Moscow Institute of Physics and Technology Phystech School of Applied Mathematics and Informatics
Department of Machine Learning and Digital Humanities

Alua Musralina

Sign Language Production using Deep Learning

Master's Thesis

Advisor: Radoslav Neychev

Moscow, 2023

Contents

Abstract

Sign language is a visually performed language with distinct signed languages and millions of deaf individuals worldwide. Sign Language Processing involves tasks such as translation and production. This work focuses on the second task, which is crucial for interpreters, educators and accessibility. Just as spoken languages have their own unique grammatical structures, different sign languages also have their own distinct grammar. Moreover, sign sequences and spoken language sentences lack direct alignment. Therefore, Gloss production as an intermediate step is crucial. In this work the German Sign Language is chosen for evaluation. Deep learning approach, like Neural Machine Translation is used and different seq2seq architectures application results are evaluated. A Fine-tuning of pre-trained model is explored as well. With the fine-tuned Transformer the results are improved from the previous methods and can be a promising step towards to close to natural Sign Language Production. At the end, for demonstration purposes the skeleton from Gloss generation is implemented.

1 Introduction

According to the United Nations in year 2022, there are as many as 300 distinct signed languages, while the World Federation of the Deaf estimates that there may be up to 70 million deaf individuals worldwide.

Sign language is a visually performed language that relies on hand gestures, body postures, and facial expressions, with the meaning of signs being determined by the combination of all these elements. The Sign Language Processing is the part of both Natural Language Processing (NLP) and Computer Vision (CV) tasks. For it, there are two main tasks can be implemented: Translation from sign language, known as Sign Language Recognition (SLR) and into sign language, known as Sign Language Production (SLP).

In this work the focus will be on the second part. The production of proficient signs is crucial for professional sign language interpreters and translators. Interpreters work in a wide range of settings, including conferences, meetings, classrooms, medical appointments, legal proceedings, and public events. Educators and sign language instructors for students with hearing impairments might use SLP system to deliver lessons, provide instructions, and foster a conducive learning environment. Sign language production is commonly used in public service announcements, news broadcasts, and emergency broadcasts to ensure accessibility for the deaf and hard-of-hearing community. Including sign language alongside spoken language ensures that important information is accessible to everyone. SLP is utilised in performance arts, such as sign language poetry, storytelling, and theatre. Deaf performers and hearing performers incorporating sign language can express their creativity, engage audiences, and promote cultural diversity and inclusivity.

Each sign in sign language is identified by a unique identifier known as a Gloss. There is a lack of direct alignment between sign sequences and sentences in spoken language. See example in Fig. 1. Please note that a single sign can have two

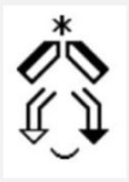
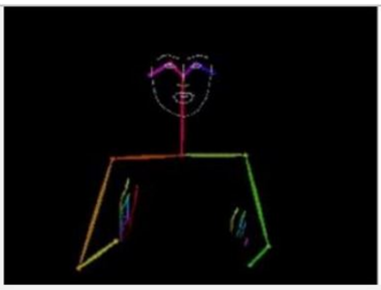
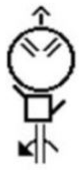
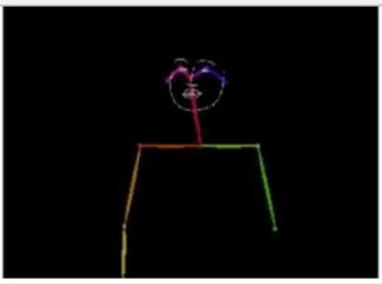
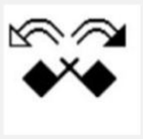
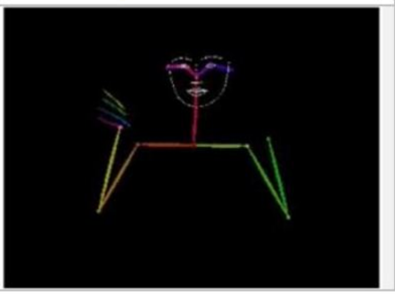
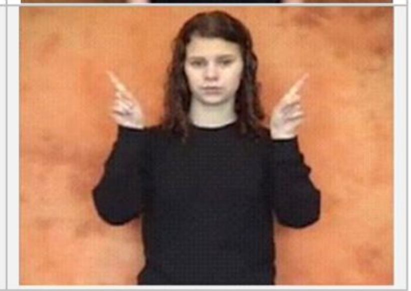
English Translation	Gloss	Notation	Pose	Video/Avatar
House	HOUSE			
What's the matter? What's wrong?	Wrong-What			
Different But	DIFFERENT BUT			

Figure 1: English translations, their Glosses and Notations for several signs [1].

unrelated Glosses, and a single Gloss can be translated into multiple valid spoken languages.

In general, sign language processing involves several tasks such as detection, identification, segmentation, recognition, translation, and production. Detection involves identifying whether sign language is present in a video, while identification is about determining which sign language is being used. The segmentation task applied to identify the signs to separate phrases or signs, while recognition aims to comprehend the signs, resulting in labels being assigned to each sign. Translation aims to provide a more natural output than just Glosses or labels, and it is an automatic process. Production, on the other hand, involves translating text into signs [2].

In this work focus is on text to sign production, but not from speech. Because, from speech it can be easily converted into the textual representation, for example, using speech-to-text technology Whisper. It is an automatic speech recognition (ASR) system.

The thesis is organised into five chapters. In Chapter 2, the literature and existing solutions for solving the problem is reviewed. Additionally, in subchapters the dataset and metrics are introduced. After explaining and introducing the theory, the problems statement is derived and explained in Chapter 3. In Chapters 4, the implementation results is presented. Results are compared and evaluated.

2 Literature overview

In general, for SLP three modules might be considered as in [3]: the first module pre-processes the input text and brakes into words with their types (noun, verb, particle, etc.) with a help of language model. In the second module, the pre-processed words are converted into sign sequences or sign Glosses, where RBMT, EBMT, SMT, HBMT, and NMT can be applied. The last module converts the sign sequences into videos, or animated avatars.

2.1 Skeleton-based approach for Sign Language Production

The skeleton points based approach in recognition and generation tasks makes it possible to focus on the signs themselves, since the subject and background variation becomes independent. For example, the work [4] uses the whole body estimator to find the skeleton points, where 133 points are whole body keypoints, 27 nodes of graph, 10 nodes for each hand and 7 nodes for upper body. In that work the sign language is additionally depicted as a continuous sequence of skeleton points, representing the person's coordinates, including both manual and non-manual attributes. In general, there are not so many researches are done for continuous sign generation.
After producing a sequence of skeleton points, given a source spoken language, author in [5] states, that transforming them into a video requires not much effort. It includes plotting the joints and connecting the relevant bones. These 3D joints can then be utilised to animate an avatar, like in [6] [7] or serve as conditioning for a GAN [8]. While the generated sign pose sequence accurately translates the provided text, it might be performed at a different pace compared to the reference data. This variation in speed is not considered as incorrect, since signers

have their own motion and tempo. However, to facilitate visual comparison with the reference sequences, Dynamic Time Warping (DTW), as described in [9], was utilized to align the temporal aspects of the generated sign pose sequences. This alignment procedure aims to enhance the temporal coherence for visualization without altering the content of the productions [5].

2.2 Transfer Learning

Transfer Learning has emerged as one of the most widely recognized and extensively explored areas in NLP. Transfer learning involves the fundamental concept of "transferring" knowledge from one task or model to another.
When it comes to representing the input for a network, word embeddings play a crucial role. There are three options available for obtaining these embeddings for any model:
1. Train from scratch: One option is to train the word embeddings from scratch as an integral part of your model. In this case, the network learns the representations of words during the training process, adjusting them according to the specific task at hand.
2. Use pretrained embeddings as static vectors: Another option is to utilize pretrained word embeddings such as Word2Vec, GloVe, or similar models. These embeddings have been precomputed on large corpora and capture general semantic and syntactic information. By using them as static vectors, they are kept fixed during the training process without updating their values.
3. Initialize with pretrained embeddings and fine-tune: The third option involves initializing the model's embeddings with pretrained vectors and allowing them to be further adjusted or fine-tuned during training. This approach combines the benefits of pretrained embeddings with the ability to adapt them specifically to the task at hand, potentially capturing more task-specific nuances.

So by utilizing pretrained embeddings, the model gains access to a vast corpus of knowledge, enabling it to acquire a broad understanding of the world. Consequently, the pretrained embeddings contribute to the model's ability to capture comprehensive information and leverage the insights extracted from the large-scale training data [10].
Firstly, from word embeddings there was improvements, such as knowing not only about word in general, but how it is utilized in context. For example, the Embeddings from Language Models (ELMo) as it is clear from its name, uses representations from a language model. ELMo representations consist of three layers. So each word is a vector, which is a weighted sum of representations from all layers. But ELMo utilizes task-specific weights to effectively combine representations from its three layers [11].
BERT has emerged as a highly influential technique that has inspired numerous variants such as XLNet, XLM, RoBERTa, ELECTRA. It has demonstrated exceptional performance and set new benchmarks for various language understanding tasks, including reading comprehension and text classification. The state-of-the-art results achieved by these models highlight the significant impact of BERT in advancing the field of natural language understanding [12].

2.3 Deep Learning approaches for Sign Language Production

There are several DL approaches are already introduced for automatic SLP. Neural Machine Translation (NMT) is a popular application of deep learning that utilises encoder-decoder architecture to translate text from one language to another. But in general, when referring to machine translation, it might be any broad sequence-to-sequence task, which involves translating between sequences of tokens, regardless of their nature. In the context of machine translation, the task involves having an input sequence and an output sequence with the possibility of differing lengths. Translation can be considered as the process of identifying the target sequence that has the highest probability, given the input. In Machine translation knowledge about a function with certain parameters is acquired and subsequently its argmax for a given input is determined:

$$p(y/x) : y' = \arg\max_{y} p(y|x, \theta) \tag{1}$$

Sequence-to-sequence tasks can be conceptualized as Conditional Language Models (CLM), which function in a similar manner to Language Models (LMs). However, CLMs also incorporate source information in their operation. As the only distinction from Language Models (LMs) lies in the inclusion of the source information, the modeling and training process for Conditional Language Models (CLMs) remains highly similar. Specifically, the high-level pipeline can be outlined as follows:
1. Pass the source and previously generated target words into a network.
2. Obtain a vector representation of the context, encompassing both the source and the previous target, from the decoder of the network.
3. Using this vector representation, generate a probability distribution for the next token [10].
There are 2 types of the SLP, as mentioned in [13]: end-to-end system and the intermediate results production. In the second method the Glosses are produced, prior to generate the signs, which described in [14] and [15]. According to [16] applying the second method reduces the performance results, and therefore the transformer-based encoder-decoder architecture was proposed there. But in this work, by refinement the Gloss production process the possible results improvements are evaluated.
The [17] introduces a seq2seq model with Luong attention that translates Text to Gloss first. The Luong et al. [18] model sequentially processes all the words in the source sentence until it encounters the end-of-sentence symbol "<eos>".

Subsequently, it generates the target sentence one word at a time. Instead of using CNN for feature extraction, the model employs the probabilities generated by the decoder to solve a Motion Graph (MG) for sign language pose data, which enables the text-to-pose translation. The MG is a computer graphics technique used for character animation and conditional image generation. To build the MG for sign language pose data, the authors divided continuous sign sequences into individual Glosses and grouped the motion sequences by Gloss. The nodes of the MG are populated by these motion sequences, and the probabilities from the NMT decoder are used to transition between nodes at each time step.

Contrary to [17] in [5] the authors do not provide the a priori information for conditional image generation. It is focused on direct or automatic signs generation, and the method learns the mapping automatically between the text and the pose sequences.

One previous approach was introduced in which the 7-frame sign was generated for every input word. However, this technique had a fixed output length and did not allow dynamic length output [19].

In case of using traditional DNNs, it would be required to be converted the input text into a fixed-length vector using methods, for example as BOW or Word2Vec. However, it should be noted that in doing so, the original word sequence is lost. This results in the model being unaware of the word order, and consequently, important information about the input is missing.

Recurrent Neural Networks (RNNs) were developed to address the issue of retaining sequence information in neural networks. RNNs are specifically designed to process sequential data, such as text, by maintaining a hidden state that captures information about the preceding inputs in the sequence. This enables the model to preserve the order of words in the input text and utilise it for improved predictions or classifications.

The encoder takes in each token/word of the input sequence sequentially and sends the final states to the decoder. Its parameters are updated using back-propagation through time. During the training phase, the decoder operates differently from the encoder. It uses the "Teacher Forcing" technique, so that the actual output/token from the previous step is fed as input to the current step, rather than the predicted output/token. It is important to note that, in the decoder, the output yt_{pred} at any step t represents the probability distribution across the entire output dataset's vocabulary. This distribution is generated using the Softmax activation function. The predicted word is chosen as the token with the highest probability. After generating predictions at each time-step, the loss is computed, and the errors are backpropagated through time to adjust the model parameters during training. The loss function employed is the categorical cross-entropy loss function, which measures the discrepancy between the predicted sequence Y_{pred} and the actual sequence Y_{true}. In testing phase it is a little different, since in a real-world scenario, there will be no access to the actual output sequence Y_{true}, and instead, there will be only the input sequence X. Consequently, it is not possible to employ the same strategy used during the training phase. During testing, the predicted output is fed from the previous time-step as input to the current time-step, rather than the actual output. The remainder of the process is similar to the training phase. To reduce the dimensionality of the input word vectors, the input sequence is passed through an embedding layer in both the encoder and the decoder components. One-hot-encoded vectors can be prohibitively large in practice, making embedded vectors a superior word representation. The embedding layer can be trained together with the model, or it can be pre-trained, such as in the case of Word2Vec embeddings [20].

Before the transformers the RNN was the base of the Neural SL translation [21]. Both GRU and LSTM are recurrent neural network variants used in seq2seq architectures. LSTM is generally considered more powerful in capturing long-term dependencies, but GRU can be computationally more efficient.

At each generation step, certain portions of the source may hold varying degrees of significance. However, in the current configuration, the decoder faces the chal-

lenge of extracting the relevant information from a fixed representation, which is by no means a straightforward task. Attention enables the model to selectively "focus" on specific segments of the input during different stages. An attention mechanism is an integral component of a neural network. At each decoder step, it determines the relative importance of different sections of the source. In this setup, the encoder is not obligated to condense the entire source into a single vector; instead, it provides representations for each individual source token, such as all RNN states rather than just the final one. The core concept is that a network can acquire the ability to discern the relative importance of different input elements at each step. Given that all components involved, including the attention function, softmax, and others, are differentiable, a model equipped with attention can be trained end-to-end. There is no need to explicitly instruct the model on which words to prioritize, as the model itself will autonomously learn to identify significant information.
In Gloss production since the input and output are texts, the pre-trained model could be applied as well. Those different architectures as seq2seq with RNN, attention and pre-trained models are applied for evaluation in this work.

2.4 Progressive Transformer

A Transformer was firstly introduced in [22]. In general, it is a model architecture that avoids using recurrence and instead relies solely on an attention mechanism to capture global dependencies between input and output. This design allows for a greater degree of parallelization and has led to significant advancements in NLP and other areas of machine learning. It can be said that the attention mechanism in the Transformer model has advantages over traditional sequential attention mechanisms, especially when it comes to capturing long-range dependencies. The traditional sequential attention mechanisms, such as global attention or local attention, typically have limitations in capturing long-range dependencies efficiently. Global attention attends to the entire source sentence for each target word, which can be computationally expensive for long sentences. Local attention, on the other hand, focuses on a subset of source words for each target word, but it still operates sequentially and may struggle with capturing dependencies that span long distances [18]. The queries in the "encoder-decoder attention" layers in Transformers come from the preceding decoder layer, while the memory keys and values are sourced from the encoder's output. Within the encoder and decoder are self-attention layers, where keys, values, and queries all originate from the same source, namely the previous encoder or decoder layer's outputs, correspondingly. Each layer within encoder and decoder not only contains attention sub-layers, but also a fully connected feed-forward network. This network operates on each position individually and uniformly and is composed of two linear transformations separated by a ReLU activation. Although the linear transformations are constant across various positions, they are configured with different parameters from layer to layer. Alternatively, these transformations can be thought of as two convolutions with a kernel size of 1. To generate predicted probabilities for the next token, a learned linear transformation and softmax function to the decoder output is applied.

The goal of the work in [5] was to determine the conditional probability p(Y/X) for generating a sign sequence with U frames, given a text X. However, there were some challenges due to the non-monotonic relationship between the source and target language, as well as the fact that sign language utilises a continuous vector space and different channels for manual and non-manual features. To address these challenges, Progressive Transformers were introduced, which made it possible to translate from a symbolic sequence to a continuous sequence. The encoder in this approach encodes the source sequence to a latent representation before being mapped to a target. Source embedding and the temporal encoding is applied to the symbolic source tokens, when Counting Decoding is applied to the target. As mentioned before, Progressive transformer is for continuous sequences, where there is no pre-defined vocabulary. Here important to mention, that it is achieved using the counter decoding, which predicts continuous sequences of different length by predicting the end of sequence. To deal with the possible prediction drifts, data augmentation can be applied. The transformer is called Progressive, since it checks the progress of sequence generation and knows the length of sequence given a source text.

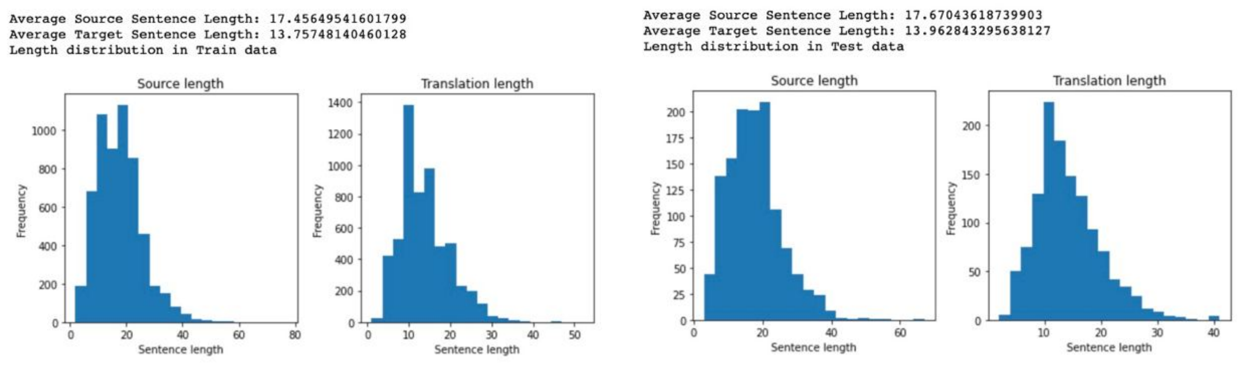
Initially, the source sentences are tokenised and transformed into embeddings to place tokens with similar meanings closer in a high-dimensional space. The purpose of embeddings is to create a dense vector representation for each word that captures its semantic and syntactic properties. Following this, positional encoding is applied to the word embeddings to account for the order of words in the sequence. Since the embedding layer does not consider the word order, positional encoding provides information about the position of each word in the sequence. This allows the model to learn the relative positions of words and their importance in the sequence. The target is also a sequence and they are continuous signs, which are represented by coordinates. Here the counter encoding is applied as well, which includes information about the frame position relative to the total sequence length. Progressive encoder and decoder have several layers, each of which consist of 2 sub-layers: Multi-head attention (MHA) and non-linear feed-forward layer, whereas the progressive decoder generates a sign frame at each time, which is represented by the joined embeddings including the counter. MHA sub-layer is applied in both sides, but in decoder additionally with some extra masking to prevent the future steps decoding [5].

2.5 Dataset

The most of the datasets are prepared for SLR task, and only some of them for SLT [2]. Phoenix German weather dataset is dedicated for SLT task and will be considered for our purpose as well. This dataset contains the pre-recorded 8257 videos of 9 different signers, with 2887 German words and 1066 sign Glosses. The distribution of the dataset see in Fig. 2. In Table 1 the examples of the dataset's source-target pairs are given.

For production purposes, an annotated dataset is required. For example, Swiss German dataset is prepared for continuous Sign Language Generation. It utilizes linguistic corpora with subunit annotation to acquire cross-language subunit models. These subunit models can be employed in machine learning applications on Gloss-based corpora. Specifically, the research focused on employing deep convolutional neural networks to classify finger and palm orientation [23]. This dataset consists of only 100 Glosses and there are only several words for the weather vocabulary. Therefore, in this work only the self-prepared roughly annotated dataset with lack of alignment is applied for skeleton points from text production.

In Fig. 2 the sentences length distribution for Source and its Translation is given,



```
Average Source Sentence Length: 17.45649541601799
Average Target Sentence Length: 13.75748140460128
Length distribution in Train data
```

```
Average Source Sentence Length: 17.67043618739903
Average Target Sentence Length: 13.962843295638127
Length distribution in Test data
```

Figure 2: Phoenix Dataset Distribution (PHOENIX-2014-T).

where the Source is the German spoken-language sentences, and Target is the sequence of Glosses. Examples of these pairs are presented in Table 1. From the given figure, the average length is around 17.5 words for Source sentences, and 14 words for the Target sequences.

Table 1: Example of the Source sentences and Target as Gloss sequences from the PHOENIX-2014-T dataset.

German sentences (SOURCE)	**Sequence of Glosses (TARGET)**
Tiefer Luftdruck bestimmt in den nächsten Tagen unser Wetter	Druck Tief Kommen
Das bedeutet viele Wolken und immer wieder zum teil kräftige Schauer und Gewitter	Es-Bedeutet Viel Wolke Und Koennen Regen Gewitter Koennen
Und nun die Wettervorhersage für morgen Freitag den sechsten Mai	Jetzt Wetter Wie-Aussehen Morgen Freitag Mai Zeigen-Bildschirm
Meist weht nur ein schwacher Wind aus unterschiedlichen Richtungen der bei schauern und gewittern stark böig sein kann	Wind Maessig Schwach Region Wenn Gewitter Wind Koennen

2.6 Metrics for evaluating the results

For tasks like machine translation or summarization, various metrics have been proposed, including BLEU, ROUGE, and METEOR. These metrics provide a way to evaluate the quality of generated text, which can be difficult due to the subjective nature of language.

In general, BLEU provides a standardized and objective way to evaluate the quality of machine translations by comparing them to reference translations. BLEU, which stands for Bilingual Evaluation Understudy, is the most widely used metric for evaluating machine translation systems. In BLEU, modified n-gram precision is used to approximate precision, while best match length is used to approximate recall.

To calculate modified n-gram precision, the fraction of n-grams in the candidate text that are present in any of the reference texts is determined. However, using only this value can lead to issues. For example, if two candidates have identical

unigrams, but different word orders, they will receive the same score. To address this, modified n-gram precision weights each n-gram based on its maximum count in any single reference text.
Additionally, BLEU incorporates a best match length measure to approximate recall. This is the length of the reference sentence which has the closest length to the candidate sentence. The closer the candidate length is to the best match length, the better the recall score will be.
ROUGE, which stands for Recall-Oriented Understudy for Gisting Evaluation, is a metric commonly used for evaluating the quality of text summarization. It provides a quantitative measure of the effectiveness of a summarization system and helps researchers to compare different models or algorithms.
ROUGE is primarily focused on recall and uses different features to calculate it. There are several types of ROUGE, including ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. ROUGE-N is a metric that relies on the concept of n-grams. For instance, ROUGE-1 focuses on matching unigrams, and so forth for higher values of n. To calculate ROUGE-N, the total count of n-grams is determined across all reference summaries, and then the number of n-grams present in the candidate summary is identified. The ratio of these two quantities represents the metric value for ROUGE-N.
In contrast, ROUGE-L, ROUGE-W, and ROUGE-S use different features such as the longest common subsequence (LCS), weighted LCS, and skip-bigram co-occurrence statistics, respectively, to calculate recall. These metrics use an F-score that combines both precision and recall values. For ROUGE-L, precision and recall values are calculated based on the length of the longest common subsequence between the candidate summary and reference summaries.
METEOR, which stands for Metric for Evaluation of Translation with Explicit ORdering, is a machine translation evaluation metric that claims to have a higher correlation with human judgment compared to other metrics such as BLEU. While BLEU is widely used, it can be problematic for evaluating individual sentences as

its scores can be influenced by the length of the entire corpus.

METEOR addresses this problem by modifying the precision and recall computations and replacing them with a weighted F-score based on mapping unigrams and a penalty function for not correct word order. To calculate the weighted F-score, the metric first finds the largest subset of mappings that can form an alignment between the candidate and reference translations. This is done by considering exact matches, matches after Porter stemming, and matches based on WordNet synonymy. Once the alignment is found, the precision and recall are calculated based on the number of mapped unigrams (m) and the length of the candidate (c) and reference (r) texts, as m/c and m/r, respectively.

In addition to the weighted F-score, METEOR also includes a penalty function for incorrect word order, which takes into account the number of unaligned words that are between aligned words. This penalty function helps to ensure that the metric penalizes translations that are not only incorrect in terms of content but also in terms of word order. In general, METEOR is a useful metric for evaluating machine translation systems, as it takes into account both content and word order to provide a more accurate evaluation [24]. The author in [5] proposes the use of back-translation as a method for evaluating sign language processing (SLP). This approach involves translating the generated sign pose sequences back into spoken language. This method is not considered in current work.

The most popular metrics in the machine translation domain are BLEU and ROUGE. Those metrics are used in previous works for evaluations too. For comparison, in this work, in addition, the METEOR is calculated as well and uploaded to the Huggingface repository (see in 4.1 for details).

3 Problem statement

As mentioned above, there is no direct alignment between sign sequences and spoken language sentences. The direct production of signs using the Progressive Transformers gave not high metric (BLEU-4 = 10.51) as well. Even when the method with generation of Glosses in between is applied, it gave BLEU-4 = 9.68 [5]. Glosses production could be still improved with different seq2seq architectures described in Section 2. This work is dedicated to evaluate those possible solutions. When the translation into Glosses achieves better results, then the mapping from Gloss to sign will be dependent mostly on dataset dedicated for sign production.

Some pre-trained models trained on German and some other language pairs might be used for translation for our task too. If the results can be improved by fine-tuning such models will be checked in this work as well. In general, to fine-tune such encoder-decoder model, typically a pre-trained model that has been trained on a large dataset or a related task is applied. The pre-training phase helps the model learn general representations of the data. However, these pre-trained models may not perform optimally on a specific task or dataset due to domain differences or task-specific requirements.

During fine-tuning, the model's parameters are updated using a smaller task-specific dataset. For SLP this dataset contains examples of German sentences and their Gloss sequences representations. By training the model on this dataset, its parameters are adapted to better fit specifically to the task, thereby improving its performance. During the fine-tuning process, some layers of the model may be frozen to prevent them from being updated. This is especially common for lower-level layers, that capture more general features. By freezing these layers, the pre-trained knowledge is preserved and fine-tuning process is focused on the task-specific layers, which are closer to the output.

4 Computational experiment

4.1 Text to Gloss production

Firsty, the tokenization method is selected. As it is mentioned in Section 2.2 by leveraging pretrained embeddings, the model gains the advantage of tapping into an extensive corpus of knowledge, granting it a comprehensive understanding of the world. This access to a vast wealth of information empowers the model to acquire a broader and deeper comprehension of various linguistic nuances and semantic relationships. BERT, RoBERTa and ELECTRA tokenizers are applied for source sentences for the NMT task, where the ELECTRA showed the best results and selected for the further evaluations. For the target, which is only the sequence of Glosses and can be assumed as no context is important, the tokenizer is applied, which tokenizes a text into a sequence of alphabetic and non-alphabetic characters.

Encoder-decoder architecture with LSTM is implemented as a baseline. During each generation step, certain segments of the source information may hold more relevance than others. However, the decoder is tasked with extracting pertinent information from the unchanging representation, which can be a challenging task. Then the model with attention and with bidirectional GRU is evaluated. Attention allows the model to "focus" on distinct sections of the input at various steps. Note, that the bidirectional GRU does not give the possibility for online simultaneous translation. As the third method, the huggingface pre-trained transformer with almost 74 mln. parameters [25] is applied. This is a fine-tuned version of "Helsinki-NLP/opus-mt-de-en" model using the dataset wmt16 for the german-english pair of languages.

OPUS-Machine Translation (MT) – are building open translation services for the World. OPUS-MT is dedicated to its mission of delivering open translation services and tools that are entirely free from commercial interests and limitations.

It is engaged in a collaborative venture with the Wiki-Media Foundation to establish translation services for generating Wikipedia content in previously unexplored languages, drawing from more comprehensive resources primarily available in English. At present, the project boasts an impressive collection of over 1,000 pre-trained translation models, which can be freely accessed and utilized. The project's ongoing endeavours are primarily focused on enhancing translation quality, broadening language coverage, and prioritising specific test cases to evaluate the applicability of their approach. The models are built upon the cutting-edge transformer-based neural machine translation (NMT) technology. A reliable and efficient NMT toolbox for training and decoding purposes, called Marian-NMT is deployed. These models are trained using parallel corpora freely available in the extensive repository known as OPUS. The architecture adheres to a standard transformer setup, consisting of six self-attentive layers in both the encoder and decoder networks, with eight attention heads in each layer. The hyperparameters are selected in accordance with the general recommendations provided in the software documentation [26].

For evaluation BLEU scores on different n-gram granularities, namely BLEU 1, 2, 3 and 4 are utilized, in order to give readers a better perspective of the translation performance. For ROUGE the ROUGE- L F1 scores are reported here. These metrics allow to directly compare with the previous works, where the same dataset is used. Different splittings to train and test sets with different word length distributions gave ±2 points differences in BLEU-4 evaluations. From the Table 2 it can be seen, that with the pre-trained model the BLEU-4 is improved by 5 points and ROUGE-L F1 by around 7 points from the previous methods.

The results of the fine-tuned model is uploaded to the Huggingface repository. The translation can be tested directly on the website:

https://huggingface.co/musralina/helsinki-opus-de-en-fine-tuned-wmt16-finetuned-src-to-trg.

Table 2: Methods evaluation results comparison table.

Method	Number of param.	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L F1
Stoll: Enc-Dec with attention	Not available	50.67	32.25	21.54	15.26	48.10
Saunder: Progressive transformer, text2gloss part	Not available	55.18	37.10	26.24	19.10	54.55
Our: Enc-Dec model without attention and with LSTM	2,436,911	26.90	16.80	11.50	8.27	31.42
Our: Enc-Dec model with attention and with bidirectional GRU	3,053,743	39.27	28.54	21.79	17.04	44.81
Our: fine-tuned pre-trained Huggingface transformer	73,886,208	64.14	45.11	32.47	24.27	61.35

4.2 Gloss to sign production

After the relatively good results of Gloss production is achieved, keypoints of the gestures from the Glosses are produced.
There is a gesture animations using HamNoSys notation might be applied. For each word, its form in the HamNoSys notation can be assigned. This notation was developed to formally describe the position of the hands, facial expressions and body in space for sign languages by the Hamburg Academy of Sciences [27]. At the end of this processing, gestures are animated using the avatar. To translate from spoken to sign language, research uses CV methods. Such developments would help compose a parallel corpus of signed and spoken languages. But for

now, these sorts of models don't work well in "natural," non-lab environments where the lighting and ambiance aren't as good. So, in one of the latest articles, it was possible to achieve a quality of about 92%, but only for dactyl (gestures of letters and numbers), and not language gestures. This was also evaluated using the MNIST dataset to classify static 24 American alphabet characters and achieved the 95% of accuracy.
To identify and monitor hand movements, one can utilize HandDetector libraries. These libraries are part of the CVzone CV package, which simplifies the execution of image processing and AI functions. CVzone relies on OpenCV and Mediapipe libraries to perform its tasks. When using the HandDetector libraries, the hands are detected and tracked, and each hand is represented by a list of 21 landmarks. Each landmark is composed of x, y, and z values. x and y are the image's width and height respectively. z represents the landmark depth with the depth at the wrist being the origin, and the smaller the value the closer the landmark is to the camera. The magnitude of z uses roughly the same scale as x.

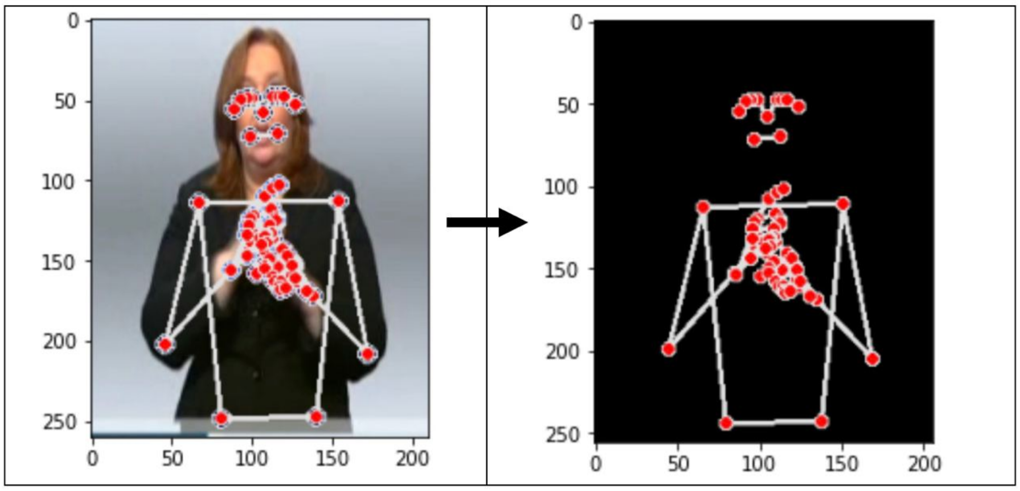


Figure 3: Example of the keypoints generation with mediapipe library.

The annotations of the images are done roughly, by calculating the average number of frames for each Gloss in sentence. 2D upper body joint and facial landmark positions are first extracted using MediaPipe library. 21 keypoints for each hand and 33 keypoints for pose are extracted from frames. See the example of keypoints detection in Fig. 3. When the lookup table was ready, the mapping from the generated Glosses to keypoints are performed. The results can be shown in videos and visually can be compared with original, otherwise expertise of the sign language interpreter is required for evaluation.

5 Summary and outlook

Fine-tuning the encoder-decoder model allows you to leverage the pre-trained knowledge while adapting the model to the specific task, leading to improved performance and better results. This was shown in Sign Language Production task, where the source is the German sentences and the target is the Glosses of German Sign Language. By fine-tuning the Huggingface pre-trained model, with small dataset (ca. 8000 pairs) a good BLEU 24.27 in translation into Glosses is achieved, which further could be used for avatar generation. The good results shows that, with more data there might be even better outcome. Additionally, for comparison, the simpler encoder-decoder models are evaluated too. By incorporating pretrained embeddings using the BERT tokenizers, the knowledge obtained from their original training data to our task-specific model effectively "transfered". This transfer allows the model to leverage the acquired knowledge encoded within the embeddings, benefiting from the insights and generalizations learned from the pre-training process. The hyperparameters as a random seed in splitting into train-test sets, which affects the average words distribution in sentences, number of epochs, batch size and optimisation methods as usually affected the results, and choice the proper hyperparameters is important as well.
At the end, the keypoints from the Glosses are generated and demonstrated. There are other challenges occur, as not smooth and not natural movements. This might be the scope of the further research.

References

[1] https://research.sign.mt, (April, 2023)

[2] Adrián Núñez-Marcos, Olatz Perez-de-Viñaspre, Gorka Labaka: A survey on Sign Language machine translation. Expert Systems With Applications, Spain (March, 2023)

[3] Navroz Kaur Kahlon, Williamjeet Singh: Machine translation from text to sign language: a systematic review. CVPR, USA (July, 2021)

[4] Songyao Jiang, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li and Yun Fu: A Skeleton Aware Multi-modal Sign Language Recognition. CVPR, USA (May, 2021)

[5] Ben Saunders, Necati Cihan Camgoz, Richard Bowden: Text2Sign: Continuous 3D Multi-Channel Sign Language Production via Progressive Transformers and Mixture Density Networks. International Journal of Computer Vision, (May, 2021)

[6] Michael Kipp, Alexis Heloir, Quan Nguyen: Sign Language Avatars: Animation and comprehensibility. International Workshop on Intelligent Virtual Agents (IVA), (2011)

[7] John McDonald: Automated Technique for Real-Time Production of Lifelike Animations of American Sign Language. Universal Access in the Information Society (UAIS), (2016)

[8] Caroline Chan: Everybody Dance Now. Proceedings of the IEEE International Conference on Computer Vision (CVPR), (2019)

[9] Donald J Berndt, James Clifford: Using Dynamic Time Warping to Find Patterns in Time Series. AAA1 Workshop on Knowledge Discovery in Databases (KDD), (1994)

[10] Lena Voita: https://lena-voita.github.io/nlp_course/transfer_learning, (April, 2023)

[11] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner: Deep contextualized word representations, CL, (March, 2018)

[12] Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, Tie-Yan Liu: Incorporating BERT into neural machine translation, ICLR, (Dec, 2019)

[13] Amanda Duarte: Cross-modal Neural Sign Language Translation. The 27th ACM International Conference, (October, 2019)

[14] Sujay S Kumar, Tenzin Wangyal, Varun Saboo, Ramamoorthy Srinath: Time Series Neural Networks for Real Time Sign Language Translation. The 17th IEEE International Conference on Machine Learning and Applications (ICMLA), (December, 2018)

[15] Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, Houqiang Li: Improving Sign Language Translation with Monolingual Data by Sign Back-Translation. CVPR, (May, 2021)

[16] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, Richard Bowden: Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. CVPR, (March, 2020)

[17] Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, Richard Bowden: Text2Sign: Towards Sign Language Production Using Neural Machine Translation and Generative Adversarial Networks. International Journal of Computer Vision, (January, 2020)

[18] Minh-Thang Luong, Hieu Pham, Christopher D. Manning: Effective Approaches to Attention-based Neural Machine Translation. EMNLP, (September, 2015)

[19] Zelinka, J., Kanis, J: Neural sign language synthesis: Words are our glossesNeural sign language synthesis: Words are our glosses. The IEEE winter conference on applications of computer vision (WACV), (May, 2020)

[20] Kriz Moses: https://medium.com/analytics-vidhya/encoder-decoder-seq2seq -models-clearly-explained-c34186fbf49b, (April, 2023)

[21] Biyi Fang, Jillian Co, Mi Zhang: DeepASL: Enabling Ubiquitous and Non-Intrusive Word and Sentence-Level Sign Language Translation. CVPR, (October, 2018)

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin: Attention Is All You Need, Computation and Language, (December, 2017)

[23] Oscar Koller, Hermann Ney, Richard Bowden: https://www-i6.informatik.rwth-aachen.de/publications/download/999/KollerOscarNey HermannBowdenRichardAutomaticAlignmentofHamNoSysSubunitsfor-ContinuousSign LanguageRecognition–2016.aux, (April, 2023)

[24] Desh Raj: https://medium.com/explorations-in-language-and-learning/ metrics-for-nlg-evaluation-c89b6a781054, (April, 2023)

[25] https://huggingface.co/mariav/helsinki-opus-de-en-fine-tuned-wmt16/tree/ main, (April, 2023)

[26] Joerg Tiedemann, Santhosh Thottingal: OPUS-MT – Building open translation services for the World. European Association for Machine Translation, (November, 2020)

[27] Rupinder Kaur, Parteek Kumar: HamNoSys generation system for sign language, ICACCI, (December, 2014)