# Sign Language Translation from Instructional Videos

Laia Tarrés[1,2] Gerard I. Gállego[1] Amanda Duarte[2] Jordi Torres[1,2] Xavier Giró-i-Nieto[3]
[1]*Universitat Politècnica de Catalunya* [2]*Barcelona Supercomputing Center* [3]*Amazon*
https://imatge-upc.github.io/slt_how2sign_wicv2023

## Abstract

*The advances in automatic sign language translation (SLT) to spoken languages have been mostly benchmarked with datasets of limited size and restricted domains. Our work advances the state of the art by providing the first baseline results on How2Sign, a large and broad dataset.*

*We train a Transformer over I3D video features, using the reduced BLEU as a reference metric for validation, instead of the widely used BLEU score. We report a result of 8.03 on the BLEU score, and publish the first open-source implementation of its kind to promote further advances.*

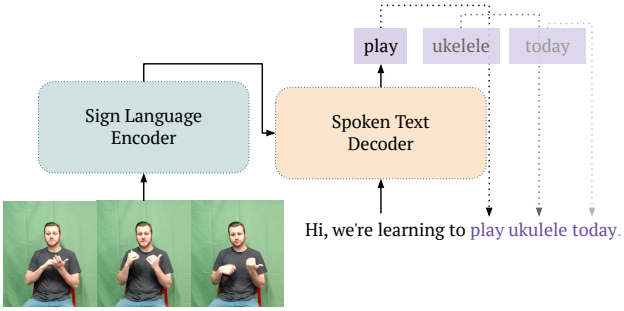


Figure 1. A basic pipeline for sign language translation.

## 1. Introduction

Sign language translation (SLT) is the task of translating continuous sign language videos into spoken language sentences. SLT is a challenging multimodal problem that requires both a precise understanding of the signer's pose and the generation of a textual transcription. The current state of the art for automatic SLT is still far away from considering the problem solved [11, 16, 18, 59, 64, 66]. Solving SLT will bring important benefits to the communication between signers and non-signers.

Recent advances in SLT have followed a trajectory similar to other computer vision and natural language processing problems: training deep neural networks on large-scale datasets. However, the availability of public sign language datasets is limited and especially reduced when considering parallel corpus of videos and their textual translations, which allow benchmarking the state of the art. Up to date, the most used dataset to assess the progress in SLT is PHOENIX-2014-T [24], with only 9.2 hours of video recordings on the restricted domain of weather forecasts.

In this work, we consider a much larger and more complex dataset, How2Sign [23], which contains almost 80 hours of instructional videos from 10 different topics. This dataset was approved by the Carnegie Mellon University Institutional Review Board. We publish the first SLT baselines for this dataset, achieving a BLEU score of 8.03.

In addition, we show that the common practice of choosing the best model checkpoint based on the BLEU metric may be misleading. This is because the model tends to capture frequent patterns, and may even provide realistic outputs completely unrelated to the input video. We empirically test how using an alternative metric [21], *reduced BLEU* (rBLEU), we can better characterize the performance of the SLT solutions and choose better checkpoints during training.

We provide open code and models of a translation system from American Sign Language (ASL) to written English, trained on How2Sign. Our implementation contains scripts to preprocess the data, train, translate, and evaluate models, which allows reproducibility and adaptation to other datasets.[1]

## 2. Related Work

Sign language video understanding has been addressed from a variety of tasks: sign language recognition (SLR) over isolated or continuous signs [1, 19, 27, 28, 41, 44, 46], sign language translation (SLT) [9, 17, 24, 32], sign language production (SLP) [49–53] or retrieval [22]. Our work focuses on sign language translation.

**Gloss supervision**. Gloss-based SLT [8, 11, 16, 66] uses an intermediate textual representation between the input video sequence and the output text. These tokens are named *glosses*. Glosses are a type of transcription of sign lan-

[1] https://github.com/imatge-upc/slt_how2sign_wicv2023

guages that must be produced by trained sign language linguists and that are available in some SLT datasets. Glosses provide supervision that helps models in their training, but their acquisition is also very time-consuming and expensive because of the scarcity of annotators.

On the other hand, gloss-free SLT [9, 12, 45, 54, 64] addresses the raw task of converting the video into text, without any intermediate gloss. Our work targets this second case, as glosses for [23] have yet to be released.

**Datasets**. Gloss-free SLT has traditionally focused on datasets that have a limited amount of data, and a restricted vocabulary [8, 32, 34, 36, 55]. Thus, the challenge to serve in real-world use cases remains.

SLT is by definition associated with *continuous* signing rather than *isolated* signing, because signers naturally concatenate one sign after the other with no resting position, similarly to how speakers concatenate spoken words. Compared to isolated SL, continuous SL videos include important effects such as prosody, which can play a crucial role in the meaning of a sentence.

Table 1 shows the current state of the art in terms of the BLEU metric for different SLT benchmarks. Reasonable scores in the range between 29 and 60 BLEU have been reported in three datasets of limited vocabulary size: KETI [33], PHOENIX-2014T [8], and CSL Daily [66].

Our work aims at the more open domain of instructional videos across 10 different topics, to set the first SLT baselines on the How2Sign [23] dataset. This dataset has been used in the past for human motion transfer, sign language retrieval [22], or topic detection [6], but never for SLT.

Our baselines are similar to those published with OpenASL [54], another dataset of similar complexity. While the scores are not directly comparable because they are different datasets, the BLEU score of 6.72 reported for OpenASL is in a similar range to the values we report in Section 5. Other works on alternative datasets of large scale obtained very poor BLEU scores: 1.0 in BOBSL [3], 0.4 in SWISSTXT-NEWS [12], 0.4 in VRT-NEWS [12], or 0.37 in SRF [57] and 0.84 in FocusNews [21] in the WMT shared task on sign language translation 2022 [43].

**Algorithms**. SLT was initially approached with rule-based systems [65] and statistical methods [7]. Since 2018, virtually all related work has basically applied the advances in deep learning to sign language translation datasets. Given that SLT can be formulated as an input sequence of video frames that is transformed into a sequence of words, it fits perfectly in the popular sequence-to-sequence (seq2seq) [56] formulation widely adopted by the Machine Translation field which employs an encoder-decoder architecture to transform the input sequence into the output one, as depicted in Figure 1.

First approaches in neural SLT used Recurrent Neural Networks (RNNs) [56] for the encoder-decoder architecture, whether with (GRUs) or LSTMs [26, 31, 33, 45].

However, RNNs present limitations in modeling long-term dependencies, an especially relevant problem when considering video input sequences captured at high frame rates. To overcome these limitations, attention-based approaches were proposed [5]. Attention mechanisms selectively focus on parts of the input during decoding, allowing to capture long-term dependencies more effectively. Camgoz et al. fed 2D CNN visual features into an RNN encoder-decoder with attention to perform translation [8].

The Transformer [58] has emerged as the preferred option for numerous Natural Language Processing (NLP) and, more recently, Computer Vision (CV) tasks. The Transformer relies on the self-attention mechanism, which allows it to process the input sequence in parallel rather than sequentially, allowing even better modeling for long-term dependencies and allowing parallelization during training. Transformers have proven to work well for the SLT task [10–12, 16, 37, 60, 64, 66]. Thus, we explore the Transformer architecture for the How2Sign dataset.

**Tokenization of sign language videos**. Sign language videos are normally tokenized to be fed into neural architectures, like the Transformer. Camgoz et al. [8, 11] used 2D CNN to extract features of frames at the gloss-level. These 2D CNN features were learned from another sign language recognition model [66].

Another commonly used features are inflated 3D convnets (I3D), developed for action-recognition [14], that can be further trained with sign language data [2, 21, 22, 37, 45, 54]. Similarly, [15] uses S3D [62] features, which have been pretrained in kinetics [30] and WLASL dataset. We focus our study on I3D video features.

Other works have also used pose estimators [13, 40] to represent the input video sequence since they contain relevant information about the motion and position of body parts. They have been particularly useful for action recognition [61]. Poses can be directly extracted, normalized and concatenated to form a video-level representation [12, 33, 43] or perform frame level processing [31]. Our work does not include baselines based on poses because our first efforts to train models using this approach were unsuccessful.

Finally, other manually designed and sophisticated multi-cue channels have been proposed for SLT. In [16] they combine raw frames and poses with a two-stream network. [64] proposes using Spatial and Temporal Multi-cue networks [67], which combines cues from image and pose (hand, face, full-frame and pose) in multiple scales. Another work uses a combination of raw frames and poses, and uses Graph Convolutional Networks to extract the tokens [25].

| Dataset | Duration(h) | | | Vocabulary(K) | | | BLEU | Domain |
|---|---|---|---|---|---|---|---|---|
| | train | val | test | train | val | test | | |
| KETI [33] | 20.05 | 2.24 | 5.70 | ← | 0.49 | → | 57.37 [33] | Emergency situations |
| PHOENIX-2014T [8] | 9.2 | 0.6 | 0.7 | 2 | 0.9 | 1 | 25.59 [60] | Weather Forecast |
| CSL Daily [66] | 20.62 | 1.24 | 1.41 | 2 | 1.3 | 1.3 | 23.92 [15] | Daily life |
| OpenASL [54] | ← | 288 | → | ← | 33 | → | 6.72 [54] | Youtube (news + vlogs) |
| How2Sign [23] | 69.6 | 3.9 | 5.6 | 15.6 | 3.2 | 3.6 | 8.03 (Ours) | Instructional |

Table 1. Comparison between SLT datasets based on the duration of the videos (in hours), number of unique words (in thousands) in the vocabulary and SOTA on SLT without glosses. ← → indicate that in some cases only statistics on the whole dataset are provided.

## 3. Data Preprocessing

One of the main challenges in SLT is the variability and complexity of sign languages, which can be influenced by a variety of factors such as the signer's background, context, and appearance. Therefore, it is important to preprocess the data to reduce this variability. This includes techniques such as visual feature extraction and normalization, as well as standardizing the format of the target data, which is text in our case.

### 3.1. Video tokenization

We choose I3D features [14] to extract video representations directly from the RGB frames, motivated by their effectiveness in the sign recognition [29, 36] and retrieval [22] tasks. I3D features consider not only visual cues, but also temporal information. As a result, they provide a dense and reliable source of visual cues as input to our models.

The original I3D network is trained on ImageNet [20] and fine-tuned for action recognition with the Kinetics-400 [30] dataset. As shown in [2, 21, 22, 37, 45, 54], further fine-tuning with sign language data is needed to properly model the temporal and spatial information present in them. We used the I3D features provided in [22].

The I3D network was trained on 16 consecutive frames, and videos were resized to 224 x 224. Color, scale, and horizontal flip augmentations were applied, and the features were extracted from the 1024-dimension activation before the pooling layer of the I3D backbone. Since they are already an output of a trained network, further processing, such as normalization, is not needed.

### 3.2. Text processing

Text preprocessing is an important step in preparing raw text data into a more suitable format for NLP models. By cleaning, normalizing, and transforming text data into a standardized format, data can be effectively utilized by NLP algorithms.

**Lowercase.** Similar to NLP pipelines, our system first converts raw text to lowercase. Lowercasing reduces the complexity of the vocabulary and minimizes the impact of irrelevant capitalization variations, thereby simplifying subsequent processing steps.

**Tokenization.** We employ the Sentencepiece tokenizer [35] to segment the lowercase text into sub-word units. This approach represents a significant improvement over the conventional method of treating each word as a unit of the sequence. Word-based tokenization leads to an expansive vocabulary and an inability to account for previously unseen words, even if they are variations of words in the vocabulary. On the other hand, sub-word tokenizers optimize the representation of words in the training data partition by identifying the most effective sub-word units, based on their frequency, while imposing a predefined vocabulary size. This allows better handling of unseen words, that can be represented as combinations of sub-words from the vocabulary. Sub-word tokenization requires specifying a fixed vocabulary size, which becomes a hyperparameter to be optimized. The choice of vocabulary size has trade-offs in terms of representation and computational efficiency. When the vocabulary size is small, all sub-words are used more frequently, potentially leading to a better representation of unseen words. However, this also results in longer sequences as more sub-words are required to represent the same inputs, which can increase computational costs. Conversely, a larger vocabulary size reduces sequence length but may have worse coverage of rare and unseen words. Therefore, selecting the optimal vocabulary size requires balancing the need for better representation against the computational cost.

**Postprocessing.** To ensure a fair assessment of the system's performance, it is necessary to compare the model outputs to the original test set without any prior processing. However, this approach may result in a lower BLEU score, as the model generates text based on preprocessed data. For instance, comparing two versions of the same sentence, one lowercase and the other not, would result in the same word being counted as two different words. Therefore, we implement a postprocessing step, that involves detokenization and truecasing [38], to restore the original capitalization and prevent this issue from arising.

## 4. Methodology

The building blocks of our implementation are depicted in Figure 2. The input video stream is tokenized with a pre-trained I3D feature extractor. These tokens are fed into the encoding layers of the Transformer. The decoder of the Transformer operates with lowercase and tokenized textual representations.

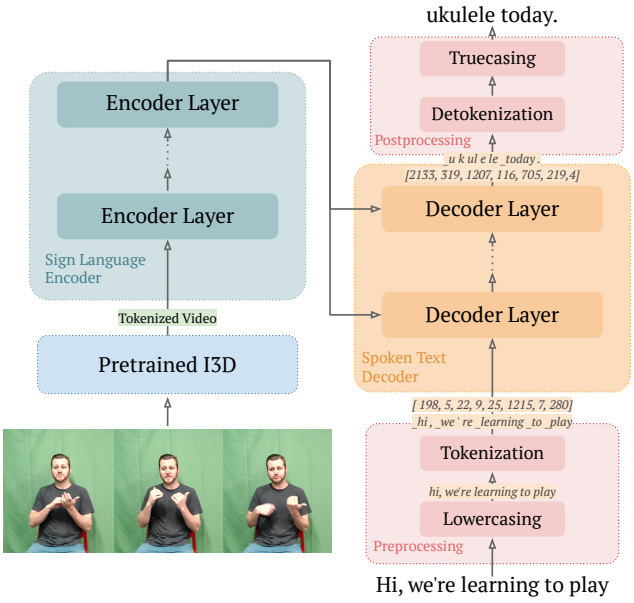


Figure 2. The input video sequence is fed into a Transformer to generate the output text sequence.

### 4.1. Neural architecture

We use a standard transformer encoder-decoder, to leverage their ability to model context and dependencies across the input sequence, as well as their sequence-to-sequence capabilities.

For the model, we choose an asymmetric encoder-decoder with six encoder layers and three decoder layers, each with four attention heads, we select an embedding dimension of 256 and feed-forward network hidden size of 1024, which corresponds to ID (17) from Table 3.

### 4.2. Implementation details

In our implementation, we first preprocess the vocabulary as described in Section 3.2, with a vocabulary size of 7000 subwords.

For training, the batch size is set to 32, and we use cross entropy loss with label smoothing of 0.1. We select the Adam optimizer, we warm-up the learning rate for the first 2000 updates, and then we apply a cosine decay from $10^{-3}$ to $10^{-7}$ with warm restart every $1.7 \cdot 10^4$ steps We train the model for $10^5$ steps, equivalent to 108 epochs. We perform validation every two epochs. Our training process takes 3.5 hours on a single NVIDIA GeForce RTX 2080 Ti GPU.

For inference, we adopt steps commonly used in machine translation. During the text generation phase, the decoder predicts the next token by sampling from the probability distribution conditioned on the previously generated tokens. Instead of only selecting the prediction with higher probability, we use the beam search algorithm to generate predictions. Beam search generates multiple candidate sequences, we choose a beam size of five.

### 4.3. Evaluation protocol

To measure the performance of our SLT models, we use BLEU score [47] a widely used metric in machine translation that measures the similarity between the predicted translation and the ground truth, at the corpus level.[2] We implement it using sacreBLEU [48].

The difficulty of the SLT task causes a bias in the model prediction towards most statistically frequent patterns, such as Example (2) and (3) in Table 4. These patterns can inflate the BLEU scores without actually translating anything meaningful. Inspired by [21] we compute reducedBLEU (rBLEU). This metric consists of removing certain words from the reference and the prediction before computing the BLEU score. We create a blacklist of words that are frequently used in the training data but do not contribute much to the meaning of the sentences, such as articles, prepositions, and pronouns. Appendix A.1 provides the complete list of the removed words, as well as the process to obtain them.

Table 4 shows a comparison between rBLEU and BLEU metrics. In general, rBLEU scores are substantially lower than BLEU. With rBLEU we are reducing the number of words of the sentences. Although How2Sign sentences are long, with an average of 11 words per sentence [23], if the reduced texts have less than four words, it is not possible to compute BLEU score and becomes zero-valued, even if the words are perfectly matching. For reference, for the test partition, after applying the reduction to the text, 40% of the sentences contained less than 3 words. Although only longer sentences contribute to the rBLEU metric, shorter sentences are known to be comparatively easier for the model, so we are able to identify models that deal better with complex examples.

Focusing on concrete examples, row (2) in Table 4, shows that both the prediction and the reference contain the phrase “In this clip I’m going to show you how to”, which is one of the frequent patterns on the instructional dataset. This pattern inflates the BLEU score, while it does not affect the rBLEU score, which is low, suggesting that sentences have different meanings. Similarly, row (3) in Table 4 con-

[2]Other SLT papers use BLEU-4 instead of BLEU. It represents the same score, we use BLEU for simplicity.

| | val | | | | | test | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **rBLEU** | BLEU-1 | BLEU-2 | BLEU-3 | **BLEU** | **rBLEU** | BLEU-1 | BLEU-2 | BLEU-3 | **BLEU** |
| Ours. | 2.79 | 35.2 | 20.62 | 13.25 | 8.89 | 2.21 | 34.01 | 19.3 | 12.18 | **8.03** |

Table 2. Best scores on How2Sign for Sign Language Translation.

tains the phrase "we're going to talk about how to", which similarly inflates the BLEU score even if prediction and the reference differ in meaning.

Experimental results indicate that rBLEU is a more reflective indicator of actual performance than traditional BLEU, for low-resource settings that also have repetitive patterns, given that it considers mostly semantically meaningful words. In order to provide comparable results with other works, we also report standard BLEU in our results.

## 5. Experiments

The performance of our proposed approach is shown in Table 2. We evaluate our models using the metrics described in Section 4.3 and provide examples of generated spoken language translation sentences.

### 5.1. Quantitative results

Our implementation achieves the machine translation metrics reported in Table 2. To the authors' knowledge, these are the first published results for SLT obtained with the How2Sign dataset. The table displays the results of our best configuration, which provides a baseline from where future work can build upon.

### 5.2. Qualitative results

We provide a qualitative assessment of the results in Table 4, showing a few spoken language translations generated by our best-performing model. Words used to compute rBLEU are in bold.

Example (1) shows the ability of our model to provide detailed translations even for complex words like *"self defense"*. Our metrics indicate both high BLEU and rBLEU scores meaning that the model is generating a good translation, considering both full sentences and meaningful words.

However, our results also suggest that this is not always the case. For instance, in Examples (2) & (3), BLEU is higher than rBLEU. We believe that this occurs because of the nature of the dataset. Given that we are working with instructional videos, there are frequent phrases like *"I'm going to show you how to"*, *"we're going to talk about how to"*, which the model learns to predict easily. And although this phrase has been correctly translated, the example has a different meaning, resulting in a lower rBLEU score. Given that high BLEU scores can be misleading due to their susceptibility to frequent phrases, we emphasize the importance of using rBLEU instead of BLEU when selecting the best checkpoint.

The provided examples suggest that the models' performance may depend on the complexity and length of the signed video. We observed that the model was able to provide reasonably accurate translations for short sentences, except for Example (5). For longer sequences, the model struggled to capture the meaning of the video. This is also evidenced by the fact that only a few words are selected to compute rBLEU.

The last example (6) illustrates the reason behind the disparity between rBLEU and BLEU metrics, explained in Section 4.3. In this case, despite obtaining a high BLEU score and an accurate translation, the corresponding rBLEU score is zero due to the reduced number of remaining words for rBLEU calculation, which is less than four.

Overall, the findings suggest that the model's quality is still suboptimal, as demonstrated by Example (4), which has comparable metrics to the overall performance. Our analysis identifies cases where BLEU-guided translations fall short, and we propose rBLEU as a validation metric that aligns more closely with translations that effectively capture the original semantic meaning.

| | Values |
|---|---|
| Text preprocessing | {**yes**, no} |
| Vocabulary size | {1k, 4k, **7k**} |
| Batch size | {**32**, 64} |
| Learning Rate (LR) | {5e-2, **1e-3**, 5e-3} |
| LR scheduler | {**cosine**, inv_sqrt} |
| Warm-up steps | {0, **2k**, 4k} |
| Warm restarts period | {0,**17k**, 22k} |
| Weight Decay | {1e-3, 1e-2, **1e-1**} |
| Label Smoothing | {0, **0.1**} |
| Dropout | {0, 0.1, 0.2, **0.3**} |
| # Layers (encoder-decoder) | {2-2, 3-3, 4-2, **6-3**} |
| Embed dim | {**256**, 512} |
| FFN dim | {512, **1024**, 2048} |
| # Attention heads | {**4**, 8} |

Table 3. Hyperparameters search space. In bold are the optimal ones that we found during validation, and underlined are defaults.

| Example | | | rBLEU | BLEU |
|---|---|---|---|---|
| (1) | Reference<br>Prediction | And that's a **great vital point technique** for **women's self defense**.<br>It's really a **great point** for **women's self defense**. | 30.29 | 38.25 |
| (2) | Reference<br>Prediction | In this **clip** I'm going to **show** you how to **tape** your **cables** down.<br>In this **clip** I'm going to **show** you how to **improve push ups**. | 24.88 | 64.53 |
| (3) | Reference<br>Ours | In this **segment** we're going to **talk** about how to **load** your still for **distillation** of **lavender essential oil**.<br>—<br>**Ok**, in this **clip**, we're going to **talk** about how to **fold** the **ink** for the **lid** of the **oil**. | 6.77 | 29.82 |
| (4) | Reference<br>Ours. | You are **dancing**, and now you are going to **need** the **veil** and you are going to just **grab** the **veil** as **far** as **possible**.<br>—<br>So, **once** you're **belly dancing**, **once** you've **got** to have the **strap**, you're going to **need** to **grab** the **thumb**, and **try** to **avoid** it. | 4.93 | 8.04 |
| (5) | Reference<br>Ours | But if you have to **setup** a **new campfire**, **there's two ways** to do it in a very **low impact**; one is with a **mound fire**, which we should in the **campfire segment earlier** and the other way to **setup** a **low impact campfire** is to have a **fire pan**, which is just a **steel pan** like the **top** of a **trash** can.<br>—<br>And other thing I'm going to **talk** to you is a little bit more **space**, a **space** that's what it's going to do, it's **kind** of a **quick**, and then I don't want to take a **spray skirt** off, and then I don't want it to take it to the **top** of it. | 0.85 | 3.79 |
| (6) | Reference<br>Ours | So, this is a very **important part** of the **process**.<br>It's a very **important part** of the **process**. | 0.0 | 61.86 |

Table 4. Qualitative examples from our best-performing model. In bold the words remaining to compute rBLEU. Corresponding manually selected input frames from examples can be found in Appendix A.2

### 5.3. Hyperparameter search

Transformer under low-resource conditions is highly dependent on hyperparameter settings [4]. Our experiments show that using an optimized Transformer improves the translation quality over 3.47 BLEU points and 1.8 reduced BLEU points compared to the default hyperparameters for SLT.

Table 3 shows the hyperparameters that we optimize, ordered by tuning order. Default hyperparameters for SLT come from [11]. Exploring all possible values in Table 3 is extremely expensive. Possible methodologies of exploration include random search or grid search. We choose a flexible grid search, which means that we try different values, and once the hyperparameter is tuned, we fix it. Since there is no guarantee that this will result in a global optimum, we analyze the results to discard or add experiments during the exploration.

As highlighted in Section 3, text preprocessing plays an important role in NLP tasks. In our experiments, we opted to lowercase our training data. To showcase its efficacy, we trained our model with both preprocessed text data and raw text data (i.e., direct production of truecased outputs). Our results indicate that lowercasing the text data yields significant improvements in the rBLEU metric, as evidenced in Table 5.

| ID | Text Preprocessing | rBLEU |
|---|---|---|
| (1) | no | 0.62 |
| (2) | yes | **0.98** |

Table 5. Impact of text preproceessing.

| ID | Vocabulary Size | rBLEU |
|---|---|---|
| (3) | 1000 | 0.85 |
| (4) | 4000 | 0.89 |
| (5) | 7000 | **0.98** |

Table 6. Impact of the vocabulary size.

| ID | Encoder Layers | Decoder Layers | Embed Dim | FFN Dim | Attention Heads | LR | LR Scheduler | rBLEU |
|---|---|---|---|---|---|---|---|---|
| (6) | 3 | 3 | 512 | 2048 | 8 | 0.001 | inverse_sqrt | 0.98 |
| (7) | 3 | 3 | 512 | 2048 | 8 | 0.001 | cosine (T=17k) | 0.89 |
| (8) | 3 | 3 | 256 | 1024 | 4 | 0.001 | cosine (T=17k) | 1.14 |
| (9) | 3 | 3 | 256 | 1024 | 4 | 0.005 | cosine (T=17k) | 0.68 |
| (10) | 2 | 2 | 256 | 1024 | 4 | 0.001 | cosine (T=17k) | 1.32 |
| (11) | 2 | 2 | 256 | 1024 | 4 | 0.005 | cosine (T=17k) | 0.72 |
| (12) | 2 | 2 | 256 | 512 | 4 | 0.001 | cosine (T=17k) | 1.37 |
| (13) | 4 | 2 | 256 | 1024 | 4 | 0.001 | cosine (T=17k) | 1.14 |
| (14) | 4 | 2 | 256 | 1024 | 4 | 0.005 | cosine (T=17k) | 0.64 |
| (15) | 6 | 3 | 512 | 2048 | 8 | 0.001 | cosine (T=17k) | 0.87 |
| (16) | 6 | 3 | 512 | 2048 | 8 | 0.001 | cosine (T=22k) | 0.75 |
| (17) | 6 | 3 | 256 | 1024 | 4 | 0.001 | cosine (T=17k) | 0.93 |

Table 7. Validation scores during the exploration of the model architecture.

As previously discussed in Section 3.2, the selection of appropriate vocabulary size is a trade-off between enhancing the representation of rare words or producing shorter sequences. In our study, we utilize the SentencePiece [35] tokenizer and experiment with dictionary sizes of 1000, 4000, and 7000 sub-words to evaluate their respective impacts on NMT performance.

Results in Table 6 show that larger dictionary improves results, in our experiments it increases 0.14 points of reduced BLEU score. Thus, we always use a Sentencepiece tokenizer with a vocabulary of 7000 sub-words.

A current observation in Transformers is that increasing the number of parameters will improve the performance. However, in low-resource languages, increasing the number of model parameters can hinder performance [64]. We study the effect of using a deeper and shallower Transformer by changing the number of layers in the encoder and decoder, the number of attention heads, the feed-forward layer dimension, and embedding dimensions.

Since the optimization of the learning rate (LR) is dependent on the number of parameters of the model, we tune it together with other hyperparameters related to the architecture size. Furthermore, we introduce the use of LR scheduling of cosine with warm restarts. This scheduler has been shown to perform better than alternatives [39]. The resetting of the learning rate acts as a simulated restart of the learning process and is defined by number of steps T.

Table 7 shows the results of our system optimizations. Experiments point to the direction that smaller models, like (12) perform better for our dataset. The loss curves indicated a substantial amount of overfitting in the larger models, which is most likely related to the *small* amount of provided data compared to the amount of data needed to tune a large number of parameters. We see gains in tuning the learning rate to improve performance. Our results indicate that it is beneficial to use four attention heads instead of eight under low-resource conditions. Due to the fact that the input data is by far more complex than the output, we choose to carry out further experiments with both the best symmetric model (12) and the best asymmetric model (17).

Given the observed overfitting, we add regularization by adding dropout, weight decay, and label smoothing. Considering it is difficult to perform data augmentation with our video features, adding regularization helps make the model more robust to overfitting.

Table 8 shows that we obtain substantial improvements by increasing regularization techniques. That is to be expected since overfitting was present in our previous experiments. Surprisingly, it appears that tuning these hyperparameters is the most effective measure to improve the model's performance. We also show that under these conditions, a larger model paired, such as (22), with regularization techniques outperforms a smaller model.

| ID | Base | Dropout | Weight Decay | Label Smoothing | rBLEU |
|---|---|---|---|---|---|
| (6) | - | 0.1 | 0.001 | 0 | 0.98 |
| (18) | (17) | 0.2 | 0.01 | 0.1 | 1.17 |
| (19) | (12) | 0.2 | 0.01 | 0.1 | 1.21 |
| (20) | (6) | 0.3 | 0.1 | 0.1 | 1.84 |
| (21) | (12) | 0.3 | 0.1 | 0.1 | 1.38 |
| (22) | (17) | 0.3 | 0.1 | 0.1 | **2.78** |

Table 8. Validation scores for different regularization techniques.

## 6. Discussion

Our experiments yielded several findings. Firstly, we observed that text preprocessing is an important step that can significantly improve performance, resulting in an increase of 0.36 rBLEU points. Secondly, we found using a greater vocabulary size led to an increase of 0.14 rBLEU.

Another finding was that choosing the correct parameters for the architecture is crucial for achieving optimal performance, resulting in a 0.39 rBLEU improvement. Furthermore, our results highlight the difficulty of finding the sweet spot where regularization techniques help but not hinder the performance of deep learning models. In our case, we boosted performance by an impressive 1.8 rBLEU points after an extensive sweep of hyperparameters and configurations.

During the qualitative analysis, we show that the model is able to produce meaningful translations. Moreover, our experiment highlights the importance of considering rBLEU as an effective metric for evaluating the best checkpoint. Higher rBLEU scores indicated a consistent correlation with the model's ability to capture the semantic meaning from the video.

While our work has shown promising results, there is still room for improvement. Our current approach only explores the use of I3D as visual feature. While other works use pose landmarks as a visual features [12, 31, 33, 43], our initial exploratory work with MediaPipe [40] poses, obtained unsatisfactory results with BLEU score of 0.8 for the test partition.

Furthermore, upon qualitative exploration, we realized the decoder was discarding the conditioning provided by the encoder and functioning solely as a language model. We hypothesize that this behavior may be due to our current approach of feeding poses as sequences of one-dimensional arrays containing only landmark coordinates. This method may not be the most effective way of processing the graph-like structure present in poses. One proposed way of tackling this is extracting optical flow features based on human pose estimation [42], which worked well for sign language detection. Similarly to [63], we recognize the need for an in-depth exploration of visual features appropriate for SLT.

We believe another exciting direction would be the exploration of using a pre-trained decoder, similar to [18], where language models that are already trained for spoken language translation are adapted for sign languages.

**Societal Impact.** Efficiently translating sign language videos can have a significant impact on accessibility, opening up a range of useful applications. However, there are also potential risks associated with this technology, including problems associated with the accuracy of models, which currently produce inaccurate or incomplete translations, biases present in the datasets, and increased risk of surveillance of signers, similarly how automatic speech recognition (ASR) technologies may affect the privacy of speakers.

## 7. Conclusions

In this work, we made an open-source implementation that serves as a first baseline for sign language translation on the How2Sign dataset, a large and complex dataset. Our approach achieved a BLEU score of 8.03, inidicating a certain level of understanding of the signed utterances, which is on par with results reported for OpenASL [54], a publicly available dataset of comparable complexity.

Additionally, our extensive hyperparameter search demonstrates the necessity of tuning to obtain the best set of parameters. The best results are obtained with an asymmetric Transformer trained with great amounts of regularization.

Our evaluations, both quantitative and qualitative, have led us to conclude that rBLEU is a suitable evaluation metric for similar benchmarks, particularly for low-resource datasets with frequent repetitive patterns. In contrast to traditional BLEU score, which may be inflated due to these patterns, rBLEU provides a more accurate evaluation that better reflects the model's performance.

Lastly, we provide the code and models to allow reproducibility and encourage further research and advancements in sign language translation field.

## Acknowledgements

This research was partially supported by research grant Adavoice PID2019-107579RB-I00 / AEI / 10.13039/501100011033, research grants PRE2020-094223, PID2021-126248OB-I00 and PID2019-107255GB-C21 and by Generalitat de Catalunya (AGAUR) under grant agreement 2021-SGR-00478.

## References

[1] Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. Bsl-1k: Scaling up co-articulated sign language recognition using mouthing cues. In *European Conference on Computer Vision*, pages 35–53. Springer, 2020. 1

[2] Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 3

[3] Samuel Albanie, Gül Varol, Liliane Momeni, Hannah Bull, Triantafyllos Afouras, Himel Chowdhury, Neil Fox, Bencie Woll, Rob Cooper, Andrew McParland, and Andrew Zisserman. BOBSL: BBC-Oxford British Sign Language Dataset. 2021. 2

[4] Ali Araabi and Christof Monz. Optimizing transformer for low-resource neural machine translation. *Proceedings of the*

*28th International Conference on Computational Linguistics*, 2020. 6

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2

[6] Alvaro Budria, Laia Tarres, Gerard I Gallego, Francesc Moreno-Noguer, Jordi Torres, and Xavier Giro-i Nieto. Topic detection in continuous sign language videos. *arXiv preprint arXiv:2209.02402*, 2022. 2

[7] Jan Bungeroth and Hermann Ney. Statistical sign language translation. 2004. 2

[8] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, 2018. 1, 2, 3

[9] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Multi-channel transformers for multi-articulatory sign language translation. In *European Conference on Computer Vision*. Springer, 2020. 1, 2

[10] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Multi-channel transformers for multi-articulatory sign language translation. In *Computer Vision–ECCV 2020 Workshops*, pages 301–319. Springer, 2020. 2

[11] 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 (CVPR)*, 2020. 1, 2, 6

[12] Necati Cihan Camgöz, Ben Saunders, Guillaume Rochette, Marco Giovanelli, Giacomo Inches, Robin Nachtrab-Ribback, and Richard Bowden. Content4all open research sign language translation datasets. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 1–5. IEEE, 2021. 2, 8

[13] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[14] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[15] Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. A simple multi-modality transfer learning baseline for sign language translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5120–5130, 2022. 2, 3

[16] Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. Two-stream network for sign language recognition and translation. In *Advances In Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2

[17] Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[18] Mathieu De Coster, Karel D'Oosterlinck, Marija Pizurica, Paloma Rabaey, Severine Verlinden, Mieke Van Herreweghe, and Joni Dambre. Frozen pretrained transformers for neural sign language translation. In *18th Biennial Machine Translation Summit (MT Summit 2021)*, pages 88–97. Association for Machine Translation in the Americas, 2021. 1, 8

[19] Mathieu De Coster, Mieke Van Herreweghe, and Joni Dambre. Isolated sign recognition from rgb video using pose flow and self-attention. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3436–3445, 2021. 1

[20] 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. 3

[21] Subhadeep Dey, Abhilash Pal, Cyrine Chaabani, and Oscar Koller. Clean text and full-body transformer: Microsoft's submission to the wmt22 shared task on sign language translation. *Proceedings of the Seventh Conference on Machine Translation*, 2022. 1, 2, 3, 4

[22] Amanda Duarte, Samuel Albanie, Xavier Giro i Nieto, and Gul Varol. Sign language video retrieval with free-form textual queries. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3

[23] Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. How2Sign: A Large-scale Multimodal Dataset for Continuous American Sign Language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 3, 4

[24] Jens Forster, Christoph Schmidt, Oscar Koller, Martin Bellgardt, and Hermann Ney. Extensions of the sign language recognition and translation corpus rwth-phoenix-weather. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1911–1916, 2014. 1

[25] Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Lei Xie, and Sanglu Lu. Skeleton-aware neural sign language translation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4353–4361, 2021. 2

[26] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Hierarchical lstm for sign language translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. 2

[27] Aiming Hao, Yuecong Min, and Xilin Chen. Self-mutual distillation learning for continuous sign language recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[28] Songyao Jiang, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li, and Yun Fu. Skeleton aware multi-modal sign language recognition. In *Challenge on Large Scale Signer Independent Isolated Sign Language Recognition (CVPR)*, 2021. 1

[29] Hamid Reza Vaezi Joze and Oscar Koller. Ms-asl: A large-scale data set and benchmark for understanding american sign language. 2019. 3

[30] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 3

[31] Youngmin Kim, Minji Kwak, Dain Lee, Yeongeun Kim, and Hyeongboo Baek. Keypoint based sign language translation without glosses. *arXiv preprint arXiv:2204.10511*, 2022. 2, 8

[32] Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. Neural sign language translation based on human keypoint estimation. *Applied Sciences*, 2019. 1, 2

[33] 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. 2, 3, 8

[34] 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, Dec. 2015. 2

[35] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Conference on Empirical Methods in Natural Language Processing*, 2018. 3, 7

[36] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1459–1469, 2020. 2, 3

[37] Dongxu Li, Chenchen Xu, Xin Yu, Kaihao Zhang, Benjamin Swift, Hanna Suominen, and Hongdong Li. Tspnet: Hierarchical feature learning via temporal semantic pyramid for sign language translation. *Advances in Neural Information Processing Systems*, 33:12034–12045, 2020. 2, 3

[38] Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. tRuEcasIng. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 152–159, Sapporo, Japan, July 2003. Association for Computational Linguistics. 3

[39] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 7

[40] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 2, 8

[41] Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen. Visual alignment constraint for continuous sign language recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11542–11551, 2021. 1

[42] Amit Moryossef, Ioannis Tsochantaridis, Roee Yosef Aharoni, Sarah Ebling, and Srini Narayanan. Real-time sign language detection using human pose estimation. 2020. https://www.slrtp.com/. 8

[43] Mathias Müller, Sarah Ebling, Eleftherios Avramidis, Alessia Battisti, Michèle Berger, Richard Bowden, Annelies Braffort, Necati Cihan Camgöz, Cristina España-Bonet, Roman Grundkiewicz, et al. Findings of the first wmt shared task on sign language translation (wmt-slt22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 744–772, 2022. 2, 8

[44] Zhe Niu and Brian Mak. Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 172–186, Cham, 2020. Springer International Publishing. 1

[45] Alptekin Orbay and Lale Akarun. Neural sign language translation by learning tokenization. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 222–228. IEEE, 2020. 2, 3

[46] Ilias Papastratis, Kosmas Dimitropoulos, Dimitrios Konstantinidis, and Petros Daras. Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space. *IEEE Access*, 8:91170–91180, 2020. 1

[47] 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, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. 4

[48] Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. 4

[49] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Everybody sign now: Translating spoken language to photo realistic sign language video. *arXiv preprint arXiv:2011.09846*, 2020. 1

[50] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Continuous 3d multi-channel sign language production via progressive transformers and mixture density networks. *International journal of computer vision*, 129(7):2113–2135, 2021. 1

[51] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Mixed signals: Sign language production via a mixture of motion primitives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1919–1929, 2021. 1

[52] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Skeletal graph self-attention: Embedding a skeleton inductive bias into sign language production. 2021. 1

[53] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Signing at scale: Learning to co-articulate signs for large-scale photo-realistic sign language production. 2022. 1

[54] Bowen Shi, Diane Brentari, Greg Shakhnarovich, and Karen Livescu. Open-domain sign language translation learned from online video. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022. 2, 3, 8

[55] Ozge Mercanoglu Sincan and Hacer Yalim Keles. Autsl: A large scale multi-modal turkish sign language dataset and baseline methods. *IEEE Access*, 8:181340–181355, 2020. 2

[56] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014. 2

[57] Laia Tarrés, Gerard I Gállego, Xavier Giró-i Nieto, and Jordi Torres. Tackling low-resourced sign language translation: Upc at wmt-slt 22. *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 2022. 2

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[59] Andreas Voskou, Konstantinos P Panousis, Dimitrios Kosmopoulos, Dimitris N Metaxas, and Sotirios Chatzis. Stochastic transformer networks with linear competing units: Application to end-to-end sl translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11946–11955, 2021. 1

[60] Andreas Voskou, Konstantinos P Panousis, Dimitrios Kosmopoulos, Dimitris N Metaxas, and Sotirios Chatzis. Stochastic transformer networks with linear competing units: Application to end-to-end sl translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11946–11955, 2021. 2, 3

[61] 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. 2

[62] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*, 1(2):5, 2017. 2

[63] Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. Including signed languages in natural language processing. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021. 8

[64] Kayo Yin and Jesse Read. Better sign language translation with stmc-transformer. *Proceedings of the 28th International Conference on Computational Linguistics*, 2020. 1, 2, 7

[65] Liwei Zhao, Karin Kipper, William Schuler, Christian Vogler, Norman Badler, and Martha Palmer. A machine translation system from english to american sign language. In *Proceedings of the Fourth Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 54–67. Springer, 2000. 2

[66] Hao Zhou, Wen gang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. Improving sign language translation with monolingual data by sign back-translation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1316–1325, 2021. 1, 2, 3

[67] Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. Spatial-temporal multi-cue network for sign language recognition and translation. *IEEE Transactions on Multimedia*, 24:768–779, 2020. 2