UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

End-to-end Sign Language Translation with Stochastic and Natural Language Processing Transformers (BERT)

Trabajo de integración curricular presentado como requisito para la obtención del título de Ingeniero en Tecnologías de la Información

Author:

Andrés Fabricio Quelal Flores

Advisor:

Manuel Eugenio Morocho Cayamcela, Ph.D.

Urcuquí, Noviembre 2024

Autoría

Yo, **ANDRÉS FABRICIO QUELAL FLORES**, con cédula de identidad 1004533160, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autor/a del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, Noviembre de 2024.

Andrés Fabricio Quelal Flores

CI:1004533160

Autorización de publicación

Yo, **ANDRÉS FABRICIO QUELAL FLORES**, con cédula de identidad 1004533160, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación

Urcuquí, Noviembre de 2024.

Firmado electrónicamente por:
ANDRES FABRICIO
QUELAL FLORES

Andrés Fabricio Quelal Flores

CI:1004533160

Dedication

To my dear parents, who have always supported me in my career and worked hard to give my siblings and me a better life. To my siblings, thank you for all the funny moments we have shared and will share. To my family, who has been with me through good times and bad. A special dedication to my uncle Jorge, who was a kind and respectable person.

Thanks for all

Andrés Fabricio Quelal Flores

Acknowledgment

This work was made possible through the guidance of my advisor, Manuel Eugenio Morocho, Ph.D. Thank you for your support in helping me continue and complete my career.

I am grateful to all the professors at Yachay Tech, especially Isidro Amaro, Erick Cuenca, Kevin Chamorro, Rigoberto Fonseca, and Cristian Iza, for helping to solidify my background in computational science.

To all my dear friends, thank you, Mike, Leo, Sebas, Rash, Romi, Mabe, Washo, Mishel, and Vins, for the conversations, projects, and unforgettable moments at the university. Gabi, thank you for the enthusiasm and good music shared in each conversation. To Dánely and Ariana, the best IT students and friends, I am proud of all you have accomplished. To Sebas, my best friend, thank you for sharing good moments and funny anecdotes with me. To Nathy, my best friend, thank you for standing by me through good and bad times, sharing meaningful conversations, dinners, and parties since we began our university journey. To Monse, thank you for your unconditional support, helping me through challenging moments, filling my days with happiness and laughter, and always being there for me.

As I always say, *Gracias totales* to all the wonderful people in my life.

Andrés Fabricio Quelal Flores

Resumen

Las personas sordas y con dificultades auditivas utilizan el lenguaje de señas para comunicarse a través de expresiones faciales, gestos y señales visuales, los cuales son esenciales para superar las barreras de comunicación y participar plenamente en la sociedad. Al hablar de la comunicación en lenguaje de señas, nos referimos a un canal de comunicación visual compuesto por gestos de las manos y expresiones faciales, con sus propias reglas de pronunciación, orden de palabras y estructura de oraciones.

Con los recientes avances en visión por computadora y aprendizaje profundo, los sistemas de reconocimiento y traducción de lenguaje de señas implementan redes neuronales convolucionales (CNN), traducción automática neuronal (NMT) o transformadores de visión (ViT) como arquitecturas para la detección y clasificación. Debido a las potentes aplicaciones de los ViT, este trabajo propone el uso de una arquitectura de transformador para realizar tareas de reconocimiento y traducción en la traducción continua de lenguaje de señas (CSLT). Incorporamos un módulo de representaciones bidireccionales de codificador de transformadores (BERT) preentrenado como codificador y añadimos una función de activación novedosa llamada local winner-takes-all (LWTA) en el módulo decodificador.

El modelo se entrenó con el conjunto de datos RWTH-PHOENIX-Weather 2014 T, se evaluó utilizando protocolos de señas a texto (S2T) y se analizó con las métricas BLEU. La evaluación con la métrica BLEU-4 arrojó un valor de 23.83, superando en un promedio de 2.1 puntos a los modelos de referencia ejecutados y comparados en este trabajo.

Palabras clave:
Traducción de Lenguaje de Señas, Reconocimiento de Lenguaje de Señas, Transformador de Visión, Traducción Continua de Lenguaje de Señas, Ganador-Toma-Todo Local, Representaciones de Codificador Bidireccional de Transformadores.

Abstract

Deaf and hard-of-hearing individuals use sign language to communicate through facial expressions, gestures, and visual signals, which are essential for overcoming communication barriers and participating fully in society. When we discuss sign language communication, we refer to a visual communication channel composed of hand gestures and facial expressions, with its own rules for pronunciation, word order, and sentence structure.

With recent advances in computer vision and deep learning, sign language recognition and translation systems implement convolutional neural networks (CNNs), neural machine translation (NMT), or vision transformers (ViTs) as architectures for detection and classification. Due to the powerful applications of ViTs, this work proposes using a transformer architecture to perform recognition and translation tasks for continuous sign language translation (CSLT). We incorporate a bidirectional encoder representations from transformers (BERT) module pre-trained as an encoder and add a novel activation function called local winner-takes-all (LWTA) in the decoder module.

The model is trained on the RWTH-PHOENIX-Weather 2014 T dataset, evaluated using sign-to-text (S2T) protocols, and assessed with BLEU metrics. The BLEU-4 metric evaluation reports a value of 23.83, exceeding the baseline models tested and compared in this work by an average of 2.1 points.

Keywords:
Sign Language Translation, Sign Language Recognition, Vision Transformer, Continuous Sign Language Translation, Local Winner-Takes-All, Bidirectional Encoder Representations From Transformers.

Contents

Dedication **iii**

Acknowledgment **iv**

Resumen **v**

Abstract **vi**

Contents **vii**

List of Tables **x**

List of Figures **xi**

1 Introduction **1**

1.1 Background 1

1.2 Problem Statement 4

1.3 Objectives 4

1.3.1 General Objective 4

1.3.2 Specific Objectives 5

2 Theoretical Framework **6**

2.1 Computer Vision 6

2.1.1 Overview 6

2.1.2 Computer Vision Tasks 7

2.1.3 Convolutional Neural Networks (CNNs) 8

2.2 Vision Transformers (ViT) 10

2.2.1 Transformer 10

2.2.2 Vision in Transformer 13
2.3 Natural Language Processing (NLP) 14
2.3.1 Overview 14
2.3.2 NLP Pipeline 15
2.3.3 Neural Machine Translation (NMT) 15
2.3.4 Pre-trained NLP 17
2.3.5 Bidirectional Encoder Representations from Transformers (BERT) . 17
2.4 Sign Language in Computer Vision 18

3 State of the Art **20**
3.1 Sign Language Recognition 20
3.2 Sign Language Translation 23

4 Methodology **26**
4.1 Phases of Problem-Solving 26
4.1.1 Description of the Problem 26
4.1.2 Analysis of the Problem 26
4.2 Model Proposal 27
4.2.1 Spatial Embedding & Word Embedding 28
4.2.2 Encoder 29
4.2.3 Decoder 30
4.3 Experiment 33
4.3.1 Dataset Description 33
4.3.2 Metric Evaluation 34
4.3.3 Translation Protocols 34
4.3.4 Implementation 34
4.3.5 Experimental Setup 35

5 Results and Discussion **38**
5.1 Baseline Models Comparison 38
5.2 Train and Validation Results 41
5.3 Model Training Loss Analysis 44
5.4 BLEU-4 Metric score 44

5.5 Overfitting 45
5.6 Translation Prediction Results 47
5.7 Objectives and Results 50

6 Conclusions **52**

Bibliography **54**

List of Tables

2.1 N-gram types used in BLEU scores with examples. 16

3.1 Summary of sign language recognition related works. 22
3.2 Summary of sign language translation related works. 25

4.1 Transformer configuration settings. 36

5.1 Evaluation metric results (BLEU Scores) from baselines models. 39
5.2 Reduced BLEU Scores in Training and Validation Stages. 42
5.3 Generated spoken language translations by our sign model at the 7000th step. 48
5.4 Generated spoken language translations by our sign model at the 7100th step. 48
5.5 Generated spoken language translations by our sign model at the 7200th step. 48
5.6 Generated spoken language translations by our sign model at the 7300th step. 49
5.7 Generated spoken language translations by our sign model at the 7400th step. 49
5.8 Generated spoken language translations by our sign model at the 7500th step. 49
5.9 Generated spoken language translations by our sign model at the 7600th step. 50
5.10 Generated spoken language translations by our sign model at the 7700th step. 50
5.11 Objectives and Corresponding Results. 51

List of Figures

2.1 Computer vision tasks. 8
2.2 Typical CNN architecture. 9
2.3 The transformer architecture. 12
2.4 The vision in transformer architecture (ViT). 14
2.5 NLP pipeline illustration. 15
2.6 BERT pre-training and fine-tuning framework. 18
2.7 End-to-end sign language transformer. 19

4.1 Proposed sign language translation architecture. 27
4.2 LWTA network structure. 31

5.1 BLEU-1 score during the evaluation stage. 39
5.2 BLEU-2 score during the evaluation stage. 40
5.3 BLEU-3 score during the evaluation stage. 40
5.4 BLEU-4 score during the evaluation stage. 41
5.5 Consolidated training BLEU scores. 43
5.6 Consolidated validation BLEU scores. 43
5.7 Training loss graphic. 44
5.8 Training and validation BLEU-4 scores. 45
5.9 BLEU-4 score progression - Test 1. 46
5.10 BLEU-4 score progression - Test 2. 46
5.11 BLEU-4 score progression - Test 3. 47

Chapter 1

Introduction

1.1 Background

In a world that is becoming more interconnected, good communication is crucial for encouraging inclusivity and guaranteeing equal opportunities for everyone. However, those who are deaf or have hearing impairments face persistent difficulty in attaining efficient communication, which not only impacts their personal life but also hinders their ability to access education, employment, and social engagement.

The World Health Organization reports that around 430 million individuals, comprising 5% of the global population, experience total hearing impairment [1]. By 2050, it is estimated that the number of people with hearing loss worldwide will exceed 700 million. To surmount these obstacles, civilizations characterized by a significant prevalence of hearing impairment have wholeheartedly adopted sign language as a distinctive means of conveying and exchanging information. Sign languages exhibit distinct grammatical and syntactic norms that vary significantly across different countries, demonstrating that sign languages are not identical.

Sign language is a nonverbal communication tool primarily used in gestures, motions, and facial expressions for persons who are deaf or hard of hearing. It creates a complex and varied language system, including about three hundred sign languages used worldwide [2]. In contrast, studies to standardize sign language, such as the "Gestuto" project [3], which seeks to create a consistent sign language with easily learnable gestures, the complicated interaction of several sign languages has made it difficult to create uniform grammatical

rules.

Because of these difficulties, technology has become a primary tool for facilitating communication within the deaf community. This technology primarily involves using and exploring physical devices and computer vision systems [3].

Research shows that conventional techniques for sign language recognition used external hardware equipment, such as the Kinect sensor, as shown in the research conducted by Aly *et al.* [4], or sensor gloves, as explored in sign language recognition by Ahmed *et al.* [5]. Nevertheless, these techniques are costly and impractical for consumers' daily contacts.

Recent research has shown that there has been a transition to using vision systems rather than hardware devices. An instance of this transition is seen in the research conducted by Sakshi *et al* [6], whose publication introduces a computer vision system that utilizes convolutional neural networks (CNNs) to interpret sign language by recognizing hand gestures. Morocho-Cayamcela *et al.* [7] suggested enhancing the precision of ASL alphabet prediction by refining two CNNs (AlexNet and GoogLeNet) using atypical compensating approaches and transfer learning. Lomas *et al.* [8] conducted a study that examined the effectiveness of CNNs in recognizing sign language. The study also focused on mitigating overfitting problems by employing transfer learning methods. In addition, a groundbreaking sign language transformer was created by Camgoz *et al.* [9] has introduced revolutionary transformer-based methods for translating sign language from start to finish.

Given these researches in the sign language field discussed above, a new field that explores translation tasks is the Neural Machine Translation (NMT) task.

Considering the prior research and developments in sign language, a new and promising avenue of exploration is the application of NMT to the domain of sign language. NMT is a machine translation task between languages where deep neural network (DNN) models [10], CNNs, and specific NMT architectures [11] have been utilized. These models have significantly enhanced translation quality, often complemented by natural language processing (NLP) techniques.

These methodologies have played a pivotal role in achieving comprehensive translations, as demonstrated by studies such as the work by Camgoz *et al.* [12], which introduces the neural sign language translation (NSLT) models and the sign language production by Rastgoo *et al.* [13] using NMT models.

Beyond traditional language translating, NMT handles the particular difficulties of translating sign language in recognition and generation. This technology has benefited many facets of sign language, from recognition to translation to jobs involving sign language interpretation.

Vision transformers (ViT) [14] have marked a significant turning point in artificial intelligence since they have transformed machine perception and interpretation of visual input. Designed chiefly for processing sequential data such as text, these ViT represent a development of the original transformer architecture. Transformer architecture has been modified and refined in ViT to excel in visual tasks.

As mentioned earlier, transformer topologies' adaptability has resulted in their application in several artificial intelligence fields, from natural language processing to computer vision [9, 15]. These transformer implementations have made particularly noteworthy strides in sign language research.

Transformer structures have established the state of the art in many vision-related activities within sign language research. These chores cover sign language translation, sign language into written or spoken language and sign language recognition, in which computers learn to recognize and interpret sign language motions. Additionally, tasks requiring sign language interpretation, hand gesture detection, facial expression recognition, real-time interpretation, and even sign language generation have seen great success for transformer-based models.

One remarkable development within the transformer architecture domain is the integration of stochastic transformers into the realm of end-to-end sign language translation, as pioneered by Vosku *et al.* [13]. This innovative approach combines the effectiveness of stochastic modeling with the transformer's ability to understand complex patterns in sign language.

When translating sign language into words, researchers often use something called "glosses". These are quick, written notes that capture the meaning of a sign in regular language. Studies include work by Camgoz *et al.* [9] show that the integration of gloss mid-level implementation inside transformer systems has shown its capacity to minimize computational complexity and boost sign language translation (SLT) performance. These developments have generated original Sign2Gloss2Text models that are different from the

conventional Sign2Text models. Transformer designs have expanded SLT technology to unprecedented degrees by using annotations as an intermediary representation, enhancing the accessibility and inclusivity of sign language communication for a larger audience [16, 9, 17].

Overall, sign language tasks like recognition, production, and translation have implemented traditional and vision systems using novel architectures such as NMT and ViT. These technologies have pushed the boundaries of sign language research and significantly improved accessibility, inclusivity, and communication for the deaf and hard-of-hearing communities worldwide. Furthermore, the fusion of traditional sign language methodologies with cutting-edge AI and machine learning techniques shows promising results, as seen in studies that have utilized NLP models within the NMT framework or the utilization of Stochastic transformers.

1.2 Problem Statement

Neural machine translation models require much input data to achieve better inference performance. In particular, sign language recognition and translation tasks face the recurring data scarcity problem. Due to the laborious nature of sign language production, insufficient sign language datasets are available. We employ the latest transformer implementations in the sign language field to address these challenges. Specifically, we incorporate novel techniques, such as using pre-trained models as encoders and stochastic transformers, with the replacement of traditional rectified linear unit (ReLU) units by local winner-take-all (LWTA) as the activation function. Research in this field focuses on solving the costly production of sign language data and getting better performance results of sign translation metrics.

1.3 Objectives

1.3.1 General Objective

To implement a robust sign language transformer to achieve better BLEU metric results with an optimal architecture by leveraging new implementations, such as efficient activation

functions, and taking advantage of natural language processing models, like pre-trained models.

1.3.2 Specific Objectives

- Integrate pretrained models as encoders to enhance the understanding and processing of sign language inputs.
- Implement stochastic transformers with LWTA activation functions to improve model performance and efficiency.
- Evaluate and optimize different model configurations to identify the most effective architecture for sign language translation.
- Develop a transformer architecture that performs the translation task without intermediate gloss application.

Chapter 2

Theoretical Framework

2.1 Computer Vision

2.1.1 Overview

Visual perception is a straightforward way to understand the environment and its behavior, although other senses also play a role. Humans deduce spatial information faster from their visual senses. Distance, depth, and volume are readily identifiable to us. This capacity to understand visual info helps us to move and interact with our surroundings.

Computer vision aims to develop systems that can process, comprehend, and make decisions based on visual data. In a nutshell, computer vision aims to replicate human vision ability by enabling machines to interpret and understand visual information from the surrounding environment. Using techniques such as image recognition, object detection, and depth estimation, computer vision systems can analyze visual inputs and make informed decisions. These systems are employed in various applications, from autonomous vehicles and facial recognition to medical imaging and augmented reality, demonstrating their potential to revolutionize multiple fields, provide accurate analysis, and save time [18].

In practice, one example of computer vision is segmenting neuron structures from microscopic images. This task has been addressed using CNNs, particularly the U-Net architecture proposed by Ronneberger *et al.* [19]. Automating this process reduces the need for human experts, leading to significant time savings.

Computer vision has been the most active application area within deep learning research

in the last few years. Despite recent advances, understanding human-level images remains a significant challenge.

2.1.2 Computer Vision Tasks

Computer vision involves various tasks that allow machines to process and interpret visual data. Key tasks within this field include:

Image Classification

Image classification is a core component of computer vision. It involves assigning a label or category to an entire image. Deep learning, mainly through CNNs, has significantly advanced this field. Image classification is vital in various fields, such as recognizing objects in photographs, diagnosing medical conditions, and sorting images into specific categories (Fig 2.1).

Object Detection

Object detection goes beyond image classification by identifying objects within an image and determining their location using bounding boxes. This means it can locate and classify multiple objects in an image. This task is crucial for applications such as autonomous driving, where systems must detect and pinpoint entities like pedestrians, other vehicles, and obstacles. Popular techniques for object detection include Faster R-CNN [20], YOLO [21], and SSD [22].

Semantic/Instance Segmentation

Semantic segmentation involves labeling each pixel in an image with a class of the object, allowing for a detailed understanding of the image content. Instance segmentation extends beyond semantic segmentation by identifying and differentiating between individual instances of the same category. For example, in Figure 2.1, the instance segmentation image demonstrates how to distinguish different objects even if they belong to the same group. It clearly outlines each object—the cat in red, the dog in blue, and the duck in green.

This task is particularly beneficial for applications like medical imaging (e.g., differentiating various tissue types) and autonomous driving (e.g., interpreting road environments).

Architectures like U-Net [19], R-CNNs [23], F-CNNs [24], SegNet [25], and DeepLab [26] are commonly used for image segmentation.

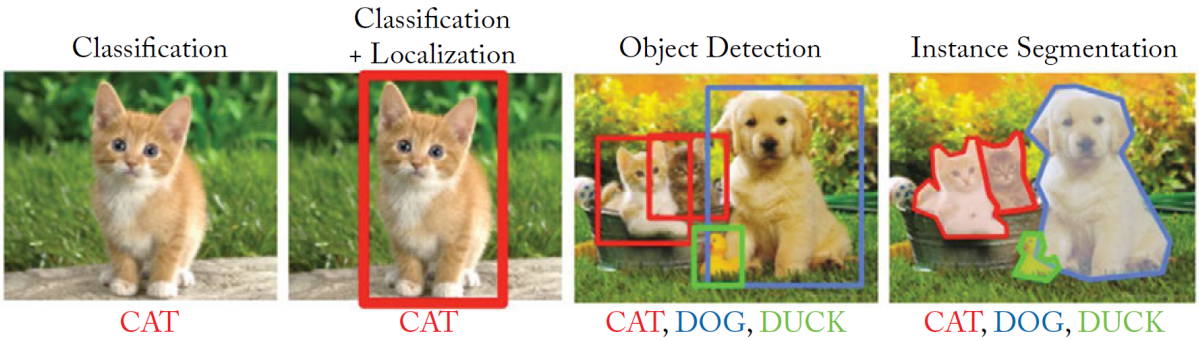

Figure 2.1: Computer vision tasks include identifying objects (classification), locating objects (localization), finding multiple objects (detection), and precisely outlining individual objects (segmentation). Taken from [27].

These and other proposed computer vision tasks aim to describe space through visual resources accurately. In particular, the tasks discussed above form the basis of computer vision and are integral to developing systems that can interpret and interact with the visual world.

2.1.3 Convolutional Neural Networks (CNNs)

Inspired by the human brain's visual processing system, CNNs introduced by LeCun *et al.* [28], have become a popular and powerful tool for handling complex visual tasks applied to computer vision systems [18]. CNNs are a deep learning model that handles data arranged in grids like images and videos. As the name suggests, this type of neural network uses a mathematical operation called "convolution."

A CNN architecture has several types of blocks and layers:

- **Convolutional Layer**: Considered the most essential layer of CNNs [27, 18], the convolutional layer extracts meaningful features from the input image through the convolutional operation that sets filters to the input image, producing feature maps that capture spatial hierarchies of features, such as edges, textures, and patterns [29]. Each filter slides over the input image and performs a dot product operation, highlighting specific patterns in the data.

- **Pooling Layers**: Due to increasing network dimension across each layers, the pooling layers helps to reduce the spatial dimensions of the feature maps, retaining the most important information while discarding less critical details. Pooling layers reduce the computational load and controlling overfitting. The most common type is max pooling, which takes the maximum value from a set of pixels within a defined window [29].

- **Fully Connected Layers**: Once the input image passes through the convolutional and pooling layers, the features extracted from this process are consolidated into a fully connected layer. Fully connected layers are typically used at the network's end to classify based on the extracted features.

- **Activation Function**: In the hidden layers, the CNN weights pass through a nonlinear activation function, allowing the model to learn the relationship between the input and the output. A widely used activation function is the ReLU. ReLU adds nonlinearity to the model, enabling it to learn complex patterns by converting the input into an output spanning zero to positive infinity [27].

This Figure 2.2 graphically explains the CNN architecture from the input layer to the output layer.

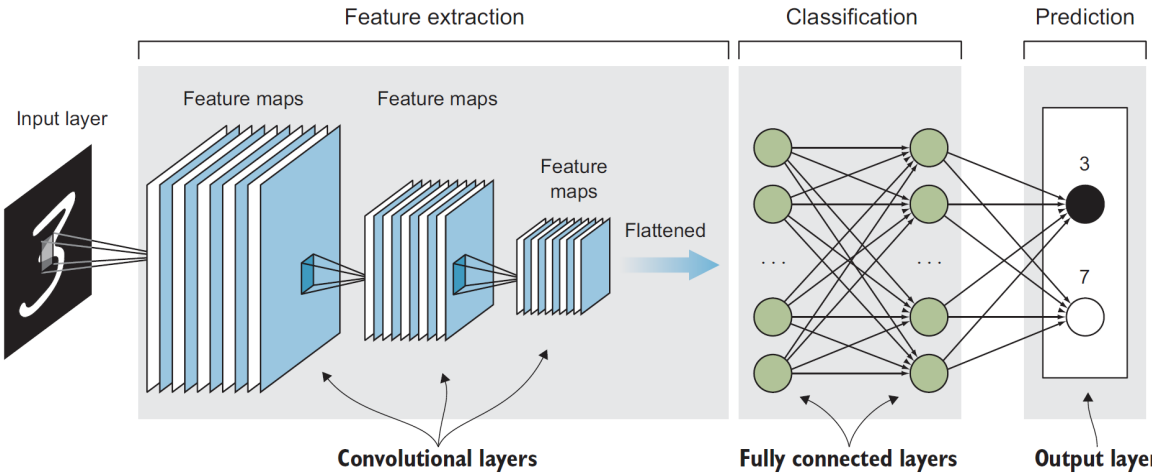

Figure 2.2: Typical CNN architecture illustrating the flow from the input layer through convolutional layers for feature extraction, followed by fully connected layers for classification, and ending with the output layer for prediction. Taken from [29].

CNN architectures like AlexNet [30], VGGNet [31], and ResNet [32] have set new performance standards in various computer vision competitions, highlighting the exceptional capabilities and strong performance of CNNs in tackling complex visual problems.

2.2 Vision Transformers (ViT)

2.2.1 Transformer

A pivotal moment in deep learning occurred in 2017 with the introduction of the transformer architecture by Vaswani *et al.* [15] from Google. This groundbreaking work, titled "Attention Is All You Need," revolutionized natural language processing by applying self-attention mechanisms to handle sequence data without recurrent networks. The performance results obtained from this work also demonstrated efficiency through faster training and reduced computational costs via parallelization [18].

Comparative studies between LSTM, RNN, and Transformer architectures have concluded the superior performance of the Transformer [33, 34, 35]. The Transformer architecture achieves stable results with faster training due to parallelization and the attention mechanism. These studies also highlight the limitations of RNNs in handling long-term dependencies [36, 33].

Transformer Architecture

This architecture's general composition is self-attention mechanisms, positional encodings, and encoder/decoder blocks. The following items explore each stage of the transformer architecture clearly.

- **Positional Encoding**: Since transformers do not inherently understand the order of input tokens, positional encodings store the position of a token in a dense vector, preserving the order of the input sequence. The resulting matrix of positional encodings, denoted as $\boldsymbol{PE}$, is mathematically defined as follows:

$$\boldsymbol{PE}_{p,i} = \begin{cases} \sin\left(p/10000^{i/d}\right) & \text{if } i \text{ is even} \\ \cos\left(p/10000^{(i-1)/d}\right) & \text{if } i \text{ is odd} \end{cases} \tag{2.1}$$

where p is the position of the token in the sentence, i^{th} is the dimension of the encoding, and d is the embedding dimension [18].

- **Self-Attention and Multi-Head attention Mechanism**: This mechanism allows the model to weigh the importance of different parts of the input sequence when making predictions, enabling it to capture long-range dependencies. In the encoder, the features vector of positional encoding passes to scaled dot-product attention equation in the attention block, as follows the equation:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{2.2}$$

where $\boldsymbol{Q}$ represents the query matrix, $\boldsymbol{K}$ represents the key matrix, $\boldsymbol{V}$ represents the value matrix, $\boldsymbol{K}^T$ is the transpose of the $\boldsymbol{K}$ matrix, and d_k is the dimensionality of the key vectors.

Once the $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ word input sentences are passed through an attention mechanism, the result feeds into the normalization layer, obtaining self-attention weights. This process is repeated to form the Multi-head attention block, expressed by the following mathematical expression:

$$\begin{aligned} &\text{MultiHead}\ (\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \ \text{Concat}\,(\ \text{head}_1, \ldots,\ \text{head}_\text{h})\,\boldsymbol{W}^O \\ &\text{where head}_\text{i} = \text{Attention}\left(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V\right) \end{aligned} \tag{2.3}$$

where $\boldsymbol{W}$ represents all the matrices of learnable parameters.

- **Feed-Forward Network and Normalization Layer**: Each transformer block incorporates a feed-forward neural network to process the information derived from the attention mechanism further. These networks typically consist of two linear layers with a ReLU activation in between [15]. Layer normalization ensures that the data going into each layer is similar in size, which helps the model learn better. Residual connections create shortcuts in the network, allowing information to skip over some layers. This makes it easier to train profound models.

Like the encoder, the decoder block comprises multi-head attention blocks, normalization layers, and feed-forward networks. However, the decoder also receives input from the encoder block's output. The target sentence words also pass through a word embedding process before entering the masked multi-head attention block [18]. These steps are repeated until the word prediction is obtained.

The following image shows the transformer architecture proposed by Vaswani *et al.*:

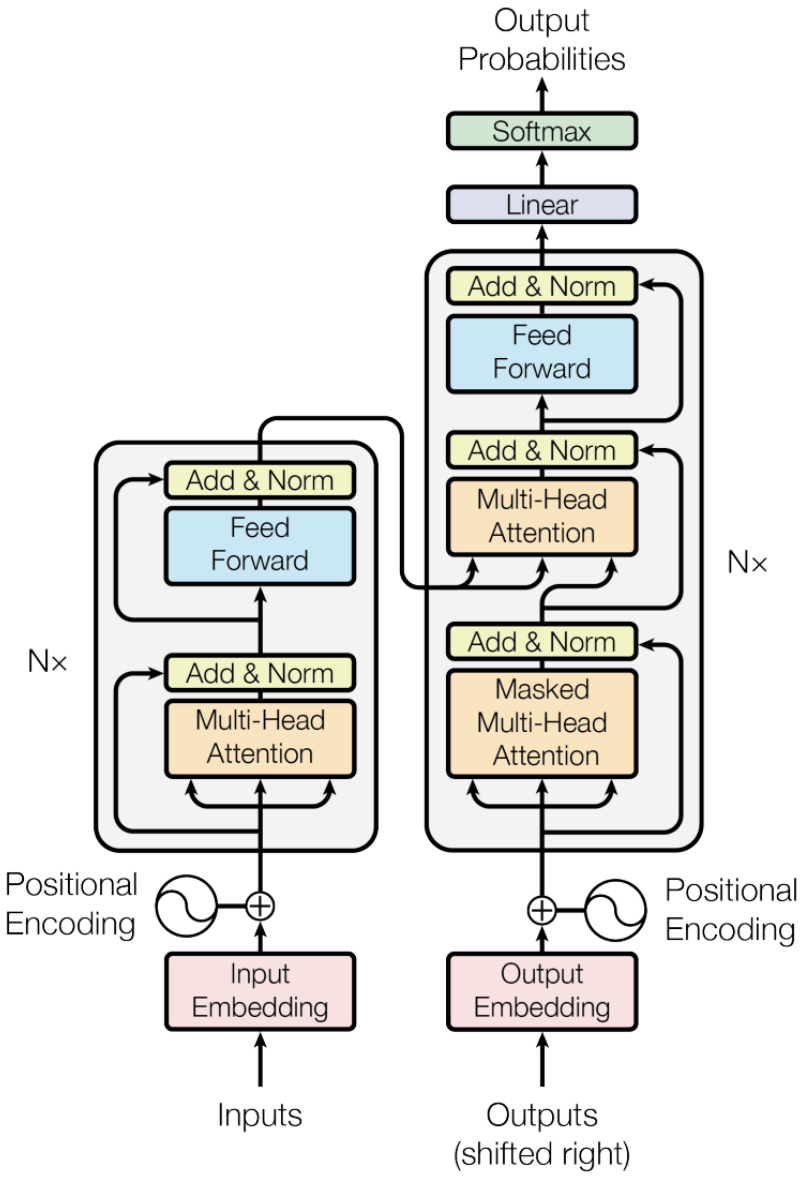

Figure 2.3: The transformer design consists of an encoder and a decoder, both of which have multi-head attention and feed-forward layers. Positional encodings are incorporated into the input and output embeddings. The encoder transforms the input into intermediate representations, while the decoder utilizes these representations along with the shifted outputs to generate predictions. Taken from [18].

2.2.2 Vision in Transformer

First introduced by Dosovitskiy *et al.* [14], vision transformers (ViT) (Fig 2.4) adapt the transformer architecture (Fig.2.3) to process images as input data for computer vision tasks such as image classification.

Following the same structure as the transformer architecture (Fig.2.3), the ViT architecture proposes the following new modules:

- **Patch and Positional Embedding**: The input image is divided into fixed-sized squares or patches. These patches are mathematically transformed into numerical representations called patch embeddings. Similar to word embeddings in the transformer architecture, positional encodings are added to these patch embeddings to provide information about the spatial positions of the patches within the image, preserving the spatial relationships for image understanding [14].

- **Self-Attention Mechanism**: Like the traditional transformer (Fig.2.3), the self-attention mechanism enables the model to establish relationships among all patches during prediction, capturing dependencies and contextual relationships within the image.

- **Classification Head**: The output from the final transformer block is passed through a multi-layer perceptron (MLP) head, which typically involves a fully connected layer to produce the final predictions [14].

ViTs have shown competitive performance on various benchmarks, sometimes surpassing traditional CNNs in tasks such as image classification, object detection, and segmentation [37]. Transformer architectures in vision systems have become a hot topic for research. Due to the evident advantages of ViTs, new benchmark results are increasingly composed of transformer architectures.

The next image show the ViT architecture proposed [14].

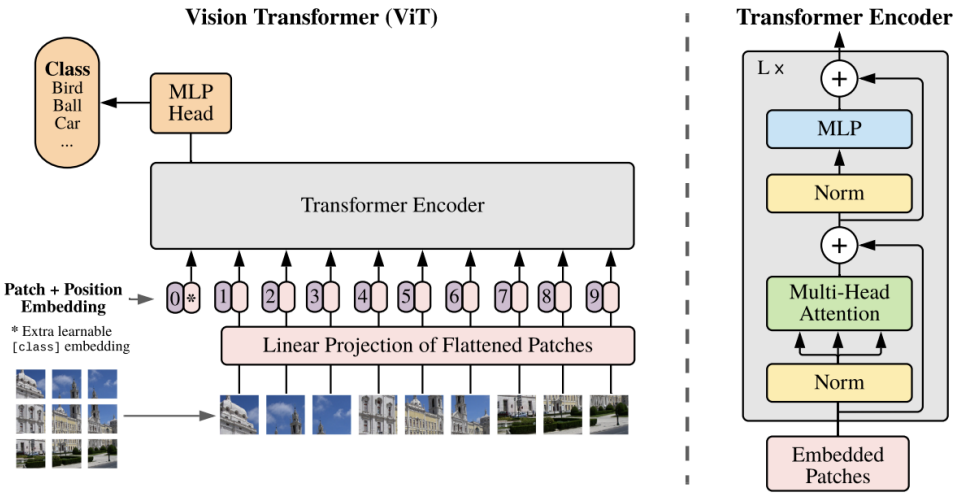

Figure 2.4: The Vision Transformer (ViT) architecture involves partitioning an image into patches, applying linear projection to them, incorporating positional embeddings, and then passing them through a transformer encoder. The ultimate representation is transferred to a Multilayer Perceptron (MLP) head for categorization. Taken from [14].

2.3 Natural Language Processing (NLP)

2.3.1 Overview

Natural language processing (NLP) is a multidisciplinary field of artificial intelligence that facilitates the interaction between computers and human language. NLP aims to enable machines to understand, interpret, and generate human language in a meaningful and valuable manner [38]. NLP encompasses text categorization, language translation, speech recognition, and sentiment analysis tasks. NLP is divided into two subfields: natural language understanding (NLU), which focuses on analyzing the semantics or intended meaning of text, and natural language generation (NLG), which concentrates on generating text by a machine.

Different NLP models, such as sequence-to-sequence models, utilize recurrent neural networks and excel at language translation. Autoregressive models have significantly advanced NLP capabilities by sequentially generating text, expanding the field's potential. Transformer models, leveraging self-attention, capture complex language structures.

From the introduction of the transformer architecture with the attention mechanism [15] and the application of transfer learning in LSTM (ULMFiT) [39] in 2018-2019, ad-

vancements in NLP have accelerated, resulting in powerful and efficient NLP systems. Models such as bidirectional encoder representations from transformers (BERT) [40] and generative pre-trained transformer (GPT) [41] have established new standards in various NLP tasks [42, 43], showcasing the effectiveness of transformers in comprehending and generating human language.

2.3.2 NLP Pipeline

Preparing raw text data for NLP analysis involves crucial steps like text pre-processing and feature extraction [40]. Following the flow of Figure 2.5, from data acquisition, the text pre-processing pipeline begins with tokenization, where the text is split into smaller units called tokens. This is followed by normalization, which involves converting all characters to lowercase, removing punctuation, and performing stemming and lemmatization to reduce words to their root or base forms. Stop words, which are common words like "the," "is", and "in" that do not add significant meaning, are also removed to focus on more meaningful words.

After text pre-processing, feature extraction converts tokens into numerical representations. Techniques for this conversion include bag of words (BoW) [44], term frequency-inverse document frequency (TF-IDF) [45], and advanced word embeddings [46]. This sequence of processes results in clean, organized data that can be effectively used in NLP models.

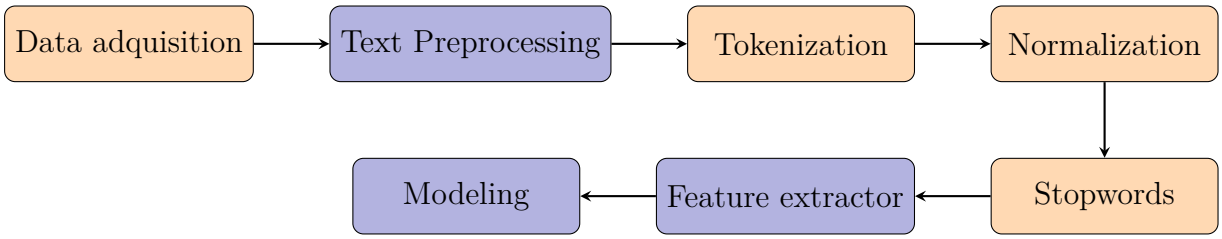

Figure 2.5: NLP pipeline illustrating the steps of data acquisition, text pre-processing, tokenization, normalization, stopwords removal, feature extraction, and modeling.

2.3.3 Neural Machine Translation (NMT)

Neural Machine Translation (NMT) utilizes neural networks to translate entire sentences simultaneously. Unlike traditional statistical machine translation methods, NMT excels at

capturing complex language patterns and dependencies, producing more fluid and human-like translations [47]. NMT architectures typically consist of encoder-decoder frameworks with attention mechanisms, embeddings, and classification layers.

Evaluation Metrics in Neural Machine Translation

Evaluating the quality of translations produced by NMT models is essential to determine their effectiveness. One of the most widely used metrics for this purpose is the Bilingual Evaluation Understudy (BLEU) score [47, 48]. The BLEU metric, introduced by Papineni *et al.* [49], measures the similarity between a machine-generated translation and one or more reference translations by calculating the precision of n-grams, which are contiguous sequences of n words.

The BLEU score is calculated using the following formula:

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log P_n\right) \tag{2.4}$$

where BP is the Brevity Penalty to account for short translations, w_n is the weight for n-gram, and P_n is the precision of n-grams; and N represents the maximum n-gram length considered, generally up to 4.

In BLEU scoring, BLEU-1 (Unigram Precision) measures individual word accuracy, while BLEU-2 to BLEU-4 evaluate coherence through sequences of 2 to 4 words. Table 2.1 provides examples of these n-gram types, illustrating how each BLEU score level captures increasingly complex language patterns with longer sequences.

Table 2.1: N-gram types used in BLEU scores with examples.

BLEU Score	**N-gram Type**	**Example**
BLEU-1	Unigram (1-gram)	*"The", "weather", "is", "sunny"*
BLEU-2	Bigram (2-gram)	*"The weather", "weather is", "is sunny"*
BLEU-3	Trigram (3-gram)	*"The weather is", "weather is sunny"*
BLEU-4	4-gram	*"The weather is sunny"*

Higher BLEU scores indicate better translation quality by reflecting a closer match to the reference translations. The BLEU score ranges from 0 (no overlap) to 1 (perfect match) [49]. For ease of interpretation, these values are generally multiplied by 100 and

presented as percentages. Typically, a score above 30% is considered good, while scores above 50% suggest high-quality translations approaching human-level performance [50, 51]. This metric is widely recognized for providing an objective and quantifiable measure of translation accuracy and fluency, making it a standard in machine translation evaluation [52, 53, 54]

In addition to BLEU, other metrics like WER and SER are essential for evaluating sign language recognition models. WER, adapted from speech recognition, measures word-level errors in model output [55, 56, 57, 58, 59, 60, 9, 61], while SER specifically assesses sign recognition accuracy [62]. In translation tasks, metrics such as ROUGE and METEOR are also commonly used; ROUGE evaluates n-gram overlap, emphasizing recall and precision, while METEOR rewards semantic similarity by considering stemming and synonyms [63, 59, 57, 11, 60, 16, 64, 62].

In the context of sign language translation, using these metrics is crucial for assessing how effectively the NMT model translates visual gestures into coherent and accurate text.

2.3.4 Pre-trained NLP

Due to the increasing number of parameters in deep learning models, robust and extensive datasets are essential for practical training. A groundbreaking development in NMT models is the introduction of pre-trained models. These models alleviate the dependency on massive amounts of task-specific data by providing strong initial representations, accelerating training, aliviate overfitting, and often improving overall performance [65].

2.3.5 Bidirectional Encoder Representations from Transformers (BERT)

Introduced by Devlin *et al.* [66] in 2018, BERT is a machine learning model designed to create deep contextual representations directly from unlabeled text data. It has become revolutionary due to its improved performance and ability to solve various NLP tasks [67, 68, 69].

BERT comprises two main frameworks, as shown in the Figure 2.6: pre-training and fine-tuning. The pre-training framework involves training BERT on large datasets to ac-

quire contextualized word embeddings, which capture the meaning of words based on their context within sentences. This framework allows BERT to predict the following sentence or the missing word in a sentence. Fine-tuning involves adapting the pre-trained BERT model to a specific task (sentiment analysis, speech-to-text translation, masked language modeling) using labeled data. Unlike traditional transformers that use separate encoder and decoder blocks, BERT employs a single encoder block to handle pre-training and fine-tuning tasks.

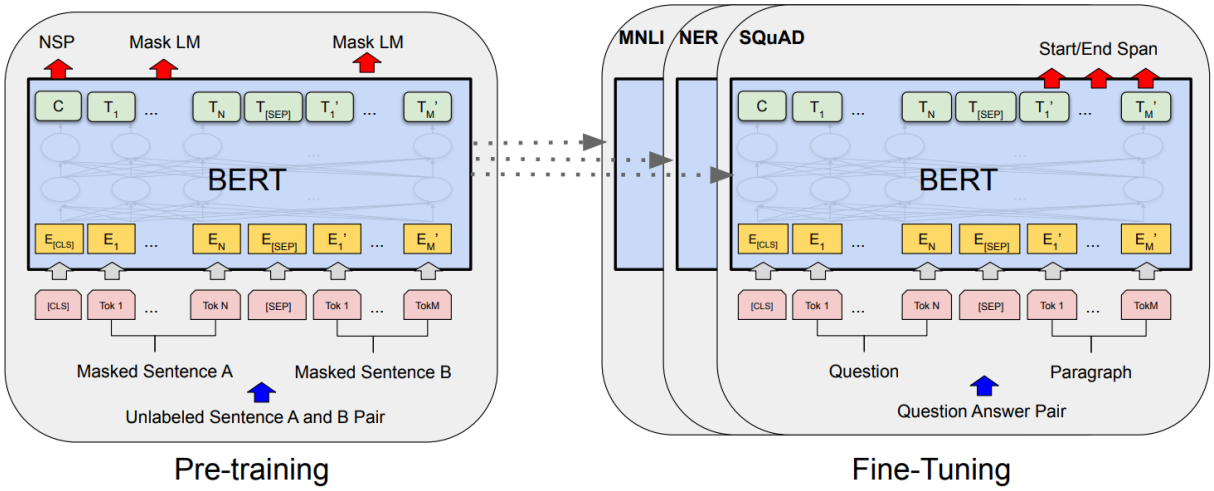

Figure 2.6: The BERT framework involves two main steps: pre-training and fine-tuning. Pre-training includes masked language modeling (MLM) and next sentence prediction (NSP) tasks. Fine-tuning adjusts the model to perform well on specific tasks by training it on labeled data. Taken from [66].

2.4 Sign Language in Computer Vision

In sign language communication, computer vision uses technology tools to lower the communication barrier separating deaf people from hearing ones. Hand gesture identification, depth detection, 3D model recognition, deep learning-based recognition tasks, and pose-based methods [10] are among the several computer vision techniques routinely investigated to improve systems for sign language categorization and detection.

There are two main approaches to sign language recognition and translation: physical devices and vision-based systems.

Physical devices like gloves and sensors record hand and finger motion and location.

For instance, Tao *et al.* [70] employed american sign language gesture recognition using a Kinect sensor. Likewise, Aly *et al.* [71] captured hand motion using a principal component analysis network on the Kinect sensor. These gadgets convert bodily motions into digital data that is fit for analysis and interpretation as sign language. These devices are effective but often inconvenient and costly, which hinders their use.

Cameras and computer vision algorithms are employed in vision-based systems to analyze and comprehend sign language. For instance, Hu *et al.* [72] proposed the SignBERT+ framework, which captures gestures using hand estimation from sign videos. Another notable work is the "Sign language transformer" by Camgoz *et al.* [9], which proposes a pure vision system for end-to-end sign language translation and recognition using transformer architecture. Figure 2.7 shows the proposed sign language model. These systems analyze sign videos and employ various methods to understand and convert the signs into text or spoken language. Vision-based systems are more user-friendly and accessible since they do not require additional hardware beyond a standard camera.

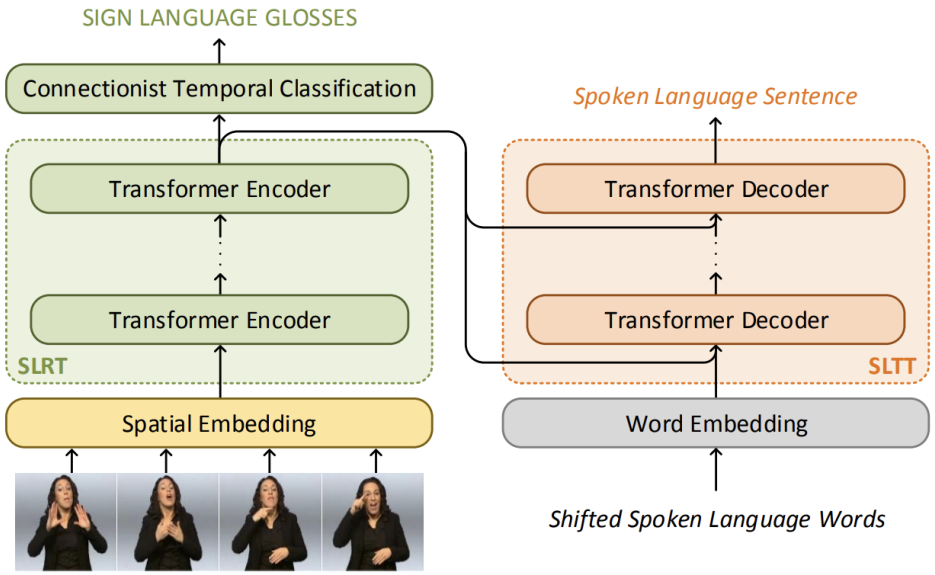

Figure 2.7: The authors Camgoz *et al.* have presented an end-to-end sign language transformer. The novel model employs spatial embedding to recognize sign language and word embedding to translate spoken language. It utilizes transformer encoders and decoders to connect both tasks. Taken from [9].

Chapter 3

State of the Art

3.1 Sign Language Recognition

Sign language recognition involves the process of recognizing and translating sign language gestures into written or spoken language. This task combines computer vision and natural language processing techniques. The primary evaluation metric in this task is the word error rate (WER).

To evaluate advancements in sign language translation, researchers rely on established benchmarks. Among these, RWTH-PHOENIX-Weather 2014 (proposed by koller *et al.* [73]) and RWTH-PHOENIX-Weather 2014 T (proposed by Camgoz *et al.* [12].) are widely used.

Capturing hand gestures and facial movements in sign language involves understanding complex temporal relationships. Learning the spatial sequence in a recognition task is a novel approach proposed by Min *et al.* [74], introducing the "Deep Radial Embedding" method. This method effectively captures complex temporal relationships. Similarly, Hu *et al.* [72] introduce SignBERT+, which captures hand pose information. This model implements a pre-trained BERT model that effectively captures gesture state and spatial-temporal position, allowing it to learn robust representations. Due to the dynamic nature of sign language, human body trajectories are also considered to identify a sign. Hu *et al.* [75] explore this idea by creating "correlation maps" or the correlation network (CorrNet) that compare nearby frames, essentially tracking hand and face movements. This allows CorrNet to gain a more complete understanding of the signs being conveyed, achieving

superior accuracy on standard datasets. In this context, it is important to interpret all sign sequences captured from sign language videos over time. Lu *et al.* [76] introduced TCNet, which integrates a trajectory module to track hand and face movements across video frames and a Correlation Module to highlight significant areas within each frame. TCNet method has the ability to capture extended sign sequences, highlighting the potential of integrating trajectory and correlation data for future SLR improvements.

Regarding the field of continuous sign language recognition (CSLR), novel models have been proposed, such as the work by Hao *et al.* [12]. They introduced the Self-mutual knowledge distillation (SMKD) method. SMKD leverages a two-stage architecture, enforcing the visual and contextual modules to capture long-term dependencies. By incorporating SMKD, the authors demonstrated the importance of the visual information treatment.

The current state of the art in CSLR is led by the work proposed by Ahn *et al.* [77]. To achieve effective spatial and dynamic feature extraction, they utilize the SlowFast network, which employs a two-pathway design: a "slow lane" that captures the broader context of the signs and a "fast lane" that focuses on the finer details of rapid hand motions. This dual-path approach helps gain a more comprehensive understanding of the sign language process, ultimately demonstrating that their network outperforms previous methods on the PHOENIX14, PHOENIX14-T, and CSL-Daily datasets.

On the other hand, there is a broad range of sign language categories, with a particularly interesting field of study being the ankara university turkish sign language (AUTSL) Dataset [78]. The next work represents the top benchmark result reported on the "Test Set Recognition Rate" metric. Ryumin *et al.* [79] implemented two deep neural networks for sign gesture recognition and audio-visual speech recognition, taking into account the audio, visual, and speech scenery. Although this work does not entirely focus on sign language applications, the efforts to implement an efficient computer vision system for gesture recognition are evident.

Framework	Contribution	Dataset	Architecture	Metric	Cited
Deep Radial Embedding	RadialCTC to enforce sequence features on a hypersphere, enhancing modeling of temporal relationships	PHOENIX14-T, Seq-MNIST, Scene Text Recognition	Transformer-based architecture with radial constraint on embeddings	WER, mAP	[74]
SignBERT+	Introduces a self-supervised pre-training framework incorporating hand pose information	RWTH-PHOENIXT	Pretrained BERT model with hand-model-aware embedding layer	PCK, AUC, P-I, P-C, WER	[72]
Correlation Network (CorrNet)	Develops correlation maps to capture trajectories in body regions	PHOENIX14, PHOENIX 2014T	CNN, BiLSTM, Correlation module	WER	[75]
TCNet	Combines a Trajectory Module and a Correlation Module to improve the comprehension of extended sign sequences	PHOENIX14, PHOENIX14-T, CSL, CSL-Daily	CNN, BiLSTM, Trajectory and correlation module	WER	[76]
Self-Mutual Knowledge Distillation (SMKD)	Utilizes a two-stage architecture to capture spatial and long-term dependencies	PHOENIX14, PHOENIX14-T	2D-CNN, 1D-CNN, BiLSTM	WER	[12]
SlowFast Network	Employs a two-pathway design to capture spatial and dynamic information	PHOENIX14, PHOENIX14-T, CSL-Daily	Two-pathway network architecture	WER	[77]
Audio-Visual Speech and Gesture Recognition	Combines gesture recognition and audio-visual speech recognition for mobile devices	LRW, AUTSL	Two deep neural networks for speech and gesture recognition	-	[79]

Table 3.1: Summary of sign language recognition related works.

3.2 Sign Language Translation

Sign language translation involves generating spoken language sentences from sign language video representations. Recent advancements in sign language translation leverage the use of ViT to address traditional machine translation challenges, achieving significant improvements in accuracy and performance in this field.

Deep learning approaches to sign language translation can be categorized into three main type: using just computer vision, using physical devices, or combining both for a comprehensive system. Nowadays, computer vision system applications are popular because it's flexible and gives great results.

Being part of the first category, the frozen pre-trained transformer proposed by De Coster *et al.* [56] presents a novel implementation in the SLT field. To address the issues of limited training data and overfitting, the authors leveraged pre-trained language models. They consolidated a sign language transformer architecture by freezing parameters to prevent overfitting. This approach benefits from the knowledge encoded in the pre-trained models. The authors reported improved results using sign to text (ST2) translation protocols, as evidenced by higher BLEU metric scores.

A common topic of discussion is the use of traditional intermediate representations, such as gloss notation. Zhou *et al.* [80] propose a gloss-free transformer architecture called "Gloss-free SLT based on visual-language pre-training" (GFSLT-VLP). By leveraging visual language pre-training, the authors developed a model that can directly map visual sign language input to textual output without the need for intervening gloss representations. The results demonstrate an enhancement in their model's performance.

Similarly, Voskou *et al.* [52] share the proposal of not using explicit glosses. They introduce a novel transformer architecture, called the stochastic transformer, which incorporates stochasticity and competition among neurons to improve model performance. By using linear competing units and stochastic weights, the authors achieve a robust and efficient model.

Focusing on sign language recognition and processing, Chen *et al.* [81] employ a novel two-stream network framework. One stream processes raw video data, while the other handles keypoint information extracted from the video. By combining these two streams,

the model effectively captures both low-level visual features and high-level semantic information. The results of this paper, reported on the "papers with code" state-of-the-art platform, place this model in second position in the best benchmark results for the BLEU-4 metric.

The best BLEU-4 benchmark was reported by the work of Guan *et al.* [82], achieving state-of-the-art performance on the PHOENIX14-T dataset. The multi-stream keypoint attention network (MSKA) was the model proposed, built entirely with attention mechanism modules. These modules focus on keypoint information extracted from video sequences, utilizing multiple streams to process different aspects of the keypoint data, such as hand, face, and body movements.

Taking information from the sign language recognition state-of-the-art section, the SignBERT+ model developed by Hu *et al.* [72] once again excels in the task of sign language translation. Through several experiments, the authors evaluated and reported results for continuous sign language translation, demonstrating a high score of 25.70 in BLEU-4 evaluation. Due to the implementation of self-supervised pretraining and fine-tuning, this framework achieves new state-of-the-art performances in the sign language field.

Now, considering the How2Sign dataset (american sign language dataset by [83]), the first baseline framework reported on this dataset is the work proposed by Tarrés *et al.* [84]. The authors propose a Transformer-based model trained on I3D video features to translate sign language into text. The I3D serves as a feature extractor that tokenizes the input video stream to feed the proposed transformer architecture. This work establishes a baseline for studying and exploring more alternatives for sign language frameworks using the How2Sign dataset.

Framework	Contribution	Dataset	Architecture	Metric	Cited
Frozen Pretrained Transformer	Address issues of limited training data and overfitting by leveraging pretrained language models.	RWTH-PHOENIX-Weather 2014T	Transformer-based architecture with frozen parameters and pretrained model.	BLEU	[56]
Gloss-free SLT based on Visual-Language Pretraining	Directly map visual sign language input to textual output without intervening gloss representations.	PHOENIX14-T, CSL-Daily	Gloss-free transformer architecture with visual language pretraining	BLEU, ROUGE	[80]
Stochastic Transformer with LWTA	Improve model performance by incorporating stochasticity and competition among neurons.	PHOENIX14-T	Transformer architecture with Linear Competing Units and stochastic weights.	BLEU	[52]
TwoStream-SLT Network	Combine raw video data and keypoint information to capture low-level visual features and high-level semantic information	PHOENIX14, PHOENIX14-T, CSL-Daily	TwoStream-SLR and TwoStream-SLT	BLEU, ROUGE	[81]
Multi-Stream Keypoint Attention Network (MSKA)	Utilize multiple keypoint attention modules to process different aspects of keypoint data.	PHOENIX14, PHOENIX14-T, CSL-Daily	Multi-stream keypoint attention network focusing on attention mechanism modules.	BLEU, ROUGE	[82]
SignBERT+	Introduces a self-supervised pre-training framework incorporating hand pose information.	RWTH-PHOENIXT	Pretrained BERT model with hand-model-aware embedding layer	BLEU, ROUGE	[72]
Transformer-based Model Trained on I3D Video Features	Establish a baseline for studying and exploring alternatives for sign language frameworks using How2Sign dataset	How2Sign	Transformer architecture with I3D feature extractor	rBLEU, BLEU,	[84]

Table 3.2: Summary of sign language translation related works.

Chapter 4

Methodology

4.1 Phases of Problem-Solving

4.1.1 Description of the Problem

Related works in end-to-end sign language translation [52, 11, 56, 72] highlight significant issues with dataset size and gloss production in machine translation tasks. There are only a few publicly available large-scale datasets in sign language production [85]. The complexity of grammatical and linguistic production in this field presents a major barrier to creating enriched datasets [56, 52]. Mid-level representations (glosses) included in SLT datasets are particularly laborious to obtain in large corpora data. Gloss annotations require expert knowledge and extensive manual effort, which limits the scalability of creating comprehensive datasets. This scarcity of gloss annotations limits the domain coverage of translation datasets, thereby hindering real-world applications [86].

4.1.2 Analysis of the Problem

The creation of gloss sequences is labor-intensive and time-consuming, requiring expert knowledge to accurately annotate each sign. Furthermore, video preprocessing in sign language translation poses additional challenges as it involves complex computer vision tasks to accurately capture and interpret the dynamic gestures in the videos. These challenges contribute to the overall difficulty of developing robust sign language translation systems. To overcome these challenges, solutions are being implemented to address the limitations of weak datasets [11, 56], improve hand gesture recognition [72], enhance sign language

production [63, 12, 67], and manage computational costs in real-time video analytics [68].

4.2 Model Proposal

Inspired by the stochastic transformers with linear competing units proposed by Voskou *et al.* [52], the pre-trained transformers for neural sign language translation introduced by De Coster *et al.* [11, 56], and the first sign language transformer that integrates recognition and translation into a unified framework by Camgoz *et al.* [9], our model aims to build a robust transformer architecture as shown in Figure 4.1.

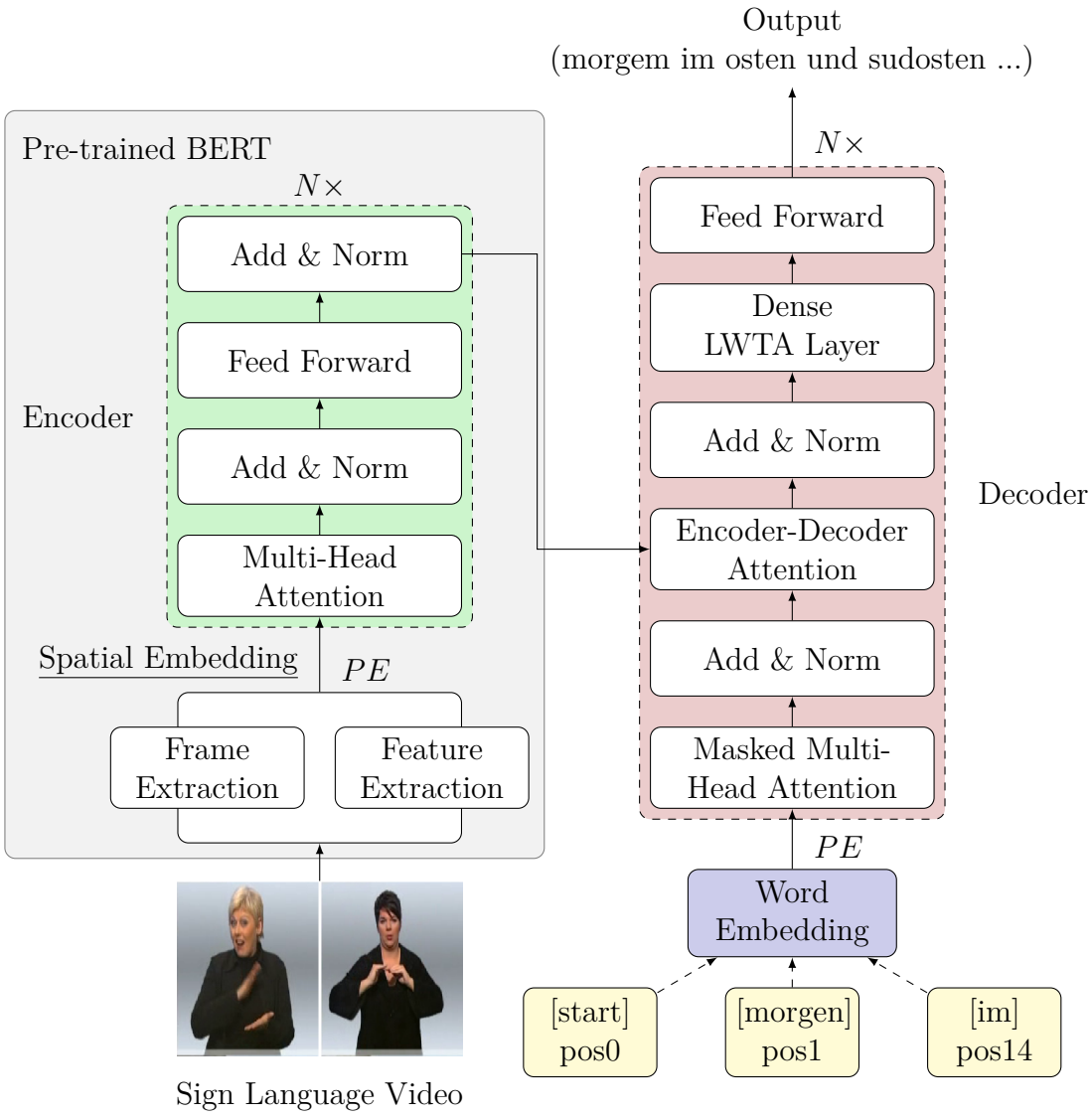

Figure 4.1: This is an overview of our proposed architecture for sign language translation, which uses a BERT encoder and stochastic transformers with LWTA activation implemented in the decoder.

4.2.1 Spatial Embedding & Word Embedding

For our model proposal, we begin by employing spatial embedding and word embedding techniques to handle video frames and target spoken language words, respectively.

Spatial Embedding

The input modality involves obtaining a series of features from complete video frames. These features are extracted on a per-frame basis from a pre-trained inception network [87] using a 2D CNNs [12]. Each video frame x_t is processed through the pre-trained CNN to generate a feature vector f_t getting a sequence of feature vectors $\{f_1, f_2, \ldots, f_t\}$ as follows:

$$f_t = \text{SpatialEmbedding}(x_t) \tag{4.1}$$

Word Embedding

In sign language translation, word embedding converts target spoken language words into numerical representations. Initially, words are represented as high-dimensional, sparse one-hot vectors. These vectors are then projected into a lower-dimensional dense space using a fully connected layer, resulting in dense vectors $\{g_1, g_2, \ldots, g_u\}$, expressed as:

$$g_u = \text{WordEmbedding}(y_u) \tag{4.2}$$

where y_u is the one-hot vector.

Integration with Positional Encoding

In the transformer architecture, it is essential to employ positional information within sequences to maintain the context. The spatial embedding f_t and word embedding g_u representations obtained from the CNN are passed through a positional encoding (PE) method, as shown below:

$$\hat{f}_t = f_t + \text{PE}(t) \tag{4.3}$$

$$\hat{g}_u = g_u + \mathrm{PE}(u) \tag{4.4}$$

The spatial embeddings $\hat{f}_t$ are fed into the transformer encoder, while the word embeddings $\hat{g}_u$ are fed into the transformer decoder.

4.2.2 Encoder

Instead of a traditional encoder transformer module, we use a pre-trained language model, BERT, combined with the encoder module to perform machine translation from sign language. The encoder consists of 4 blocks: multi-head attention, add & norm, feed forward, and add & norm.

Pre-trained Transformer: BERT

While sign language data is often limited, this model leverages pre-trained language models to enhance the encoder module's learning process [88, 56, 11, 69]. Since pre-trained models like BERT are typically trained on written text data, adaptations might be necessary for sign language translation.

We employ a pre-trained BERT model in the encoder. The spatial embeddings ($\hat{f}_t$) are fed into this encoder. BERT, trained on large text datasets, converts these embeddings into contextually rich representations. These are processed through multi-head attention mechanisms, allowing the model to focus on different input parts and capture diverse features for sign language translation. The multi-head attention mechanism is expressed in this equation 2.2.

The following equation illustrates how the BERT encoder transforms the spatial embeddings:

$$\hat{h}_t = \mathrm{BERT}(\hat{f}_t, \mathrm{PE}_t) \tag{4.5}$$

Here, t denotes the time step (frame index), $\hat{f}_t$ represents the spatial embedding for frame t, PE_t is the positional encoding for frame t (if used), and $\hat{h}_t$ denotes the contextually enriched representation produced by the BERT encoder.

4.2.3 Decoder

Once the encoder model was defined, we began to design the decoder transformer architecture. The decoder consists of six blocks: masked multi-head attention, add & norm, encoder-decoder attention, add & norm, dense LWTA layer, and feed forward.

To generate the output sequence, the decoder processes the target spoken language words, which are first converted into word embeddings $\hat{g}_u$. These embeddings are input into the masked self-attention layer of the decoder. The masked self-attention mechanism ensures each position in the decoder can only attend to previous positions, preserving the autoregressive property.

The output from the masked self-attention layer is then passed through subsequent layers of the decoder, ultimately generating the translated spoken language sentence.

The decoder module includes a dense LWTA layer [52] above the encoder-decoder attention layer, as shown in Figure 4.1. This replaces the traditional deterministic activation functions in encoder/decoder modules, such as ReLU layers, where the activation is given by:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

This novel activation function has been implemented in data modeling and dataset construction, replacing the ReLU layer implementation. It promotes generalization, improves robustness, and enhances performance, achieving superior results in neural machine translation tasks, sign language translation tasks, meta-learning, and network robustness [89, 85, 90, 55, 91].

Stochastic Transformer: Local Winner-Take-All

Local Winner-Take-All Networks

To establish local competition among neurons, we consider a network N with n input neurons $(x_1, \ldots, x_n)$ and n output neurons $(y_1, \ldots, y_n)$. The network N is divided into B blocks arranged in layers: $N = \{B_1, B_2, \ldots, B_B\}$, where each block B_i (for $i = 1$ to B) consists of m neurons: $B_i = \{N_{i1}, N_{i2}, \ldots, N_{im}\}$, where N_{ij} represents the j-th neuron in

the i-th block.

Each block B_i produces an output vector $\mathbf{y}_i$ based on local interactions among neuron activations:

$$\mathbf{y}_i = g(a_i^1, a_i^2, \ldots, a_i^m), \tag{4.7}$$

where $g(\cdot)$ is the interaction function within each block.

The activation a_i^j of the j-th neuron in block i is given by:

$$a_i^j = f(\mathbf{w}_{ij}^T \mathbf{u}_i), \tag{4.8}$$

where $\mathbf{u}_i$ is the input vector to block i from the previous layer, $\mathbf{w}_{ij}$ is the weight vector for neuron j in block i, and $f(\cdot)$ is a non-linear activation function.

The resulting activations $\mathbf{y}_i$ serve as inputs to the next layer.

$$y_i^j = \begin{cases} a_i^j & \text{if } a_i^j \geq a_i^k, \forall k = 1, \ldots, m \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

This study employs a winner-take-all function, detailed graphically in Figure 4.2.

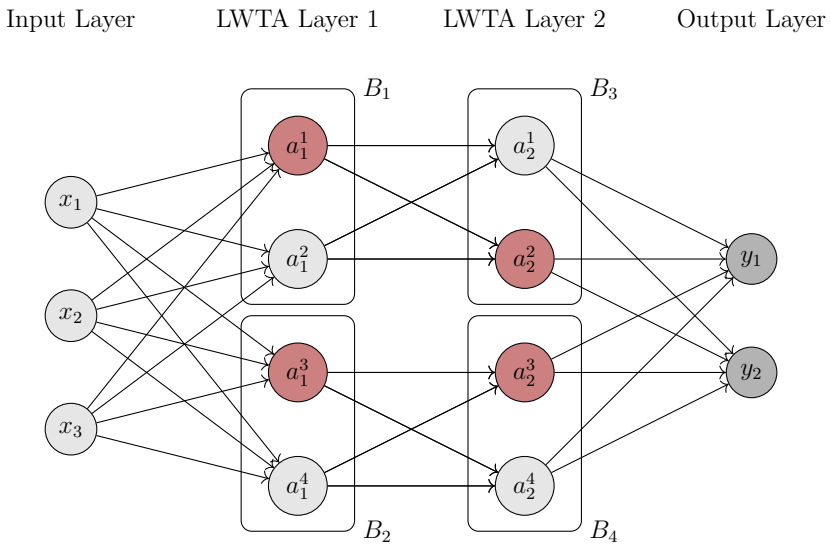

Figure 4.2: Local Winner-Take-All (LWTA) networks are composed of three layers: input, LWTA, and output. Within each LWTA layer, neurons are organized into clusters, with only the neuron exhibiting the highest level of activation in each cluster being chosen to transmit its signal to the subsequent layer.

Stochastic Local Winner-Take-All

Now, let $\boldsymbol{x} \in \mathbb{R}^J$ be the input vector with J features. We consider a three-dimensional matrix $\boldsymbol{W} \in \mathbb{R}^{J \times B \times U}$ representing the related weights, where B is the number of blocks and U are the linear competing units used. Following the Voskou *et al.* [52] LWTA approach, the activations $h_{b,u}$ of each u^{th} linear unit within the b^{th} block are computed as follows:

$$h_{b,u} = \sum_{j=1}^{J} w_{j,b,u} \cdot x_j \in \mathbb{R} \tag{4.10}$$

These activations are then passed through a softmax function to obtain the concatenated activation probability values.

Introducing the principles of stochastic competition, the output vectors $\boldsymbol{y} \in \mathbb{R}^{B \cdot U}$ of the competition encoded by LWTA layers are passed through a discrete latent vector $\boldsymbol{\xi}_b \in \text{one\_hot}(U)$ that corresponds to the winner unit among the U components. Thus, the winner unit determined by the stochastic LWTA operation is represented as:

$$y_{b,u} = \xi_{b,u} \sum_{j=1}^{J} \left(w_{j,b,u} \cdot x_j \right) \in \mathbb{R} \tag{4.11}$$

$$y_{b,u} = \xi_{b,u} \cdot h_{b,u} \tag{4.12}$$

where $h_{b,u}$ represents the linear activation of the u^{th} unit in block b before applying the one-hot encoding, $\xi_{b,u}$ is the element of the one-hot encoded vector indicating the winning unit within block b, and $y_{b,u}$ is the activation of the winning unit in the b-th block after incorporating the stochastic competition.

The winner unit indicator ξ_b shows that a unit with higher linear activation has a greater chance of being selected as the winner. This creates a data-driven competition within an LWTA block, where the probability of winning increases with the unit's linear output. This relationship is mathematically represented by the discrete distribution given in Equation 4.13.

$$q\left(\boldsymbol{\xi}_b\right) = \text{ Discrete } \left(\boldsymbol{\xi}_b \mid \text{softmax} \left(\sum_{j=1}^{J} \left[w_{j,b,u} \right]_{u=1}^{U} \cdot x_j \right) \right) \tag{4.13}$$

The LWTA stochastic mechanism introduces sparsity in neural activations by allowing only the most competitive units to be active. This reduces computational load and improves generalization by mitigating overfitting. This approach captures more discriminative features from the input data, which is crucial for handling the intricate details in sign language videos.

4.3 Experiment

4.3.1 Dataset Description

A crucial component of our end-to-end sign language translation model is the selection of an appropriate sign language dataset for recognition and translation tasks. For this work, the RWTH-PHOENIX-Weather 2014 T dataset was selected. Introduced by Camgoz *et al.* [12], this dataset is a comprehensive parallel corpus of German sign language translations based on the Phoenix-2014 Dataset [92]. It includes sign-gloss annotations, sign language videos, and corresponding German translations, all derived specifically from weather forecast broadcasts on a German public television station.

The RWTH-PHOENIX-Weather 2014 T dataset includes:

- **Videos**: 684 video sequences recorded between 2009 and 2011, each with a resolution of 210 x 260 pixels and a frame rate of 25 frames per second. These videos capture weather forecasts presented by sign language interpreters.
- **Segments**: The dataset is divided into three subsets: training with 7,096 segments, development with 519 segments, and testing with 642 segments, to facilitate systematic training and evaluation of models.
- **Sentences**: It consists of 8,257 parallel sentences in German sign language, each paired with its corresponding German text translation.
- **Gloss Composition**: Each sentence is annotated with glosses, which are textual representations of the signs used. The dataset includes 1,066 unique glosses for training, 393 for development, and 411 for testing. These glosses represent the different signs used across the dataset.

- **Vocabulary Size**: The dataset comprises 1,066 unique glosses for sign language and 2,887 unique words for German text, supporting extensive translation and recognition tasks.

- **Frames**: A total of 947,756 frames are included, providing detailed visual data for each second of the signing process. These frames are divided into 827,354 for training, 55,775 for development, and 64,627 for testing, ensuring a comprehensive dataset for model training and evaluation.

4.3.2 Metric Evaluation

The BLEU metric is used as the primary reference for assessing the performance of our sign language translation model during the training and validation stages. Throughout our evaluation, we track BLEU-1, BLEU-2, BLEU-3, and BLEU-4 scores, focusing on BLEU-4 due to its higher accuracy in reflecting translation quality. This focus on BLEU-4 allows us to ensure that our model captures individual words or short phrases accurately and maintains the coherence and meaning of longer text segments.

4.3.3 Translation Protocols

Further evaluation protocols such as sign to text (S2T), gloss to text (G2T), sign to gloss to text (S2G2T), and sign to gloss and text (S2(G+T)) have been proposed for sign language experimental evaluations. These protocols aim to better evaluate and improve translation models by considering various stages and representations in the translation process. Due to our proposed SLT transformer architecture, we use the S2T protocol instead of G2T, S2G2T, or S2(G+T). Implementing the BERT pre-trained encoder bypasses the need for gloss annotations by directly mapping sign language videos to spoken language text. This means the model bypasses the sign language representation to text without intermediary gloss representation.

4.3.4 Implementation

The main code was developed as a new adaptation of the JoeyNMT toolkit [93], combining the pre-trained transformers by [11, 56] and the stochastic transformer by [52] for sign

language translation. This modified version integrates the pre-trained BERT encoder with a decoder transformer utilizing the LWTA activation function. It was implemented in the PyTorch framework, and the results and graphical visualizations were produced using the TensorBoard framework.

The computational resources used in this experiment included an NVIDIA RTX 3060 6GB GPU, 24GB RAM, and a Ryzen 9 6900HS processor. The programming environment was set up using Visual Studio Code.

4.3.5 Experimental Setup

Baseline Models

To compare the performance of our model, we run and compare it with the following baseline models:

- Stochastic Transformer Networks by Voskou *et al.* [52] (S. Transformer)
- Leveraging Frozen Pretrained Written Language Models by De Coster *et al.* [11, 56] (FP. Transformer)

Following the S2T translation protocols, these models were executed locally using the computational resources described above. Table 4.1 explains the configuration settings used for the baseline models and our transformer configuration. These baseline configurations were taken from the best benchmark results obtained from the S2T Stochastic and BERT2RND experiments by [52, 56]. Essential values in our architecture, such as the LWTA units, were selected based on [52], where several experiments demonstrated that the value 4 was the most efficient, achieving the best BLEU score reported in 2022. BERT configuration values [11] are based on the best BLEU-4 values reported in 2021, obtained from three experiments combining BERT and MBART models.

Table 4.1: Transformer configuration settings.

Settings	**FP. Transformer**	**S. Transformer**	**Our model**
		Data	
Feature Size	1024	1024	1024
Max Sentence Length	400	400	400
		Training	
Eval Metric	BLEU, ROUGE, CHRF	BLEU	BLEU
Optimizer	Adam	Adam	AdamW
Learning Rate	0.0003	0.001	0.001
Batch Size	32	32	64
Epochs	5000000	500	80
		Model	
Encoder			
Type	BERT	Transformer	BERT
Pretrained Name	bert-base-uncased	-	bert-base-uncased
Layers	2	2	1
Attention Heads	12	8	12
Embedding Size	768	512	768
Hidden Layer Size	768	512	768
Feed-Forward Size	2048	2048	3072
Dropout	0.1	0.2	0.3
Decoder			
Type	Transformer	Transformer	Transformer
Layers	3	2	2
Attention Heads	8	8	8
Activation	-	LWTA	Dense LWTA
LWTA Competitors	-	4	4
Embedding Size	768	512	768
Hidden Layer Size	768	512	768
Feed-Forward Size	2048	2048	2048
Dropout	0.1	0.2	0.3

Training and Evaluation

During the training and evaluation stages, the learning rate establishes the early stopping of the model. The training stage stops if the model does not achieve better results. Due to computational resource constraints, the final result underwent training with checkpoints established at each step, following the configuration described by [93].

Training Setup

Key configurations and parameters are outlined below.

Data Configuration: The input data feature size is set to 1024, and the maximum sentence length is 400 tokens. Glosses input are discarded.

Training Parameters: We use BLEU as the primary evaluation metric. The AdamW optimizer adjusts the model's weights with a learning rate 0.001. A batch size 64 enhances computational efficiency, and the model is trained for 80 epochs.

Encoder: Our encoder utilizes a pre-trained BERT model (bert-base-uncased) to leverage rich contextual representations. It consists of one layer with 12 attention heads, an embedding size of 768, and a hidden layer of 768. The feed-forward layers are sized at 3072, with a dropout rate of 0.3 to prevent overfitting.

Decoder: The decoder is based on the Transformer architecture with two layers and eight attention heads. It employs a dense LWTA activation function with four competitors to promote sparsity and improve generalization. The decoder shares the same embedding size (768), hidden layer size (768), and feed-forward size (2048) as the encoder, with a dropout rate of 0.3.

Chapter 5

Results and Discussion

This chapter presents the results of our model's performance. The transformer configuration values were adjusted throughout the training and evaluation stages to achieve optimal performance. In the first section, we show the comparison results of baseline models. The second section details the results of various experiments based on configuration changes. In the final section, we present our model's translation prediction results.

5.1 Baseline Models Comparison

We compared our model's performance to two baseline models: a Frozen Pre-trained Transformer (FP Transformer) and a Stochastic Transformer (S Transformer). The training process for each model took approximately 7 hours and involved 8,000 training/evaluation steps. Notably, training for the FP Transformer stopped early (at step 4900) due to a lack of improvement.

Table 5.1 summarizes the maximum BLEU scores (BLEU-1 to BLEU-4) achieved by all three models. Our model consistently outperforms the baselines across all BLEU metrics, achieving a BLEU-1 score of 51.48, surpassing the FP Transformer (46.65) and S Transformer (48.82). Similarly, in BLEU-2 and BLEU-3 scores, our model leads, demonstrating its ability to capture individual word translations (unigrams) and short phrases (bigrams, trigrams). The most significant advantage of our model is evident in the BLEU-4 score (23.83), highlighting its effectiveness in capturing the nuances and contextual depth crucial for accurate sign language translation. In contrast, the FP Transformer (20.79) and S Transformer (22.67) fall short in this metric.

Table 5.1: Evaluation metric results (BLEU Scores) from baselines models.

Baseline models	Evaluation			
	BLEU-1	BLEU-2	BLEU-3	BLEU-4
FP Transfomer	46.65	33.05	25.60	20.79
S Transformer	48.82	35.68	27.89	22.67
Our model	51.48	37.38	29.12	**23.83**

The stochastic transformer network does not display training results when running the main code from the official repository. This is the primary reason why Table 5.1 only shows evaluation results.

Figures 5.1 to 5.4 visually represent the BLEU scores. While the S Transformer initially exhibits a rapid rise in BLEU-4 (Figure 5.4), our model demonstrates a more consistent upward trend throughout training, achieving the best performance across all BLEU metrics. Our model's approach, which incorporates a dense LWTA activation function in the decoder and a pre-trained BERT encoder, has proven to be highly effective in capturing the complexities of sign language translation. This observation underscores the impressive capabilities of our model.

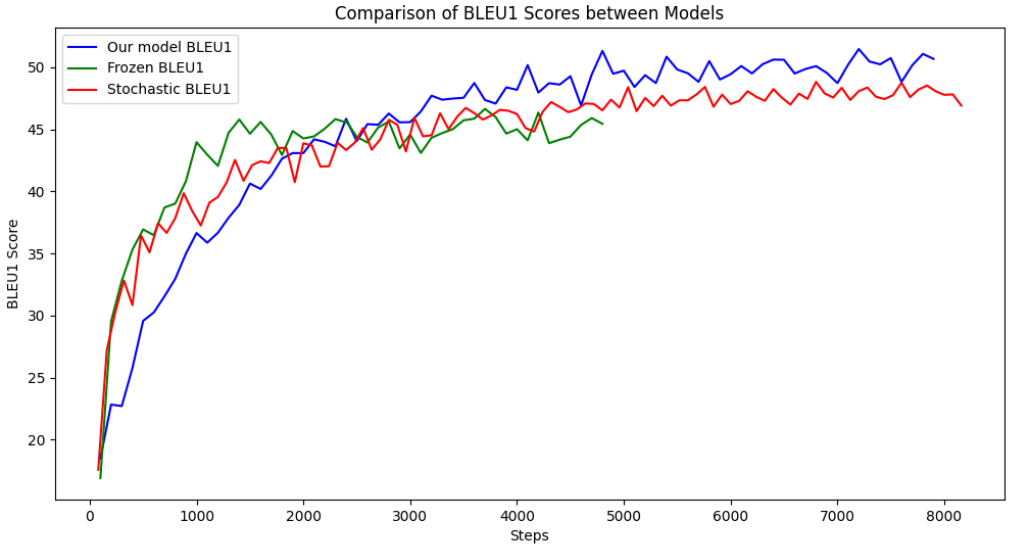

Figure 5.1: BLEU-1 score during the evaluation stage.

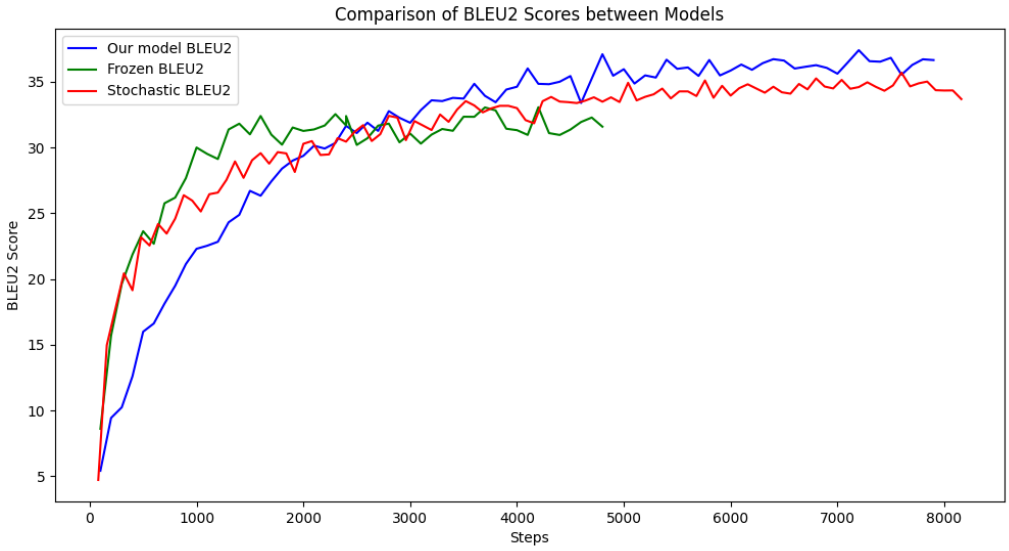

Figure 5.2: BLEU-2 score during the evaluation stage.

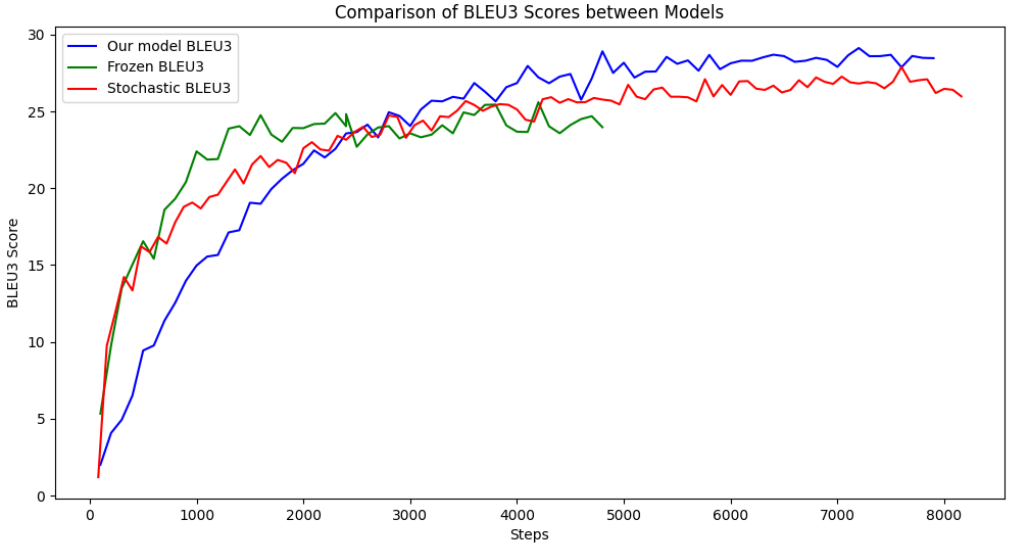

Figure 5.3: BLEU-3 score during the evaluation stage.

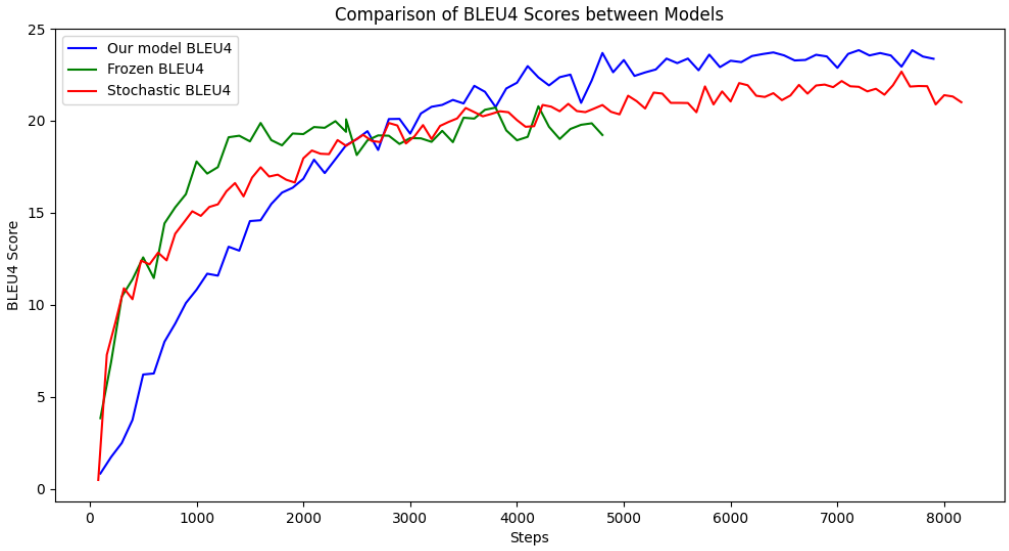

Figure 5.4: BLEU-4 score during the evaluation stage.

Baseline studies [52, 56] using the S2T approach report BLEU-4 scores of 22.25 and 23.65 for the FP Transformer and S Transformer, respectively. However, when we implemented these models with our computational resources, we obtained different results, as shown in Table 5.1. These variations likely stem from differences in computational resources and the specific setup environments used in our experiments compared to the previous studies.

5.2 Train and Validation Results

Throughout our model's training and validation process, we generated the reduced Table 5.2 for the BLEU metric scores. Due to the extensive data, the table displays results for every 500 steps only.

Our sign language translation model, configured as shown in Table 4.1, demonstrates consistent improvement in both training and validation BLEU scores (BLEU-1 to BLEU-4) throughout the training process, as evidenced by the results in Table 5.2. This indicates effective learning from the training data and good generalization to unseen data. For instance, at step 7200, we achieved training/validation BLEU-4 scores of 30.58/23.83, respectively.

Table 5.2: Reduced BLEU Scores in Training and Validation Stages.

Step	BLEU-1		BLEU-2		BLEU-3		BLEU-4	
	Train	Valid	Train	Valid	Train	Valid	Train	Valid
0	18.50	18.50	5.38	5.42	1.82	0.84	0.53	0.00
500	29.15	29.58	15.67	15.99	9.01	9.44	5.73	6.21
1000	36.83	36.66	22.64	22.29	14.90	14.97	10.56	10.82
1500	42.35	40.64	28.36	26.70	20.34	19.06	15.55	14.55
2000	44.90	43.09	31.25	29.36	23.17	21.59	18.17	16.84
2500	44.85	44.06	32.59	31.09	24.76	23.65	19.83	18.99
3000	48.30	45.59	34.68	31.87	26.51	24.05	21.30	19.30
3500	49.80	47.55	36.33	33.71	28.21	25.83	22.95	20.93
4000	51.98	48.19	38.56	34.61	30.42	26.84	25.11	22.06
4500	53.89	49.29	40.50	35.42	32.30	27.43	26.85	22.50
5000	54.37	49.73	41.08	35.94	34.45	28.17	27.50	23.29
5500	55.01	49.83	41.48	36.67	35.46	28.10	28.56	23.12
6000	55.14	49.45	41.97	35.84	36.14	28.13	28.23	23.25
6500	56.39	50.61	43.17	36.60	37.32	28.60	29.37	23.55
7000	55.70	50.22	42.64	36.48	34.46	27.90	28.87	22.87
7200	57.81	51.48	44.57	37.38	36.26	29.12	30.58	23.83
7500	57.67	50.75	44.71	36.81	36.51	28.68	30.89	23.54
8000	57.67	50.69	44.71	36.64	36.51	28.46	30.89	23.36

Graphically, Figures 5.5 and 5.6 display the BLEU scores on training and validation data across 8,000 training steps. As expected, the training BLEU scores (Figure 5.5) exhibit a steady rise across all metrics (BLEU-1 to BLEU-4), indicating successful learning from the data. The validation BLEU scores (Figure 5.6) demonstrate a similar positive trend to the training scores (Figure 5.5), albeit at a slightly lower rate. This difference reflects the model's exposure to unseen data during validation, testing its generalization capabilities. Nevertheless, the validation scores still reach significant values (BLEU-1 $\sim$ 50, BLEU-4 $\sim$ 20), highlighting our model's robustness in translating sign language under different conditions.

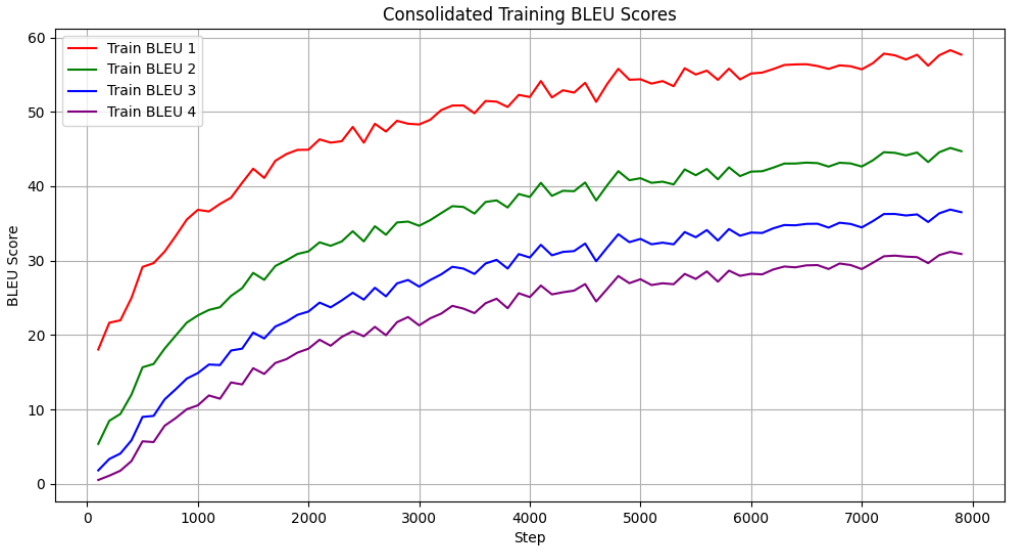

Figure 5.5: Consolidated training BLEU scores.

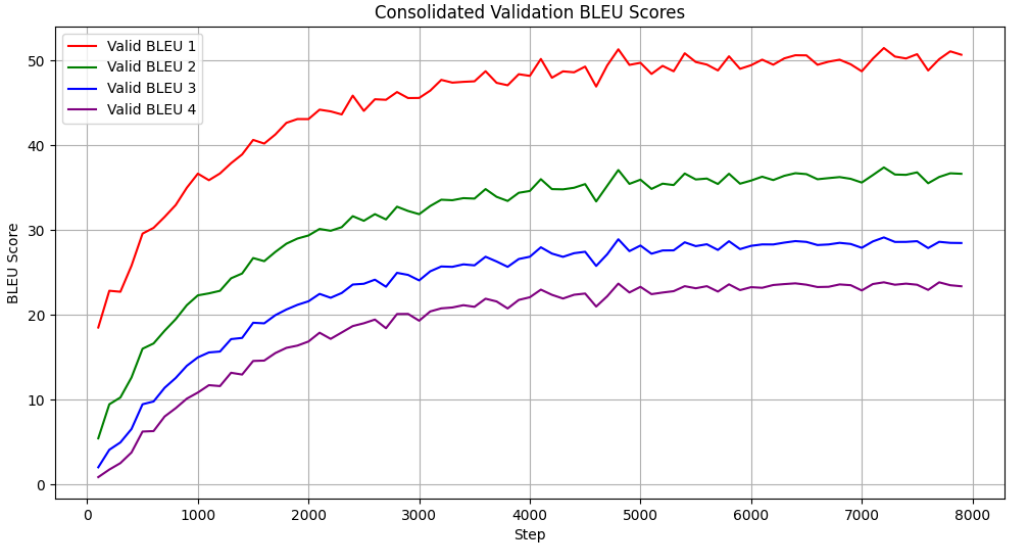

Figure 5.6: Consolidated validation BLEU scores.

5.3 Model Training Loss Analysis

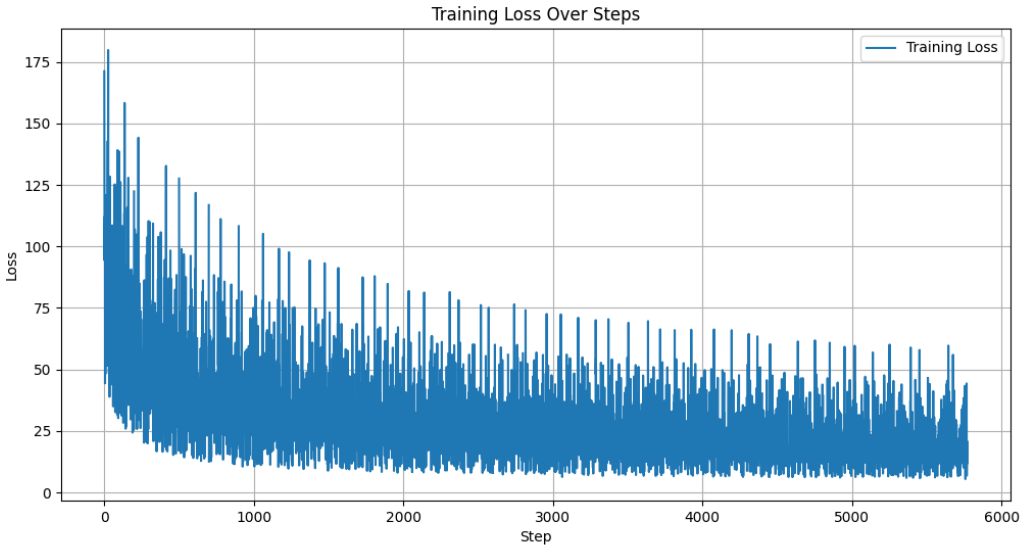

Figure 5.7: Training loss graphic.

Figure 5.7 shows the training loss of our machine learning model over 8,000 steps. The loss initially decreases rapidly and then stabilizes as the model captures data patterns and reaches diminishing returns. Fluctuations are present due to the stochastic training process. The loss curve suggests that the model is approaching optimal performance, but we identified the need for techniques such as early stopping and regularization to prevent overfitting and ensure generalization.

5.4 BLEU-4 Metric score

To achieve translation precision in longer sign sequences, the BLEU-4 metric is the primary focus of our analysis. Figure 5.8 shows training scores (blue line) steadily increasing, indicating effective learning from the data. The initial rapid rise signifies the model capturing basic patterns, while the later slowdown suggests refinement in understanding more complex patterns. Validation scores (green line) initially improve but then plateau, suggesting potential overfitting. The widening gap between training and validation scores further underscores this issue.

Despite the plateau in the validation curve, the high BLEU-4 scores demonstrate good generalization capability, highlighting the effectiveness of our proposed model approach.

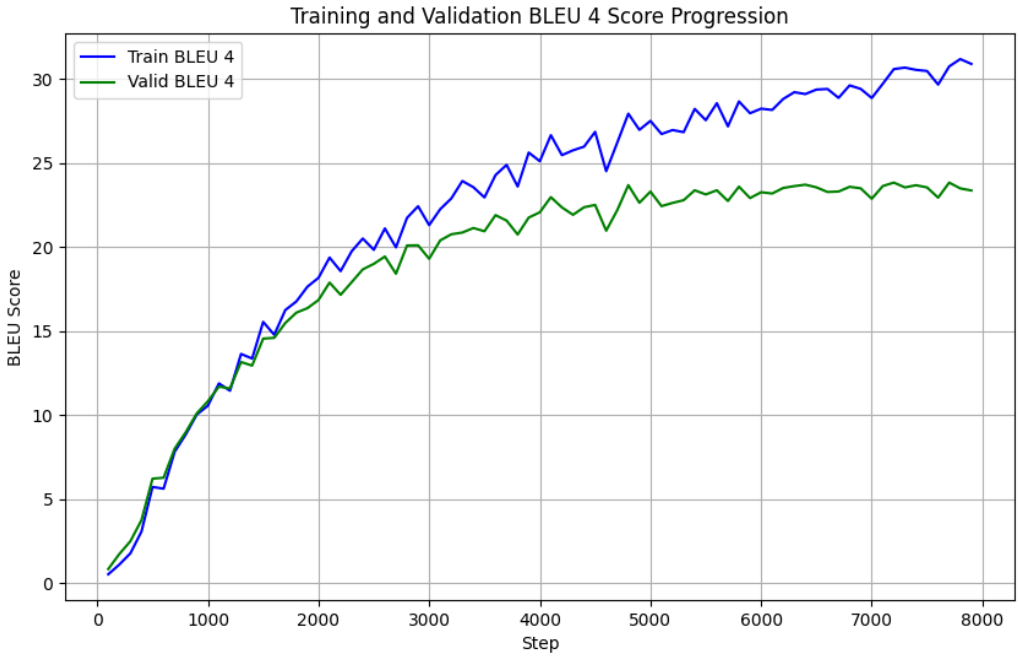

Figure 5.8: Training and validation BLEU-4 scores.

5.5 Overfitting

In the initial configuration and testing of our proposed model, a concerning trend emerged in the training and validation results.

Figures 5.9 to 5.11 display the progression of BLEU-4 scores during the training and validation phases in the initial research stage. We observed a substantial gap between the training and validation curves in the first testing stages (see Figures 5.9 and 5.11), which suggested overfitting. This issue likely arose due to the scarcity of sign language data resources, which affected the model's ability to generalize. To mitigate this, techniques such as frozen layers [11, 56], conditional sentence generation [94], and the implementation of pre-trained models [72, 68] were proposed.

The overfitting issues observed in initial testing were effectively resolved by implementing the optimized settings in Table 4.1. Incorporating a pre-trained BERT encoder and a dense LWTA layer in the decoder reduced the gap between training and validation BLEU-4

scores (see Table 5.2 and Figure 5.10), indicating improved generalization and robustness in sign language translation.

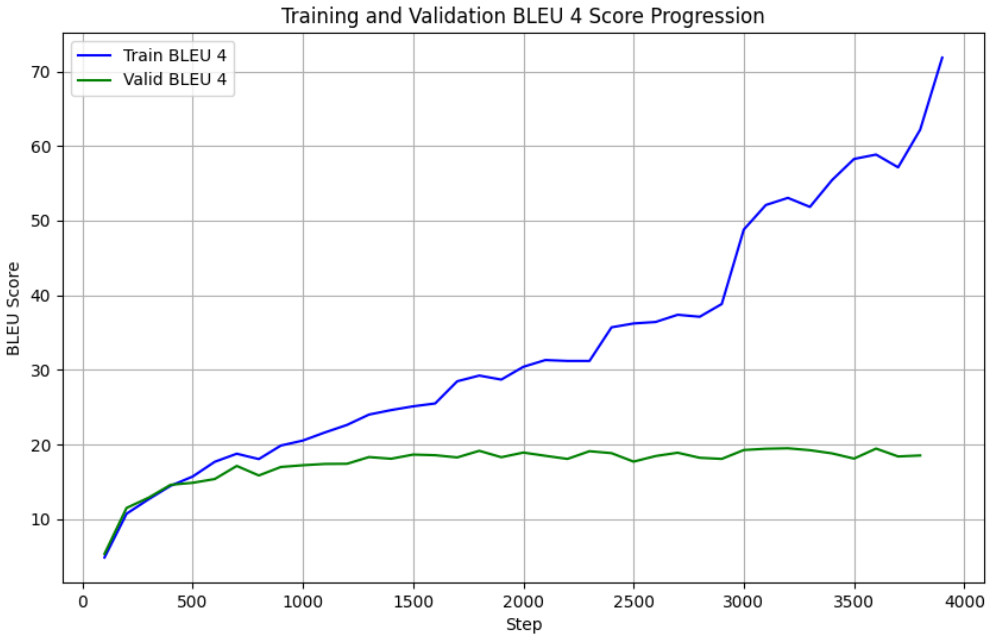

Figure 5.9: BLEU-4 score progression - Test 1.

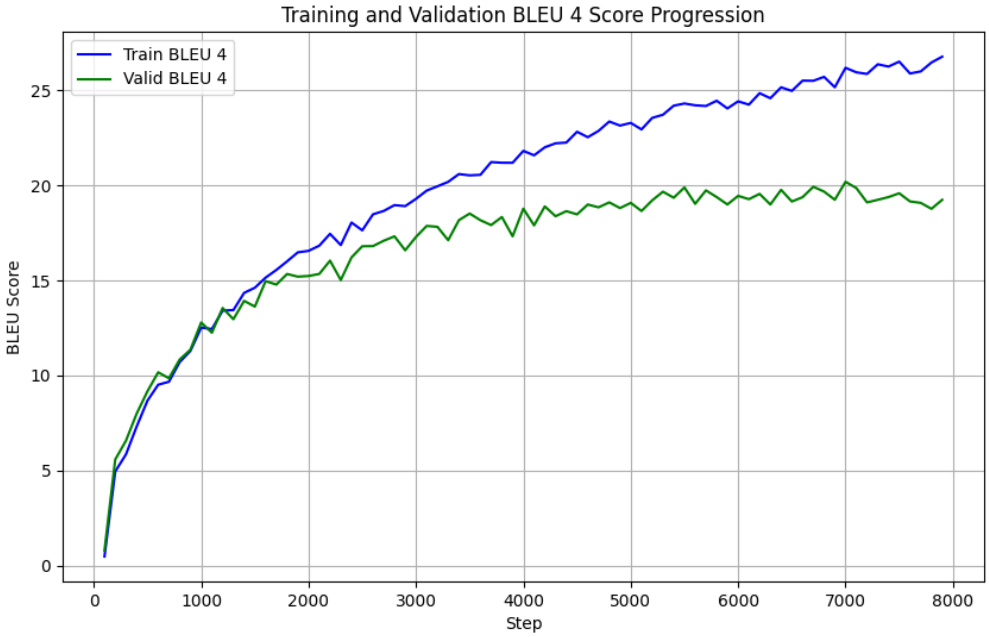

Figure 5.10: BLEU-4 score progression - Test 2.

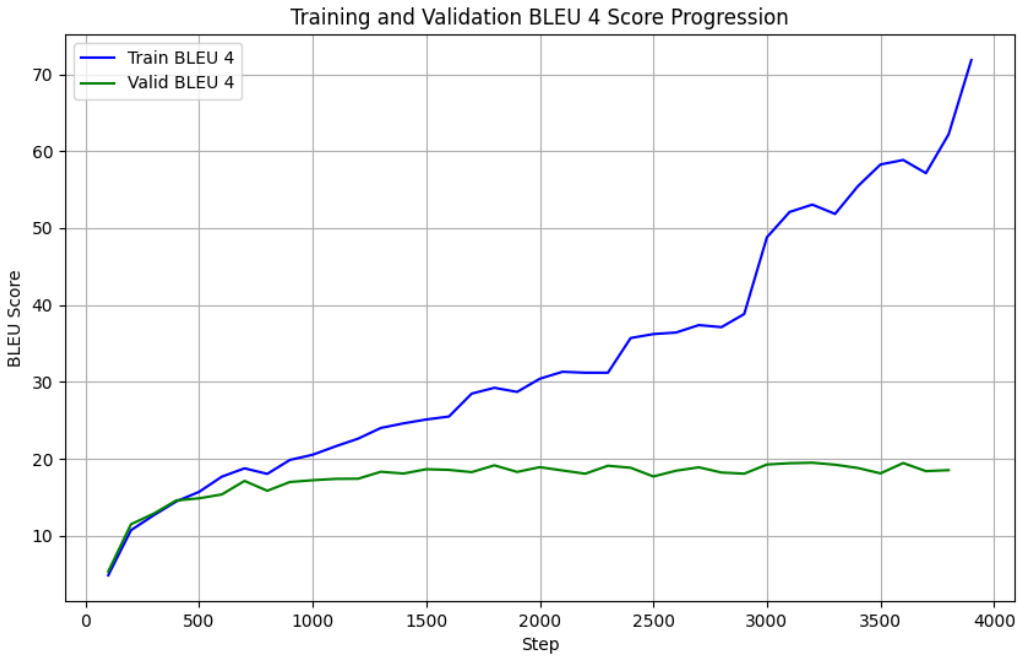

Figure 5.11: BLEU-4 score progression - Test 3.

5.6 Translation Prediction Results

Our evaluation examined the sign language translation model trained using the JOEY NMT framework [93]. For the PHOENIX14T German dataset, the produced or predicted sentence must be translated into English.

The model showed promise with straightforward sentences, accurately capturing the overall meaning and translating common expressions (Tables 5.3, 5.4 and 5.10). This demonstrates its potential for essential communication. However, there is room for improvement in handling complex sentences. The model struggled with specific geographic references like place names and directional phrases (Tables 5.5, 5.6, and 5.7). Additionally, it had trouble with nuanced weather conditions, particularly details like the sequence of events and variations in visibility (Tables 5.8 and 5.9).

These results typically appear in models like ours. Previous works [11, 9] show the same challenges with geographical information, specific numbers, and locations. Overall, our model demonstrates good capabilities in translating standard phrases. We will focus on enriching the training data with these specific aspects. This will allow the model to

understand and translate complex sign language sentences better.

	Spoken language translations
Text Reference:	Von Dresden runter bis zum Alpenrand null bis minus sechs Grad im Norden bleibt es aber frostfrei da haben wir es morgen auch wieder mild.
Translation:	*From Dresden down to the edge of the Alps, it's zero to minus six degrees in the north, but it stays frost-free so it'll be mild again tomorrow.*
Text Hypothesis:	In Bayern sind nur einstellige Temperaturen an den Alpen im Norden wird es im Norden wieder. milder
Translation:	*In Bavaria there are only single-digit temperatures in the Alps in the north it will be again in the north.*

Table 5.3: Generated spoken language translations by our sign model at the 7000th step.

Text Reference:	Und nun die wettervorhersage für morgen donnerstag den vierundzwanzigsten november.
Translation:	*And now the weather forecast for tomorrow, Thursday the twenty-fourth of November.*
Text Hypothesis:	Und nun die wettervorhersage für morgen donnerstag den vierundzwanzigsten november.
Translation:	*And now the weather forecast for tomorrow, Thursday the twenty-fourth of November*

Table 5.4: Generated spoken language translations by our sign model at the 7100th step.

Text Reference:	Kuhle sieben grad heute nacht im allgäu bis sechzehn an der nordsee.
Translation:	*Cool seven degrees tonight in the Allgäu until sixteen on the North Sea.*
Text Hypothesis:	Kuhle sieben grad heute nacht am alpenrand sechzehn an der ostsee.
Translation:	*Cool seven degrees tonight on the edge of the Alps, sixteen degrees on the Baltic Sea.*

Table 5.5: Generated spoken language translations by our sign model at the 7200th step.

Text Reference:	Am tag dreizehn grad bei dauerregen und einundzwanzig grad am oberrhein.
Translation:	*On the day it was thirteen degrees with constant rain and twenty-one degrees on the Upper Rhine.*
Text Hypothesis:	Am tag dreizehn grad an der ostsee und bis zu zwanzig grad an der see..
Translation:	*During the day thirteen degrees on the Baltic Sea and up to twenty degrees on the sea.*

Table 5.6: Generated spoken language translations by our sign model at the 7300th step.

Text Reference:	Richtung osten kann sich ab und zu auch die sonne blicken lassen bevor der regen am abend auch dort ankommt.
Translation:	*Towards the east you can occasionally see the sun before the rain arrives there in the evening.*
Text Hypothesis:	Im osten scheint häufig die sonne später k nnen sich auch ein paar tropfen bringen k nnen.
Translation:	*The sun often shines in the east, later it can also bring a few drops.*

Table 5.7: Generated spoken language translations by our sign model at the 7400th step.

Text Reference:	In der s dh hälfte weht der wind schwach aus unterschiedlichen richtungen im norden m ig an den k sten frisch aus westlichen richtungen.
Translation:	*In the south half the wind blows weakly from different directions in the north, moderately on the coast, fresh from westerly directions.*
Text Hypothesis:	Im s den schwacher bis niger wind aus s d bis s dwest an den k sten weht er schwach bis m ig
Translation:	*In the south the wind is weak to moderate, from the south to the southwest on the coast it blows light to moderate.*

Table 5.8: Generated spoken language translations by our sign model at the 7500th step.

Text Reference:	Am sonntag nur noch wenig schnee im norden zeigt sich teilweise die sonne.
Translation:	*On Sunday there is only a little snow left in the north and the sun is partly visible.*
Text Hypothesis:	Am sonntag nur wenig schnee im norden viel sonne.
Translation:	*On Sunday only a little snow in the north and lots of sun*

Table 5.9: Generated spoken language translations by our sign model at the 7600th step.

Text Reference:	Ich wünsche einen schönen abend und machen sie es gut.
Translation:	*I wish you a nice evening and take care.*
Text Hypothesis:	Ihnen einen schönen abend und machen sie es gut.
Translation:	*Have a nice evening and do well.*

Table 5.10: Generated spoken language translations by our sign model at the 7700th step.

5.7 Objectives and Results

Using a pre-trained BERT encoder and a decoder with dense LWTA activation functions, we built a robust model that achieved a BLEU-4 score of 23.83. This score surpassed both the FP Transformer (20.79) and the S Transformer (22.67), exceeding the benchmarks set in 2022. By incorporating the pre-trained BERT model, we enriched contextual embeddings, which led to higher translation accuracy and improved BLEU scores across all metrics. The use of dense LWTA layers enhanced the model's ability to handle data variability, resulting in better performance and generalization.

Table 5.11 provides a clear summary, linking each specific objective to its corresponding result. This overview highlights the main achievements of our work and demonstrates our contributions to the field of sign language translation.

Table 5.11: Objectives and Corresponding Results.

Objectives	**Results Obtained**
Implement a sign language transformer for better BLEU scores, using efficient activation functions and pre-trained NLP models.	Developed a robust translation model with a BERT encoder and LWTA decoder, achieving a BLEU-4 score of 23.83, surpassing 2022 benchmarks.
Integrate pre-trained models as encoders for enhanced sign language processing.	Used BERT as an encoder for enriched contextual embeddings, leading to more accurate translations.
Implement stochastic transformers with LWTA for improved performance.	Employed LWTA layers in the decoder, enhancing performance and data variability handling.
Optimize model configurations to find the best architecture.	Extensive testing identified optimal configurations, achieving better BLEU scores.
Develop a model translating directly without intermediate gloss.	Built a model translating sign language videos to spoken language without gloss, matching linguistic structures.

Chapter 6

Conclusions

In this work, we explore two main machine translation implementations: pre-trained modules fed into transformer architectures and stochastic processes, specifically stochastic transformers. These implementations avoid using ground truth gloss sequences of sign languages, achieving the best benchmark results in 2022. These novel implementations help us to create and implement a sign language translation model that uses a pre-trained model as the encoder and a decoder with dense LWTA layers.

For the spatial and word embedding process, the sign language videos are converted to spatial vectors through CNN feature vectors, with positional encoding added in the final process. Similarly, in the word embedding process, the target spoken language is converted to dense vectors with positional encoding.

Leveraging the large corpus of data trained by pre-trained models, the encoder uses a BERT pre-trained model, which creates enriched word embeddings that capture contextual information. The decoder, enhanced with dense LWTA layers, further processes these embeddings to generate accurate translations. This approach allows the model to benefit from both spatial and textual contexts, improving the overall performance of sign language translation.

Throughout the experimental stage, the optimal values were selected by finding the most suitable parameters featured in previous works. This proposed sign language translation model was addressed using the PHOENIX14T German dataset, which is a benchmark database in this research area. It is important to note that machine translation models typically suffer from overfitting problems, making it a critical topic to address.

The primary problems that arise in these models stem from the data source. Sign language and spoken language have different grammatical and linguistic structures, making the generation of sign language datasets a challenging and complex task. In a nutshell, the lack of sign data is the main problem. Several alternatives, such as the implementation of pre-trained models, new hand gesture frameworks, and new tokenization processes, have been explored. We have adopted these alternatives to develop an optimal and improved sign language translation model.

The results obtained from our model achieved better BLEU scores compared to the pipeline models we evaluated. Training and evaluating our model against stochastic transformers and pre-trained transformer models showed superior performance, particularly in BLEU-4 with a result of 23.83 on step 7200. Please note that the experiments were performed in a newly created environment using computational resources that we have available.

Despite the good results obtained from our models, there is still room for optimization. Freezing parameters, using transfer learning, or implementing new transformer models focused on sign language tasks could enhance performance. The configurations set up in our models can be modified and tested in other situations to achieve better results. Due to the ease of implementation of transformers, changing the encoder or decoder modules could help improve performance. Recent works propose changes in the information fed to the encoder, such as sign videos where new hand gesture and video modeling techniques have achieved the latest benchmark results.

Bibliography

[1] G. Vaughan, "Deafness and hearing loss," 2024, september 23th, 2024. [Online]. Available: https://www.un.org/en/observances/sign-languages-day

[2] A. Sultan, W. Makram, M. Kayed, and A. A. Ali, "Sign language identification and recognition: A comparative study," *Open Computer Science*, vol. 12, no. 1, pp. 191–210, 2022. [Online]. Available: https://doi.org/10.1515/comp-2022-0240

[3] P. Martins, H. Rodrigues, T. Rocha, M. Francisco, and L. Morgado, "Accessible options for deaf people in e-learning platforms: Technology solutions for sign language translation," *Procedia Computer Science*, vol. 67, pp. 263–272, 2015, proceedings of the 6th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915031166

[4] W. Aly, S. Aly, and S. Alaybani, "User-independent american sign language alphabet recognition based on depth image and pcanet features," *IEEE Access*, vol. PP, pp. 1–1, 09 2019.

[5] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. b. Lakulu, "A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017," *Sensors*, vol. 18, no. 7, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/7/2208

[6] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," *Expert Systems with Applications*, vol. 182, p. 115657, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421010484

[7] M. E. Morocho-Cayamcela and W. Lim, "Fine-tuning a pre-trained convolutional neural network model to translate american sign language in real-time," 01 2019, pp. 100–104.

[8] M. S. Lomas, A. Quelal, and M. E. Morocho-Cayamcela, "Implementation of a lightweight cnn for american sign language classification," in *Doctoral Symposium on Information and Communication Technologies*, K. Abad and S. Berrezueta, Eds. Cham: Springer International Publishing, 2022, pp. 197–207.

[9] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," 2020.

[10] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," *Expert Systems with Applications*, vol. 182, p. 115657, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421010484

[11] M. De Coster and J. Dambre, "Leveraging frozen pretrained written language models for neural sign language translation," *Information*, vol. 13, no. 5, 2022. [Online]. Available: https://www.mdpi.com/2078-2489/13/5/220

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

[13] R. Rastgoo, K. Kiani, S. Escalera, and M. Sabokrou, "Sign language production: A review," 2021.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[16] K. Yin, "Sign language translation with transformers," 04 2020.

[17] T. Ananthanarayana, P. Srivastava, A. Chintha, A. Santha, B. Landy, J. Panaro, A. Webster, N. Kotecha, S. Sah, T. Sarchet, R. Ptucha, and I. Nwogu, "Deep learning methods for sign language translation," *ACM Transactions on Accessible Computing*, vol. 14, pp. 1–30, 12 2021.

[18] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. O'Reilly Media, Inc., 2019.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016. [Online]. Available: https://arxiv.org/abs/1506.01497

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016. [Online]. Available: https://arxiv.org/abs/1506.02640

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *SSD: Single Shot MultiBox Detector*. Springer International Publishing, 2016, p. 21–37. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_2

[23] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014. [Online]. Available: https://arxiv.org/abs/1311.2524

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1411.4038

[25] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2016. [Online]. Available: https://arxiv.org/abs/1511.00561

[26] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," 2017. [Online]. Available: https://arxiv.org/abs/1606.00915

[27] S. Khan, H. Rahmani, and S. A. A. Shah, *A Guide to Convolutional Neural Networks for Computer Vision*. Morgan & Claypool Publishers, 2018.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[29] M. Elgendy, *Deep Learning for Vision Systems*. Manning Publications, 2020. [Online]. Available: https://books.google.com.ec/books?id=6gkLzAEACAAJ

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[33] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and lstm encoder decoder models for asr," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 8–15.

[34] A. Katrompas, T. Ntakouris, and V. Metsis, "Recurrence and self-attention vs the transformer for time-series classification: A comparative study," in *Artificial Intel-*

ligence in Medicine, M. Michalowski, S. S. R. Abidi, and S. Abidi, Eds. Cham: Springer International Publishing, 2022, pp. 99–109.

[35] D. Y. Wu, D. Lin, V. Chen, and H.-H. Chen, "Associated learning: an alternative to end-to-end backpropagation that works on CNN, RNN, and transformer," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=4N-17dske79

[36] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, September 2021. [Online]. Available: https://ideas.repec.org/a/spr/elmark/v31y2021i3d10.1007_s12525-021-00475-2.html

[37] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, p. 87–110, Jan. 2023. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2022.3152247

[38] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimedia Tools and Applications*, vol. 82, no. 3, p. 3713–3744, Jul. 2022. [Online]. Available: http://dx.doi.org/10.1007/s11042-022-13428-4

[39] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018. [Online]. Available: https://arxiv.org/abs/1801.06146

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[41] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49313245

[42] L. Tunstall, L. von Werra, and T. Wolf, *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. O'Reilly Media, 2022. [Online]. Available: https://books.google.com.ec/books?id=pNBpzwEACAAJ

[43] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Q. Liu and D. Schlangen, Eds. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6

[44] W. A. Qader, M. M. Ameen, and B. I. Ahmed, "An overview of bag of words;importance, implementation, applications, and challenges," in *2019 International Engineering Conference (IEC)*, 2019, pp. 200–204.

[45] *Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 1.* Springer Singapore, 2021. [Online]. Available: http://dx.doi.org/10.1007/978-981-33-6977-1

[46] A. Tabassum and D. R. R. Patil, "A survey on text pre-processing & feature extraction techniques in natural language processing," 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:235211496

[47] S. Mohamed, A. Elsayed, Y. Hassan, and M. Abdou, "Neural machine translation: past, present, and future," *Neural Computing and Applications*, vol. 33, pp. 1–13, 12 2021.

[48] J. Wieting, T. Berg-Kirkpatrick, K. Gimpel, and G. Neubig, "Beyond bleu: Training neural machine translation with semantic similarity," 2019. [Online]. Available: https://arxiv.org/abs/1909.06694

[49] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, P. Isabelle, E. Charniak, and D. Lin, Eds. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040

[50] S. Seljan, M. Brkić, and T. Vičić, "BLEU evaluation of machine-translated English-Croatian legislation," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 2143–2148. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/233_Paper.pdf

[51] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, D. Lin and D. Wu, Eds. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 388–395. [Online]. Available: https://aclanthology.org/W04-3250

[52] A. Voskou, K. P. Panousis, D. Kosmopoulos, D. N. Metaxas, and S. Chatzis, "Stochastic transformer networks with linear competing units: Application to end-to-end sl translation," 2021. [Online]. Available: https://arxiv.org/abs/2109.13318

[53] Z. Liang, H. Li, and J. Chai, "Sign language translation: A survey of approaches and techniques," *Electronics*, vol. 12, no. 12, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/12/2678

[54] B. Natarajan, E. Rajalakshmi, R. Elakkiya, K. Kotecha, A. Abraham, L. A. Gabralla, and V. Subramaniyaswamy, "Development of an end-to-end deep learning framework for sign language recognition, translation, and video generation," *IEEE Access*, vol. 10, pp. 104 358–104 374, 2022.

[55] K. P. Panousis, S. Chatzis, and S. Theodoridis, "Stochastic local winner-takes-all networks enable profound adversarial robustness," 2021. [Online]. Available: https://arxiv.org/abs/2112.02671

[56] M. De Coster, K. D'Oosterlinck, M. Pizurica, P. Rabaey, S. Verlinden, M. Van Herreweghe, and J. Dambre, "Frozen pretrained transformers for neural sign language translation," in *1st International Workshop on Automated Translation for Signed and Spoken Languages*, August 2021.

[57] N. Aloysius, G. M, and P. Nedungadi, "Incorporating relative position information in transformer-based sign language recognition and translation," *IEEE Access*, vol. 9, pp. 145 929–145 942, 2021.

[58] D. M. Madhiarasan and P. P. P. Roy, "A comprehensive review of sign language recognition: Different types, modalities, and datasets," 2022. [Online]. Available: https://arxiv.org/abs/2204.03328

[59] Y. Chen, F. Wei, X. Sun, Z. Wu, and S. Lin, "A simple multi-modality transfer learning baseline for sign language translation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5110–5120.

[60] P. Xie, M. Zhao, and X. Hu, "Pisltrc: Position-informed sign language transformer with content-aware convolution," 2021. [Online]. Available: https://arxiv.org/abs/2107.12600

[61] N. Cihan Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Subunets: End-to-end hand shape and continuous sign language recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[62] N. Kahlon and W. Singh, "Machine translation from text to sign language: a systematic review," *Universal Access in the Information Society*, vol. 22, 07 2021.

[63] A. Orbay and L. Akarun, "Neural sign language translation by learning tokenization," 2020. [Online]. Available: https://arxiv.org/abs/2002.00479

[64] K. Yin and J. Read, "Better sign language translation with stmc-transformer," 2020. [Online]. Available: https://arxiv.org/abs/2004.00588

[65] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Science China Technological Sciences*, vol. 63, no. 10, p. 1872–1897, Sep. 2020. [Online]. Available: http://dx.doi.org/10.1007/s11431-020-1647-3

[66] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[67] W. Zhao, H. Hu, W. Zhou, J. Shi, and H. Li, "Best: Bert pre-training for sign language recognition with coupling tokenization," 2023. [Online]. Available: https://arxiv.org/abs/2302.05075

[68] Z. Zhou, V. W. L. Tam, and E. Y. Lam, "Signbert: A bert-based deep learning framework for continuous sign language recognition," *IEEE Access*, vol. 9, pp. 161 669–161 682, 2021.

[69] K. Imamura and E. Sumita, "Recycling a pre-trained BERT encoder for neural machine translation," in *Proceedings of the 3rd Workshop on Neural Generation and Translation*, A. Birch, A. Finch, H. Hayashi, I. Konstas, T. Luong, G. Neubig, Y. Oda, and K. Sudoh, Eds. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 23–31. [Online]. Available: https://aclanthology.org/D19-5603

[70] W. Tao, Z.-H. Lai, M. Leu, and Z. Yin, "American sign language alphabet recognition using leap motion controller," 05 2018.

[71] W. Aly, S. Aly, and S. Almotairi, "User-independent american sign language alphabet recognition based on depth image and pcanet features," *IEEE Access*, vol. 7, pp. 123 138–123 150, 2019.

[72] H. Hu, W. Zhao, W. Zhou, and H. Li, "Signbert+: Hand-model-aware self-supervised pre-training for sign language understanding," *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, vol. 45, no. 09, pp. 11 221–11 239, sep 2023.

[73] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, pp. 108–125, 2015, pose Gesture. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314215002088

[74] Y. Min, P. Jiao, Y. Li, X. Wang, L. Lei, X. Chai, and X. Chen, "Deep radial embedding for visual sequence learning," in *Computer Vision – ECCV 2022*, S. Avidan,

G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 240–256.

[75] L. Hu, L. Gao, Z. Liu, and W. Feng, "Continuous sign language recognition with correlation network," 2023. [Online]. Available: https://arxiv.org/abs/2303.03202

[76] H. Lu, A. A. Salah, and R. Poppe, "Tcnet: Continuous sign language recognition from trajectories and correlated regions," 2024. [Online]. Available: https://arxiv.org/abs/2403.11818

[77] J. Ahn, Y. Jang, and J. S. Chung, "Slowfast network for continuous sign language recognition," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 3920–3924.

[78] O. M. Sincan and H. Y. Keles, "Autsl: A large scale multi-modal turkish sign language dataset and baseline methods," *IEEE Access*, vol. 8, p. 181340–181355, 2020. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2020.3028072

[79] D. Ryumin, D. Ivanko, and E. Ryumina, "Audio-visual speech and gesture recognition by sensors of mobile devices," *Sensors*, vol. 23, no. 4, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/4/2284

[80] B. Zhou, Z. Chen, A. Clapés, J. Wan, Y. Liang, S. Escalera, Z. Lei, and D. Zhang, "Gloss-free sign language translation: Improving from visual-language pretraining," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 20 871–20 881.

[81] Y. Chen, R. Zuo, F. Wei, Y. Wu, S. Liu, and B. Mak, "Two-stream network for sign language recognition and translation," 2023. [Online]. Available: https://arxiv.org/abs/2211.01367

[82] M. Guan, Y. Wang, G. Ma, J. Liu, and M. Sun, "Multi-stream keypoint attention network for sign language recognition and translation," 2024. [Online]. Available: https://arxiv.org/abs/2405.05672

[83] A. Duarte, S. Palaskar, L. Ventura, D. Ghadiyaram, K. DeHaan, F. Metze, J. Torres, and X. G. i Nieto, "How2sign: A large-scale multimodal dataset for continuous american sign language," 2021. [Online]. Available: https://arxiv.org/abs/2008.08143

[84] L. Tarrés, G. I. Gállego, A. Duarte, J. Torres, and X. G. i Nieto, "Sign language translation from instructional videos," 2023. [Online]. Available: https://arxiv.org/abs/2304.06371

[85] A. Voskou, K. P. Panousis, H. Partaourides, K. Tolias, and S. Chatzis, "A new dataset for end-to-end sign language translation: The greek elementary school dataset," 2023. [Online]. Available: https://arxiv.org/abs/2310.04753

[86] K. Lin, X. Wang, L. Zhu, K. Sun, B. Zhang, and Y. Yang, "Gloss-free end-to-end sign language translation," 2023. [Online]. Available: https://arxiv.org/abs/2305.12876

[87] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016. [Online]. Available: https://arxiv.org/abs/1602.07261

[88] T. Miyazaki, Y. Morita, and M. Sano, "Machine translation from spoken language to sign language using pre-trained language model as encoder," in *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, E. Efthimiou, S.-E. Fotinea, T. Hanke, J. A. Hochgesang, J. Kristoffersen, and J. Mesch, Eds. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 139–144. [Online]. Available: https://aclanthology.org/2020.signlang-1.23

[89] A. Voskou, C. Christoforou, and S. Chatzis, "Transformers with stochastic competition for tabular data modelling," 2024. [Online]. Available: https://arxiv.org/abs/2407.13238

[90] R. K. Srivastava, J. Masci, S. Kazerounian, F. Gomez, and J. Schmidhuber, "Compete to compute," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26.

Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/8f1d43620bc6bb580df6e80b0dc05c48-Paper.pdf

[91] K. Kalais and S. Chatzis, "Stochastic deep networks with linear competing units for model-agnostic meta-learning," 2022. [Online]. Available: https://arxiv.org/abs/2208.01573

[92] J. Forster, C. A. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. H. Piater, and H. Ney, "Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus," in *International Conference on Language Resources and Evaluation*, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:2516961

[93] J. Kreutzer, J. Bastings, and S. Riezler, "Joey nmt: A minimalist nmt toolkit for novices," 2020. [Online]. Available: https://arxiv.org/abs/1907.12484

[94] J. Zhao, W. Qi, W. Zhou, N. Duan, M. Zhou, and H. Li, "Conditional sentence generation and cross-modal reranking for sign language translation," *IEEE Transactions on Multimedia*, vol. 24, pp. 2662–2672, 2022.