Sign Language Translation

Riyanshi Goyal(12041240), Vanisha Agrawal (12041670), Vidhi Mittal(12041730)
IIT Bhilai

I. Introduction

Sign language is a visual form of communication used by deaf and hard of hearing people. It uses hand gestures, body movements and facial expression to communicate. One of the most important challenges of sign language is that it is a multi-channeled language and a non-written language. For converting the spoken language that is available in the form of text to sign language that is mostly available in the form of videos or images, an intermediate representation known as glosses are used. In glosses, signs are labeled with words from the corresponding spoken language. They prove to be valuable in the current state of Sign Language Translation, particularly for applications like interpreter training and education. Additionally, glosses are the only Sign language representation with sufficiently large parallel corpora for MT training, offering insights for future treatment of more suitable representations. Our focus lies on the text-to-gloss translation aspect, a crucial step in generating sign animations, and despite previous improvements, significant breakthroughs in this domain are yet to be achieved. We focus on improving text to gloss translation by the following contributions:

- Different kinds of data pre-processing and data augmentation techniques are used to tackle the scarcity of parallel corpora.
- Incorporate syntax information in the input sentences.

II. Problem Statement

Our problem statement involves converting the spoken language texts into glosses, which act as an intermediate step in building MT systems for translating from spoken language to SLs. This would further help overcome the linguistic barrier between deaf or hard-of-hearing communities and the hearing population.

SLs are Low-Resource Languages regarding MT, since there is little parallel data. So here, we explore the effects of different methods on text-to-gloss translation to increase its performance.

III. Earlier works

The early stages of text-to-gloss translation systems utilized Statistical Machine Translation (SMT) with the goal of translating spoken language into sign language glosses and representing them through 3D avatars. While initial evaluations showed positive results based on limited data and automatic metrics, deaf users provided contrasting assessments. Recent advancements in Neural Machine Translation (NMT) have led to more promising systems, employing RNNs or integrated into end-to-end transformer systems. Notably, [1] Li et al. (2021) introduced a transformer architecture with an editing agent, [2] Walsh et al. (2022) explored the impact of tokenization techniques and embeddings on translation performance. In contrast to these methods, the present work proposed syntax-aware transformers. [3] Egea Gómez et al. (2021) proposed model leverages lexical dependency information to learn grammatical rules for translating from text to glosses. The technique of paraphrasing the text is also proposed to give more information about the input text to the encoder.

IV. Novelty

A. Syntax Aware

The uniqueness lies in the augmentation of input embeddings to the Encoder by incorporating lexical dependency information. The motivation for this augmentation stems from the observation, as indicated in Table 1, that gloss production from spoken text primarily involves operations like word permutations, stemming, and deletions. These transformations are often influenced by the syntactical functions of words. For example, in many instances, determiners are consistently removed to generate glosses. The belief underlying this model is that word dependency tags can be valuable in modeling the syntactic rules that are inherent in the process of gloss production. By including information about the dependencies between words, the model aims to capture the relationships and structural aspects of the language that contribute to the creation of sign language glosses. In essence, the incorporation of word dependency tags is seen as a way to enhance the model's understanding of the syntactic nuances involved in the translation from spoken language to sign language glosses.

Table 1: T2G production examples

Spoken	Später breiten sich aber nebel oder hochnebelfelder aus (EN) Later, however, fog or high-fog fields are widening
Gloss	ABER IM-VERLAUF NEBEL HOCH NEBEL IX[4] (EN) BUT IN-COURSE FOG HIGH FOG IX

B. Paraphrasing Input Sentences

Paraphrasing is another data augmentation technique where the input sentences are paraphrased. It means expressing the same or similar ideas using different words or sentence structures while retaining the original meaning. It involves finding alternative ways to convey the information to improve clarity, style, or emphasis. The intuition behind this was that

we thought paraphrased sentences when fed as input along with the original sentences to the model will provide more information to the model as to how same meaning can be expressed in different ways. This adds to the robustness of the model. This helps the model generalize better and handle variations in input more effectively, also makes the model performs better on unseen data.

V. Experimental architecture

Our experimentation starts with data pre-processing and setting the baseline. Following this, we explore different methods, such as data augmentation and using a syntax-aware transformer, evaluating their effectiveness through automatic metrics.

A. Baseline

For our baseline model [4], we use Marian (NMT) framework that provides various models for machine translation tasks. It typically consists of an encoder-decoder model based on transformer architecture, similar to the original model introduced in the "Attention is All You Need" paper [5].

The architecture includes:

1) Encoder: This part of the model processes the input sequence. It employs a stack of transformer encoder layers that capture the contextual information of the input tokens. Each encoder layer contains self-attention mechanisms and feed-forward neural networks.
2) Decoder: Responsible for generating the output sequence based on the encoder's representations. It comprises transformer decoder layers that attend to the encoded information to produce the target sequence. The decoder layers also involve self-attention and cross-attention mechanisms.
3) Attention Mechanism: Marian utilizes attention mechanisms, particularly self-attention and cross-attention, allowing the model to weigh different parts of the input sequence while decoding. This mechanism aids in understanding the relevant context during translation.
4) Transformer Layers: Both the encoder and decoder consist of multiple transformer layers, each containing multi-head self-attention and position-wise feed-forward neural networks. These layers contribute to capturing and processing the input information effectively.

B. Data Augmentation

Data augmentation is a common technique used for low resource conditions by adding synthetically generated data from various sources. Here, we focus on the following methods:

Back-translation involves acquiring new source-side data by translating a monolingual dataset in the target language using a target-to-source model. The created source sentences are then paired with their corresponding target side data to form a synthetic parallel dataset. In our case, due to the absence of a monolingual glosses dataset, we utilize a gloss-to-text system to convert the already present target-side glosses from the training data into spoken language text. This generates a modified corpus version which is then augmented to the original training data.

Tagging A special token is added at the beginning of each synthetic source sentence in the training data to inform the model Which sentences are original and which are synthetic, as the augmented data may be of lesser quality.

Paraphrasing As mentioned above we are changing the structure of the original sentence while retaining the same meaning. Since our input dataset is Phoenix which consists if German language, to paraphrase them we first convert the German sentences to the corresponding English sentences using Google Translate Api, then paraphrase the English sentences using a transformer, then finally convert the paraphrased English sentences back to German using google translate. For paraphrasing the english sentences, we use a fine tuned version of T5 Transformer. T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format.

C. Syntax aware T2G model

The described model is a specialized Encoder-Decoder architecture designed for Text2Gloss translation. The key innovation lies in augmenting the input embeddings to the Encoder by incorporating lexical dependency information. This augmentation is motivated by the observation that gloss production from spoken text often involves word permutations, stemming, and deletions, with transformations dependent on the syntactical functions of words. For instance, determiners are consistently removed to generate glosses. The neural architecture of the model is based on multi-attention layers, specifically three such layers with four attention heads each for both the Encoder and the Decoder. The internal dimensions for the fully connected network are set to 1024, and the output units are set to 512. This architecture has shown excellent results in modeling long input sequences. The Encoder transforms inputs into latent vectors, and the Decoder generates word probabilities based on these encoded latent representations. To enhance the discriminative power of the embeddings inputted to the Encoder, the model aggregates syntactic information with word embeddings. Unlike a previous approach, which added encoders to manage injected features, this model integrates an additional table containing vector embeddings for syntactic tags. The sum of word and syntax embeddings produces an aggregated embedding that serves as input to the Encoder. Both tables have a vector length of 512. The input text is processed using subword tokenization, and dependency tags are generated using the spaCy library's model, specifically the core news model. The incorporated dependency tags originate from the TIGER dependency bank, designed for categorizing words in German. A Sentence Piece model is trained for tokenization with a vocabulary size of 3000, keeping some tokens for control.

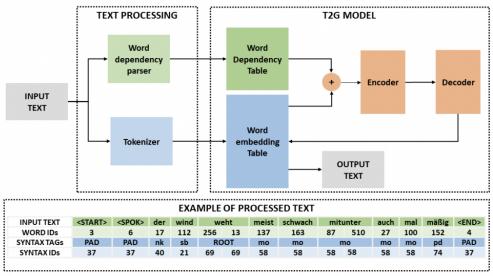

Figure 1: Syntax-Aware Text2Gloss model

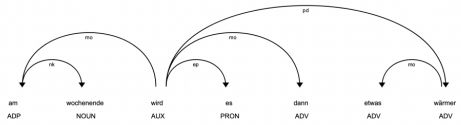

Figure 2: Lexical dependency tree diagram of the sentence "On the weekend it gets a little warmer". Key to tags: ep = expletive *es*, mo = modifier, nk = noun kernel element, pd = predicate

VI. Experimental settings

A. Dataset

RWTH-PHOENIX-Weather2014T (Camgoz et al., 2018), abbreviated as PHOENIX, is a parallel corpus of SL containing weather forecasts. The original language was German, translated into DGS by professional interpreters and then annotated with DGS glosses. We use the provided split of parallel train-, dev- and test-set with respective sizes of 7,096, 519 and 642 sentences.

B. Data preprocessing

For the data preprocessing, at source side, we perform tokenization, lowercase conversion and lemmatization and set them with lowercased glosses. We then apply Byte Pair Encoding (BPE; Sennrich et al., 2016b) to decompose the words and build vocabulary.

C. Baseline

1) Model Settings:
   - Encoder and Decoder architecture: We take the advice of some paper that indicates in LRLMT scenarios with small data size, the model performance increases when the number of encoders/decoders are reduced compared to the original transformer architecture. So, our model has with 1 encoder and 2 decoders with LSTM cells.
   - Number of attention heads: 8
   - Activation function in the feedforward network: ReLU
   - Optimizer: Adam
   - Batch size : 16
   - Learning rate: 0.0003
   - Loss function: Cross-Entropy Mean Words. It computes the average cross-entropy loss per word in the output sequence.
   - Byte Pair Encoding (BPE) with 32,000 merge operations is used for subword tokenization
   - Validation and testing: the generated translations for the validation and test sets are evaluated using SacreBLEU for BLEU4 score

D. Back-translation

We first train simple gloss-to-text translation models for the corpus and then they generate sets of new source sentences from the target-side glosses. A token <BT>is added to the beginning of each of the newly generated sentences. The synthetic texts are paired appropriately and then mixed with the original dataset.

E. Syntax aware T2G model

1) Model Settings: BART model is used with the following specifications
   - Optimizer: Adam
   - Learning rate: 10^5
   - Batch size: 16
   - Loss function: Cross Categorical Entropy
   - Beam size: 5

F. Paraphrasing

For paraphrasing we are using T5_Paraphrase_Paws model that has been fine-tuned on Google PAWS dataset for paraphrasing task. After rephrasing the input sentences, they along with the original sentences are fed as input to the model. Thus we have a total of 7096*2=14192 sentences, we apply the baseline model as described above on the input data. We use the same hyperparameters for the baseline model.

-

VII. Results and analysis

We have presented the results of the various methods on the SL dataset in Table 2.

System	BLEU
Baseline (paper)	22.78
Baseline (Ours)	22.37
Back + Tag (Paper)	23.62
Back + Tag (Ours)	21.82
Baseline + Syntax	24.11
BART + Syntax	51.44
Baseline + Paraphrase	24.86

TABLE I: BLEU scores of different systems

1) For PHOENIX dataset, data augmentation using back-translation in the paper has shown a significant improvement in comparison with the baseline but the same has not been reflected in our experimentation.
2) The BLEU score has increased significantly on the baseline model after adding syntax embeddings as we can see from the table. When we used the BART model with syntax embeddings we achieved the best BLEU score i.e. 51.
3) We see that paraphrasing the input sentences, improves the baselines model performance. Thus this proves to be a promising approach.

REFERENCES

[1] D. Li, C. Xu, L. Liu, Y. Zhong, R. Wang, L. Petersson, and H. Li, "Transcribing natural languages for the deaf via neural editing programs," 2021.

[2] H. Walsh, B. Saunders, and R. Bowden, "Changing the representation: Examining language representation for neural sign language production," 2022.

[3] S. Egea Gómez, E. McGill, and H. Saggion, "Syntax-aware transformers for neural machine translation: The case of text to sign gloss translation," in *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*, R. Rapp, S. Sharoff, and P. Zweigenbaum, Eds. Online (Virtual Mode): INCOMA Ltd., Sep. 2021, pp. 18–27. [Online]. Available: https://aclanthology.org/2021.bucc-1.4

[4] D. Zhu, V. Czehmann, and E. Avramidis, "Neural machine translation methods for translating text to sign language glosses," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 12 523–12 541. [Online]. Available: https://aclanthology.org/2023.acl-long.700

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.