, guided by the BLEU score, progressed from the basic ONMT Encoder/Decoder model to the advanced ONMT Transformer model, outperforming both our earlier models and referenced literature. The results achieved is attributed to strategic techniques like shared vocabulary and copy attention, tailored to sign language glosses. While challenges persist in fully integrating these strategies, our work contributes valuable insights to sign language translation research.In addition, we undertook a supplementary initiative by leveraging our English-to-English gloss model to create a small Hindi-to-Hindi gloss dataset. This modest effort contributes to the establishment of a Hindi gloss corpus, broadening the scope of our research impact.

References

[1] Neural Machine Translation Methods for Translating Text to Sign Language Glosses, Dele Zhu , Vera Czehmann and Eleftherios Avramidis

[2] Approaching Sign Language Gloss Translation as a Low-Resource Machine Translation Task, Xuan Zhang Kevin Duh

[3] English To Indian Sign Language: Rule-Based Translation System Along With Multi-Word Expressions and Synonym Substitution, Abhigyan Ghosh Radhika Mamidi

[4] Better Sign Language Translation with STMC-Transformer , Kayo Yin Jesse Read

[5] Rubén San-Segundo, Juan Montero, Ricardo Cordoba, V. Sama, Fernando Fernández-Martínez, Luis D'Haro, Verónica López-Ludeña, D. Sánchez, and A. García. 2012. Design, development and field evaluation of a spanish into sign language translation system. Pattern Analysis and Applications, 15.

[6] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. ArXiv, abs/1706.03762.