# CONTINUOUS SIGN LANGUAGE TRANSLATION ON THE NEW EDUCATIONAL TURKISH SIGN LANGUAGE DATASET (E-TSL) USING NEURAL MACHINE TRANSLATION METHODS

# YENİ EĞİTİMSEL TÜRK İŞARET DİLİ VERİ KÜMESİ (E-TSL) KULLANARAK NÖRAL MAKİNE DÖNÜŞÜMÜ YÖNTEMLERİ İLE SÜREKLİ İŞARET DİLİ ÇEVİRİSİ

**ŞÜKRÜ ÖZTÜRK**

**ASSOC. PROF. DR. HACER YALIM KELEŞ**

**Supervisor**

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the Degree of Master of Science

in Computer Engineering

May 2024

# ABSTRACT

## CONTINUOUS SIGN LANGUAGE TRANSLATION ON THE NEW EDUCATIONAL TURKISH SIGN LANGUAGE DATASET (E-TSL) USING NEURAL MACHINE TRANSLATION METHODS

**Şükrü ÖZTÜRK**

**Master of Science, Computer Engineering**

**Supervisor: Assoc. Prof. Dr. Hacer YALIM KELEŞ**

**May 2024, 67 pages**

Sign language is a fundamental communication tool for people with hearing and speech disabilities. However, it is not widely known by others, making communication challenging for those who rely on it. Additionally, sign language varies by country and evolves over time, which further complicates communication among sign language users. To address these challenges, recent technological advances have led to numerous studies on sign language processing. These studies focus on sign language translation (sign to text) and production (text to sign). As with most deep learning models, large datasets are essential. Widely used datasets include Phoenix-2014, Phoenix-2014T, SWISSTXT-NEWS, VRT-NEWS, BSL Corpus, and How2Sign ASL. Turkish Sign Language (TSL) datasets are also available, but they are typically isolated. Currently, there is no Turkish Sign Language dataset suitable for continuous sign language translation.

In this thesis, we created the Educational Turkish Sign Language (E-TSL) dataset, featuring Turkish secondary school courses, to promote continuous sign language translation methods for TSL. The dataset includes 1,410 video clips with 11 different signers, totaling nearly 24

hours of content. Due to the agglutinative nature of Turkish, sign language translation faces additional challenges. After lemmatizing the words, the E-TSL dataset's dictionary shows that 64% of the words are singletons, and 85% are rare words appearing less than five times, posing significant challenges for translation.

To address these challenges, we developed transformer-based pose-to-text (P2T-T) and graph neural network-based transformer (GNN-T) models. Despite the dataset's complexity, our GNN-T model achieved ROUGE-L, BLEU-1, and BLEU-4 scores of 22.93, 21.01, and 3.49, respectively. These results highlight the difficulty of the E-TSL dataset compared to others. To validate our models, we used the Phoenix-Weather 2014T dataset as a benchmark, providing comparative results. Finally, we evaluated the performance of our E-TSL dataset against other commonly used datasets.

**Keywords:** Continuous Sign Language Translation, Graph Neural Networks, Graph Pooling, Graph Convolution, Transformers, E-TSL Dataset

# ÖZET

## YENİ EĞİTİMSEL TÜRK İŞARET DİLİ VERİ KÜMESİ (E-TSL) KULLANARAK NÖRAL MAKİNE DÖNÜŞÜMÜ YÖNTEMLERİ İLE SÜREKLİ İŞARET DİLİ ÇEVİRİSİ

**Şükrü ÖZTÜRK**

**Yüksek Lisans, Bilgisayar Mühendisliği**

**Danışman: Assoc. Prof. Dr. Hacer YALIM KELEŞ**

**Haziran 2024, 67 sayfa**

İşaret dili, işitme ve konuşma engelli bireyler için temel bir iletişim aracıdır. Ancak, diğer insanlar tarafından çok iyi bilinmediği için işaret dili kullanan bireylerin iletişimi zorlaşmaktadır. Ayrıca, işaret dili ülkeden ülkeye farklılık göstermekte ve zamanla değişim göstermektedir, bu da işaret dili kullanıcıları arasında iletişimi daha da zorlaştırmaktadır.

Bu zorlukları aşmak için, teknolojik gelişmelerin de yardımıyla işaret dili işleme üzerine birçok çalışma yapılmıştır. Bu çalışmalar, işaret dili çevirisi (işaretten metne) ve üretimi (metinden işarete) üzerine odaklanmaktadır. Çoğu derin öğrenme modeli gibi, büyük veri kümeleri bu alanda da önemlidir. Yaygın olarak kullanılan veri kümeleri arasında Phoenix-2014, Phoenix-2014T, SWISSTXT-NEWS, VRT-NEWS, BSL Corpus ve How2Sign ASL bulunmaktadır. Alanda Türk İşaret Dili (TSL) veri kümeleri de bulunmakla birlikte, var olan veri kümeleri ayrık formdadır. Şu anda sürekli işaret dili çevirisi için uygun bir TSL veri kümesi bulunmamaktadır.

Bu tezde, sürekli işaret dili çeviri yöntemlerini TSL ile ilgili yaygınlaştırmak amacıyla, ortaokul düzeyindeki Türkçe derslerinden oluşan Eğitim Türk İşaret Dili (E-TSL) veri

kümesini oluşturduk. Veri kümesi, 11 farklı işaretçiden oluşan 1,410 video klibi içermekte ve toplamda yaklaşık 24 saatlik içerik sunmaktadır. Türkçenin sondan eklemeli yapısı nedeniyle, işaret dili çevirisi ek zorluklar içermektedir. Kelimeleri lemmatize ettikten sonra, E-TSL veri kümesinin sözlüğünde kelimelerin %64'ü tekil, %85'i ise beş kereden az görünen nadir kelimelerden oluşmaktadır, bu da çeviri için önemli zorluklar yaratmaktadır.

Bu zorlukları aşmak için, transformer tabanlı pozdan metne (P2T-T) ve grafik sinir ağı tabanlı transformer (GNN-T) modeller geliştirdik. E-TSL veri kümesinin karmaşıklığına rağmen, GNN-T modelimiz ROUGE-L, BLEU-1 ve BLEU-4 skorlarında sırasıyla 22.93, 21.01 ve 3.49 değerlerini elde etmiştir. Bu sonuçlar, E-TSL veri kümesinin diğerlerine kıyasla zorlu olduğunu ortaya koymaktadır. Modellerimizi doğrulamak için, karşılaştırmalı sonuçlar sunarak Phoenix-Weather 2014T veri kümesini bir kıstas olarak kullandık. Son olarak, E-TSL veri kümemizin performansını diğer yaygın kullanılan veri kümeleri ile karşılaştırarak değerlendirdik.

**Keywords:** Sürekli İşaret Dili Çevirisi, Çizge Sinir Ağları, Çizge Ortaklama, Çizge Dönüşümü, Transformatörler, E-TSL Veriseti

# ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Assoc. Prof. Dr. Hacer YALIM KELEŞ for trusting me and allowing me to enter the field of sign language translation and for giving me the opportunity to contribute to this field. Thank you for constantly supporting me in this difficult process, for encouraging me, and for helping me overcome difficulties when I'm in trouble and when I start to lose hope.

I would like to thank my managers and colleagues in the Radio and Television Supreme Council, as well as the Presidency of Repuclic of Türkiye Directorate of Communications, for always supporting me.

I would like to thank TRT EBA TV, which enabled us to generate the data as part of this research, and all the TRT employees.

Most of the numerical calculations involved in this research were carried out at the TUBITAK ULAKBIM, High Success and Grid Computing Centre (TRUBA sources). I'd like to thank TUBITAK ULAKBIM for providing me with this infrastructure and equipment.

I am forever grateful to my mother, Fadime ÖZTÜRK, and my father, Ömer ÖZTÜRK, who raised me, who contributed so much to my coming to this day, who supported me unconditionally in the work I have done throughout my life.

Finally, I would like to thank my wife Burcu ÖZTÜRK and my son Kerem ÖZTÜRK for supporting and encouraging me in this process, as well as for sacrificing the time I have to dedicate to them.

# CONTENTS

**Page**

# TABLES

# FIGURES

# ABBREVIATIONS

| | | |
|---|---|---|
| **ASL** | : | **A**merican **S**ign **L**anguage |
| **BLEU** | : | **Bi**-**L**ingual **E**valuation **U**nderstudy |
| **CNN** | : | **C**onvolutional **N**eural **N**etwork |
| **CSL** | : | **C**hineese **S**ign **L**anguage |
| **CSLR** | : | **C**ontinuous **S**ign **L**anguage **R**ecognition |
| **E-TSL** | : | **E**ducational **T**urkish **S**ign **L**anguage |
| **GNN** | : | **G**raph **N**eural **N**etwork |
| **LCS** | : | **L**ongest **C**ommon **S**ubsequence |
| **LSTM** | : | **L**ong-**S**hort **T**erm **M**emory |
| **NLP** | : | **N**atural **L**anguage **P**rocessing |
| **NMT** | : | **N**eural **M**achine **T**ranslation |
| **RNN** | : | **R**ecurrent **N**eural **N**etwork |
| **ROUGE** | : | **R**ecall-**O**riented **U**nderstudy for **G**isting **E**valuation |
| **SL** | : | **S**ign **L**anguage |
| **SLR** | : | **S**ign **L**anguage **R**ecognition |
| **SLT** | : | **S**ign **L**anguage **T**ranslation |
| **TRT** | : | **T**ürkiye **R**adio **T**elevision Agency |
| **TSL** | : | **T**urkish **S**ign **L**anguage |
| **VIA** | : | **V**GG **I**mage **A**nnotation |
| **WER** | : | **W**ord **E**rror **R**ate |
| **WHO** | : | **W**orld **H**ealth **O**rganization |

# 1. INTRODUCTION

The most fundamental means of communication among people is speaking. Unfortunately, not everyone possesses this ability. The majority of those unable to speak are born with hearing impairments. These impairments can result from genetic factors, infections during pregnancy, or oxygen deficiency at birth. Additionally, some individuals develop hearing impairments later in life due to various complications during birth. Hearing impairment is defined as a loss of at least 35 decibels in the better hearing ear [2]. According to the WHO, approximately 430 million people worldwide are hearing impaired, representing about 5% of the global population. Experts predict that by 2050, 1 in 10 people will experience hearing impairment, a truly alarming statistic.

In Türkiye, as of 2023, there are 836,000 individuals with hearing impairments and 507,000 with speech impairments [3]. This accounts for nearly 2% of the total population. One of the most significant challenges for hearing-impaired individuals is their difficulty in communicating with others. These individuals often rely on sign language to bridge this communication gap. While sign language is typically associated with the hearing impaired, it is also a crucial communication tool for those unable to speak due to issues with their mouth, tongue, or vocal cords. This thesis defines sign language as a communication tool for both the hearing and speech impaired. Unfortunately, very few speaking individuals are familiar with sign language, which hampers communication, leads to isolation, and complicates the daily lives of those with speech disabilities. Contrary to common belief, sign language is not universal; it varies by region, much like spoken languages. This variation further complicates communication not only between the speech-impaired and others but also among sign language users themselves.

Some manufacturers produce various devices to address hearing problems. These devices generally assist individuals with mild hearing loss, making it easier for them to hear, and are usually affordable. However, some devices are designed to help those who have lost their hearing either at birth or later in life to hear and learn to speak. These devices are both

very expensive and hard to access. Considering that approximately 80% of hearing-impaired individuals have low income, as reported by the WHO [2], most hearing-impaired people cannot afford these costly devices. Therefore, there is a need for accessible and easy-to-use solutions for the majority of hearing-impaired individuals.

In recent years, advancements in sign language recognition (SLR), sign language translation (SLT), and related fields have provided a more affordable and accessible alternative for the hearing impaired. These technological developments offer promising solutions that can bridge the communication gap and improve the quality of life for those with hearing and speech impairments.

The first studies on SLR began in the 1970s, with studies such as creating sign language letters with a robotic hand and recognizing sign language using gloves that can process motion signals. Later, with the development of computer vision methods, using more advanced techniques such as Hidden Markov Models [4] or neural networks [5], or recognition continuous signs and sentences (Continuous Sign Language Recognition - CSLR) studies have been carried out [6, 7].

Sign language generation is the translation from speech or text to sign language. In the first studies in this domain, letters or isolated signs were created using avatars or anime characters [8]. Later, thanks to technological developments in the field, continuous signs and glosses were created using syntactic analysis or semantic representation methods [9].

SLT, unlike CSLR, involves automatically translating sign language into spoken language or text. Early studies in this field primarily utilized statistical models and rule-based methods [10]. Significant advancements have been achieved with the development of NMT methods, which can learn from large parallel corpora consisting of sign language videos and their corresponding text or speech transcripts [11]. The transformer architecture proposed by Vaswani et al. [1] has been a milestone, significantly increasing research in this area.

For SLT, a continuous sign language dataset is required. Although there are large-scale TSL datasets available for isolated sign recognition [12–14], these are not suitable for

continuous recognition tasks. Currently, there is no continuous TSL dataset suitable for the SLT task. Therefore, in this thesis, we created a new Educational Turkish Sign Language (E-TSL) dataset specifically designed for sign language translation. We also proposed two transformer-based deep models to serve as baselines for future research utilizing the E-TSL dataset.

## 1.1. Scope of the Thesis

This thesis involves developing and introducing a new dataset for TSL, specifically designed for educational purposes. We address the challenges faced by people with hearing and speech impairments, focusing on their communication difficulties and the role of sign language as an essential communication tool. Although sign language recognition and translation technologies have become widespread globally, there remains a significant gap in continuous TSL datasets. We aim to fill this gap by presenting the E-TSL dataset. Additionally, we propose two transformer-based models trained on the E-TSL dataset, providing a benchmark for future research in SL and TSL translation.

## 1.2. Contributions

The most important contributions of this thesis are:

- The first continuous Turkish Sign Language Translation (SLT) dataset, E-TSL, containing 1410 video segments and spoken language translations, totaling nearly 24 hours are collected and utilized in this research.
- Two basis models, namely Graph Neural Network-based Transformer (GNN-T) and Sign Pose2Text Transformer (P2T-T) models, are developed to generate baseline performances on the new E-TSL benchmark. These works and our early results are published [15].
- Comparison of the preliminary results obtained with P2T-T and GNN-T models on the E-TSL dataset with other state-of-the-art benchmark datasets are provided.

## 1.3. Organization of the Thesis

The chapter-by-chapter organization of the thesis is listed below.

- Chapter 1 presents our motivation, contributions and the scope of the thesis.
- Chapter 2 provides background on neural machine translation
- Chapter 3 gives a related works
- Chapter 4 introduces performance and evaluation metrics
- Chapter 5 demonstrates dataset preparation
- Chapter 6 explains the proposed method
- Chapter 7 shows experimental results
- Chapter 8 states the summary of the thesis and possible future directions.

# 2. NEURAL MACHINE TRANSLATION

NMT is a method used for automatic translation between languages. It is based on deep learning and artificial neural networks (ANNs). NMT models are trained on large datasets to capture complex relationships between source and target texts. By leveraging deep learning techniques, NMT can provide highly accurate translations by understanding context and nuances in language. Previously, machine translation was usually performed using canonical-based [16] or statistical methods [17, 18]. However, these methods had some limitations and had difficulty accurately translating especially long and complex sentences.

Kalchbrenner et al. [19] proposed an encoder-decoder architecture to perform machine translation using neural networks. They used an approach containing Convolutional n-Gram Model (CGM) and inverse CGM&RNN as encoder and decoder, respectively. Cho et al. proposed an architecture in which they use RNNs in both the encoder and decoder [20]. RNNs are good at capturing short-term dependencies, but they struggle to preserve information at distant points over time. This causes difficulties in translating long sentences and texts.

One of the first studies using the NMT definition is Sutskever et al.'s article titled "Sequence to Sequence Learning with Neural Networks" published in 2014 [21]. In this article, they use deep neural networks (DNNs) to convert a series of inputs in one language to a series of outputs in another language. They proposed an LSTM-based model formed an early example of the "encoder-decoder" architecture. LSTM has achieved successful results by covering the shortcomings of RNNs because they are very successful in preserving and extracting short- and long-term correlations. The publication of this article was a milestone for the development of NMT, leading many researchers and companies to work on improving NMT models and translation quality.

The requirement that the lengths of NMT's input vectors be fixed reduces the effectiveness of the model in translating sentences and texts of variable lengths. To avoid this problem, Bahdanau et al. proposed the "attention mechanism" [22]. This "attention mechanism"

allows, at each step, the model to focus on specific parts of the input data while producing output. Therefore, the length of the input vector does not matter because the model can dynamically focus on each input component. In the attention mechanism, the model improves translation quality by paying more attention to certain parts. This can be very useful, especially when working with long and complex sentences. Thanks to this mechanism, the problem of fixed length of input vectors is overcome and the model becomes more flexible. This provides better performance in tasks such as NMT.

The number of studies using attention mechanisms has increased day by day, and the article "Attention is All You Need" presented by Vaswani et al. in 2017 [1] made revolutionary contributions to the field of NMT and introduced the Transformer model seen in Figure 2.1, which is the basic algorithm. This paper has extended the attention mechanism to all interactions between encoder and decoder, rather than examining only the relationship between input and output. This formed the basis for a more flexible and scalable model that performs better over many language pairs.

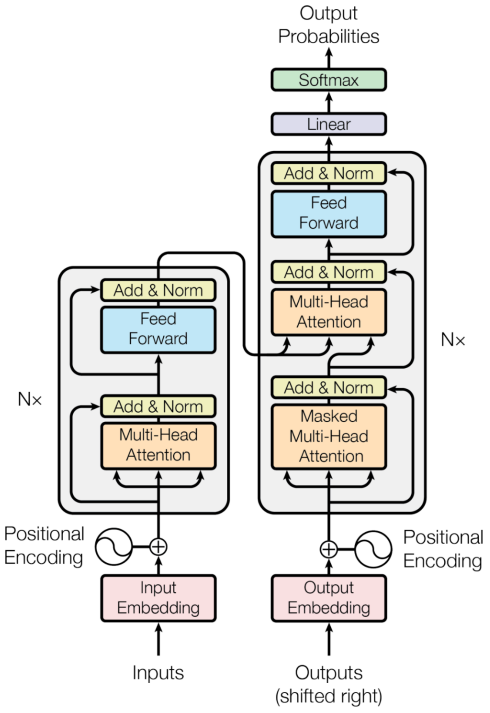


Figure 2.1 Transformer Model Architecture[1]

Transformer offers significant advantages over traditional RNN and LSTM-based NMT models. Firstly, thanks to its parallel computing capability, the Transformer model can handle long-term dependencies more effectively and enables faster training. Additionally, the attention mechanism seen in Figure 2.2 allows the model to focus on specific parts of the input data at each translation step, which improves translation quality and allows better processing of long sentences.

Additionally, Transformer increases parallelization by ensuring that the encoder and decoder are independent of each other. This increases the scalability of the model and provides better performance on large datasets. Another important feature is the non-grammar-based attention mechanism. This mechanism pays balanced attention to all the encoder's inputs at each translation step, thus providing more general context information beyond grammar.

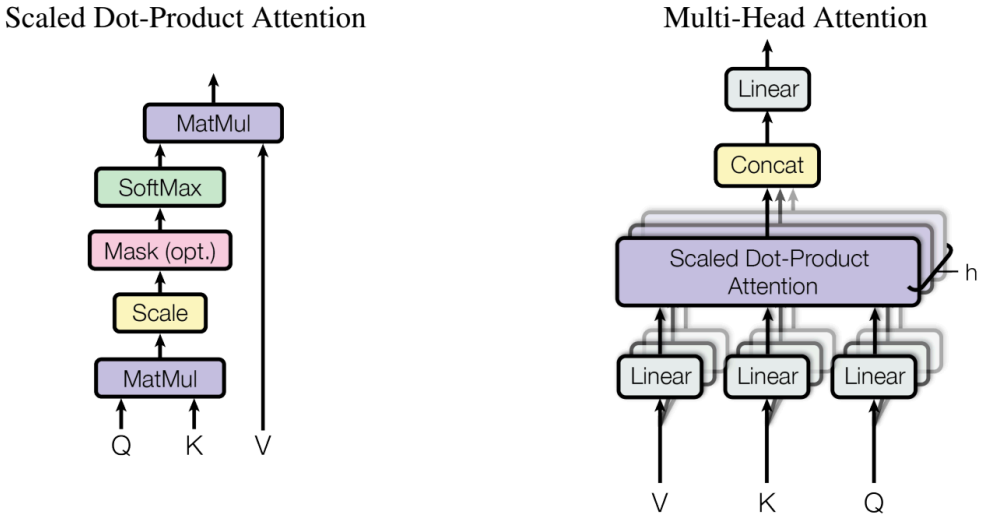


Figure 2.2 Attention Mechanisms[1]

In conclusion, the article "Attention is All You Need" provides a revolutionary advance in the field of NMT, laying the foundation for a more flexible, scalable and non-grammar-based model. This article is considered a milestone in NMT research and forms the basis of many modern translation systems.

# 3. RELATED WORKS

Researchers have conducted numerous studies in the areas of SLR, SLT and sign language production (SLP). In this chapter, we describe various studies done in these areas.

Wen et al. [23] presented a system that includes a triboelectric smart glove, deep learning, and a virtual reality interface that allowed sign language to be recognized and voiced after it has been translated into text. They chose 50 words and 20 sentences from the American sign language book, which they believe are the most used in daily life, and translated the sign language movements into signals with the help of the triboelectric smart glove. Afterwards, they completed the training process with the deep learning model. They aimed to facilitate the communication of two people who know sign language and those who do not by separating sign language sentences into words, first converting them into text with the help of the model they trained, and then making them into sentences again and making vocalisations in the virtual environment. With this system, they achieved 86.67% accuracy.

Gokul et al. [24] released a 4.6K hour dataset in isolated form, which they call SignCorpus, containing 9 different sign languages. They also proposed the graph-based Sign2Vec model, which they pretrained on this dataset. Third, they created "MultiSign-ISLR", which consists of pose keypoint sequences obtained from 11 labeled datasets containing 7 different sign languages. The "MultiSign-FS" dataset they created includes finger-spelling divided into training and test sets in 7 different languages. By fine-tuning the Sign2Vec model on these datasets, they obtained a multilingual isolated sign language recognition model. They tested this model, which has multilingual support, on datasets used as benchmarks to measure the effectiveness of transfer learning. They achieved an average accuracy of 74.3% on the 9 datasets used as benchmarks.

Bull et al. [25] studied the problem of subtitles not being aligned in sign language videos. To solve the alignment problem, they proposed a Transformer architecture trained on a manually labeled and aligned dataset containing 17.7 hours of video and more than 15K captions. In this transformer architecture, BERT subtitle embeddings are used as encoder input, and CNN

signals and frame information containing subtitles are used as decoder input. They obtained predicted sequence of frame information containing subtitles as output. With the transformer model they trained, they reported 76% accuracy on the BSL Corpus dataset and 55.62% accuracy on the BOBSL dataset.

Chen et al. [26] proposed TwoStream-SLR for SLR and TwoStream-SLT for SLT tasks. Both models take both RGB Video and keypoint as input that is why they are called as TwoStream models. Both models use CNN architecture. Unlike TwoStream-SLR, TwoStream-SLT also includes a translation network layer. They used PHOENIX14**T** and CSL Daily datasets and obtained ROUGE: 51.59, BLEU-4: 26.71 using Sign2Gloss2Text and ROUGE: 53.48, BLEU-4: 28.95 using Sign2Text on PHOENIX14**T** and obtained ROUGE: 54.92, BLEU-4: 24.13 using Sign2Gloss2Text and ROUGE: 55.72, BLEU-4: 25.79 using Sign2Text on CSL Daily dataset.

Saunders et al. [27] proposed the first model of SLP that translates from spoken language text into a sign language pose sequence. The model they developed using a transformer based architecture is suitable for both Text2Sign and Text2Gloss2Sign tasks. Surprisingly, the Text2Sign results of their proposed model are better than the Text2Gloss2Sign results. They applied data augmentation methods such as "Future Prediction" and "Gaussian Noise" to enable the model to learn better and increase its performance. They obtained 32.02 ROUGE and 10.43 BLEU-4 scores for the Text2Sign task on the PHOENIX14**T** dataset.

Camgöz et al. [11] released the first publicly accessible CSLT dataset. The "RWTH-Phoenix-Weather 2014T" dataset they present consists of weather news videos from the Phoenix TV channel. PHOENIX14**T** dataset includes spoken language texts and gloss level annotations. They proposed a model with RNN-based encoder and decoder architecture to perform the first experiments on this dataset. With this model and dataset, they obtained a 19.26 BLEU-4 score in the Gloss2Text experiment, a 18.13 BLEU-4 score in the Sign2Gloss2Text experiment, and a 9.58 BLEU-4 score in the Sign2Text experiment.

Camgöz et al. [7] created improved version of the transformer-based JoeyNMT [28] which provides an end-to-end solution for Sign2Gloss, Sign2(Gloss+Text) and Sign2Text tasks.

Moreover, their proposed model is suitable for both CSLR and SLT. They performed experiments on the Phoenix14**T** dataset using their proposed model. They reported a Word Error Rate (WER) of 24.49% and 21.80 BLEU-4 score for Sign2Gloss2Text in CSLR. In SLT, their model achieved a WER of 26.16% and 21.32 BLEU-4 score for Sign2Gloss2Text outputs. For Sign2Text task, they reported 20.17 BLEU-4 score.

As with other deep neural networks, large datasets are required for SLT. Unfortunately, the number of datasets in this domain is low and the size of the existing datasets is a little too small to train the models. To overcome this bottleneck, Chen et al. [29] proposed the transfer learning method with pre-trained models in the general domain of action recognition and machine translation. They used the mBART model based on transformer architecture for transfer learning. They used PHOENIX14**T** and CSL Daily datasets to test the efficiency of their method. They achieved 51.43 ROGUE and 21.46 BLEU-4 on the Sign2Gloss2Text task on the CSL Daily dataset. In the Sign2Text task, they achieved 53.25 ROGUE and 23.92 BLEU-4. On the PHOENIX14**T** dataset, they obtained scores of 52.54 ROGUE and 26.70 BLEU-4 in the Gloss2Text task, 49.59 ROGUE and 24.60 BLEU-4 in the Sign2Gloss2Text task, and 52.65 ROGUE and 28.39 BLEU-4 in the Sign2Text task.

Cheng et al. [30] introduced a fully convolutional network for the online Sign Language Recognition task. Using their proposed network, they aimed to extract spatial and temporal features from weakly annotated video sequences with only sentence-level annotations. They added a "Gloss Feature Enhancement (GFE)" module to the network to improve sequence alignment learning. With their proposed FCN, they obtained a word error rate (WER) of 3% in the CSL dataset and 23.9% WER in the RWTH dataset.

Hosain et al. [31] introduced a methodology aimed at word-level SLR in ASL through video analysis. Their method incorporates both motion and hand shape indicators, maintaining resilience against variations in execution. It leverages information regarding body posture, as inferred by a readily available pose estimator. Spatial and temporal feature maps from various layers of a 3D CNN are aggregated, with the pose serving as a guiding element. Separate classifiers are trained using pose-guided aggregated features from different resolutions,

and their prediction scores are combined during testing. The proposed methodology demonstrates performance improvements of 10% on WLASL100, 12% on WLASL300, 9.5% on WLASL1000, and 6.5% on WLASL2000 subsets.

Hu et al. [32] presented SignBERT, the pioneering self-supervised pre-trainable SLR framework that incorporates a model-aware hand prior. Their approach incorporates both hands and treats hand poses as a visual token. Before feeding the visual token into the framework, they enrich it with information about gesture state, temporal, and hand chirality. To begin, they engage in self-supervised pre-training using a vast array of hand poses, employing token masking and reconstruction techniques. During the pre-training phase, their framework comprises a Transformer encoder and a hand-model-aware decoder. They meticulously devise multiple masking strategies to better capture hierarchical contextual information, supplemented by the decoder's incorporation of hand priors. As a result, their pre-trained framework is fine-tuned for SLR tasks. They extensively evaluate their method on NMFs-CSL, SLR500, MSASL and WLASL datasets. They achieved successful results on all 4 datasets.

Kan et al. [33] inspired by NMT, which is based on sequence-to-sequence learning in most SLT studies, proposed the "Hierarchical Spatio-Temporal Graph Neural Network (HST-GNN)" model, which performs SLT using the important visual features of sign language. HST-GNN processes the features obtained from sign language images as high-level and fine-level graphs. High-level graph represents the area where the signer's hands and face are located on the screen and uses the relationship between each other. The fine-level graph represents some specific landmarks on the signer's hands and face, independently of each other. Hierarchical graph pooling was applied to each graph obtained. Afterwards, the graphs were combined to obtain a single graph. Graph convolution was applied to this graph. Finally, hierarchical pooling has been applied to both high-level and fine-level graphs. The resulting vector was fed to the transformer network and the Sign2Gloss2Text task was applied. Using this method, they reported 22.3 BLEU-4 score on the PHOENIX14**T** dataset and 17.8 BLEU-4 score on the CSL dataset.

Voskou et al. [34] presented the Elementary23 dataset, which contains various lecture videos and translations at the elementary school level. The dataset consists of 71 hours of Greek Sign Language videos. All videos were shot in HD quality in the most suitable environments using professional equipment. Using a transformer-based model, they obtained 11.5 BLEU-1, 2.85 BLEU-2, 1.05 BLEU-3 and 0 BLEU-4 with the Elementary23 dataset. Then, by applying some methods to this dataset, they produced a subset called Elementary23-SLT, which is approximately 11 hours long, as being most suitable for SLT. They reported that they made a significant improvement by obtaining scores of 19.9 BLEU-1, 11.10 BLEU-2, 7.68 BLEU-3 and 5.69 BLEU-4 with the Elementary23-SLT dataset they produced.

# 4. PERFORMANCE AND EVALUATION METRICS

Automatic text evaluation is crucial in fields such as machine translation, summarization, and NLP. Various metrics are used to measure and evaluate the quality of text translations. The most commonly used metrics are WER (Word Error Rate) [35], BLEU (BiLingual Evaluation Understudy) [36], and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [37]. These metrics are designed to reflect human evaluations and objectively measure the quality of machine-generated texts. Each metric assesses different aspects of the translation, making it important to choose the appropriate metric for the specific problem. However, a single metric often fails to fully capture translation quality, so it is generally recommended to use multiple evaluation metrics.

In this thesis, we used BLEU and ROUGE metrics to evaluate the translation performance of our transformer-based models. We utilized all BLEU scores to measure the models' performance across various n-grams. Additionally, we used the ROUGE-L F-measure to evaluate the longest common subsequence (LCS) between the text predicted by the models and the reference text.

## 4.1. Word Error Rate (WER)

WER is a widely used metric to measure the performance of systems such as NMT, automatic speech recognition (ASR), and optical character recognition (OCR). It measures the difference between the original or reference text and the recognized, or translated text in terms of word-level errors. To calculate the Word Error Rate, we compare two texts and count the number of errors. These errors can be of three types:

**Substitutions:** This refers to the number of incorrectly predicted words that are different from the corresponding words in the reference text.

**Additions:** This shows the extra words count found in the translated text but not in the reference text.

**Deletions:** This indicates the words count missing from the translated text but found in the reference text.

Once these errors are counted, the Word Error Rate can be calculated using the following formula:

$$WER = \frac{Substitutions + Additions + Deletions}{Total\,number\,of\,words\,in\,the\,reference\,text} \tag{1}$$

Usually, a percentage represents the result. A lower WER means the model performs better.

In most languages, in addition to the correctness of the word, the place of the word in the sentence is also important in terms of the meaning of the sentence. In the translation process, a synonymous word may have been guessed, and the meaning of the sentence may not have changed. WER, on the other hand, is not concerned with where the word is in the sentence or whether the words are synonyms. Therefore, while WER is a useful metric, it is often not sufficient by itself to measure the performance of the model.

## 4.2. BiLingual Evaluation Understudy (BLEU)

The BLEU score is a metric used to evaluate the quality of the output of machine translation systems. Introduced in a study published by IBM researchers in 2002 [36], BLEU measures how well a translation system's predictions match reference texts (usually created by human translators). It is a popular method for evaluating the performance of machine translation systems. Here's how to calculate the BLEU score:

**N-gram Precision:** BLEU calculates precision scores using n-grams, which are contiguous sequences of words in the predicted and reference translations. It takes into account n-grams of different lengths (unigrams, bigrams, trigrams, etc.). The n-gram typically takes values from 1 to 4. To calculate precision, we divide the total n-grams' count in the predicted translation by the n-grams' count in the reference translation.

**Brevity Penalty:** We apply a brevity penalty to account for length differences between predicted and reference translations. The brevity penalty encourages the system to create translations of similar length as references.

**BLEU Score Calculation:** We calculate the BLEU score by combining n-gram precisions. Each n-gram precision is weighted equally, i.e., the score equally considers the precision of n-grams of different lengths. We then multiply the combined precision scores by the brevity penalty to obtain the final BLEU score.

The BLEU score is a numerical value between 0 and 1, and 1 is a perfect match with reference translations. Typically, the literature expresses it as a percentage. For this reason, for example, if the BLEU score is calculated as "0.2442", it is mentioned as "24.42" in the results. The machine translation improves with a higher BLEU score.

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log p_n) \tag{2}$$

In this formula:

p stands for n-gram precision

w is the weight for each precision, and it is usually set to 1/N, where N is the total n-grams' count considered.

We calculate the Brevity Penalty (BP) as follows:

$$if c > r => BP = exp(1 - r/c) \tag{3}$$

$$if c \leq r => BP = 1 \tag{4}$$

In this formula:

c: total n-grams' count in the predicted translation

r: total n-grams' count in the reference translations

BLEU determines whether the words and their sequence are correct; it is not concerned with the placement of the word or word groups in the sentence. Consequently, BLEU may incorrectly calculate a correctly predicted and inverted sentence. We need to consider such situations when making an evaluation.

## 4.3. Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

As the name implies, automatic text summarization systems often use ROUGE as a measurement metric to evaluate their performance. However, besides its widespread use in evaluating automatic text summarization systems, ROUGE also serves as a tool for evaluating machine translation. This metric measures how similar the system-generated summary or translation is to the reference summary or translation (usually written by humans)[37].

The ROUGE metric works with a combination of important metrics: Recall and Precision. Recall refers to the proportion of text fragments that are correctly predicted by the system, while Precision refers to how much of the text fragments predicted by the system are actually correct.

ROUGE is generally used in three main versions, ROUGE-N, ROUGE-L, and ROUGE-W. ROUGE-N calculates n-gram similarity and is usually expressed as ROUGE-1, ROUGE-2, and ROUGE-3 (using 1, 2 and 3 as the n value). ROUGE-L measures the LCS similarity. ROUGE-W, on the other hand, takes into account the sequential matching length of words.

For instance, the system calculates the ROUGE-1 metric by dividing single words count predicted by the model by the total words count in the actual reference text. For binary words, ROUGE-2 does the same. ROUGE-L measures LCS similarity; that is, it finds the LCS between the text predicted by the model and the reference text and calculates the length of this string.

In conclusion, people frequently compare the quality of systems for natural language processing tasks like machine translation, summarization, and more using the ROUGE metric.

# 5. PROPOSED E-TSL DATASET

First of all, one of the primary purposes of this thesis is to intensify research on Turkish Sign Language. This is due to several factors, including the laborious nature of producing a dataset for sign language translation, the low number of produced datasets, the need for more data for translation models to learn effectively, and our belief that the current number of datasets related to Turkish Sign Language is insufficient. Additionally, since there is no TSL dataset suitable for Continuous SLT in the literature, we created the E-TSL dataset.

This chapter provides a detailed, step-by-step explanation of the preparation process for the E-TSL dataset. We believe this study will serve as a good guide and encourage those who will prepare new data sets in the future, given the laborious and time-consuming nature of dataset preparation. At the end of this chapter, we compare the E-TSL dataset we created with other datasets frequently used for sign language translation.

## 5.1. Topic Selection

At present, the majority of TSL datasets are in an isolated form, containing gloss-based examples. In our study, we are performing CSLT, which necessitates that the dataset consist of continuous videos. Creating continuous video recordings requires expert signers, intensive labor, time, and money. Therefore, we have directed our focus towards public areas that are easily accessible and available.

Broadcasts and productions such as main news bulletins, some series, and films, which involve SLT and typically display a signer in a corner of the screen, are available. However, in these broadcasts and productions, the signers' movements have long pauses and halts proportional to the broadcast's flow. Frequent interruptions of signer movements both reduce the clean broadcast rate obtained from the unit-length broadcast and significantly extend video editing processes. Furthermore, because the signer's location is typically small in these broadcasts and productions, cropping the signer portion from the video results in a low-resolution video. This has a negative impact on image processing.

TRT EBA TV, which broadcasts lectures for elementary, secondary, and high school students, influenced our decision to use their videos for reasons such as broadcast quality and resolution, easy accessibility and the signer's significantly fewer pauses compared to other broadcasts.

In elementary school-level lecture videos, teachers speak slowly and repeat certain words frequently. In high school-level lecture videos, teachers' speeches are faster; hence, the movements of the signers are also faster. This complicates the pose estimation process in the videos. For these reasons, we decided that secondary school is the most suitable level for our dataset. For the subject, we chose Turkish lectures, as we thought they would contain less foreign language. We made this choice to enhance the model's understanding of the Turkish language and its structure. Some of the lecture videos contain lectures over slides, while others contain the teacher's lecture on the smart board. For a more reliable pose extraction process, we preferred videos where the signer and the teacher are on opposite sides of the screen as much as possible. This helps to minimize any potential interference between the signer and the teacher in the video.

In line with these criteria, we have created a dataset that includes Turkish lecture videos for 5th, 6th, and 8th grades.

## 5.2. Video Editing

In the video editing phase, we first remove the introduction and closing parts of the video because they do not contain the signer. To clean up other people and materials on the screen and to focus only on the signer, we cropped a 190x230 pixel (width x height) section from the bottom right of the screen where the signer is located, ensuring that the signer's hands stay within the screen as much as possible.

## 5.3. Video and Audio Labeling

The attention mechanism in Transformer networks, matrix multiplication operations, and connections of the layers necessitate that the data we send as input to the model be of the same size. Furthermore, having data of the same size contributes to improving training and generalization performance, as well as optimizing memory usage. Therefore, in the Transformer model, we need to bring the input and output data to the same size by applying padding or truncation, as in most artificial intelligence models. Input data that is too short and padded and input data that is too long and truncated can make it difficult for the model to learn and may cause some features in the data to be lost. To minimize these problems, we decided to divide all videos into average 1-minute segments. In this way, we aimed to minimize padding and truncation operations and enable the model to learn better.

For video and audio labeling processes, we utilized the open-source VGG Image Annotator (VIA) tool [38] developed by Oxford University. Since VIA tool is an open source software, it offers a structure that can be customized according to the user's purpose. We used this tool in its original form without any customization, as it met our requirements. Using this tool shown in Figure 5.1, we performed the following operations sequentially in each lecture video:

- Parts where the signer stopped and there was no sound for at least 1.5 - 2 seconds were marked and discarded.
- Parts where there was sound but the signer did not translate were marked and discarded.
- After the first two items, the cleaned videos were marked to be divided into average 1-minute segments, taking into account the completion of movements of the signer and the completion of the sentence.

As a result, we've got about one-minute video clips and audio files.

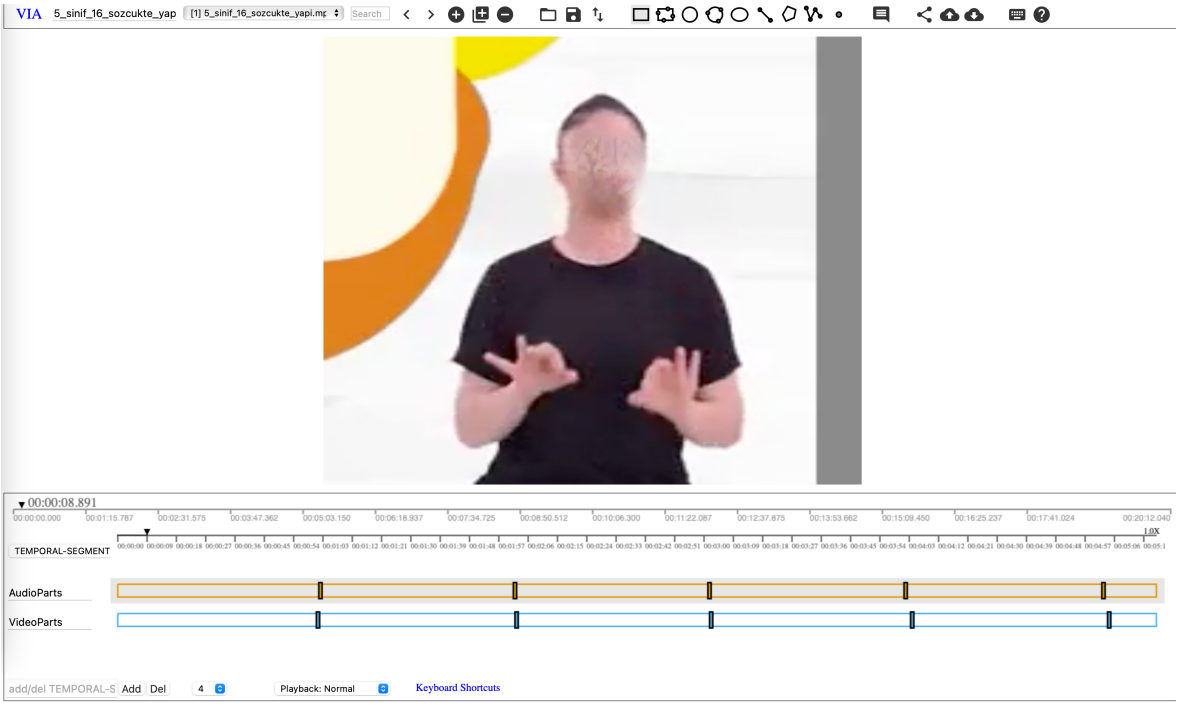


Figure 5.1 VIA Tool

## 5.4. Signer Alignment

In each of the videos, signers do not appear in exactly the same pixel range. So in crop videos, signers don't always appear right in the middle of the screen. This could make it difficult for transformers and other deep learning models to learn from coordinate and pixel data. To overcome this problem, we shifted the image so that the signer was in the middle of the screen. We filled the empty part on the right or left of the video with gray color.

## 5.5. Text Editing

We converted audio files corresponding to about 1 minute of video resulting from video and audio tagging to text with the help of the Google Speech to Text API. Then we manually checked each text file, listened to the audio file it was associated with, and fixed the

errors. Finally, to improve our Transformer models' performance, we removed all existing punctuations except the "periods".

## 5.6. Video Segmentation

Videos often contain elements like a smart board screen, a slide display, or a library in the background of the signer, and rarely videos where the cursor and the teacher are on the same side. We implemented the segmentation process using SOLO V2 because the images in the signer's background would be unnecessary. This way, we made sure that only the signer was on the screen.

## 5.7. Final Dataset

As a result of all the processes we have done, we have produced 1410 video clips, each with an average length of 1 minute, and 23.89 hours of data from 18 5th grade, 27 6th grade, and 15 8th grade videos. Videos are 25 frames per second and 190x230 pixels in size. Unlike the datasets in the literature that contain a video for each sentence, our video clips provide text with an average of 120 words each, corresponding to an average of 1 minute. Thus, as seen in Figure 5.2, we obtained a balanced word distribution with the number of words varying between 80 and 160.

To make the dataset ready for training, we randomly divided it into train, development, and test sets using the numbers and proportions shown in Table 5.1.

Table 5.1 Dataset Partitions

| | Number of Videos | Percentile |
|---|---|---|
| Train | 1.057 | 75 |
| Development | 141 | 10 |
| Test | 212 | 15 |

We compare our E-TSL dataset with commonly used datasets in the sign language translation domain. We chose the PHOENIX14**T**, SWISSTXT-NEWS, and VRT-NEWS datasets for

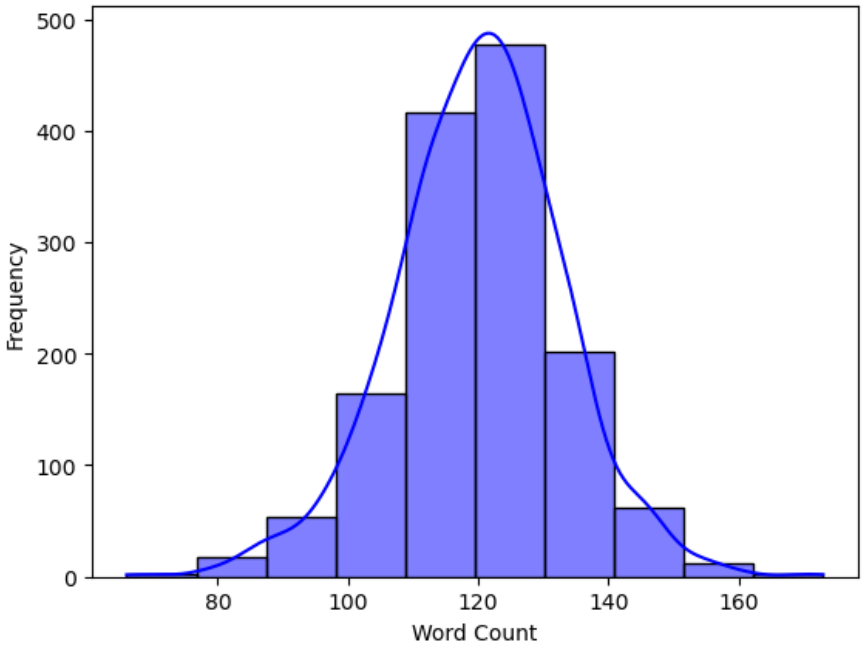


Figure 5.2 Word Distribution of E-TSL Dataset

comparison. We also include the Elementary23-SLT dataset in this comparison because it is both up-to-date and similar in content to the E-TSL dataset. Visual statistics for all of these datasets are included in Table 5.2. Unfortunately, the paper by Camgöz et al. does not contain details about the count of signers in the SWISSTXT-NEWS and VRT-NEWS datasets. Apart from these, there are 9 signers in the PHOENIX14**T** and Elementary23-SLT datasets. The E-TSL dataset offers slightly more diversity, with 11 signers. ”Total hours” and ”frames” information for the Elementary23-SLT dataset is not available, but Voskou et al.[34] stated that it is similar in size to the PHOENIX14**T** dataset. Therefore, we assumed that the Elementary23-SLT and PHOENIX14**T** datasets have similar properties. Comparing the dataset durations reveals that the E-TSL dataset, with a duration of nearly 24 hours, has more than twice the content of the others. In terms of video resolution, it is clearly seen that E-TSL has the worst resolution. It is obvious that the PHOENIX14**T** dataset also has poor resolution. With their HD resolutions, the SWISSTXT-NEWS, VRT-NEWS, and Elementary23-SLT datasets offer much better image quality. Therefore, our E-TSL dataset is more challenging compared to these datasets.

Table 5.2 Visual Statistics of Datasets

| | Signers | Total Hours (h:m:s) | Frames | Pixel Resolution |
|---|---|---|---|---|
| PHOENIX14**T**[11] | 9 | 10:31:50 | 947.756 | 210x260 |
| SWISSTXT-NEWS[39] | - | 09:24:28 | 1.693.391 | 1280x720 |
| VRT-NEWS[39] | - | 09:00:30 | 810.738 | 1280x720 |
| Elementary23-SLT[34] | 9 | - | - | 1280x720 |
| E-TSL (Ours) | 11 | 23:53:06 | 2.149.650 | 190x230 |

Table 5.3 Word Base Statistics of Datasets

| | Total Words | Vocabulary Size | Singletons | Rare Words $< 5$ |
|---|---|---|---|---|
| PHOENIX14**T**[11] | 99.081 | 2.287 | 1.077 (47%) | 1.758 (77%) |
| SWISSTXT-NEWS[39] | 72.892 | 10.561 | 5.969 (57%) | 8.779 (83%) |
| VRT-NEWS[39] | 79.833 | 6.875 | 3.405 (50%) | 5.334 (78%) |
| Elementary23-SLT[34] | 83.327 | 8.202 | 3.327 (41%) | 6.155 (75%) |
| E-TSL (Ours) | 169.356 | 6.980 | 4.466 (64%) | 5.936 (85%) |

In order to further compare the E-TSL dataset with other datasets, we present the data on vocabulary-based metrics in Table 5.3. Because the PHOENIX14T dataset only includes weather news, it contains much less vocabulary, singletons, and rare words than other datasets. It is not surprising that E-TSL is the dataset with the most total words, as it has the longest video content. A large vocabulary size, singletons ratio, and rare words ratio is a factor that reduces SLT performance. SWISSTXT-NEWS stands out as the dataset with the largest vocabulary size, with 10,561 words. Elementary23-SLT dataset has the second largest vocabulary size with 8,202 words. The vocabulary size of the E-TSL and VRT-NEWS datasets is almost the same, with 6,980 and 6,875 words, respectively. The dataset with the highest rate of both singletons and rare words is the E-TSL dataset. The high rates are one of the most important factors that make it difficult for our models to learn. Models will have difficulty learning words they encounter less frequently. Additionally, the rate of words that are in the test set but not in the train set is close to 10%. This shows the ratio of the words

that the model encountered in the test set for the first time. Therefore, these rates show us that the E-TSL dataset can be challenging.

# 6. PROPOSED METHODS

In this research, we formulated two baseline models. The first one is a Graph Neural Network based Transformer (GNN-T) model, an extension of the approach introduced by Kan et al. [33], which integrates body pose data to recognize the role of arm, hand, and facial gestures in sign language. Our GNN-T model incorporates techniques such as graph pooling and graph convolution, as depicted in Figure 6.3.

The second model, referred to as Pose to Text Transformer (P2T-T), draws inspiration from the framework proposed by Camgöz et al. [7]. We customized this model to accommodate pose modality and implemented landmark normalization to enhance its efficacy. Notably, our model lacks gloss annotation, thus it undergoes end-to-end training for translation purposes. To enable Sign2Text functionality, we adapted the model by directly connecting the transformer encoder's output to the transformer decoder.

Below, we provide a exhaustive overview of the important components of our models, underlining the technical details and functionalities essential to their performance.

## 6.1. Pose Extraction

Current state-of-the-art SLT is performed using computer vision and machine learning techniques. One of these techniques, pose extraction, is crucial in sign language translation. The technique of pose extraction determines a person's posture and movements. This technique analyzes a person's hand movements, facial expressions, and body posture. Deep learning models typically perform pose extraction. These models are trained on large amounts of labeled data, and these data allow the model to accurately determine a person's body posture and movements. People usually use these models to identify specific points in a person's body, like his hands, body, and face.

In sign language, body and face movements are very important, including a person's hand movements. In sign language translation, pose extraction is used to capture the signer's hand, face, and body movements to determine what the signer says in sign language.

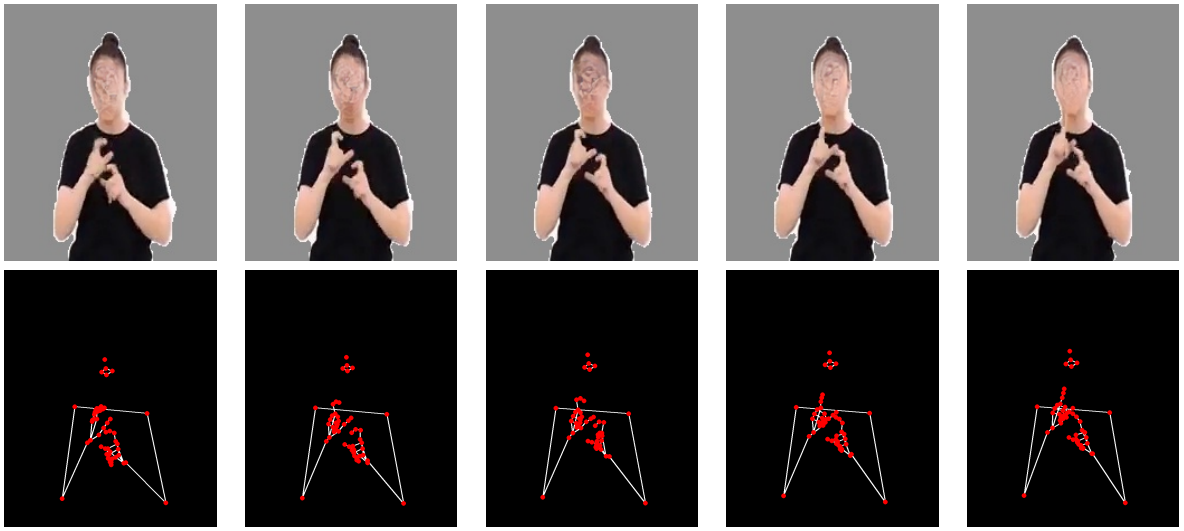

Figure 6.1 Sample Sign Sequence and Corresponding Pose Data from E-TSL Dataset.

In this thesis, we employed the MediaPipe library[40] for capturing sign language gestures and depicting the instantaneous positions of the hands, face, and specific body components as graphical representations. The MediaPipe framework, developed by researchers at Google, is designed to identify various anatomical features from both videos and images, presenting them to users in graphical format. This framework comprises a graph structure encompassing 468 points for facial features, 33 points for body components, and 21 points for each hand. It provides x and y coordinate information for each point. In our work, all 21 points for both the right and left hands, 6 points for body parts, and 5 points for facial features were utilized, as illustrated in Figure 6.1. Since facial expressions are important in sign language, using all the facial landmarks that MediaPipe offers us can have a positive effect on the model's performance. However, due to lack of GPU memory, we had to use a very small portion of the facial landmarks. We only used 4 landmarks for lip movements and 1 landmark on the nose for head movements. Additionally, for the body, only 6 poses involving shoulders and arms were considered. Given the paramount significance of hands in sign language communication, all 21 poses for hand gestures were incorporated. Consequently, we employed total of 53 poses in this study.

## 6.2. Data Preprocessing

With regard to the signer's rapid movement in sign language videos and image quality, MediaPipe is unable to detect pose landmarks for some frames. In addition to this technical challenge, it unfortunately increases the number of landmarks that could not be detected with natural difficulties, such as the signer moving his hands out of the screen, holding his hands together while he is not moving, joining his hands downward in a way that is invisible to the screen, or preventing one hand from passing in front of the other. In sign language, the movements of the hands are of particular importance. In our analysis, we found that the ratio of frames that MediaPipe can't detect from the video in our data is about 25%. When we looked at the statistics in more detail, we found that the average number of frames that MediaPipe could not detect landmarks in a row was 18. This indicates that the data loss was less than one second on average. Looking at the frames before and after these empty frames, we decided to apply the linear interpolation procedure. This works perfectly if the signer's hand moves linearly from point A to point B. However, in nonlinear movements, the empty frames are filled with misleading coordinate data. This makes it difficult for our model to learn. We set a time of half a second to minimize this negative situation. If the number of frames that MediaPipe could not detect landmarks in succession was less than 13, we filled the empty landmarks applying linear interpolation. We left the ones that were more than 13 as they were. Following these operations, the empty landmark rate has been reduced to approximately 9%.

## 6.3. Graph Pooling

Graph pooling is a technique that facilitates the analysis of larger and complex graph structures by reducing or summarizing the size of graph data. This method is frequently used in deep learning and graph-based learning algorithms, making operations on graph data more efficient.

Graph pooling is often employed when processing graph data within a layered neural network. Graph data is typically represented as nodes and edges, creating a structure

that illustrates the relationships between nodes of a graph. However, while deep learning algorithms generally operate with fixed-size data, graphs can vary in size, posing a challenge in processing graph data. This is where graph pooling comes into play.

In graph pooling, subgraphs of a graph are usually created, and the properties of these subsets are summarized into a smaller size. This summarized information is then integrated into a larger chart structure or utilized in a separate analysis phase. This process allows the graphic to be reduced to a smaller size while preserving its essence.

Graph pooling can enhance the deep learning algorithms' performance on graph data and render the analysis of larger-scale graph structures more manageable. However, it is important to choose appropriate pooling strategies and carry out the pooling process in a way that preserves the graph structure. To best address the problem studied in this thesis, we applied average graph pooling to the landmarks obtained as a result of pose extraction, averaging over every three frames. We applied average pooling separately for both hands, face, and body graphs and applied it after all of them were combined into a single graph.

## 6.4. Graph Convolution

Graph neural networks (GNNs) employ graph convolution as a methodology for conducting operations and extracting features from data structured in graphs. Graphs, comprised of nodes interconnected by edges, serve as mathematical constructs to depict relationships and associations among entities. Within this dissertation, we utilized graphs to symbolize various anatomical features, including hands, facial components, and specific body parts. Before the application of graph convolution, we connected distinct graphs representing the face, body, and hands into a unified graph. Thus, we established a connection between the lips, body and each hands.

Contrary to traditional CNNs that execute convolutional operations on uniform grid-like data, such as images, graph-structured data lacks a fixed grid structure, and nodes can possess varying connection counts. Graph convolution extends the notion of convolution to graphs by generalizing it to operate on both nodes and edges of the graph. Fundamentally, it involves

aggregating information from neighboring nodes to update a node's representation iteratively, thereby facilitating the dissemination of information across the entire graph. This iterative process, akin to message passing, entails an aggregation step wherein neighboring nodes' attributes are amalgamated in a meaningful manner. Various techniques, including feature addition, averaging, or more intricate operations like weighted summations, can be employed for this purpose. In this investigation, we adopted the averaging technique.

Graph convolutional architectures frequently incorporate multiple layers of graph convolution to capture information from successive hops within the graph. Each layer accumulates information from the local neighborhood of a node, and as information propagates through successive layers, it captures increasingly global relationships inherent in the graph. Notably, in this thesis, we restricted our focus to a singular graph convolutional process, foregoing the stacking of multiple layers.

## 6.5. Landmark Normalization

As a result of the pose extraction process, MediaPipe gives us the coordinates of the landmarks we have identified as values between 0 and 1, normalized according to the image width and height. These data can, of course, be used directly to feed the transformer model. However, it will work if the signer is always the same person, the screen is always in the same place, and the camera is always at the same distance. Each person's body size is different. The data obtained from different people performing the same movement and normalized according to the width and height of the image will be different. A large number of signers means that we will get different landmarks for the same sign for different people. This complex data structure poses a challenge for the transformer model to learn. Various methods of normalization are being applied to overcome this challenge. In this thesis, in order to overcome this difficulty, we identified the distances of each landmark to the signer's neck. We also determined the neck point using the signer's right-and-left shoulder coordinates, since MediaPipe does not give a landmark for its neck. Then we normalized the data by dividing every value we found by the shoulder width of the signer. We've done this for each

frame separately. So we've minimized situations such as changing the signer, moving in a different area of the screen, or getting closer or farther away from the camera, as shown in Figure 6.2

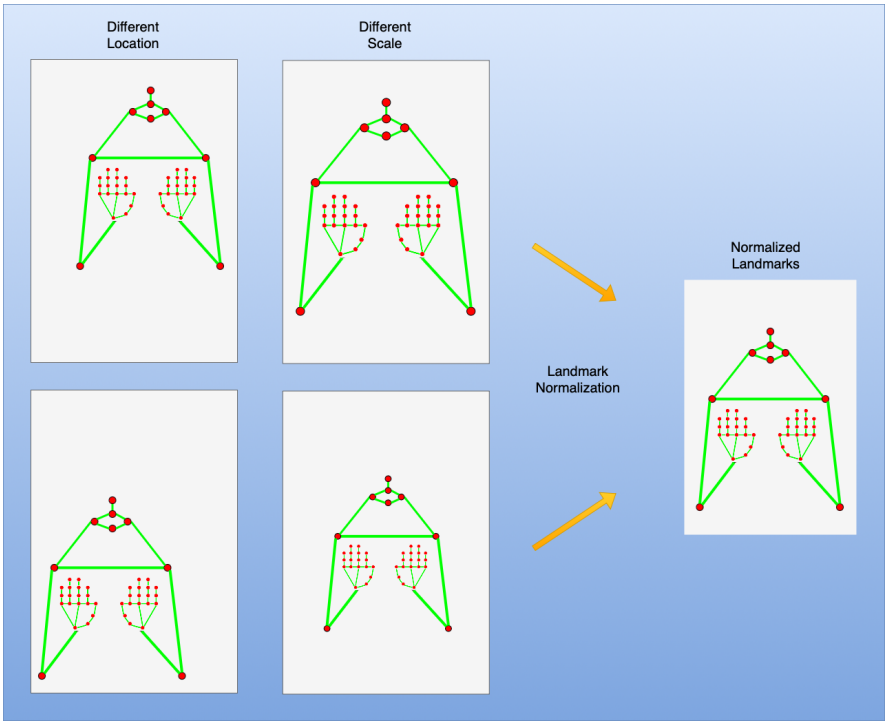


Figure 6.2 Landmark Normalization

## 6.6. Transformer Model

Transformer architecture, which emerged with the article "Attention Is All You Need" published by Vaswani et al. in 2017[1], has become a milestone in the field of NLP.

The transformer model replaces traditional methods such as CNNs and RNNs, which are often used in sequence modeling, to process sequential data such as sentences or words. Core to the Transformer model lies the self-attention mechanism, alternatively termed scaled dot product attention, or simply attention. When analyzing a string, self-attention allows the model to evaluate the importance of different words or tokens. It extracts the correlation between each word and all other words, thereby preserving context not solely within a single

sentence but across the entire text. Through the stacking of multiple layers comprising self-attention and feedforward NNs, the Transformer model adeptly captures long-term dependencies and intricate patterns within sequential data, rendering it a potent architecture for a diverse array of NLP tasks.

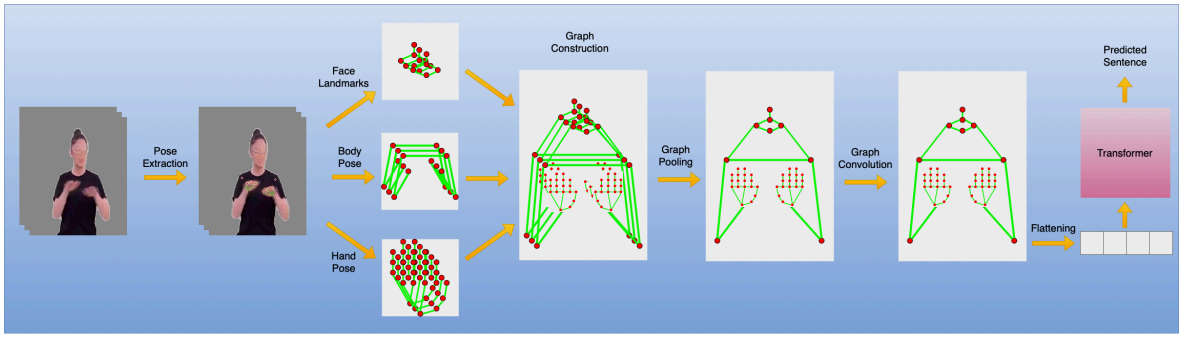


Figure 6.3 Architecture of the GNN-T Model

In this thesis, we used transformers for the translation process in both our P2T-T and GNN-T models. Our final proposed model for GNN-T is shown in Figure 6.3.

## 6.7. Training

For all of our experiments, we consistently established the dropout ratio as 0.1 in order to mitigate the risk of overfitting the model. We employed the Adam [41] optimizer to train our networks, utilizing a learning rate of $5\text{x}10^{-5}$ ($\beta_1 = 0.9, \beta_2 = 0.98$). We employed plateau learning rate (LR) scheduling, which monitors the performance of the development set. If there was no enhancement in the development score after 7 epochs, we decreased the LR by a factor of 0.7. We proceeded with this procedure until the LR reached $10^{-6}$. We used the batch size as 16 for the GNN-T model and 4 for the P2T-T model.

# 7. EXPERIMENTAL RESULTS

In this chapter, we present the quantitative and qualitative results of the experiments we conducted with our P2T-T and GNN-T baseline models on the E-TSL dataset we created within the scope of this study. We explain in the ablation study section how we reached the best parameters of our GNN-T model, which yielded the best results. To verify the performance of our P2T-T and GNN-T baseline models, we compare them with the baseline results of the PHOENIX14**T** dataset. Then, we compare the preliminary results we obtained with the E-TSL dataset with other datasets. In all experimental results in this chapter, ROUGE-L and BLEU metrics are presented as percentage (%).

## 7.1. Quantitative Results on E-TSL Dataset

Table 7.1 displays the preliminary results of our training process employing two models, namely P2T-T and GNN-T, on the E-TSL dataset. GNN-T model had superior performance in capturing longer matching sequences, as evidenced by its higher ROUGE-L score. Moreover, the GNN-T model exhibited superior performance in terms of BLEU scores, which can be attributed to the efficient message passing provided by the graph convolution method employed in this model. In addition, we used graph pooling to decrease the sequence length by a factor of three, which resulted in shorter training time and a simpler model. In general, the GNN-T model not only achieved a better score compared to the P2T-T model, but also offered more convenience due to its shorter sequence length.

Table 7.1 Model Comparison on E-TSL Dataset

| | **P2T-T** | | **GNN-T** | |
|---|---|---|---|---|
| | DEV | TEST | DEV | TEST |
| ROUGE-L | 22.42 | 22.09 | **23.42** | **22.93** |
| BLEU-1 | 18.26 | 18.09 | **21.26** | **21.01** |
| BLEU-2 | 8.27 | 8.20 | **9.51** | **9.13** |
| BLEU-3 | 4.82 | 4.78 | **5.48** | **5.20** |
| BLEU-4 | 3.28 | 3.23 | **3.65** | **3.49** |

## 7.2. Qualitative Results on E-TSL Dataset

In this section, we present a comparison of the sentences predicted by our GNN-T model, which achieved the best results on the E-TSL dataset, with the reference sentences in Table 7.2. In the table, we have presented the predicted sentences in order from best to worst. In reality, we train our model with sentence groups (texts) rather than individual sentences. However, since each text would be too long, we have chosen to present these examples in the form of sentences.

As can be seen from the examples in the table, sentences and words that are repeated frequently are predicted much better by the model. Considering the BLEU-4 score (3.49) of our GNN-T model, it is undeniable that there is a significant number of incorrectly predicted words and sentences. Moreover, most of the errors arise from the incorrect prediction of suffixes, which highlights the additional challenge posed by the agglutinative structure of the Turkish language for sign language translation tasks.

Considering all these results, we observe that the high proportion of singletons (64%) and rare words (85%) in our dataset makes it challenging for our models to learn effectively. We believe that adding new data with similar content to our dataset will improve the performance of our models.

## 7.3. Ablation Study

We conducted various experiments to determine the parameters of our best GNN-T model, the results of which we present in Table 7.1. In the upcoming sections, we present the results of different experiments that we have conducted for parameters such as the embedding size, the number of heads, the number of encoder and decoder layers, the number of neurons in the feed forward network, the dropout rate and the learning rate. When evaluating the experimental results, we based on the BLEU-4 scores we obtained on the test set.

While performing these experiments, we used the parameters in the 7.3 Table that we determined empirically. With the parameters we used as the basis, our GNN-T model reached

Table 7.2 Predicted and Reference Text Comparisons

| | |
|---|---|
| Reference: | kendinize iyi bakın sağlıkla kalın hoşça kalın.<br>(Take care of yourself, stay healthy and goodbye.) |
| Predicted: | kendinize iyi bakın sağlıkla kalın hoşça kalın. |
| Reference: | konularımızı tekrar edelim konularla ilgili testler çözelim ve kitap okumayı ihmal etmeyelim.<br>(Let's review our topics, solve related tests, and make sure we don't forget to read books.) |
| Predicted: | konularımızı tekrar edelim konuları ilgili testler çözelim ve kitap okumayı ihmal etmeyelim. |
| Reference: | bir dahaki dersimize kadar eba tvden dersleri takip edelim.<br>(Follow the lessons on eba tv until our next lesson.) |
| Predicted: | beni dahaki dersimize kadar eba tvden dersleri takip edelim |
| Reference: | merhaba çocuklar ben türkçe öğretmeniniz fatih badak.<br>(Hello children, I am your Turkish teacher Fatih Badak.) |
| Predicted: | bu sevgili ben türkçe öğretmeniniz fatih badak. |
| Reference: | verilenlere göre aşağıdakilerden hangisi kesinlikle yanlıştır hemen bilgilerimizi okuyalım.<br>(According to the information given, which of the following is absolutely wrong? Let's read our information immediately.) |
| Predicted: | aşağıdaki cümleler göre aşağıdaki hangisi kesinlikle yanlıştır hemen bakalımmizi okuyalım. |
| Reference: | bu metinde vurgulanmak istenen düşünce aşağıdakilerden hangisidir<br>(Which of the following is the idea that is wanted to be emphasized in this text?) |
| Predicted: | merhaba metinde aşağıdakilanmak istenen düşünce aşağıdakilerden hangisi getiril |
| Reference: | a seçeneğimiz gezi yazısı b haber yazısı c hikaye d anı.<br>(a option is a travel article b a news article c a story d a memoir.) |
| Predicted: | a seçeneğindezde yazısı b seçeneğin c seçeneği unsurlarıye d sevgili |
| Reference: | aşağıdaki cümleler ile yay ayraç içinde verilen sözcükleri eşleştiriniz denilmiş.<br>(You are asked to match the following sentences with the words in brackets.) |
| Predicted: | aşağıdakilerinden ilgili ayraç içinde verilen bilgilerler ve görevlielim.ilmiş. |

Table 7.3 Base Model Parameters

| Embedding | Heads | Layers | FF Dimension | Dropout | LR |
|---|---|---|---|---|---|
| 1024 | 2 | 6 | 2048 | 0.1 | 0.00005 |

a BLEU-4 score of 3.49. We left all other parameters the same except the corresponding parameter in each experimental step below. Thus, we aimed to measure the effect of each parameter on the learning of the model. We also verified that the parameters we set were the best ones.

### 7.3.1. Embedding Size

Embedding size specifies the vector's size used to represent a word or input element. Embedding size determines the model's learning capacity. A larger embedding size can increase the capacity of the model to learn more complex relationships. However, this also means that the model will need more parameters and computational resources.

Embedding size can affect how the model generalizes to data. A larger embedding size generally provides a more general and generalizing representation. However, too large an embedding size may increase the risk of overfitting. Larger embedding sizes can be more sensitive to finer details because they carry more information in more dimensions. However, this can make it more susceptible to data noise and variances in the training data. When choosing the embedding size, the model's learning ability, generalization ability, computational resources, and the specific tasks for which the model will be used should be considered.

Table 7.4 Embedding Size Comparison

| Embedding Size | BLEU-4 Score |
|---|---|
| 256 | 2.69 |
| 512 | 2.76 |
| **1024** | **3.49** |

In this experiment, we tested sizes 256, 512 and 1024 as embedding size. As seen in Table 7.4, we achieved the best result with 1024 embedding size. Our hardware was not sufficient to test more than 1024 embedding size.

### 7.3.2. Number of Heads

The number of heads affects how deeply the model can learn information. The heads' count makes the attention mechanism included in the model more complex. More heads can help capture a broader context by considering more features. However, this increase may cause the model to need more parameters and longer training times. Increasing the number of heads can often provide better performance on more complex and diverse text data, but this may increase the risk of overfitting on smaller and simpler datasets.

Table 7.5 Number of Heads Comparison

| Number of Heads | BLEU-4 Score |
|---|---|
| **2** | **3.49** |
| 4 | 3.37 |
| 8 | 3.28 |

We conducted our experiments for 2, 4 and 8 heads. As seen in the Table 7.5, we achieved the best result with 2 heads. We observed that the performance decreases as the number of heads increases.

### 7.3.3. Number of Layers

The number of encoder-decoder layers significantly affects the learning ability and overall performance of the Transformer model. Increasing the encoder-decoder layers' count allows the model to have more learning capacity. A deeper model can learn more complex relationships and understand a more general and broader context. However, this requires more parameters and computational resources. More layers may increase the risk of

overfitting. More layers may cause the model to overfit, especially if the training data is limited or noisy. More layers can increase training time because a deeper model requires more calculations.

Table 7.6 Number of Layers Comparison

| Number of Layers | BLEU-4 Score |
|---|---|
| 2 | 2.71 |
| 3 | 2.73 |
| 4 | 3.09 |
| **6** | **3.49** |

It is common to use 2 or 3 layers in transformer models. We carried out our experiments for 2, 3, 4 and 6 layers. As shown in Table 7.6, the model with 6 layers provided the best performance with a BLEU-4 score of 3.49.

### 7.3.4. Feed-Forward Dimension

The feed-forward layer processes the outputs from the attention layer in a non-linear manner. This layer transforms the context vectors from the attention layer into higher-level features. The feed-forward layer increases the learning capacity of the model. By processing context vectors from attention layers in a more complex way, it allows the model to better learn more complex relationships and structures.

Table 7.7 Feed Forward Dimension Comparison

| Dimension | BLEU-4 Score |
|---|---|
| 512 | 2.94 |
| 1024 | 3.05 |
| **2048** | **3.49** |

In this experiment, we trained our model for 512, 1024 and 2048 dimentions, as presented in the Table 7.7. We achieved the best result with 2048. We could not experiment beyond 2048 due to lack of hardware.

### 7.3.5. Dropout Ratio

Dropout is a technique of regularization used to reduce overfitting in neural network models. The basic idea is that randomly selected units (usually neurons) are ”removed” or “dropped” during training. This allows the model to be trained on a different network subset for each training example. Dropout prevents the network from overfitting and helps it learn a more general representation.

Table 7.8 Dropout Ratio Comparison

| Dropout Ratio | BLEU-4 Score |
|---|---|
| **0.1** | **3.49** |
| 0.2 | 2.88 |
| 0.3 | 3.40 |

In these experiments, we tested 0.1, 0.2 and 0.3 dropout values and obtained the best result with a ratio of 0.1, as seen in the Table 7.8.

### 7.3.6. Learning Rate

Learning rate (LR) is the parameter that determines the speed of weight updates of neural networks during the training process. A high learning rate may make it difficult for the model to converge to the global minimum, as weight updates will be made in large steps. Besides, a high LR helps prevent the model from overfitting.

If the learning rate is too low, it causes the model to take longer to reach the global minimum, thus prolonging the training process. Additionally, a low LR may cause the model to overfit.

We used LRs of 0.001, 0.0001, 0.00005 and 0.00001 in our experiments. As seen in the Table 7.9, we achieved the best result with a LR of 0.00005. We found that the model could not learn with a LR of 0.001.

Table 7.9 Learning Rate Comparison

| Learning Rate | BLEU-4 Score |
|---|---|
| 0.001 | 0.0 |
| 0.0001 | 3.48 |
| **0.00005** | **3.49** |
| 0.00001 | 2.76 |

### 7.3.7. Best Model Parameters

As a result of the experiments we conducted on the E-TSL dataset with our GNN-T model, we present the model parameters with which we achieved the best performance in the Table 7.10.

Table 7.10 Best Model Parameters

| Embedding | Heads | Layers | FF Dimension | Dropout | LR |
|---|---|---|---|---|---|
| 1024 | 2 | 6 | 2048 | 0.1 | 0.00005 |

## 7.4. Model Validation with PHOENIX14T Dataset

We employed the PHOENIX14**T** dataset, widely recognized as a benchmark in the domain of SLT, to validate our P2T-T and GNN-T models, which we propose for use on the newly established E-TSL dataset. We present initial results on the PHOENIX14**T** dataset without extensively fine-tuning the parameters of our P2T-T and GNN-T models in Table 7.11.

In parallel with the results obtained on the E-TSL dataset, our GNN-T model outperformed our P2T-T model on the PHOENIX14 dataset. Notably, this time, in addition to BLEU scores, our GNN-T model yielded superior results compared to our P2T-T model in terms of ROUGE-L score.

There have been studies that conducted comprehensive research on the PHOENIX14**T** dataset and attained relatively high scores. Given that we utilize baseline models in this thesis, we deem it appropriate to compare the performances of the models with the baseline

Table 7.11 Model Comparison on PHOENIX14**T** Dataset

| | TwoStream - SLT[26] | JEE-SLT[7] | SMM - TLB[29] | NSLT[11] | Our P2T-T | Our GNN-T |
|---|---|---|---|---|---|---|
| | DEV SET | | | | | |
| ROUGE-L | **54.08** | - | 53.10 | 31.80 | 34.16 | **34.98** |
| BLEU-1 | **54.32** | 45.54 | 53.95 | 31.87 | 30.72 | **31.76** |
| BLEU-2 | **41.99** | 32.60 | 41.12 | 19.11 | 17.65 | **18.62** |
| BLEU-3 | **34.15** | 25.30 | 33.14 | 13.16 | 11.22 | **12.38** |
| BLEU-4 | **28.66** | 20.69 | 27.61 | 9.94 | 7.85 | **9.02** |
| | TEST SET | | | | | |
| ROUGE-L | **53.48** | - | 52.65 | 31.80 | 33.79 | **35.46** |
| BLEU-1 | **54.90** | 45.34 | 53.97 | 32.24 | 27.97 | **32.34** |
| BLEU-2 | **42.43** | 32.31 | 41.75 | 19.03 | 16.47 | **19.28** |
| BLEU-3 | **34.46** | 24.83 | 33.84 | 12.83 | 10.63 | **12.81** |
| BLEU-4 | **28.95** | 20.17 | 28.39 | 9.58 | 7.31 | **8.93** |

model results of PHOENIX14**T**. Our GNN-T and P2T-T models achieved ROUGE-L scores of 35.46 and 33.79, respectively, on the test set. Both models exhibited superior performance compared to the NSLT baseline model, which attained a ROUGE-L score of 31.80. Additionally, the 32.34 BLEU-1 score and 19.28 BLEU-2 score obtained with our GNN-T model surpassed NSLT's 32.24 BLEU-1 score and 19.03 BLEU-2 score. While 12.81 BLEU-3 score and 8.93 BLEU-4 score achieved with our GNN-T model are competitive, they slightly lag behind 12.83 BLEU-3 score and 9.58 BLEU-4 score of the NSLT model. These results collectively indicate the effectiveness of both our GNN-T and P2T-T models.

## 7.5. Datasets Comparison

Table 7.12 presents our highest BLEU-4 scores, providing a comparative analysis between the E-TSL dataset and other datasets commonly employed SLT researchs. Additionally, we include the Elementary23-SLT dataset, which consists of lesson videos at the elementary school level, in our evaluation due to its thematic similarities with our dataset.

In the statistics presented in Table 5.3, it is noteworthy that despite having a vocabulary size similar to that of our E-TSL dataset, the SWISSTXT-NEWS and VRT-NEWS datasets exhibit a lower singletons ratio and rare words ratio. Furthermore, there is a

Table 7.12 Benchmarking BLEU-4 Scores

| | DEV | TEST |
|---|---|---|
| PHOENIX14**T**[11] | 9.94 | 9.58 |
| Elementary23-SLT[34] | 6.67 | 5.69 |
| Elementary23-RAW[34] | 0.36 | 0.00 |
| SWISSTXT-NEWS[39] | 0.46 | 0.41 |
| VRT-NEWS[39] | 0.45 | 0.36 |
| Our E-TSL (P2T-T) | 3.28 | 3.23 |
| Our E-TSL (GNN-T) | 3.65 | 3.49 |

distinction between the datasets in terms of image quality, with the SWISSTXT-NEWS and VRT-NEWS datasets offering HD resolution as shown in Table 5.2. Despite all this, upon initial examination of the outcomes, it becomes evident that the BLEU-4 scores for the SWISSTXT-NEWS and VRT-NEWS datasets notably lag behind those of other datasets, including ours. This observation suggests that these benchmarks, being more generalized and possibly lacking sufficient data, may not effectively train state-of-the-art models, thereby potentially compromising their reliability for future SLT researches.

As shown in Table 5.2, the 210x260 resolution of the PHOENIX14**T** dataset promises slightly better quality than the 190x230 resolution of our E-TSL dataset. The PHOENIX14**T** dataset exhibits considerably narrower thematic content, exclusively comprising weather news videos, consequently resulting in a more limited vocabulary as shown in Table 5.3. Consequently, the baseline model's BLEU-4 score for the PHOENIX14**T** dataset slightly surpasses the BLEU-4 score attained in our dataset. It is pertinent to note that the data modalities employed in our model and the baseline model described in [11] differ.

Comparatively, the baseline model outcomes of the Elementary23-SLT dataset appear superior to those of our dataset's baseline model. The Elementary23-SLT dataset is a tailored subset derived from the larger Elementary23 dataset [34] comprising approximately 71 hours of data. The creation of the Elementary23-SLT dataset involved various preprocessing techniques applied to the Elementary23 RAW dataset, such as singleton reduction, augmentation of frequently used words, retention of diverse course content, and selection of the most relevant content for SLT purposes. Under these conditions,

employing the Elementary23 RAW dataset yielded 0.36 and 0.00 BLEU-4 scores for the development and test sets, respectively. However, with the curated SLT subset, notably higher BLEU-4 scores of 6.67/5.69 were reported for the development and test sets, respectively. As shown in Table 5.3, although the Elementary23-SLT dataset possesses a marginally larger vocabulary size compared to the E-TSL dataset (8.202/6.980), its singleton rates (41%/64%) and rare word rates (75%/85%) are lower than those of the E-TSL dataset. Moreover, it can be seen at Table 5.2 that the resolution (1280x720) of videos in the Elementary23-SLT dataset significantly exceeds that of videos in our dataset (190x230). Additionally, videos in the Elementary23-SLT dataset were captured under optimal conditions with a single-color background, facilitating clearer perception of signer movements. These divergences elucidate the performance disparity observed between the baseline models of both datasets.

# 8. CONCLUSION

This thesis contributes to the field of Continuous SLT, particularly for TSL. We introduce the E-TSL dataset, which is designed to meet the specific needs of this area. This dataset provides researchers with important resources for training and evaluating their models, helping to advance the field.

Our work goes beyond just creating a dataset; we have also developed and trained two new models, P2T-T and GNN-T, using the E-TSL dataset. These models are specifically designed for the Sign Pose2Text task, an important part of sign language translation. The results we have achieved show promising performance, suggesting that our approach is effective.

To confirm the robustness and generalizability of our models, we evaluated them using the PHOENIX14**T** dataset, a widely recognized benchmark in the field. This comparative analysis validates the performance of our models and offers insights into their adaptability across different datasets.

We have identified several areas for future research and improvement based on our findings. One important focus is improving how videos are segmented within our dataset, aiming to divide them into coherent sentences. By achieving this finer granularity, we can perform more detailed analyses and set benchmarks using 1-minute segments, which would allow for more precise evaluation metrics.

Moreover, we plan to explore the utilization of various pose extraction tools other than MediaPipe, aiming to optimize model performance further. By investigating different approaches, we seek to uncover potential enhancements that could be integrated into our framework.

Additionally, we plan to explore how combining our models with other state-of-the-art techniques can enhance the overall performance of SLT systems. This effort aims to push the boundaries of what's currently possible in Continuous Sign Language Translation.

# REFERENCES


[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., **2017**.

[2] Deafness and hearing loss. `https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss`. Accessed: 04-13-2024.

[3] Disabled and elderly statistics bulletin. `https://aile.gov.tr/media/135432/eyhgm_istatistik_bulteni_nisan_23.pdf`. Accessed: 2024-04-13.

[4] A. O. Tur and H. Y. Keles. Isolated sign recognition with a siamese neural network of rgb and depth streams. In *IEEE EUROCON 2019 -18th International Conference on Smart Technologies*, pages 1–6. **2019**. doi:10.1109/EUROCON.2019.8861945.

[5] A. O. Tur and H. Y. Keles. Evaluation of hidden markov models using deep cnn features in isolated sign recognition. *Multimedia Tools and Applications*, 80:19137 – 19155, **2021**. doi:https://doi.org/10.1007/s11042-021-10593-w.

[6] D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudrealt, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef, C. Vogler, and M.R. Morris. Sign language recognition, generation, and translation: An interdisciplinary perspective. pages 16–31. **2019**. ISBN 978-1-4503-6676-2. doi:10.1145/3308561.3353774.

[7] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2020**.

[8] M. Huenerfauth, L. Zhao, E. Gu, and J. Allbeck. Evaluation of american sign language generation by native asl signers. *ACM Trans. Access. Comput.*, 1(1), **2008**. ISSN 1936-7228. doi:10.1145/1361203.1361206.

[9] M. Kipp, Q. Nguyen, A. Heloir, and S. Matthes. Assessing the deaf user perspective on sign language avatars. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '11, page 107–114. Association for Computing Machinery, New York, NY, USA, **2011**. ISBN 9781450309202. doi:10.1145/2049536.2049557.

[10] S. Morrissey and W. Andy. An example-based approach to translating sign language. In *Machine Translation Summit*. **2005**.

[11] 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*, pages 7784–7793. **2018**. doi:10.1109/CVPR.2018.00812.

[12] O. M. Sincan and H. Y. Keles. Autsl: A large scale multi-modal turkish sign language dataset and baseline methods. *IEEE Access*, 8:181340–181355, **2020**. doi:10.1109/ACCESS.2020.3028072.

[13] O. M. Sincan and H. Y. Keles. Using motion history images with 3d convolutional networks in isolated sign language recognition. *IEEE Access*, 10:18608–18618, **2022**. doi:10.1109/ACCESS.2022.3151362.

[14] O. Özdemir, A. A. Kandıroğlu, N. C. Camgöz, and L. Akarun. Bosphorussign22k sign language recognition dataset. 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*, pages 181–188. European Language Resources Association (ELRA), Marseille, France, **2020**. ISBN 979-10-95546-54-2.

[15] Ş. Öztürk and H. Y. Keles. E-TSL: A Continuous Educational Turkish Sign Language Dataset with Baseline Methods. In *Proceedings of the 6th International*

*Congress on Human-Computer Interaction (ICHORA)*. **2024**. Also available as Arxiv preprint: https://doi.org/10.48550/arXiv.2405.02984.

[16] W. Weaver. Translation. In William N. Locke and A. Donald Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA, **1949/1955**. Reprinted from a memorandum written by Weaver in 1949.

[17] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, **1990**.

[18] Y. Bengio, R. Ducharme, and P. Vincent. A neural probabilistic language model. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, **2000**.

[19] N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 3:1700–1709, **2013**.

[20] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, **2014**.

[21] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, **2014**.

[22] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, **2014**.

[23] F. Wen, Z. Zhang, T. He, and C. Lee. Ai enabled sign language recognition and vr space bidirectional communication using triboelectric smart glove. *Nature Communications*, 12:5378, **2021**. doi:10.1038/s41467-021-25637-w.

[24] N. C. Gokul, L. Manideep, N. Sumit, S. Prem, K. Pratyush, and M. K. Mitesh. Addressing resource scarcity across sign languages with multilingual

pretraining and unified-vocabulary datasets. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. **2022**.

[25] H. Bull, T. Afouras, G. Varol, S. Albanie, L. Momeni, and A. Zisserman. Aligning subtitles in sign language videos, **2021**.

[26] Y. Chen, R. Zuo, F. Wei, Y. Wu, S. LIU, and B. Mak. Two-stream network for sign language recognition and translation. In *Advances in Neural Information Processing Systems*. **2022**.

[27] B. Saunders, N. C. Camgoz, and R. Bowden. Progressive transformers for end-to-end sign language production. In *Computer Vision – ECCV 2020*, pages 687–705. Springer International Publishing, Cham, **2020**. ISBN 978-3-030-58621-8.

[28] J. Kreutzer, J. Bastings, and S. Riezler. Joey NMT: A minimalist NMT toolkit for novices. *CoRR*, abs/1907.12484, **2019**.

[29] 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)*, pages 5110–5120. **2022**. doi:10.1109/CVPR52688.2022.00506.

[30] K. L. Cheng, Z. Yang, Q. Chen, and Y.W. Tai. *Fully Convolutional Networks for Continuous Sign Language Recognition*, pages 697–714. **2020**. ISBN 978-3-030-58585-3. doi:10.1007/978-3-030-58586-0_41.

[31] A. A. Hosain, P. S. Santhalingam, P. Pathak, H. Rangwala, and J. Košecká. Hand pose guided 3d pooling for word-level sign language recognition. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3428–3438. **2021**. doi:10.1109/WACV48630.2021.00347.

[32] H. Hu, W. Zhao, W. Zhou, Y. Wang, and H. Li. Signbert: Pre-training of hand-model-aware representation for sign language recognition. pages 11067–11076. **2021**. doi:10.1109/ICCV48922.2021.01090.

[33] J. Kan, K. Hu, M. Hagenbuchner, A. Tsoi, M. Bennamoun, and Z. Wang. Sign language translation with hierarchical spatio-temporal graph neural network. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2131–2140. IEEE Computer Society, Los Alamitos, CA, USA, **2022**. doi:10.1109/WACV51458.2022.00219.

[34] 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. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1958–1967. IEEE Computer Society, Los Alamitos, CA, USA, **2023**. doi:10.1109/ICCVW60793.2023.00211.

[35] A. Ali and S. Renals. Word error rate estimation for speech recognition: e-WER. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. **2018**. doi:10.18653/v1/P18-2004.

[36] P. Kishore, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318. Association for Computational Linguistics, USA, **2002**. doi:10.3115/1073083.1073135.

[37] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. 10, pages 74–81. **2004**.

[38] A. Dutta and A. Zisserman. The via annotation software for images, audio and video. In *27th ACM International Conference on Multimedia*. **2019**.

[39] N. C. Camgoz, B. Saunders, G. Rochette, M. Giovanelli, G. Inches, R. Nachtrab-Ribback, and R. Bowden. Content4all open research sign language translation datasets. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 1–5. IEEE Computer Society, Los Alamitos, CA, USA, **2021**. doi:10.1109/FG52635.2021.9667087.

[40] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. L. Chang, M. Yong, J. Lee, W. T. Chang, W. Hua, M. Georg, and M. Grundmann. Mediapipe: A framework for building perception pipelines, **2019**.

[41] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, **2014**.