

CONTINUOUS SIGN LANGUAGE TRANSLATION WITH PRETRAINED
LANGUAGE MODELS FOR TURKISH SIGN LANGUAGE (TİD)



KARAHAN ŞAHİN

BOĞAZİÇİ UNIVERSITY

2025

CONTINUOUS SIGN LANGUAGE TRANSLATION
WITH PRETRAINED LANGUAGE MODELS FOR TURKISH SIGN
LANGUAGE (TİD)

Thesis submitted to the
Institute for Graduate Studies in Social Sciences
in partial fulfillment of the requirements for the degree of

Master of Arts
in
Cognitive Science

by
Karahana Şahin

Boğaziçi University

2025

Continuous Sign Language Translation
with Pretrained Language Models for Turkish Sign Language (TID)

The thesis of Karahan Şahin
has been approved by:

Assoc. Prof. Kadir Gökğöz
(Thesis Advisor)

Prof. Lale Akarun
(Thesis Co-Advisor)

Prof. Murat Saraçlar

Assist. Prof. Ümit Atlamaz

Assoc. Prof. Bahtiyar Makaroğlu
(External Member)

January 2025

DECLARATION OF ORIGINALITY

I, Karahan Şahin, certify that,

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature:

Date:

ABSTRACT

Continuous Sign Language Translation with Pretrained Large Language Models for Turkish Sign Language (TID)

Sign Language Translation (SLT) is a machine learning task that aims to provide accessibility systems for the deaf and hard-of-hearing community. With the recent advances within the area, SLT technologies have shifted towards large-scale training with pretrained large language models (LLMs). However, most technologies and datasets are provided for American Sign Language (ASL) and British Sign Language (ASL). To achieve similar advances for Turkish Sign Language (TID) translation, we are proposing the first large-scale open-source translation dataset for TID, comprising over 500 hours of video footage and aligned Turkish translations by utilizing publicly available YouTube content. In this work, our dataset provides a scalable and reproducible framework for SLT research. We propose a comprehensive translation system with various inter-changeable components designed for SLT on an agglutinative language to compensate for the learning alignment between both low-resource languages, namely TID and Turkish. This body of work compares feature extraction strategies (Linear layers, Multi-layer Perceptrons, and Convolutional encoders), model sizes, and their pretraining approaches (multilingual versus Turkish-specific) to the effect on sign translation performance. Innovative training and evaluation techniques, including paraphrasing-based reference generation and sign language-specific augmentations, are also introduced for developing an SLT benchmark for TID.

ÖZET

Ön-eğitilmiş Dil Modelleri ile Türk İşaret Dili için İşaret Dili Çevirisi

İşaret Dili Çevirisi (Sign Language Translation - SLT), işitme engelli ve işitme güçlüğü çeken bireyler için erişilebilirlik sistemleri sağlamayı amaçlayan bir yapay öğrenme görevidir. Bu alandaki son gelişmelerle birlikte, SLT kapsamında geliştirilen teknolojiler, büyük dil modelleri (Large Language Models - LLMs) ile büyük ölçekli eğitime doğru kaymıştır. Ancak, mevcut teknolojiler ve veri setlerinin büyük çoğunluğu Amerikan İşaret Dili (ASL) ve İngiliz İşaret Dili (BSL) için mevcuttur. Türk İşaret Dili (TID) için bu yeni teknolojilerin uygulanabilmesini sağlamak amacıyla, 500 saatten fazla video içeriği ve bunlara hizalanmış Türkçe çevirilerden oluşan, kamuya açık YouTube içeriklerini kullanarak oluşturulmuş ilk büyük ölçekli açık veri setini öneriyoruz. Bu veri seti, SLT araştırmaları için ölçeklenebilir ve tekrarlanabilir bir çerçeve sunmaktadır. Bu çalışma, işaret dili çeviri performansı üzerindeki etkilerini inceleyerek farklı özellik çıkarım stratejilerini (Lineer katman, Çok Katmanlı Algılayıcı (MLP), Evrişim (Conv1D) kodlayıcılar), model boyutlarını ve ön eğitim yaklaşımlarını (çok dilli veya Türkçe'ye özgü) karşılaştırmaktadır. Ayrıca, dil odaklı uygulamalara yönelik teorik bir temelden türetilen, referans çeviri artımı için yeniden ifade (paraphrasing) ve işaret dili odaklı veri artımı (augmentation) yöntemleri ile bu çalışmada Türk İşaret Dili'nden Türkçe'ye çeviri için denektaş sistemi sağlamaktadır.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisors, Prof. Lale Akarun and Assoc. Prof. Kadir Gökğöz, for their invaluable guidance, support, and patience throughout this work. I am especially thankful to Prof. Lale Akarun for accepting me as a master's student and providing me the opportunity to work on various projects that ultimately made this thesis possible. I am equally grateful to Assoc. Prof. Kadir Gökğöz for his unwavering support and insightful guidance, starting from my undergraduate years, that made this interdisciplinary work on Turkish Sign Language, forming the core of this thesis. It has been an honor to work with them on this work, prior projects, and possible future projects.

I would also like to extend my sincere thanks to Prof. Murat Saraçlar for his support, not only as a committee member but also for his invaluable feedback throughout this work. I am particularly grateful for the opportunities he provided to work on base systems, which served as an inspiration for the work presented in this thesis. My heartfelt thanks go to Assoc. Prof. Bahtiyar Makaroğlu and Assist. Prof. Ümit Atlamaz for their valuable feedback on both the theoretical and applicational aspects of this work. Their perspectives and suggestions have been crucial in refining this work.

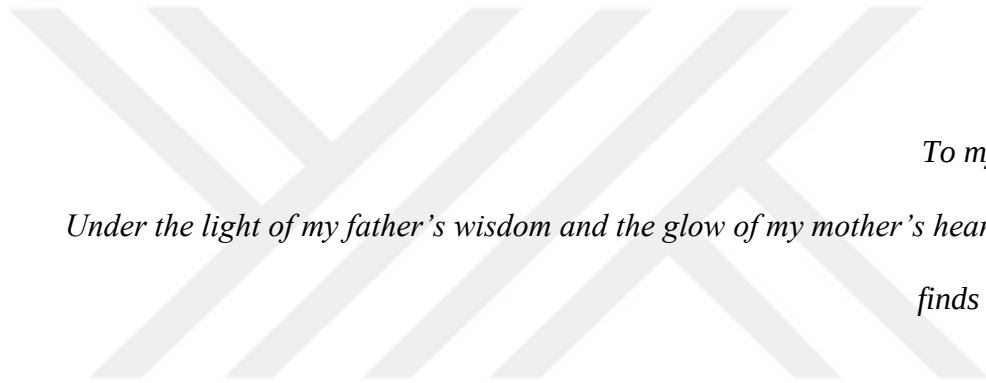
I am deeply grateful to the TABILAB group at Boğaziçi University's Computer Science Department for their help in training their model, TURNA, which played a key role in this work. I would like to specifically thank Prof. Tunga Güngör, Prof. Suzan Üsküdarlı, and Dr. Onur Güngör for their invaluable insights and guidance on training and evaluating the model. I am also thankful to Gökçe Uludoğan and Zeynep Yirmibeşoğlu for their support and contributions to the

implementation of the model, which significantly facilitated the progress of this research.

I extend my sincere thanks to my fellow researchers in our lab, PILAB, particularly Oğulcan Özdemir and Mehmet Özer, for their unwavering support and insightful brainstorming sessions. Their ideas and collaborative spirit have been incredibly valuable to the development of this work.

The computations in this thesis are partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources), and EuroHPC LUMI, whose funding and resources were instrumental in completing this research. We acknowledge EuroHPC Joint Undertaking for awarding project ID EHPC-AI-2024A03-075 access to LUMI at CSC, Finland.

Finally, I owe my deepest gratitude to my parents, Elif and Orhan, and my little brother, Dağhan, whose endless support and compassion have been the foundation of my pursuit of science. Their unwavering belief in my abilities and their encouragement through every challenge have been my greatest source of strength.



To my parents....

*Under the light of my father's wisdom and the glow of my mother's heart, each step
finds its purpose.*

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION	1
1.1 Task definition.....	3
1.2 Task application	5
1.3 Task significance	9
1.4 Task limitations	10
1.5 Contributions	14
1.6 Thesis overview.....	15
CHAPTER 2: RELATED WORKS.....	17
2.1 Datasets	17
2.2 Sign language recognition.....	25
2.3 Sign language translation	26
CHAPTER 3: DATASET	36
3.1 Dataset statistics	37
3.2 Dataset collection	40
3.3 Dataset pre-processing	42
3.4 Dataset post-processing.....	47
CHAPTER 4: METHODOLOGY	49
4.1 Neural machine translation task	49
4.2 Feature transformation	51
4.3 Feature normalization.....	56
4.4 Pre-trained large language models (LLMs).....	57
4.5 Scoring	60
4.6 Paraphrasing.....	61

4.7 Pose augmentation.....	63
CHAPTER 5: EXPERIMENTAL SETUP.....	65
5.1 Pose estimation.....	65
5.2 Dataset configuration	69
5.3 Training configuration.....	71
5.4 Generation configuration.....	72
5.5 Evaluation protocol	72
5.6 Tools.....	73
CHAPTER 6: EXPERIMENTAL RESULTS	74
6.1 Comparison between model sizes	75
6.2 Comparison between encoder types	79
6.3 Addition of paraphrasing and pose augmentation in training	84
CHAPTER 7: DISCUSSION.....	90
7.1 Input modality	90
7.2 Paraphrasing and linguistic difference	92
CHAPTER 8: FUTURE WORK.....	100
8.1 Corpus creation	100
8.2 Improvements on architecture	102
CHAPTER 9: CONCLUSION.....	104
APPENDIX A: OVERALL EXPERIMENTS.....	106
REFERENCES.....	110

LIST OF TABLES

Table 1. Overview of Sign Language Translation Datasets.....	24
Table 2. Dataset Composition: Unique Videos, Clips, and Duration	38
Table 3. Clip Length Analysis Across Dataset Subsets	38
Table 4. Model Parameters for Initializing MediaPipe Holistic Pipeline	65
Table 5. Hyperparameters of mT5-small, mT5-large, and TURNA Models.....	69
Table 6. Statistics of the SC Split.....	70
Table 7. Statistics of the LC Split	70

LIST OF APPENDIX TABLES

Table A1. Experimental Results for SLT on SC Dataset (Development and Test Scores	106
Table A2. Experimental Results for SLT on SC Dataset with Augmentations (Development and Test Scores.....	107
Table A3. Experimental Results for SLT on LC Dataset (Development and Test Scores	108
Table A4. Experimental Results for SLT on LC Dataset with Augmentations (Development and Test Scores.....	109

LIST OF FIGURES

Figure 1. Overview of the SLT task.....	1
Figure 2. Common architectures proposed in the literature for SLT	30
Figure 3. The dataset content of unique signers.....	36
Figure 4. Distribution of clip durations across subsets	39
Figure 5. The category distribution across clips on both SC and LC datasets.....	40
Figure 6. Illustration of audio caption and signing span alignment: blue shows the audio-caption span, green the signing span, red null signing, and yellow the added offset for full signing coverage	45
Figure 7. The figure shows the overall pipeline of the fine-tuning stage	50
Figure 8. Visualization of normalized vs. unnormalized pose sequences: first column (red) shows unnormalized skeletal data, while the second (purple) displays normalized cues—pose, left hand, right hand, and face.....	57
Figure 9. The original Transformer architecture.....	59
Figure 10. Components of experimental setup in serial order	65
Figure 11. BLEU-4 results on test set for all experiments, with a vertical red line- separating models trained on SC or LC subset	74
Figure 12. BLEU-1 and BLEU-2 performance comparison by model type across SC development and test sets	76
Figure 13. BLEU-3 and BLEU-4 performance comparison by model type across SC development and test sets	77
Figure 14. BLEU-4 performance comparison by model type across LC development and test sets.....	79

Figure 15. BLEU-4 performance comparison by encoder type across LC development and test sets	80
Figure 16. BLEU-1 and BLEU-2 performance comparison by encoder type across SC development and test sets	81
Figure 17. BLEU-3 and BLEU-4 performance comparison by encoder type across SC development and test sets	83
Figure 18. Comparison of BLEU-1 to BLEU-4 scores between paraphrased and non- paraphrased SC and LC test sets	84
Figure 19. BLEU-4 performance comparison on effect of augmentation across model and encoder type configurations for SC subset	86
Figure 20. BLEU-4 performance comparison on effect of augmentation across model and encoder type Configurations for LC subset.....	87
Figure 21. The width distribution of annotated bounding boxes across videos.....	90

ABBREVIATIONS

ASL	American Sign Language
ASR	Automatic Speech Recognition
BERT	Bidirectional Encoder Representations from Transformer
BSL	British Sign Language
CNN	Convolutional Neural Network
Conv1D	1-dimensional Convolutional Neural Network
CRF	Conditional Random Field
CSL	Chinese Sign Language
CSLR	Continuous Sign Language Recognition
CV	Computer Vision
DGS	German Sign Language (Deutsche Gebärdensprache)
E-TSL	Educational Turkish Sign Language
GCN	Graph Convolutional Networks
GELU	Gaussian Error Linear Unit
GPT	Generative Pretrained Transformer
HMM	Hidden Markov Models
HOG	Histogram of Oriented Gradients
I3D	Inflated 3D Convolutional Networks
IDT	Improved Dense Trajectory
LLAMA	Large Language Model Meta AI
LLM	Large Language Model
MBART	Multilingual Bidirectional and Auto-regressive Transformer
MLP	Multi-layer Perceptron
MT	Machine Translation

NLP	Natural Language Processing
NMM	Non-manual Marker
NMT	Neural Machine Translation
OF	Optical Flow
RNN	Recurrent Neural Network
SLL	Sign Language Linguistics
SLP	Sign Language Production
SLR	Sign Language Recognition
SLT	Sign Language Translation
ST-GCN	Spatio-Temporal Graph Convolutional Networks
STMC	Spatio-temporal Multi-cue
T5	Text-to-Text Transfer Transformers
TID	Turkish Sign Language (Türk İşaret Dili)
UL2	Unifying Language Learning Paradigms
VK-G2T	Vision and Context Knowledge enhanced Gloss2Text
VLLM	Vision Large Language Model

CHAPTER 1

INTRODUCTION

Sign Language Translation (SLT) is a multi-disciplinary field, spanning various research domains, including Computer Vision (CV), Natural Language Processing (NLP), and Sign Language Linguistics (SLL) (Bragg et al., 2019). The core objective of SLT is to translate sign language input into spoken or written text, converting linguistic information from visual modality to textual as shown in Figure 1. This task has been studied for over two decades to develop systems for the deaf and hard-of-hearing communities, aiming to overcome the communication barrier between sign language users and those who rely on spoken languages.

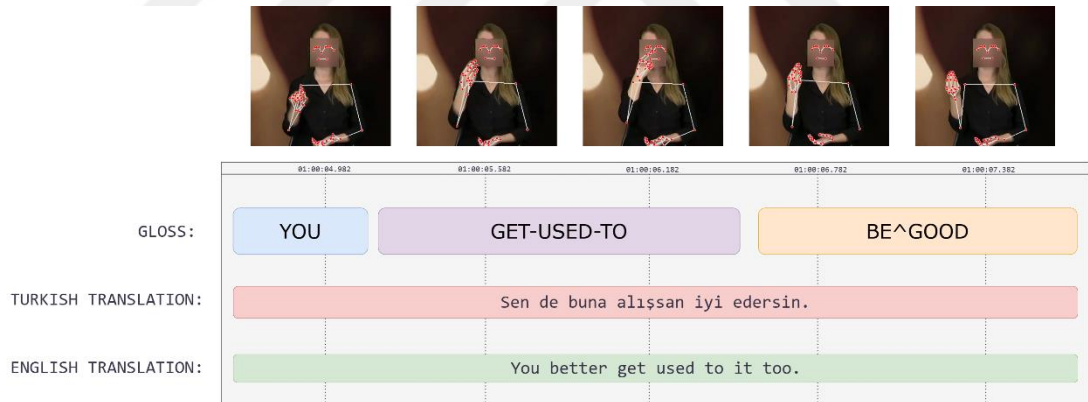


Figure 1. Overview of the SLT task

However, this is not an easy challenge since we need to transform a continuous visual language into another spoken discrete language with limited to no intermediate information, as shown in Figure 1. One of the challenges stems from the complexity of the sign language modality. Unlike spoken languages, sign languages are multi-cue, meaning there are simultaneous articulators, such as hand gestures,

facial expressions, body movements, and other non-manual markers to convey different linguistic information (Brentari, 1998, 2010; Stokoe, 1980; Wilbur, 2013). Developing systems for understanding this complex information set requires a large-scale dataset where the models learn various information, which is shown to be an effective approach in the recent advancements in artificial intelligence systems such as ChatGPT (OpenAI et al., 2024), and LLaMa (Touvron et al., 2023). However, these models are trained on billions of sentences to learn representations for language (Naveed et al., 2023), while it is not the same case for sign languages, where the data is scarce. Because of both the complexity and the low-resource nature of sign languages, they often result in challenges to the development of real-time usable sign language translation systems. Nevertheless, SLT research has improved significantly over the past two decades. From earlier approaches, models relied on heuristic methods and manual labor-intensive annotations to represent sign language information. These systems often lacked robustness and generalizability regarding their data and methodology. With advancements in machine learning, current methodologies employ pre-trained transformer-based architectures (Gueuwou et al., 2023; Rust et al., 2024; Tanzer & Zhang, 2024; Uthus et al., 2023; Zhao et al., 2023a) that leverage deep learning models in textual domains to adapt to sign language modality. The research has demonstrated that these models learn representations of sign languages about their spatial and temporal information and semantic information in alignment with their spoken language counterparts.

Despite these advancements, SLT still has several persistent challenges. As we have discussed, one major limitation is the scarcity of large, diverse datasets necessary for training machine learning models. However, this challenge has been tackled, and exponential improvements have been obtained within the last 3-4 years

with the development of open-access data collection from YouTube or by archives of TV channels (Albanie et al., 2020, 2021; Gueuwou et al., 2023; Uthus et al., 2023). Yet sign languages vary widely across different regions and communities, and collecting comprehensive datasets that capture this diversity is resource-intensive, and researchers working on less-studied sign languages may lack access to large-scale resources similar to the aforementioned studies.

Moreover, accurately representing the complex features of sign languages—such as simultaneous gestures and subtle non-manual cues—still remains a challenging task. This complexity requires sophisticated representation techniques and architectures that can learn representations from continuous visual information in sign language data.

Addressing these challenges is imperative not only for technological progress but also for promoting inclusivity and accessibility systems in general. Improved SLT systems have the potential to bridge communication gaps, and empower the deaf and hard-of-hearing communities; however, there are almost no technologies provided with real-world applications. By advancing research at the intersection of CV, NLP, and SLL, we can develop more robust translation models that meet the practical needs of their users. In this section, we will delve into applications regarding sign language translation and the challenges posed by current literature.

1.1 Task definition

The primary objective of SLT is to enable the automatic translation of spoken language text from continuous sign language input. This process involves converting sign language data, such as American Sign Language (ASL) or Turkish Sign Language (TID), presented in modalities like video recordings, gloss annotations, or

linguistic notations into the written representations of their respective spoken languages, such as English or Turkish.

Unlike word-level translation, namely Sign Language Recognition (SLR), which learns a mapping between individual glosses (word-level units in sign language) from given sign language input, continuous sign language translation captures the full grammatical structure of both signed phrases and their spoken language counterparts.

Both end-to-end recognition and translation systems rely on encoding representations into latent representations derived from visual input, such as pretrained image or video encoder representations or latent features from video sequences. Alternatively, they may use intermediate representations, such as glosses or notation systems, to encode continuous sign language into discrete sequences (Liang et al., 2023; Rastgoo et al., 2021).

The difference between SLR and SLT systems is that the SLR system does not require learning information about spoken language. This comes from the nature of the SLR task, as the model learns a many-to-one mapping between continuous sequences of sign language and a single gloss. In the case of SLT, models learn a many-to-many mapping between continuous sequences of sign language and a discrete sequence of written text.

This latter process can be framed as a sequence-to-sequence task, a term that can be applied to neural machine translation (NMT) but requires different formulations due to the multi-modal nature of SLT systems (Liang et al., 2023). Unlike traditional Machine Translation (MT), SLT involves learning the source language from visual information but also with temporal changes within the information, making the task closer to video captioning and video understanding

tasks if the problem is formulated as an end-to-end translation problem. However, due to the end goal of generating coherent text(s), SLT is also closely aligned with natural language generation tasks. These problems are defined in Section 1.2.2.

1.2 Task application

The transformation from visual to textual modality requires specialized pipelines and architectures that can encode linguistic information of a sign language from the visual input and decode linguistic information of textual output if the task is formulated as a translation task where the alignment between a sign language and a spoken language text is at issue. To explain how this problem has been formulated in the literature, we start defining our problem from SLR systems, which focus on encoding information about sign language, while SLT systems simultaneously decode textual information.

1.2.1 Sign language recognition

SLR has emerged as the earlier and foundational component where the sign language translation systems benefit greatly (Rastgoo et al., 2021). This task focuses on classifying lexical items, or single discrete units of meaning, such as gloss(es), within sign language videos in isolated forms, in most cases.

Datasets supporting SLR typically pair video input as well as other notation forms in earlier forms (Bragg et al., 2019), with corresponding labels or glosses, such as the sign "come" which is the uninflected form of the verb "to come" or other lexical items representing specific concepts in the sign language. In some cases, these datasets may also include static images, such as representations of the sign language alphabet or other static signs (Dreuw et al., 2006; Zimmermann & Brox,

2017). These datasets have limited representativeness of the syntactic and semantic structures or any alternation in the sentence if the collection has isolated signs. Yet the literature on SLR has provided benchmarks on various methodologies and insights on training models to learn spatiotemporal representations for sign language understanding, as we will discuss the importance of the task in detail in Section 2.2

Beyond the detection of isolated lexical items, the usage of SLR systems is further extended in the alignment of sign language sequences with spoken language sequences on the creation of automatic datasets (Albanie et al., 2020, 2021), making automatic gloss annotation an invaluable tool for SLT datasets. Similarly, the usage of recognition has been extended to translation applications. In Continuous Sign Language Recognition (CSLR) systems, the translation is done by predicting the possible gloss sequences in a given sign language video for extracting an intermediate representation from a given video. However, these methodologies are dependent on prior gloss-prediction and are bound in the methodologies (Camgoz et al., 2018; Jing et al., 2024; Orbay & Akarun, 2020). These techniques will be explained in more detail in Section 2.1.2

1.2.2 Sign language translation

SLT involves converting structured linguistic information in sign language into corresponding spoken or written language forms. This task requires both the spatial and temporal dynamics of sign language, including body movements and facial expressions and aligning them with the syntactic and semantic structures of the target spoken language.

The task of sign language translation is generally considered a sign-to-text task, whereby sign language input is translated into spoken or written language

output. Text-to-Sign translation, also known as Sign Language Production (SLP), involves generating signed language output based on spoken language input. However, we are solely focused on the sign-to-text applications in this work.

The recent literature on SLT models has a greater emphasis on end-to-end approaches for direct translation from sign language video input to spoken language text without intermediate information. Intermediary tasks, such as CLSR, were more prevalent in the earlier literature (see Liang et al. (2023)).

In addition, recent advancements in large language models (LLMs) have significantly influenced and utilized SLT research. Models such as BERT (Devlin et al., 2019), GPT (Radford, 2018), and LLAMA, known for their robust textual language representations, have been integrated into SLT systems to enhance translation quality. These models have also been influenced by the training methodology, where the SLT system has started to adapt self-supervised training on SLs (Gueuwou et al., 2024; Rust et al., 2024; Wong et al., 2024).

1.2.2.1 Gloss-to-text translation

In this approach, models translate sequences of glosses—written representations of sign language signs—into spoken language text. Here, the gloss provides a structured, symbolic representation of sign language, which acts as an intermediary between visual input and textual output. By transforming the continuous video signal into a sequence of discrete units, the translation task becomes a text-to-text problem. However, the task is limited to the discrete representations with limited encoding of various information, often omitting non-manual markers like eyebrow-raising, mouth gestures, and mouthing cues, which are utilized for the construction of grammatical features such as questions or negations as well as the lexical distinction in continuous

context (Moryossef et al., 2021). To improve the symbolic representation quality, various intermediate representations diverge from gloss-level units, such as HamNoSys (Prillwitz et al., 2008a), and SignWriting (Sutton, 2000). In Section 1.2.3, we will delve into the specific applications and design choices for these intermediate representations.

1.2.2.2 Sign-to-text translation

This end-to-end approach translates raw sign language signals directly into spoken language text without relying on intermediate representations, such as glosses. In this approach, the sign language information is extracted from the continuous visual input, from raw pixel information learned within the architecture, or from any latent representation extracted from visual input (Liang et al., 2023; Rastgoo et al., 2021; Varol et al., 2017).

Moreover, these architectures not only require encoding sign language input but also decoding coherent grammatical textual output. This requires a text decoder with an understanding of the contextual representation of the spoken language. This can be achieved either via alignment-based architectures trained from scratch or the usage of a pre-trained text decoder, which is more common in recent literature with the emergence of large language models (Devlin et al., 2019; Radford, 2018; Touvron et al., 2023).

Indeed, recent literature indicates that such approaches often leverage pre-trained LLMs to enhance translation quality (Gong et al., 2024; Li et al., 2020; Uthus et al., 2023). By utilizing the learned linguistic information embedded in spoken language, these models provide the advantage of initialization of textual representation with more advantage in these models.

1.3 Task significance

By advancing real-time translation between signed and spoken languages, SLT aims to achieve effective communication, empowering Deaf and Hard-of-Hearing individuals to participate fully in the society as well as promoting the cultural identity of this community. The work on this task also drives broader inclusion and representation of signed languages within artificial intelligence research and the overall scientific community in both studies on language and the inclusion of deaf and hard-of-hearing members in the research as language experts.

Historically, signed languages have been underrepresented in the NLP literature, leaving a gap in technological advancements for this linguistic modality. SLT contributes to rectifying this disparity by developing models and datasets that capture the linguistic richness of sign languages and improving the representation and involvement of the deaf community in this area.

1.3.1 Importance to deaf and hard-of-hearing community

The SLT task has been studied for the development of technologies on accessibility and inclusivity for the Deaf and Hard-of-Hearing communities, who often struggle in daily communication within a spoken language-dominant world. For over 1.5 billion individuals globally with hearing impairments (nearly 20% of the population), communication challenges lead to limited access to essential services, social interactions, and educational opportunities. Among these, approximately 430 million people experience disabling hearing loss, where traditional spoken communication may be insufficient or inaccessible, according to the World Federation of the Deaf (2024). In Turkey, there were 836,000 registered Deaf and Hard-of-Hearing individuals in 2023, representing 1.1% of the population (The Ministry of Family

and Social Services, 2023). While not all of these individuals use TID as their primary mode of communication, TID is the primary language for a majority of this community.

1.3.2 Importance to multi-modal language modeling

The continuous SLT task is also critical for advancing multi-modal language modeling, an area that examines diverse sources of natural language input beyond text, such as audio, images, and videos. Sign language inherently aligns visual information with linguistic information, offering a unique test case for multi-modal applications. Developing robust SLT systems further amplifies to multi-modal NLP advancements by deepening our understanding of how to handle non-textual input, specifically human action input (Varol et al., 2017). As SLT models evolve, they not only serve the Deaf and Hard-of-Hearing communities but also contribute to multi-modal advancements that benefit fields like video understanding, augmented reality, and cross-linguistic translation.

1.4 Task limitations

Various challenges stem from the complexity of the input modality of sign languages when compared to the input modality of spoken languages, where spoken language-based NLP research has been a long-studied area with large amounts of resources and benchmarking systems. In contrast, computational research on sign language is not a widely studied area with limited dataset access, where we see at most 50-hour-long datasets until the late 2010's (Aung et al., 2020; Rastgoo et al., 2021).

When we look at the data size required to train textual models and sign language models, we see that recent pretrained text models are trained on billions to

trillions of sentences using unsupervised learning approaches, followed by further training on various generative tasks such as text classification, summarization, and translation. For sign language, by comparison, we are mostly limited to less than a million instances per sign language. Yet, this limitation of data does not stem solely from its size, but as explained in the following sections, SLT literature suffers from various other limitations related to sign language datasets as well as the technologies for benchmarking on these datasets.

1.4.1 Limitations of sign language datasets

As we have discussed, the datasets are not only about their respective size when compared to the data required for training textual models for generalization. These datasets often discard the natural language variety used within the dataset.

This is particularly evident in earlier datasets, such as the RWTH Phoenix Weather dataset, where the vocabulary is limited to weather-related terms. Similarly, the Korean Educational Videos dataset only includes vocabulary from course lectures, discarding practical language usage. More recent datasets, like the JWSign dataset, limit their scope to translations of the Bible in different languages, restricting vocabulary to a single domain and category.

This restriction has been addressed in the creation of later datasets, where the domain of signing is considered in the curation process. For instance, datasets like CSL-Daily, How2Sign, and YoutubeASL have included open-domain categories in their data selection. In this current study, we have ensured careful consideration of the language used in the final development of our dataset. However, this is not the only issue found in dataset curation, as the collection environment of the dataset also

restricts both the language production within the dataset and the maximum amount of data that can be collected.

Up until the most recent literature, most of the data collected for SLT came from controlled lab environments, where the signer was asked to sign specific words or sentences. Only a few datasets record conversations between signers with multiple camera angles (Duarte et al., 2021; Hanke et al., 2020). Although this approach provides clean data resources and involves the deaf community actively, these datasets are often limited in size due to the labor-intensive nature of their collection processes. This limitation is partially alleviated in more recent works, where open-source videos from YouTube or television channels have been used for dataset creation.

The limitation is still a challenge for Turkish Sign Language (TID). For TID, the first continuous sign language translation dataset was introduced in late 2024, namely the E-TSL dataset, which consists of 24 hours of sign language videos. However, the dataset is limited to the domain of language use, being restricted to the educational domain, similar to datasets like KETI or RWTH. To solve this issue, we propose the first large-scale open-domain dataset for TID in this work.

1.4.2 Limitations of benchmarking

The literature on SLT offers a range of architectures proposed with unique solutions to the task, addressing challenges from input representation to translation architectures. These methodologies primarily focus on handling the complexity of input representations and aligning spoken language with sign language understanding through different modalities. Approaches vary from image- or video-based encoders (Albanie et al., 2021; Camgoz et al., 2018; Koller et al., 2016) to text-aligned

contrastive training strategies for contextual representation of pose information (Gong et al., 2024; Wong et al., 2024).

While this variation in methodologies enriches the field, it also creates challenges in selecting optimal benchmarks for developing baseline methodologies and comparing results across new datasets. The lack of standardized benchmarking methods complicates the evaluation and comparison of SLT systems, particularly as datasets and input modalities become more diverse.

Some efforts, such as those implemented in the JoeyNMT framework, have contributed to SLT benchmarking, particularly for small-scale datasets (Kreutzer, Bastings, & Riezler, 2019). More recently, the emergence of large-scale datasets, such as the YoutubeASL dataset (Uthus et al., 2023), has introduced significant improvements in baseline methodologies. For instance, the adoption of pretrained T5 (Raffel et al., 2020) models for SLT with a single projection layer have established a robust baseline for evaluating new representation models in YoutubeASL, and it serves as a benchmark for new datasets. However, this method provides only a singular approach and lacks a systematic comparison of key components, such as optimal encoder layers, model size, or pretraining language.

This work conducts comparative experiments on each component to address this gap, including input representation strategies, model configurations, and pretraining approaches. By doing so, we provide a comprehensive benchmarking framework and establish baseline scores for our proposed TID dataset, ensuring a robust foundation for future research and dataset evaluation.

1.5 Contributions

The first contribution of this body of work is the introduction of the largest dataset ever proposed for SLT in TID after the E-TSL dataset (Öztürk & Keleş, 2024), which consists of 24 hours of lecture footage focused on 5th to 8th-grade science education, whereas this newly proposed dataset expands coverage to over 500 hours of video footage of daily language from open-source content on YouTube. This increase in size and diversity in TID data enables various research on not only TID-based applications but any SLT research in the future.

The thesis also provides extensive methodological work on Sign Language Translation (SLT) using pre-trained large language models (LLMs) in terms of dataset availability, model evaluation, and linguistically informed methodologies. This analysis investigates the impact of key components such as feature encoders, pretrained model size, and language resources of the pretrained models. The comparison covers architectures that process various input layers from linear to convolutional ones.

Additionally, this work evaluates pretrained models of different sizes and multilingual capabilities, ranging from smaller models like mT5 (Raffel et al., 2020; Xue et al., 2021) to larger architectures like TURNA (Uludoğan et al., 2024), which is specifically trained on Turkish data. The training and evaluation protocols are modified to be specific to Turkish and TID, with postprocesses introduced with their theoretical justification.

Beyond quantitative results, this thesis also undertakes qualitative evaluation grounded in sign language linguistics, with a focus on the grammatical properties of the Turkish language and TID. By this approach, we are able to offer a critical

analysis of current evaluation protocols in SLT research and underline their limitations.

1.6 Thesis overview

In Chapter 2, we will explain related works, focusing on datasets, SLR, and SLT.

This section provides an extensive comparison between datasets ranging from small, domain-restricted corpora to open-domain, large-scale resources, as well as a methodological comparison of the developed translation systems.

In Chapter 3, the proposed large-scale dataset, namely BUTID, is introduced. This chapter provides detailed statistics, describes the collection process using YouTube content, and outlines preprocessing and postprocessing steps on both pose estimation and translation texts.

In Chapter 4, the thesis presents the methodology, including pose normalization and transformation techniques. It explains the adaptation of pre-trained LLMs for SLT, explores feature encoder strategies, and describes data augmentation methods, such as paraphrasing and pose-based augmentations.

In Chapter 5, the experimental setup is detailed, covering the model, dataset, and training configurations. It compares feature extractors (Linear, MLP, Conv1D), evaluates pre-trained LLMs (mT5, TURNA), specifies hyperparameters in these experiments, and finishes with evaluation protocols.

In Chapter 6, experimental results are analyzed quantitatively and qualitatively. The chapter evaluates model performance using BLEU (Papineni et al., 2002) scores and examines linguistic accuracy and contextual appropriateness in the translations.

In Chapter 7, we discuss challenges in SLT, including input modality, contextual understanding, and linguistic representation. We evaluate the impact of paraphrasing and identify areas with cross-linguistic comparisons between Turkish and TID, providing insights into the development of future models and datasets.

In Chapter 8, future work is proposed, including various improvements ranging from increasing dataset quality to architectural enhancements for the ultimate development of a high-quality, multi-disciplinary research domain on TID combining efforts of both computer science and linguistic research.

In Chapter 9, the thesis concludes by summarizing its contributions, including the introduction of the first large-scale open-source dataset for TID and a detailed evaluation of SLT models developed using this dataset, concluding with the possible directions of future research following this body of work.

CHAPTER 2

RELATED WORKS

Developing Sign Language Translation (SLT) datasets is crucial for creating models that translate sign languages into spoken languages. These datasets are necessary for training and testing models to recognize and translate the unique features of sign languages, making them a key part of any SLR and SLT system.

However, as outlined in Chapter 1.4, the development of sign language recognition and translation systems presents significant challenges. One of the primary obstacles lies in the low-resource nature of sign languages, which makes it difficult to curate large, diverse, and representative datasets necessary for representing sign languages and training robust models (Bragg et al., 2019).

In this section, we will extensively examine the production of these systems, beginning with the challenges and methodologies involved in dataset curation. We will discuss how the scarcity of labeled data, variations in signing styles, and limited access to high-quality annotated resources have been overcome in the literature.

Following this, we will delve into the design of computational models that leverage these curated datasets to effectively process and translate sign language, addressing both visual and linguistic complexities.

2.1 Datasets

Starting as early as the late 1900s, a large body of datasets have been introduced for various sign languages, such as American Sign Language (ASL), British Sign Language (BSL), German Sign Language (DGS), and Chinese Sign Language (CSL). These resources are used in training and evaluating SLR/SLT systems.

Early sign language recognition datasets were primarily developed to classify individual lexical items or specific signs within images or video sequences, namely for SLR systems. These datasets were typically collected in controlled environments, such as laboratories or indoor settings, where open-source data acquisition was not available at the time. Recording methods included the use of RGB cameras (Camgoz et al., 2016; von Agris & Kraiss, 2010), Kinect cameras (Zhang, 2012), and, in earlier applications, motion sensors or sensor-equipped gloves attached to the signers for data collection (Starner et al., 1998). While sensor gloves provided detailed motion data, they often restricted the natural movement of signers, potentially affecting the authenticity of sign production (Bragg et al., 2019; Rastgoo et al., 2021).

2.1.1 Recognition datasets

One of the first works on SLR datasets dates back to the 1990s when datasets were collected in a controlled laboratory environment using a combination of webcams and wearable sensors (Fang & Gao, 2002; Starner et al., 1998). As camera technology advanced, datasets moved beyond sensor-based systems to improved video capture methods with high-resolution video cameras, enabling the collection of quality datasets.

For German Sign Language, datasets such as the DGS Corpus (Prillwitz et al., 2008b), which comprises 560 hours of recorded material with 657,000 annotated tokens collected from 330 signers across Germany, are the first large-scale datasets seen in the SLR literature.

In the case of American Sign Language (ASL), datasets like ASLLVD (Athitsos et al., 2008) include over 3,300 ASL signs, each produced by 1-6 native

signers, totaling nearly 9,800 tokens. Later on, with their increasing scale, MS-ASL (Joze & Koller, 2018) offered more than 25,000 annotated videos, covering 1,000 signs performed by over 200 signers. WLASL (Li et al., 2020) provided 68,129 videos of 20,863 ASL glosses. Both of these datasets are the earliest examples of large-scale open-source datasets in SLR literature.

In Turkish Sign Language (TID), Kinect sensors have been utilized to capture depth information and body joint landmarks, resulting in datasets such as BosphorusSign and BosphorusSign22k (Camgoz et al., 2016; Özdemir et al., 2020). These datasets include 735 sign and phrase samples from the health, finance, and everyday life domains, collected using the Microsoft Kinect v2 depth sensor. The AUTSL dataset (Sincan & Keles, 2020) contains 226 signs performed by 43 different signers, totaling 38,336 isolated sign video samples. These samples were recorded in RGB and in depth via improved cameras, providing the first datasets for TID on the SLR domain.

Similarly, BSLDict (Fenlon, 2010) offers a comprehensive collection of signing data for British Sign Language (BSL). Even multilingual datasets have emerged, such as Dicta-Sign (Matthes et al., 2012) in the early literature on SLR datasets. Each of these datasets focuses on mostly isolated signs, but some include continuous signing, providing earlier resources for SLR development.

Each dataset proposed demonstrates the importance of increasing the quality of the data collection technologies, starting from sensory gloves to Kinect cameras, yielding comprehensive datasets for SLT methods reflecting on the quality of translation datasets.

2.1.2 Translation datasets

One of the foundational datasets in Sign Language Translation (SLT) is the RWTH-Phoenix Weather Dataset for German Sign Language (DGS) (Camgoz et al., 2018; Forster et al., 2012). Initially developed from low-resolution video footage of German weather forecasts, this dataset established an early benchmark for SLT research. The original version included only 3.25 hours of video footage with 1,980 sentence pairs, later expanding to 11 hours of footage and approximately 9,000 sign-sentence pairs. Another DGS dataset is the SIGNUM dataset (von Agris & Kraiss, 2010), which increased the data size to 55 hours of footage and 33,000 translation pairs.

These datasets also include gloss annotations for sentence-level representations. These annotations are done for the different task formulations, such as gloss-to-text-based SLT or Continuous Sign Language Recognition (CSLR). Another early contribution to SLT datasets is the BSL Corpus (Schembri et al., 2013), a resource focused on continuous British Sign Language (BSL) production. This corpus diverges from the BSL SignBank (Fenlon, 2010), which is a lexical resource that includes isolated sign annotations. The BSL Corpus focuses on sentence-level annotations for continuous signing, making it one of the earliest resources for gloss-to-text SLT application and for supporting phonological and linguistic research. Both the RWTH-Phoenix Weather Dataset and the BSL Corpus employed extensive manual annotations, which are both resource- and labor-intensive.

As the field progressed, newer datasets began to emerge in terms of both size and scope. For example, SLT datasets for Chinese Sign Language (CSL) introduced greater variability in content and domains. The CSL-Daily dataset (Zhou et al., 2021b) contains 21,000 sequences, which equates to 23 hours of footage, while the

CSL100 dataset (Han et al., 2021) features 25,000 sentences with a total duration of 100 hours.

In ASL, How2Sign (Duarte et al., 2021) has 79 hours of footage and approximately 40,000 translation pairs with extensive data collection that was obtained with multiple cameras and panoptic studio data included. Another example is the OpenASL dataset (Shi et al., 2022) provides 288 hours of ASL videos across multiple domains from over 200 signers.

Most early sign language datasets were collected in controlled environments where signers were instructed to produce specific signs or sentences. Similarly, most of them are instruction-based production, meaning that the signing does not occur naturally but is selected by the curators, which limits the lengths of collecting data, where the largest collected dataset in the lab environment is limited to 100 hours.

The recent introduction of transformer-based architectures has significantly influenced the collection and utilization of sign language data, forcing SLT systems to develop large-scale data training with the curation of this data. Pretrained large language models, such as BERT (Devlin et al., 2019) GPT (Radford, 2018), LLaMa (Touvron et al., 2023) have demonstrated the ability to perform complex tasks when trained on extensive datasets. This trend has motivated the development of newer datasets with substantially increased sizes when compared to earlier resources.

One of the first datasets produced on this scale for BSL, the BSL-1K dataset (Albanie et al., 2020) represents a major leap forward, containing 1 million sentence pairs and covering 1,060 hours of footage. The BOBSL (Albanie et al., 2021) dataset further extended over the BSL-1K dataset, featuring 1.2 million sentence pairs and 1,467 hours of video content sourced from BBC broadcasts. In this dataset, automatic sentence alignment was employed to map spoken language content to sign language

videos, which significantly reduced manual annotation efforts while maintaining large-scale coverage (Bull et al., 2021).

For ASL, the YouTubeASL dataset (Uthus et al., 2023) stands out as a notable resource. This open-source dataset initially provided 900 hours of footage translated into English. Recent expansions of the dataset, namely Youtube-SL-25 (Tanzer & Zhang, 2024) has increased the dataset size to 2,800 hours, covering more than 25 sign languages. The other similar large-scale datasets, including multilingual SLT, are the AfriSign dataset (Gueuwou et al., 2023) which has 160 hours of footage of different sign languages in Africa, and JWSign (Gueuwou et al., 2023) which has 2530 hours of Bible translations of 98 sign languages to demonstrate further the importance of the scale of pretraining data in SLT.

In the context of TID, the development of SLT datasets has been more limited compared to other languages such as ASL, BSL, or DGS. The Educational Turkish Sign Language (E-TSL) dataset (Öztürk & Keles, 2024) presents the first continuous dataset for TID. Although not matching the scale of larger ASL datasets, E-TSL comprises approximately 24 hours of TID footage, totaling 1,418 video clips featuring 11 different signers. This dataset focuses on educational content, specifically Turkish language lessons for 5th, 6th, and 8th grades. Additionally, the YouTube-SL-25 dataset includes 18 hours of TID videos, contributing to the available resources for Turkish Sign Language. While it is not employed for SLT purposes, we also have a corpus that can be utilized for the development of SLT systems for TID, specifically the TID Dictionary (Makaroğlu & Dikyuva, 2017), which contains approximately 11,000 sequences accompanied by gloss annotations and translations. While these datasets are smaller in scale compared to extensive

ASL datasets, they represent significant steps toward advancing SLT research for TID.

As shown in Table 1, SLT datasets have shown an exponential increase in the recent literature. The figure also includes ranking our proposed dataset in terms of the number of hours.



Table 1. Overview of Sign Language Translation Datasets

Dataset Name	Sign Language	Spoken Language	# Instances	# Hours	# Signers	# Words	Vocab.
Youtube-SL-25	25 SL	Multi-lingual	2.16M	3207	>3000	N/A	N/A
JWSign	98 SL	Multi-lingual	N/A	2530	>1500	N/A	729K
BOBSL	BSL	English	1.2M	1467	39	11.4M	78K
BSL-1K	BSL	English	1M	1060	40	9M	59K
YoutubeASL	ASL	English	610K	984	>2519	N/A	N/A
BUTID	TID	Turkish	234K	529	27	3.3M	125K
OpenASL	ASL	English	288	280	200	N/A	33K
AfriSign	6 SL	English	N/A	152	160	N/A	20K
BSL Corpus	BSL	English	N/A	125	N/A	N/A	N/A
CSL100	CSL	Chinese	25K	100	10	N/A	2K
How2Sign	ASL	English	35K	79	11	598K	16K
SIGNUM	DGS	German	33K	55	25	N/A	450
KETI	KRSL	Korean	N/A	28	14	N/A	N/A
E-TSL	TID	Turkish	N/A	24	N/A	N/A	6K
CSL-Daily	CSL	Chinese	21K	23	10	10K	2K
ISL	ISL	English	9K	18	5	44K	N/A
Phoenix2014T	DGS	German	8257	10	9	99K	2K

2.2 Sign language recognition

The field of Sign Language Recognition (SLR) has seen significant evolution over the past two decades, beginning with its early developments in the late 1990s (Bauer et al., 1999; Starner et al., 1998).

In the initial stages, SLR systems were constrained by limited computational and technological resources, leading researchers to rely heavily on handcrafted features and intermediate representations such as HamNoSys (Fenlon, 2010) and SignWriting (Sutton, 2000). These notational systems provided discrete sequences of representations for the recognition of isolated signs or phrases. However, the lack of detailed motion and spatial information limited the effectiveness of these approaches for SLR technologies and their real-life applications.

With advancements in computer vision and camera technology, the SLR domain experienced a shift towards utilizing RGB and Kinect cameras. These technologies enabled the capture of fine-grained motion data, including depth maps, which provided richer spatial and temporal features for sign language input. During this phase, computer vision techniques such as Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005), Improved Dense Trajectory (IDT) (Wang & Schmid, 2013), and Object Flow (OF) (Laptev et al., 2008) are utilized in the feature representation of sign language input. Pre-trained architectures such as ImageNet (Krizhevsky et al., 2012), 2D and 3D ResNet (Camgoz et al., 2018), or (I3D) network (Carreira & Zisserman, 2017) embeddings were employed in SLR systems with the advancements in image and video representation models.

These representations are further utilized by the various proposed sequential models, including Hidden Markov Models (HMMs) (Vogler & Metaxas, 1999), Conditional Random Fields (CRFs) (Yang et al., 2008). Convolutional Neural

Networks (CNNs) (LeCun, Bottou, Bengio, & Haffner, 1998) and Recurrent Neural Networks (RNNs) (Giles et al., 1994) are used with the emergence of Deep Learning-based applications in the literature (LeCun et al., 2015). More recent architectures have emerged with the adoption of attention mechanisms through encoder-decoder architectures using transformer architecture as its backbone (Bahdanau et al., 2016; Luong, 2015; Vaswani et al., 2017).

The development of pose-estimation technologies has broadly impacted SLR research. These advancements enable extracting skeletal information from video frames, providing a structured representation of a signer's body, hands, and facial movements. Pose-based recognition systems became dominant due to their ability to abstract raw video data into meaningful skeletal features with dense and interpretable information for continuous representation of human action. A graph representation of skeletal information leverages the inherent structural properties of pose estimation since the motion is restricted and interactive within the selected skeletal positions.

These models treat pose landmarks as nodes and their relationships as edges, which makes it possible to represent spatial information in frame-level connections while representing temporal dependencies in inter-frame edges for sign language. By utilizing graph convolutional networks (GCN) (Kipf & Welling, 2016) and their derivatives, such as spatio-temporal graph neural networks (ST-GCN) (Yan et al., 2018), researchers have achieved state-of-the-art performance in recognizing both isolated and continuous sign language sequences.

2.3 Sign language translation

Sign Language Translation (SLT) presents significant challenges due to the limited size and quality of available datasets and the intricate nature of sign language, which

conveys complex linguistic information in a continuous, multi-modal manner. As discussed in earlier sections, SLT combines the complexities of several research domains, including CV, NLP, automatic speech recognition (ASR), and action recognition.

Researchers began to explore this task in detail in the early 2000s, initially focusing on SLR and its extension to Continuous Sign Language Recognition (CSLR). These early studies relied on manually crafted features and template-based generation techniques, laying the groundwork for the development of more sophisticated SLT systems (Bauer et al., 1999; Fang & Gao, 2002; Huenerfauth, 2004).

Early SLT research was limited in scope and primarily utilized manually annotated features to create linguistic representations. Template-based translation systems were among the earliest approaches, constrained by the availability of small, carefully curated datasets. These systems generated translations by matching input data to predefined linguistic templates, which were effective for highly controlled environments but lacked the generalizability needed for real-world applications (Bragg et al., 2019). Over time, the field transitioned to data-driven approaches, leveraging advancements in statistical machine translation, encoder-decoder architectures, and transformer-based models, which have since become the foundation of modern SLT research.

2.3.1 Template-based translation

In the early stages of SLT research, rule-based systems were relatively limited and primarily focused on the translation of spoken languages into sign languages rather than the reverse direction.

One notable example of rule-based systems is the work of Marshall and Sáfár (2002), which developed a system for translating English into British Sign Language (BSL). This system utilized the Head-Driven Phrase Structure Grammar (HPSG) formalism to model grammar rules, allowing the translation of simple English sentences into corresponding BSL structures. Similarly, Huenerfauth (2004) introduced a rule-based framework for translating English text into ASL. The system utilizes linguistic rules and concepts to map English sentences into ASL structures. Another example can be from Van Zijl and Barker (2003) works which proposes another rule-based concept for translating English text to South African Sign Language (SASL).

2.3.2 Statistical machine translation

The advent of statistical models in machine translation marked a significant turning point for Sign Language Translation (SLT), enabling more flexible and data-driven approaches to the task. Unlike rule-based systems, which relied heavily on predefined linguistic rules and handcrafted templates, statistical models allowed for learning patterns directly from data.

One of the earliest statistical approaches in SLT was introduced by Bauer et al. (1999) proposing a statistical machine translation system by focusing on CSLR for DGS. Their system utilized handcrafted feature representations in HMM-based models to perform gloss-to-text translation, which is the earliest work on translation systems in the literature.

Similarly, Bungeroth and Ney (2004) proposed a system for translating DGS glosses into spoken German using a similar HMM-based classifier system. Another

text-to-sign system proposed by Morrissey et al. (2010) employs additional linguistic information for English-to-ISL translation for sign language production purposes.

Further advancements in machine translation for SLT included the use of visual features. For example, Dalal and Triggs (2005) combined HOG features with n-gram language modeling for SMT, while Buehler et al. (2009) applied machine learning methods for weakly aligned visual information by using a feature extractor from visual information. Yang et al. (2008) have proposed a spotting-based CLSR system with a CRF algorithm for SLT.

2.3.3 Neural machine translation (NMT)

With the progress in neural machine translation (NMT), encoder-decoder-based architectures with attention mechanisms introduced by Bahdanau et al. (2016) and Luong (2015) significantly advanced SLT methodologies. We have assembled the main methodologies of NMT-based SLT as in Figure 2.

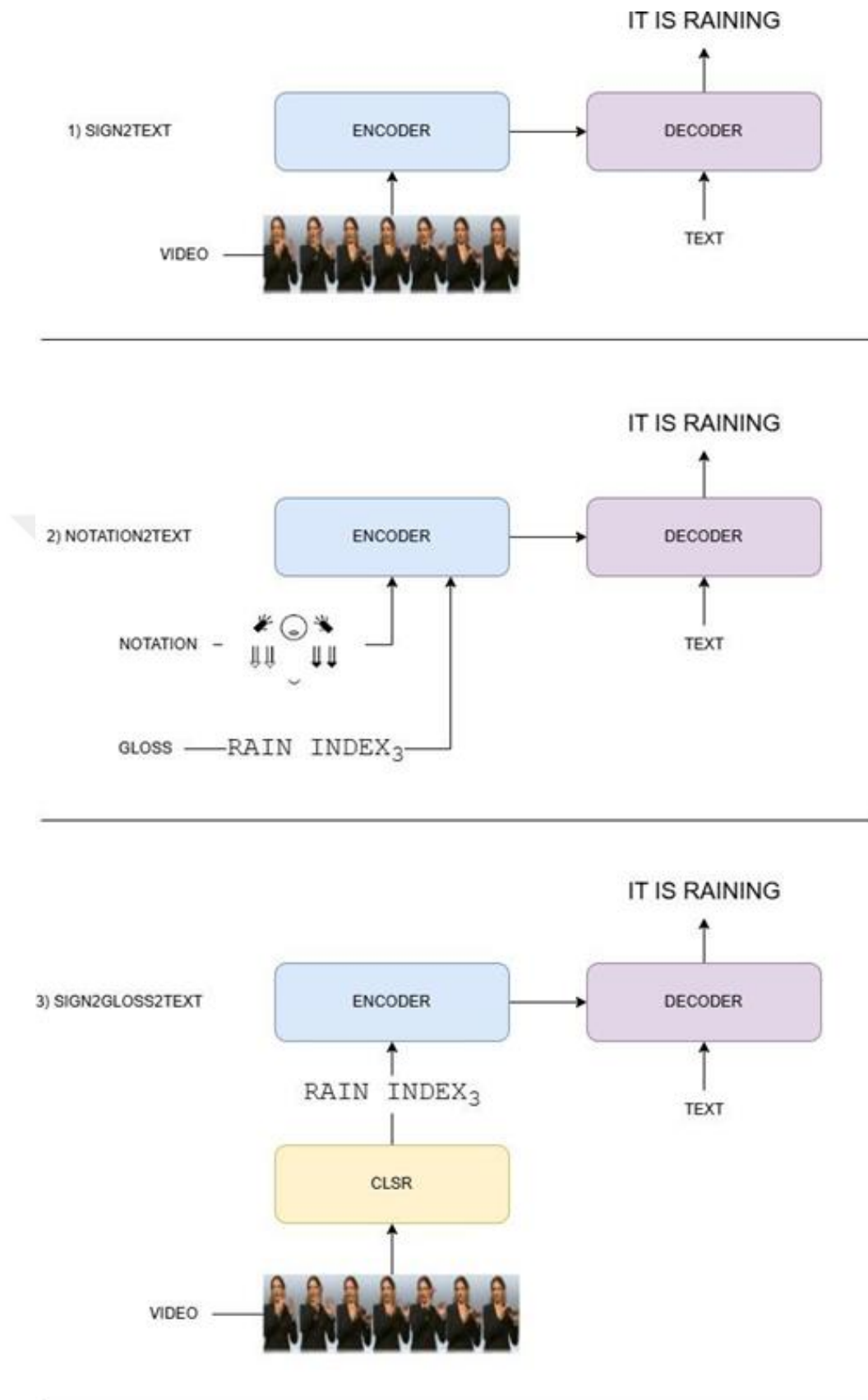


Figure 2. Common architectures proposed in the literature for SLT

2.3.3.1 Gloss-to-text

Gloss-to-Text (Gloss2Text) translation is one of the main methodologies in SLT that utilizes an intermediary step that converts sign language gloss annotations into coherent spoken language sentences.

A notable example is Ko et al. (2019), who worked on Korean Sign Language by estimating human key points to extract glosses. They then employed sequence-to-sequence (seq2seq) models for translating these glosses into spoken language, demonstrating the potential of keypoint-based gloss extraction in translation tasks. Similarly, Arvanitis et al. (2019) applied seq2seq models to the ASLG-PC12 dataset (Othman & Jemni, 2012), translating ASL glosses into spoken English.

Data scarcity has long been a critical limitation in Gloss2Text models, as gloss annotations are labor-intensive and not widely available for all sign languages. Moryossef et al. (2021) and Zhang and Duh (2021) identified this challenge and proposed solutions such as backtranslation. Inspired by Sennrich and Haddow (2016), back-translation involves translating spoken language texts into glosses to create synthetic parallel data.

The latest advancements in Gloss2Text research have focused on leveraging pre-trained LLMs, such as BERT and T5, for sign language translation. Chen et al. (2022) explored fine-tuning LLMs specifically for the Gloss2Text task. With the latest models like LLaMa (Touvron et al., 2023) or GPT (Radford, 2018), Fayyazsanavi et al. (2024) has investigated the zero-shot performance of these models as well as the latest fine-tuning techniques like Low-Rank Adapters (LoRA) (Hu et al., 2021).

2.3.3.2 Sign2(Gloss2)Text

In SLT, glosses—written representations of sign language movements—serve various roles. They are utilized as intermediate outputs in a sequential pipeline, as joint training targets to enhance shared learning between recognition and translation, or as supplementary inputs alongside video data to improve contextual accuracy.

Chen et al. (2022) introduced a method that enhances this pipeline through in-domain pretraining with synthetic data. There are two models: a visual encoder to encode human action in alignment with gloss and the translation model adopting Multilingual Bidirectional and Auto-regressive Transformers (MBART).

Zhou et al. (2021a) introduced a method that enhances this pipeline through a sign back-translation (SignBT) approach, which incorporates massive spoken language texts into SLT training. Their approach involves back-translating monolingual text to its gloss sequence using a text-to-gloss translation model.

Additional innovations included leveraging gloss information as supervision; for example, Orbay and Akarun (2020) demonstrated improved translation performance using gloss annotations while Camgoz et al. (2020b) introduced the Sign2Gloss2Text framework, jointly training SLR and SLT tasks to enhance the performance of these models.

Jing et al. (2024) recently presented Vision and Context Knowledge-Enhanced Gloss2Text (VK-G2T), emphasizing optimizing the Gloss2Text stage. VK-G2T leverages visual content from sign language videos to learn the properties of the target sentence and exploits contextual knowledge to facilitate adaptive translation of gloss words.

2.3.3.3 Sign2Text

This method is the end-to-end SLT methodology, which does not rely on any intermediate representation to align sign language with spoken language. These methodologies utilize various encoders to embed sign language information to decode textual information.

Camgoz et al. (2018) have shown NMT-based end-to-end SLT systems with the first public dataset, Phoenix2014T, using a 2D-CNN + RNN-based approach.

Subsequent advancements in end-to-end translation systems, also known as sign-to-text models, introduced various novel methodologies. In appearance-based or visual feature-based models, Koller, et al., (2016) proposed a hybrid CNN-HMM approach, while Camgoz et al., (2017) introduced SubUnets for SLT. Li et al. (2020) developed a temporal semantic pyramid for SLT tasks.

Video-level networks have also been employed to capture movement dynamics comprehensively. Guo et al. (2019) proposed Dense Temporal Convolutional Networks (DenseTCN) to capture detailed sign language movements across short-term and long-term contexts. They utilized a 3D-CNN framework for extracting visual representations and designed temporal convolutional networks to capture local context information.

With advancements in human pose estimation technologies such as OpenPose (Cao et al., 2019) and MediaPipe (Lugaresi et al., 2019), pose-based representation models have gained prominence. For instance, Zhou et al. (2021b) proposed the spatiotemporal multi-cue model (STMC) to uncover hidden information in sign language videos, while Kan et al. (2022) developed a novel approach using graph neural networks (GNNs) to transform sign language features into hierarchical spatiotemporal graphs. Building upon Camgoz et al. (2018), Kim et al. (2020)

proposed a similar end-to-end system but employed body key-point coordinates as input for their translation networks, evaluating their methods on a Korean Sign Language dataset. Recent literature on SLT with LLM-based heavily relies on the pose information in their sign language modality.

The performance of transformer architectures and pretrained models has highlighted the importance of large-scale data usage in SLT tasks. This is further supported by the emergence of the vision large language models (VLLMs) incorporating the image understanding task into the language model pretraining strategies (Liu et al., 2023; Radford et al., 2021).

To utilize these latest pre-trained architectures, Zhou et al. (2021a) augment SLT datasets by introducing monolingual text data for large-scale training. Moe et al., (2020) investigated unsupervised neural machine translation for low-resource languages like Myanmar Sign Language while De Coster et al. (2021) proposing a frozen pretrained transformer model by initializing a transformer translation model with pretrained BERT-based (Devlin et al., 2019) and mBART-50 (Chipman et al., 2022) models. Their subsequent work confirmed that performance improvements were primarily driven by the written language corpora used for pretraining rather than architectural changes. To address the low-resource problem, Zhou et al. (2023) introduced a contrastive learning model for SLT and Chen et al. (2023) proposed a progressive pretraining approach that utilized large quantities of external data to improve model performance incrementally.

As large language models (LLMs) have advanced, the importance of pretraining has become increasingly evident, especially when paired with large-scale resources such as BOBSL (Albanie et al., 2021), YASL (Tanzer & Zhang, 2024; Uthus et al., 2023), and JWSign (Gueuwou et al., 2023). Transformer-based

pretraining strategies have become standard practice for SLT tasks. Uthus et al. (2023) demonstrated the potential of using the YouTube-ASL dataset for large-scale ASL translation model training. They fine-tuned a T5 model (Raffel et al., 2020) on YouTube-ASL and subsequently fine-tuned it on smaller benchmark datasets like How2Sign (Duarte et al., 2021). Rust, et al., (2024) further improved performance on How2Sign by pretraining a video encoder on YouTube-ASL, initialized from a self-supervised video model (HieraBase) Ryali et al. (2023) trained with a masked autoencoding objective. Similarly, SignMusketeers architecture developed by Gueuwou et al. (2024) employed a multi-stream model trained with the DINOv2 model (Oquab et al., 2024) on the YouTube-ASL dataset, leveraging distinct streams for hands, face, and body inputs.

Even in isolated sign language recognition, self-supervised pretraining approaches have shown promise. SignBERT and SignBERT+ (Hu et al., 2021, 2023) treat hand poses as visual tokens and use masked prediction of hand joints as a pretraining objective. An extension of this approach, BEST (Zhao et al., 2023b), employs quantization techniques for improved performance during pretraining, further enhancing the robustness of isolated sign recognition systems.

The most recent approaches employ pretraining strategies for SLT that go beyond translation-based objectives. Zhang et al. (2024) proposed multiple training objectives, such as alignment, while Wong et al. (2024) suggested a pseudo-gloss pretraining strategy that automatically extracts pseudo-glosses from sentences to pretrain the sign encoder. Similarly, Gong et al. (2024) demonstrated the effectiveness of a vector quantization-based pretraining strategy with representation alignment pretraining with pretrained LLMs, introducing various methodologies in SLT research with the recent capabilities of large language models.

CHAPTER 3

DATASET



Figure 3. The dataset content of unique signers

In this chapter, we present a novel dataset for Turkish Sign Language (TID) translation, the largest dataset for Turkish-to-TID, and the sixth-largest dataset in overall SLT literature in terms of both duration and sentence count. This dataset includes 501 hours of sign language video, automatically aligned with corresponding Turkish text, providing a large corpus for both linguistic and computer science research.

The dataset comprises 235,251 video clips from 1700 videos on YouTube with manually added text captions alongside signed translations dedicated to accessible content for individuals with disabilities. The content is coming from a single channel, namely the "Engelsiz TRT" channel, where the sign language translations are from CODA (Children of Deaf Adults) signers, and manually captioned content is owned by the Turkish Radio and Television Corporation.

The content within videos is accompanied by sign language interpretation embedded within the video. The dataset includes 15 main categories of television series, consisting of various genres from science fiction to historical dramas, resulting in large lexical variation suitable for comprehensive SLT model training. A total of 27 unique signers are featured throughout the dataset, with diversity in terms of signing styles and individuals with age and gender variations. We provide 2 main subsets for our dataset to address different challenges stemming from the duration of sign language and translation content subsets. We obtained these subsets after prior filtering based on the lexical usage of the video domain and available annotations.

These two subsets are defined below:

- Short Context (SC): The subset includes short context sentences from video content classified as a generic category, meaning categories belonging to a TV series such as Romantic Comedy, Drama, etc.
- Long Context (LC): The subset is the long context correspondent of the SC subset. The average clip length is 3 times longer than the main category. This subset is provided for learning contextual representations from longer contexts in a sign language video.

3.1 Dataset statistics

As it can be in Table 2, the Short Context (SC) and Long Context (LC) subsets contain a comparable number of training instances; however, the Long Context (LC) subset offers a significantly greater total duration—approximately three times that of the Short Context (SC) subset.

Table 2. Dataset Composition: Unique Videos, Clips, and Duration

Subset	Num Videos	Num Clips	Duration (h)	Num Token	Num Vocab
SC	1732	116,004	143	674K	52K
LC	1729	118,498	385	2.6M	112K
Total	1732	234,502	529	3.3M	125K

We suggest that these subsets should be evaluated separately due to the grammatical structure of Turkish. We argue that evaluation metrics that are used in SLT systems, namely the BLEU metric, are not robust enough to capture the translation quality in the shorter sequences. Although we see length variation in other datasets, in the case of Turkish, an agglutinative language with word-order variation, the outcome of the metrics should be investigated separately to demonstrate the effect of each architectural component of the proposed SLT system in Chapter 4.

3.1.1 Clip length analysis

Table 3. Clip Length Analysis Across Dataset Subsets

Info	Short Context	Long Context
Mean	4.43	11.84
Std. Dev.	1.94	5.9
Min.	3	3
Median	4	10
Max	25	25

The clip-length distribution of the dataset across the four subsets is given in Table 3. On average, the clips in the SC subset have a duration of 4.43 seconds. The Long Context (LC), designed for capturing longer contexts, show significantly larger average clip durations, at 11.84 seconds, respectively.

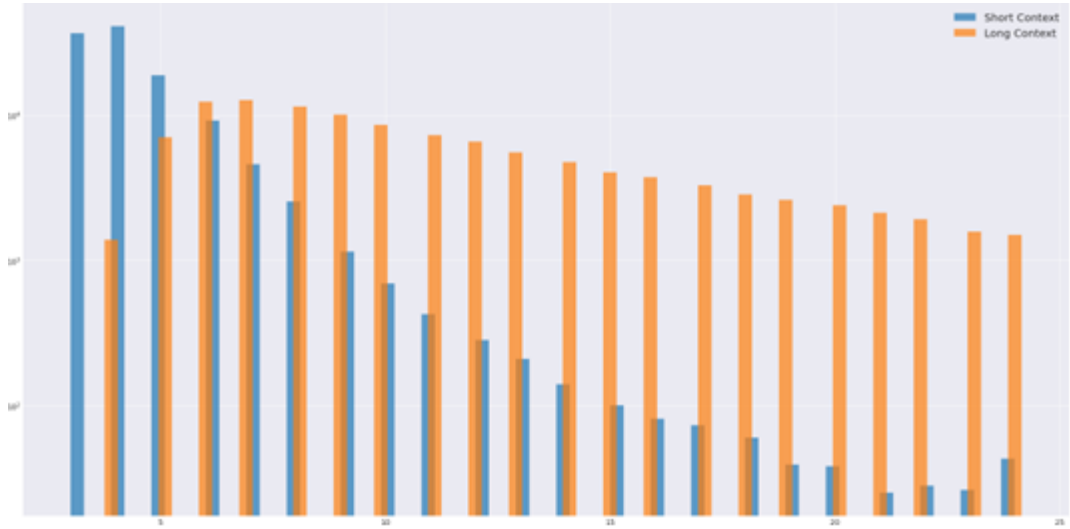


Figure 4. Distribution of clip durations across subsets

The distribution of clip durations across subsets reveals some overlap, particularly between the SC and the LC subsets. While the SC subsets share a concentration of short-duration clips ranging from 3 to 6 seconds, the LC subsets exhibit overlap in the mid-range of durations, particularly between 10 and 20 seconds.

This overlap is intentional and stems from the design of the dataset, where the SC subsets were curated to focus on short context sentences in both lengths of signed sequence as well as the lengths of translations, demonstrating that the long-term dependency is not affected by the duration of sign but by spoken language correspondent in signing content, such as dialogue-heavy scenes or explanatory narratives.

3.1.2 Category distribution

Based on the conversational content of these videos, we have manually annotated them into generic categories, as illustrated in Figure 5.

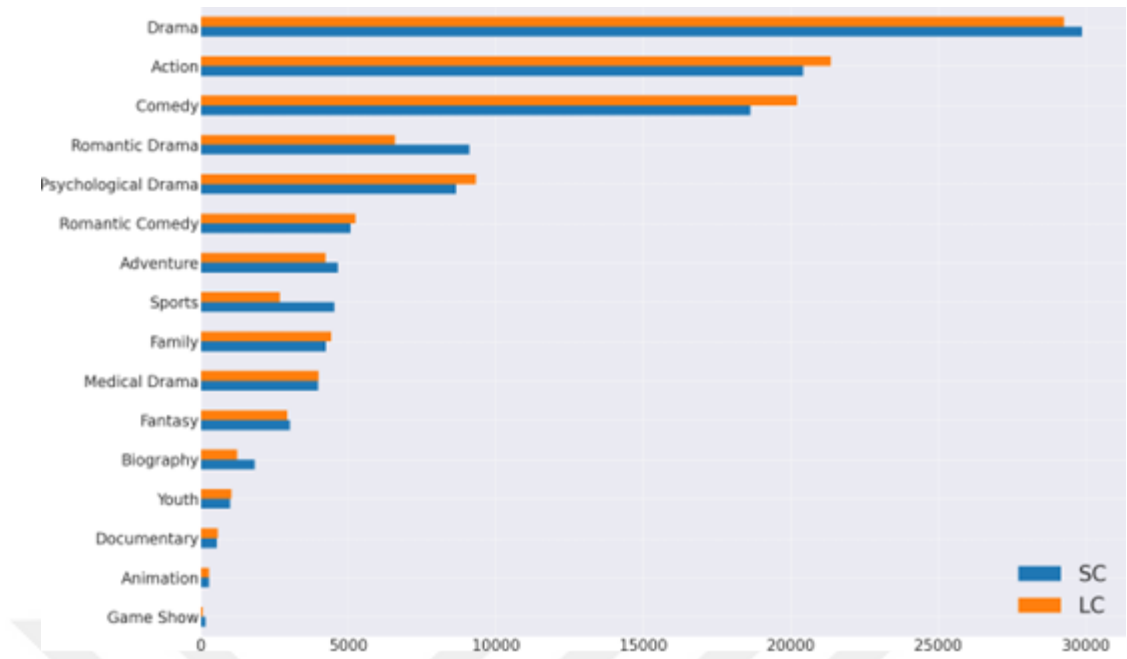


Figure 5. The category distribution across clips on both SC and LC datasets

We will be going into detail about the curation of these subsets in terms of both signed content and spoken language translations, as well as solutions for general alignment problems.

3.2 Dataset collection

The dataset was collected using the YouTube Data API, adhering strictly to YouTube’s terms of service and regulations for data access. The collection is done in compliance with all legal and ethical guidelines for the usage and distribution of public data.

3.2.1 Captions

The spoken language translations are provided by the content creators to make content accessible for Deaf and Hard-of-Hearing audiences. These translations are

aligned with the audio and formatted in standard subtitle files, such as .srt, which contain time-stamped captions corresponding to the spoken dialogue.

The absence of exact alignment between the signed content and captions is the most challenging aspect in the development of sign language translation datasets. This misalignment is due to the inherent differences in the structure and timing of signed versus spoken languages. However, given the conversational nature of the video content in the dataset, there are frequent pauses or gaps in dialogue. These gaps provide natural breaks that can be utilized in the semi-automated alignment of short- and long-signed video segments with the corresponding captions.

For the selection of captions, we utilize a series of preprocessing operations on translation text to reduce the noisy captions. These operations include the removal of descriptive captions such as "[Music]", and speaker references such as the names of the subjects in parentheses, as well as the filtering of multi-speaker captions, which are the translations of multiple speakers talking with each other. The latter operation is required to remove the turn-based translation, which requires the use of space and referential pointing in the translation process. For our data, the translator's limited space duration of the recording restricts translators from performing such translations. In addition, we applied prior filtering to the content, excluding videos that have historical language from Ottoman-based translations, as these exhibit significant lexical variation compared to Modern Standard Turkish.

3.2.2 Signed video content

The signed content is embedded within the video and appears as an overlay in the bottom-right corner. One challenge of the collection of precise visual information is that there is no predefined bounding box that consistently isolates the signed content

across the videos. To address this, we employed a semi-automated object detection approach to annotate bounding boxes. A preliminary automated detection process was applied, followed by manual approval and refinement for accuracy. For cases where the automated approach was insufficient, manual bounding box annotations were completed to provide precise localization of the signer within the video.

In the next section, we will be explaining in detail the processing operations to maximize the alignment between spoken language and signed language, as well as extracting the appropriate areas in the video content to reduce the noise that can occur while extracting the latent information about signed content.

3.3 Dataset pre-processing

We employed three main preprocessing methods: caption and sign translation alignment, signer bounding box detection, and subtitle merging. These processes are applied for exact temporal alignment between captions and signed content, detecting signers within videos, and creating subsets with longer context translations, respectively.

3.3.1 Signer bounding box selection

Given that the video content has various visual distractors, which can be considered noise to the pose estimation model, and co-occurring behavior in videos where the signers appear in the bottom-right corner of the videos, the bounding box detection must be done for isolating signing activity. Since manually annotating bounding boxes for all frames is labor-intensive, we adopted a semi-automatic annotation approach, which proceeds as follows:

- **Caption-Based Sampling:** Extract caption sequences identified in the alignment process for one video.
- **Object Detection Sampling:** Randomly sample 10 time-intervals from each video and apply an object detection model to each frame.
- **Bounding Box Aggregation:** Detect the human figure whose bounding box center falls within the bottom-right region of the frame (one-third of the y-axis and two-thirds of the x-axis). Aggregate these detections and compute the mean bounding box dimensions, adding 10% padding to each side.
- **Bounding Box Refinement:** Adjust the bounding box dimensions to form a square shape, ensuring consistency and ease of use in subsequent applications.

This method reduced the need for exhaustive manual annotation while providing reliable bounding box information on signer localization. Yet, each annotation is manually controlled, and if classified as invalid crops, the manual annotation is performed using a custom annotation application implemented using the OpenCV (Itseez, 2015) library.

3.3.2 Caption and sign translation alignment

Aligning captions with the corresponding sign sequences was one of the most challenging issues of dataset creation. The exact boundaries of signed sequences often do not overlap with caption start and end points, resulting in the extraction of mismatched and noisy content pairs, which would reduce the performance of the trained models on this data.

To address this, we designed an alignment algorithm that uses gaps in the text captions and pose estimation while detecting exact boundaries as the signer initial

transformer to a null position (hands down and not moving for multiple frames), also known as the active frames or selection process of temporal pruning (Özdemir, et al., 2023), which is utilized in SLR training. The alignment process is carried out as follows:

- Caption Gap Detection: Identify captions with gaps of more than two seconds before the next caption sequence and captions themselves with durations exceeding two seconds.
- Sequence Extraction: Extract the video sequence starting from the caption's timestamp and extend it by 1500 ms beyond the caption's end to account for potential interpretation length without overlapping with the next sequence.
- Active Signing Detection: Apply pose estimation within the signer's bounding box to identify active frames where the signer's hands are positioned above the chest, indicating signing activity.
- Subsequence Refinement: Use the leftmost and rightmost active frames to determine the signing subsequence.

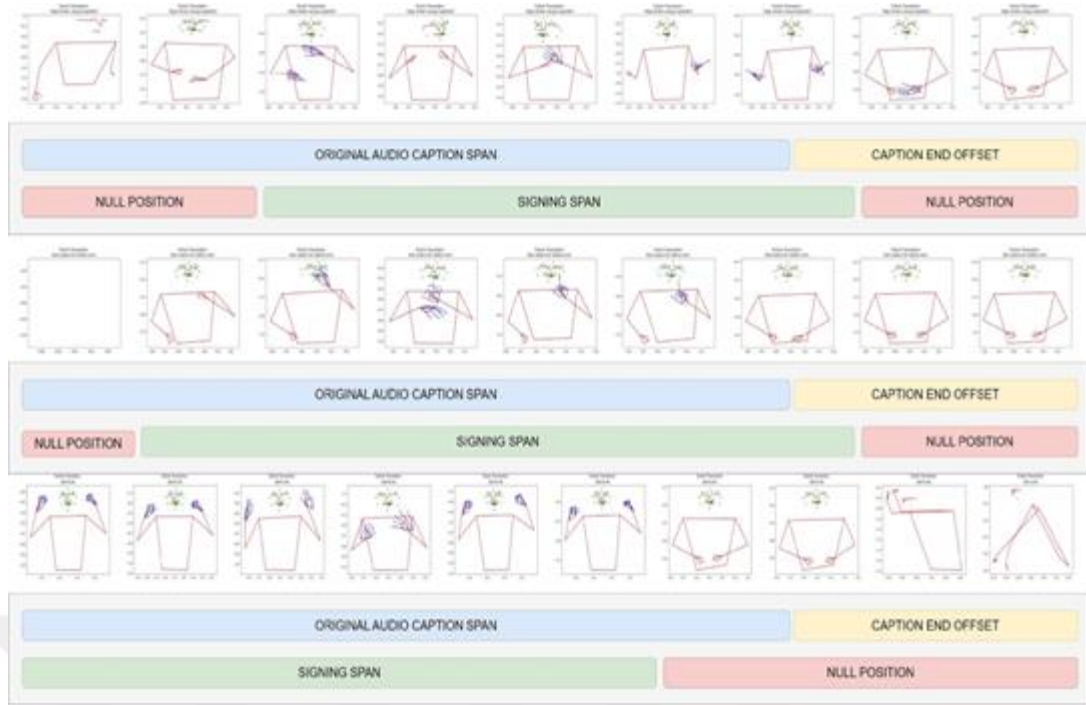


Figure 6. Illustration of audio caption and signing span alignment: blue shows the audio-caption span, green the signing span, red null signing, and yellow the added offset for full signing coverage.

The alignment process illustrated in Figure 6 includes three examples to showcase the key steps:

The audio-caption span (blue) initially covers a broader timeframe than the actual signing activity. Using pose estimation, the null positions (red) are identified, indicating frames where the hands are inactive and at rest (null position). The signing span (green) is determined by detecting active frames, where hand positions deviate significantly, indicating signing motion. A caption end offset (yellow) is applied to ensure the signing sequence is fully captured, even if it exceeds the audio-caption span. As we have discussed, the translation might require a longer duration before the start of the next caption sequence; the end offset enables the algorithm to detect accurate boundaries with the additional padding.

The second case illustrates a scenario where the signer temporarily disappears and reappears due to video editing, such as the insertion of signing segments during

longer gaps. After pose estimation, the null positions are observed before and after the signing activity. Our algorithm would not only drop initial undetected motion but also empty frames as well.

The third example demonstrates two distinct issues at the same time. First, it shows a sequence that begins with direct signing, which is identified as the initial start of the signing activity, with prior padding applied to ensure context is preserved. The second and more significant case demonstrates the algorithm's effectiveness as a noise reduction mechanism. In scenarios involving noisy inputs, such as overlapping background figures or the temporary absence of the signer, the algorithm accurately identifies unsigned sequences. By analyzing both the signer's movements and background detections, it effectively excludes irrelevant or noisy frames from the signing span.

For instance, the last two frames are from the physical activity of a character from the relevant show that is translated. The algorithm excludes those since they differ from the resting position of the signer.

3.3.3 Subtitle merging

The creation of the LC subset required merging multiple caption sequences to provide longer contexts for the translation task. This merging process leveraged the conversational dynamics of the overall video duration, where extended dialogues with minimal gaps between turns are seen. Here are the steps that we followed for subtitle merging:

- **Gap Detection:** Identify caption sequences with more than a two-second gap before the current sequence. Begin merging sequences with less than 400 ms gaps between consecutive captions.

- Merged Sequence Creation: Continue merging until the final sequence has a gap exceeding two seconds after it.
- Alignment and Duration Check: Apply the same alignment method as in the first preprocessing step. Retain sequences with durations less than or equal to 20 seconds to reduce extensive-sized inputs

3.4 Dataset post-processing

To prepare the dataset for model training, we implemented several post-processing techniques to enhance data quality and variability.

3.4.1 Initialization normalization in TID

In sign languages, introducing a new proper name is typically done by fingerspelling each letter of the name. However, repeatedly fingerspelling the full name can be cumbersome in both daily communication and extended translations. To address this issue, a grammatical strategy known as initialization is employed, where only the initial letter of the name is used after the first full spelling (Makaroğlu, 2021). This process involves using the handshape corresponding to the first letter of the name as the handshape for the sign, a practice observed in various sign languages (Kelepir, 2020; Wilbur, 2013).

For example, in our dataset, the sentence "Ben de Zehra'yı çok iyi tanıyorum." ("I also know Zehra very well.") would be transformed into "Ben de Z'yi çok iyi tanıyorum." ("I also know Z very well.") after the initial introduction of the name. This post-processing operation is sign language-aware and aligns with the grammatical conventions of Turkish Sign Language (TID) as well as the fairer ground sentence for BLEU score calculation.

3.4.2 Paraphrasing

As part of our post-processing operations, we employ paraphrasing to generate additional references for the target translations. This process utilizes a pretrained large language model fine-tuned on a paraphrasing dataset, enabling it to produce idiosyncratic translations. Further details on the paraphrasing model are provided in Chapter 4.6. The primary motivation for this operation stems from the evaluation metric used for translation scoring, which relies on n-gram matching. While this approach works effectively for isolating and strict word-order languages like English, it imposes heavy penalties on languages such as Turkish, which are agglutinative and exhibit free-word order. By introducing paraphrased references, we aim to mitigate these penalties and achieve a more accurate evaluation. The challenges and implications of these evaluation issues are discussed in detail in Chapter 7.2.

CHAPTER 4

METHODOLOGY

In this section, we present our methodology for sign language translation using transformer-based pre-trained large language models (LLMs) as shown in Figure 7. We will be dissecting each component of the proposed architecture in detail, starting from how we define our machine translation task to further understand sign language translation and its components.

To establish a clear foundation, we first define the task as an encoder-decoder neural machine translation (NMT) problem and outline its key components. Each component is then examined in the context of its adaptation for SLT, highlighting the specific novelties introduced in this work. These innovations include modifications to handle the unique challenges of sign language input, such as continuous multi-channel data and pose-based representations, as well as enhancements to the encoder-decoder framework to improve translation performance.

4.1 Neural Machine Translation task

Originally, Neural Machine Translation (NMT) was the task of automatically translating text from one language to another using computational models. In its simplest form, the task involves encoding an input sequence in the source language and decoding the target sequence in the target language for given source-target language pairs such as Turkish to English or English to French, etc.

Let $X = \{x_1, x_2, \dots, x_n\}$ be the input sentence in the source language, where x_i denotes the i -th token. The model predicts the target sentence $Y = \{y_1, y_2, \dots, y_m\}$, where y_j represents the j -th token in the target language.

The goal is to maximize the conditional probability $P(Y | X)$, which can be factorized as:

$$P(Y | X) = \prod_{j=1}^m P(y_j | y_1, y_2, \dots, y_m) \quad (4.1)$$

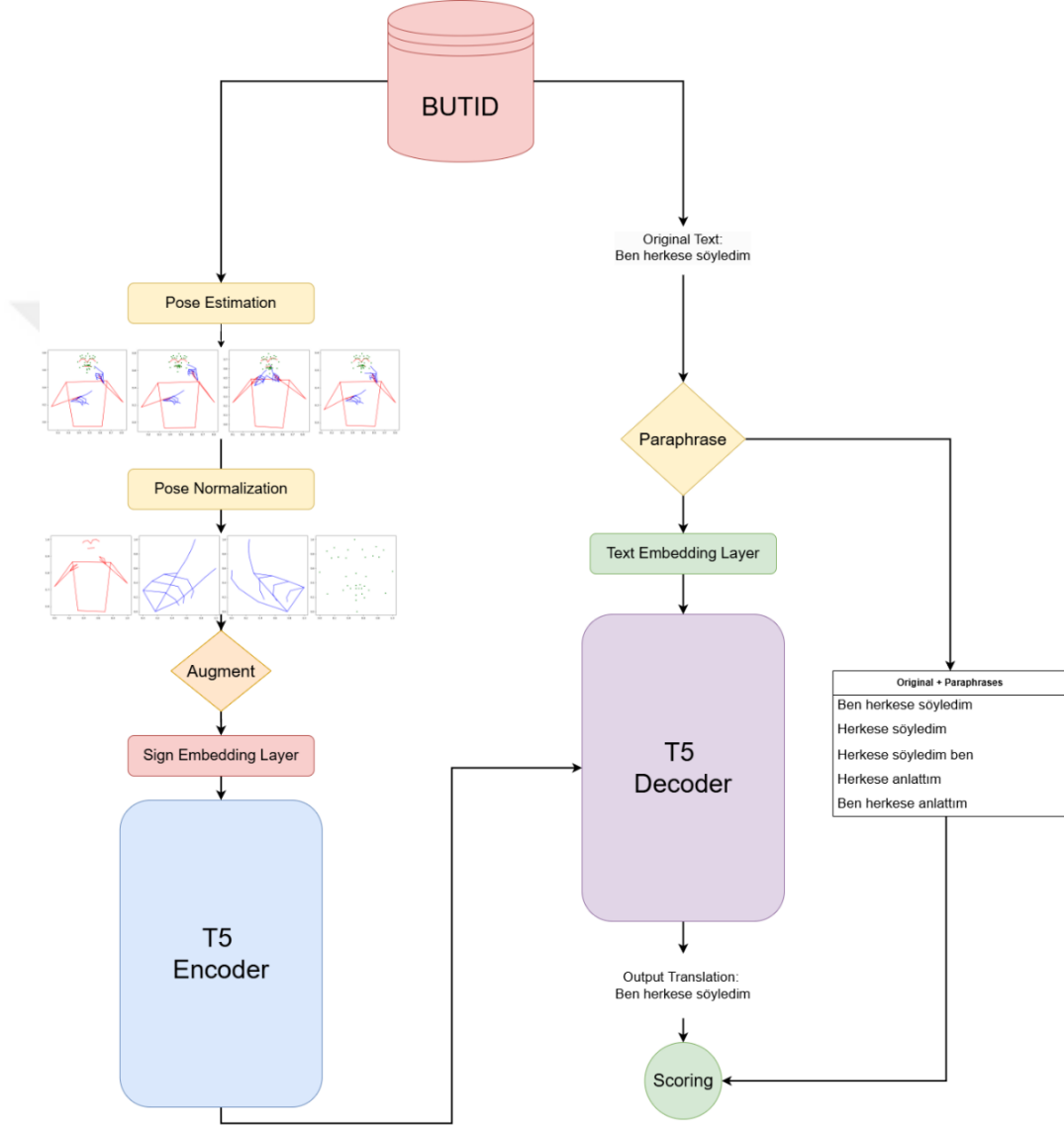


Figure 7. The figure shows the overall pipeline of the fine-tuning stage

Here $P(y_j | y_1, y_2, \dots, y_m)$ represents the probability of generating the j-th token in the target sentence, given all the previously generated tokens and the input sentence

X. This is modeled using encoder-decoder architectures with attention mechanisms, such as those in transformer-based models.

The SLT application modifies the machine translation task by changing the input modality, where the input is not text but a sequence of visual features extracted from sign language videos. These features ideally encode extensive linguistic information about sign language from visual input to maximize the output of an unseen sequence in the target spoken language from this information.

In SLT, let $V = \{v_1, v_2, \dots, v_t\}$ be the input sequence of visual features, where v_t denotes the feature vector extracted at time t .

The task is then to predict the target sentence $Y = \{y_1, y_2, \dots, y_m\}$ maximizing the conditional probability $P(Y | V)$:

$$P(Y | V) = \prod_{j=1}^m P(y_j | y_1, y_2, \dots, y_{j-1}, V) \quad (4.2)$$

Deriving from the base implementation of the neural machine, we can formulate our end-to-end SLT task via four main components:

4.2 Feature transformation

In Uthus et al.'s (2023) paper, the feature transformation consists of mapping visual features extracted from sign language videos into the token space of the pre-trained language model using a linear layer before feeding into the encoder layer. This extraction layer inputs a matrix of visual features, defined as $V \in R^{T \times D_v}$, extracted from a sign language video, where T denotes the number of time steps (frames) and D_v is the dimensionality of the visual feature vectors.

To further improve the capacity of this transformation layer, we are proposing two different layer types on top of the baseline methodology for maximizing the encoded representations of the visual inputs. These encoders are the Multi-layer Perceptron (MLP) encoder, which is applied for learning non-linear transformations from the input, and the One-dimensional Convolutional (Conv1D) encoder for extracting the temporal information via convolutional filters across multiple frames for embedding motion information from the shared matrix across the overall sequence.

4.2.1 Linear encoder (Uthus et al., 2023)

In the original methodology, the author uses a learnable weight matrix $W \in R^{D_v \times D_e}$, where D_e corresponds to the dimensionality of the language model's input embeddings defined in Equation 4.3 to match the dimensionality with the sign language sequence and the input embedding layer. Equation 4.3, where $V' \in R^{T \times D_e}$ is the resulting matrix of embedded visual features, and $b \in R^{D_e}$ is a bias vector added to each transformed feature vector.

$$V' = VW + b \quad (4.3)$$

This linear mapping enables each visual feature vector v_t at a time step t to be projected into the embedding space compatible with the pre-trained language model, thereby encoding sign language inputs within the model's architecture.

4.2.2 Multi-Layer Perceptron (MLP) encoder

The MLP encoder comprises a linear layer followed by a Gaussian Error Linear Unit (GELU) activation function (Hendrycks & Gimpel, 2016) and a dropout layer. The GELU activation function is defined in Equation 4.4, where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution.

$$\text{GELU}(x) = x \cdot \Phi(x) \quad (4.4)$$

The final formulation of MLP Encoder is defined in Equation 4.5

$$V' = \text{Dropout}(\text{GELU}(VW + b)) \quad (4.5)$$

This feature extractor would provide a non-linear transformation to sign language compared to the linear encoder layer, where we expect better performance on the translation task.

4.2.3 Convolutional (Conv1D) encoder

The Convolutional Encoder consists of three sequential blocks comprising an one-dimensional convolutional layer (Conv1D), batch normalization, and a GeLU activation function. A one-dimensional convolutional layer applies a convolution operation over a single spatial or temporal dimension. The output y at position t is computed in Equation 4.6, where x is the input sequence, w is the convolutional kernel of size k , and b is the bias term.

$$\text{Conv1D}(x) = \sum_{i=1}^k x_{t+i-1} \cdot w_i + b \quad (4.6)$$

The final formulation of Conv1D Encoder is defined in Equation 4.7

$$V' = \text{Dropout} \left(\text{GELU}(\text{Conv1d}(x)) \right) \quad (4.7)$$

As we utilize learned information from a pretrained encoder, our new representation V' can now be defined as $V' = R^{T \times D_e}$.

4.2.4 Encoder

The encoder processes the projected visual representation V' , often using pose-based feature extraction or learned representations from a pre-trained model, and maps it to a high-dimensional latent space using the linear or convolutional layer, as in Figure 4.8, where $H = \{h_1, h_2, \dots, h_T\}$ represents the encoded representations.

$$H = f_{enc}(V') \quad (4.8)$$

4.2.5 Decoder

The decoder generates the target sentence Y based on the latent representations H and previously generated tokens.

$$P(Y | V) = \prod_{j=1}^m P(y_j | y_1, y_2, \dots, y_{j-1}; H) \quad (4.9)$$

4.2.5.1 Beam search decoding

Beam search decoding (Tillmann & Ney, 2003) is an algorithm often used for generating sequences in sequence-to-sequence models. Unlike greedy decoding, which selects the highest probability token at each time step, beam search explores multiple potential sequences simultaneously, maintaining a set of top-k candidate sequences (the beam) based on their cumulative probabilities to output a sequence with maximum probability.

At each decoding step t , beam search maintains k candidate sequences and their corresponding cumulative log probabilities. For each sequence in the beam, the model generates a probability distribution over the vocabulary V for the next token. The $k \times |V|$ possible continuations are scored, and the top-k candidates are selected to form the updated beam.

At the initial decoding step, where $t = 1$, the beam consists of the start token $\langle \text{sos} \rangle$ with a log probability of $\log(1) = 0$ where the initial beam set can be shown as $\mathcal{B}_0 = \{(\langle \text{sos} \rangle, 0)\}$.

For each candidate sequence $y_{1:t}$ in the beam at step t , the model predicts a probability distribution $P(y_t | y_{1:t-1}, X)$ where X is the input sequence (e.g., pose or video embeddings).

The cumulative log probability for a sequence $y_{1:t}$ is computed as:

$$\text{Score}(y_{1:t}) = \sum_{i=1}^t \log P(y_i | y_{1:i-1}, X) \quad (4.10)$$

The top-k sequences with the highest cumulative scores are retained:

$$\mathcal{B}_t = \text{argmax}(\text{Score}(y_{1:t}): y_{1:t} \in \mathcal{B}_{t-1} \times V) \quad (4.11)$$

The process continues until the end-of-sequence token (<eos>) is generated for all sequences in the beam or a predefined maximum sequence length is reached. The final sequence is selected based on the highest cumulative log probability.

To further decode diversity and avoid overly short or repetitive sequences, beam search decoding also incorporates the “Length Penalty” parameter, adjusting the score to favor longer sequences, and the “Repetition Penalty”, which penalizes sequences with repeated n-grams by reducing their scores.

4.3 Feature normalization

For the pose estimation step, we apply min-max normalization to the x and y coordinates of each landmark cue (hands, pose, and face) independently, rather than normalizing the entire set of landmarks across all cues diverging from the

4.3.1 Cue-based normalization

Our dataset categorizes the landmarks into three distinct groups: hands, pose (upper body), and face. Each cue is normalized separately to retain its specific spatial distribution. For each cue, we apply frame-level normalization as shown in Equation 4.12, where the minimum and maximum x and y coordinates limit the maximum span of each axis between [0, 1]. The results of normalization are shown in Figure 8.

$$L_{norm}^{cue} = \frac{L^{cue} - \min(L_{x,y}^{cue})}{\max(L_{x,y}^{cue}) - \min(L_{x,y}^{cue})} \quad (4.12)$$

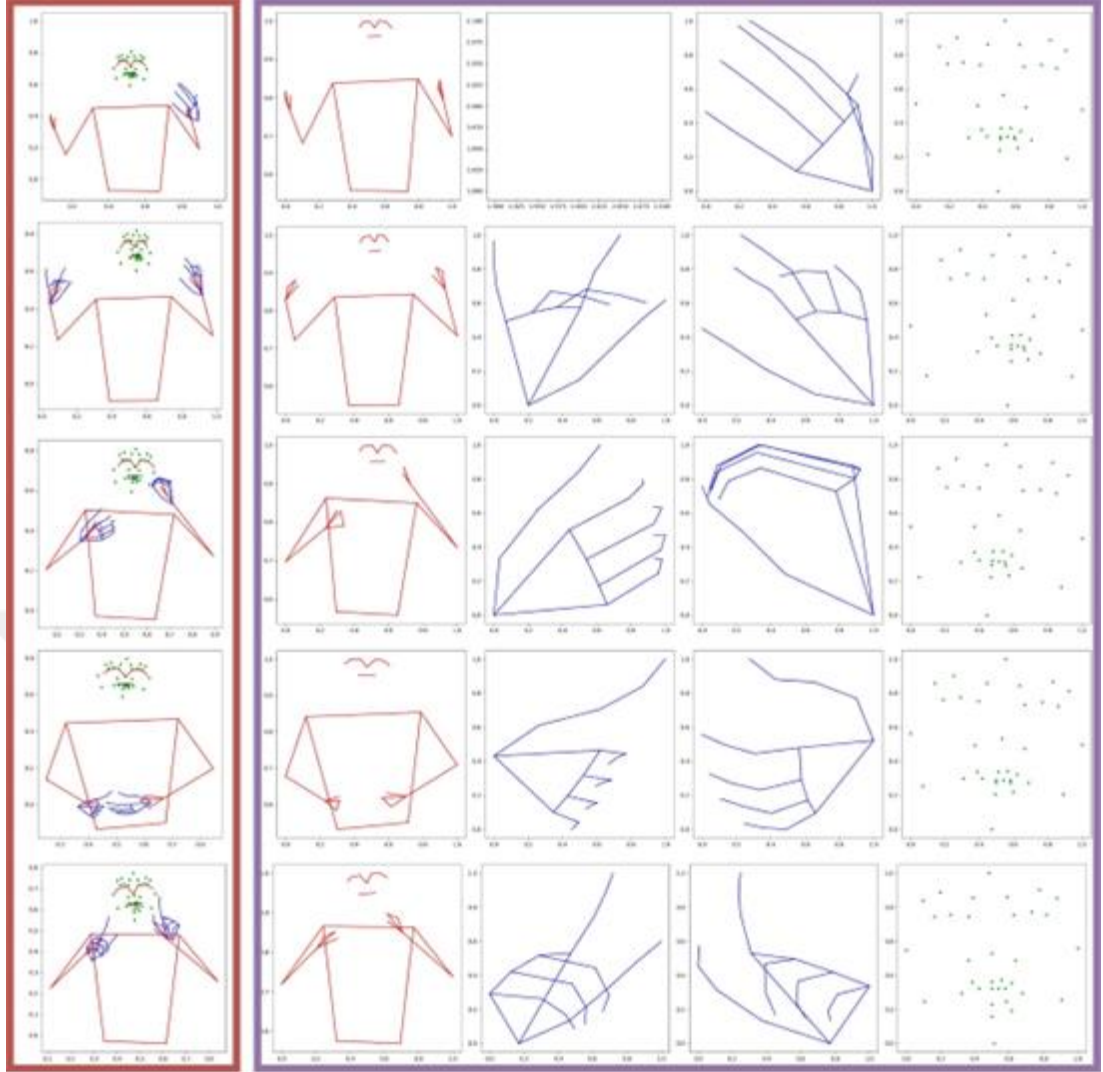


Figure 8. Visualization of normalized vs. unnormalized pose sequences: first column (red) shows unnormalized skeletal data, while the second (purple) displays normalized cues—pose, left hand, right hand, and face

4.4 Pre-trained Large Language Models (LLMs)

In our sign language translation framework, we adopt pre-trained LLMs as the foundational components of our encoder-decoder NMT architecture for SLT. As discussed in the previous section, establishing a robust baseline architecture is essential. Training a large architecture from scratch is both computationally expensive and prone to overfitting due to the high number of parameters, as observed in various studies employing different architectural bases. The findings of Uthus et al. (2023) demonstrate that utilizing a model initialized from a pre-trained checkpoint

rather than a newly initialized model significantly improves SLT performance. This behavior can be attributed to the decoder’s pre-trained contextual representations of the target language, which are learned during pre-training on extensive datasets.

However, before we continue with the rest of our methodology, we should first tackle what the pre-training strategy is in the T5 architecture to understand how this pre-trained model operates.

4.4.1 T5 model architecture and training

The T5 model, or Text-to-Text Transfer Transformer (Raffel et al., 2020), utilizes a Transformer-based encoder-decoder architecture to process and generate text sequences. This architecture comprises an encoder that processes the input text and a decoder that generates the corresponding output text, which aligns with our methodology of encoder-decoder-based NMT architecture. In this architecture, both encoder and decoder components utilize self-attention mechanisms and feed-forward neural networks.

4.4.1.1 T5 architecture

The T5 model utilizes the foundational architecture of the Transformer model proposed by Vaswani et al. (2017) as shown in Figure 9, with variations in parameter configurations for different components, such as the number of encoder and decoder layers, the number of self-attention heads, and hidden dimensions. While the core structure of the architecture remains consistent with the original Transformer design, the T5 model differentiates itself primarily through its innovative training strategies and objectives, which we will explain in the next section.

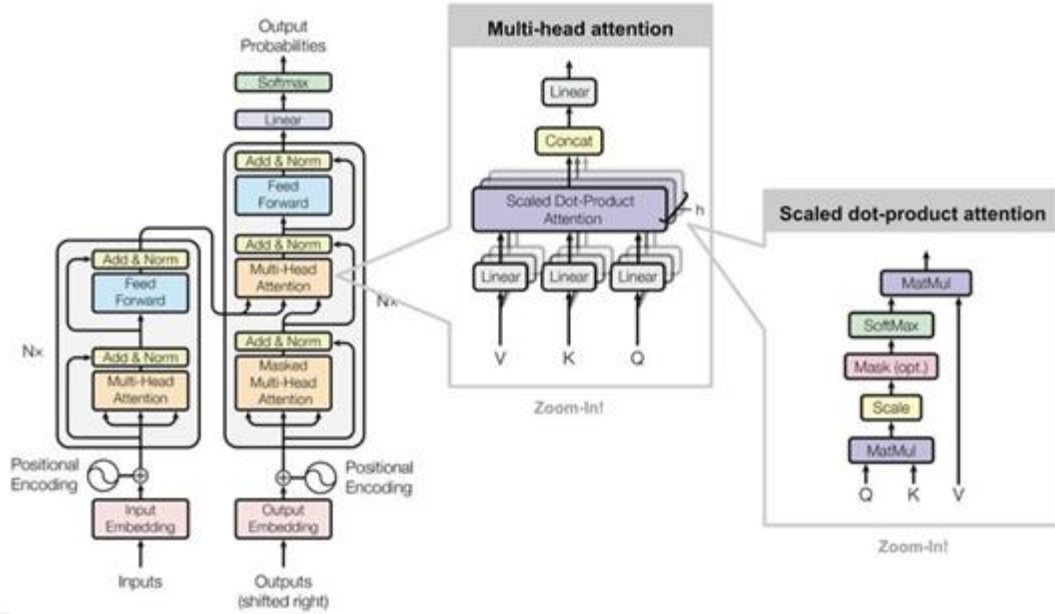


Figure 9. The original Transformer architecture

Notably, each architecture employed in this work, including mT5-small, mT5-large and TURNA adhere to the original Transformer-based architecture during training. These models leverage the same encoder-decoder structure, incorporating modifications in parameter settings and additional hyperparameters tailored for specific tasks. Detailed descriptions of these hyperparameters and their configurations are provided in Chapter 5.

4.4.1.2 T5 pretraining

During pre-training, T5 employs a denoising objective that involves corrupting input text sequences and training the model to reconstruct the original content. The denoising objective is implemented by randomly selecting spans of tokens within the input sequence, replacing them with unique sentinel tokens (e.g. `<extra-id-0>`, `<extra-id-1>`) and tasking the model with generating the missing spans. Consider the original sentence as “I love looking at the mountains”. A possible masked input could be: “I love `<extra-id-0>` the mountains”

Here, the span "looking at" is replaced with the sentinel token $\langle extra-id-0 \rangle$. The target output for the model to generate would be: " $\langle extra-id-0 \rangle$ looking at" The model's task is to predict the missing span "looking at" corresponding to the sentinel token $\langle extra-id-0 \rangle$. The loss is computed using cross-entropy loss between the decoder's output and the target sequence. By training on such corrupted sequences on a large set of corpora, T5 learns contextual representations in correlation to autoregressive generation capabilities.

4.5 Scoring

In this research, we use the BLEU (Bilingual Evaluation Understudy) for the evaluation of model performance.

The BLEU score is a widely used metric for evaluating the quality of machine-translated text by comparing it to one or more reference translations. It is designed to assess both fluency and adequacy by measuring how well the candidate translation matches the references. BLEU relies on n-gram precision, the brevity penalty, and a geometric mean to combine results.

The precision for n-grams, which are contiguous sequences of n words, is computed as Equation 4.14:

$$P_n = \frac{\text{\# of overlaps between candidate and reference}}{\text{\# of n-grams in the candidate}} \quad (4.14)$$

To prevent inflating scores through repeated words, BLEU uses clipped precision, which limits the count of each n-gram in the candidate to its maximum count in the references. This is mathematically expressed as:

$$P_{norm}^{clipped} = \frac{\sum_{n \in candidate} \min(c(candidate), c(reference))}{\text{Total n-grams in candidate}} \quad (4.15)$$

The n-gram precisions up to N are then combined using the geometric mean, defined as:

$$P = \exp\left(\frac{1}{N} \sum_{n=1}^N \log(P_n^{clipped})\right) \quad (4.16)$$

To address cases where the candidate translation is shorter than the references, a brevity penalty (BP) is applied. The brevity penalty is defined in Equation 4.17, where c is the length of the candidate translation, and r is the effective reference length for the cases where c is smaller than or equal to r.

$$BP = \exp\left(1 - \frac{r}{c}\right) \quad (4.17)$$

If the candidate sequence is longer than the value the penalty is assigned BP =

1. Finally, the BLEU score is computed by combining the geometric mean of n-gram precisions and the brevity penalty:

$$\text{BLEU} = BP \cdot \exp\left(\frac{1}{N} \sum_{n=1}^N \log(P_n^{clipped})\right) \quad (4.18)$$

4.6 Paraphrasing

Paraphrasing presents significant improvements in both the evaluation of translation quality and the diversity of training data. Sign languages exhibit structural differences from spoken languages, leading to multiple valid translations for the

same sign sequence. These variations depend on the translator's interpretation and the contextual nuances of the content. We have grouped the possible reasons that spoken language's grammatical structures are where this valid translation stems from. In detail, we will discuss the linguistic motivation behind the paraphrasing in Chapter 7.2.

4.6.1 Neural paraphrasing tool

To solve the challenges from the language variation, we implemented an automated solution utilizing the TURNA model, (Uludoğan et al., 2024) which we will explain in detail in Section 5.1.4. The fine-tuned model, which is trained on the Turkish subset of the OpenSubtitles2018 dataset (Lison, et al., 2018) comprises translated movie subtitles across multiple languages, including Turkish, aligning closely with the language used in our dataset.

By utilizing TURNA, we generated five unique paraphrases for each sentence in our dataset, thereby providing multiple reference translations for evaluation. This approach mitigates the limitations of evaluation metrics like BLEU, which may not fully capture semantic equivalence due to surface-level differences.

For example, the original sentence "Bedava versen giymeyeceğim", which is an actual training instance found in our dataset, was paraphrased into:

- "Bedava versen giymem."
- "Bedavaya giymem."
- "Bedava versen bile giymem."
- "Bedavaya giymeyeceğim."
- "Bedava versen bile giymeyeceğim."

Moreover, incorporating these paraphrases is not only considered for creation in the evaluation reference set for a fair BLEU score calculation but also serves as a data augmentation technique. While mapping of the encoder input remains the same, output representations from decoder attention would result in word-order indifference in encoder-decoder attention, focusing on semantic content.

4.7 Pose augmentation

Pose augmentation introduces variability into the input data to improve the model's ability to handle diverse scenarios. Three main augmentation methods are applied in our pipeline: random drop, skip frames, and cue masking. Each method addresses specific challenges observed in sign language datasets, particularly those involving low-resolution videos and incomplete data.

4.7.1 Random drop

Random drop reduces the length of input sequences by removing a percentage of frames. This operation mimics scenarios where frames are missing or skipped during video recording or preprocessing. A random percentage of frames, between 10% and 30%, is selected for removal. Frames are dropped uniformly across the sequence to maintain the temporal structure of the remaining data. This augmentation pushes the model to learn contextual relationships in the sequence, even when parts of the input are missing.

4.7.2 Skip frames

Skip frame augmentation reduces the frame rate by selecting every other frame in the input sequence, effectively halving the original frame rate. This method is particularly useful for handling YouTube-based datasets, where videos often have low resolution and inconsistent frame rates. Starting from the first frame, every second frame is sampled, an artificial replication of the undetected pose from the frames with blur or high-velocity movement. This augmentation is done for a more robust system on low-resolution data instances.

4.7.2 Cue masking

Cue masking applies artificial occlusion to specific cues, such as hand, pose, or face landmarks. A random percentage of cues, ranging from 10% to 30%, is selected for masking. This operation is applied independently to different landmark groups, such as hands or facial features, simulating scenarios where parts of the signer's body are obscured by motion blur, poor lighting, or camera angles. By introducing partial occlusions, cue masking prepares the model to infer gestures from incomplete information.

4.7.4 Controlled augmentation strategy

To apply augmentations consistently across training iterations, random seeds are set based on dataset indices. Each augmentation uses an independent seed, allowing multiple augmentations to create diverse outputs without compromising replication. This semi-random approach ensures that the augmentations are applied in a systematic and controlled manner, providing a diverse yet predictable set of augmented data.

CHAPTER 5

EXPERIMENTAL SETUP

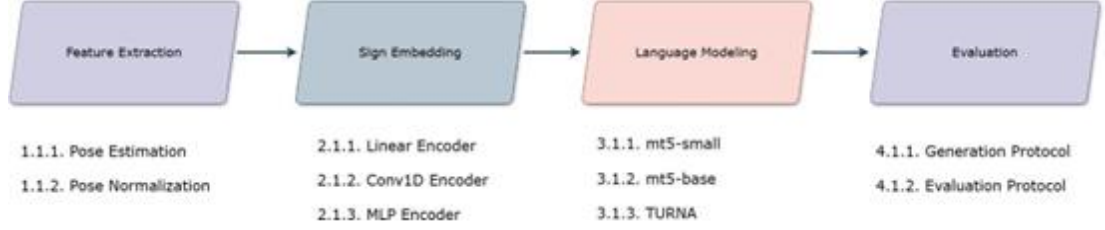


Figure 10. Components of experimental setup in serial order

In our experimental setup, we designed a comprehensive pipeline as shown in Figure 10.

5.1 Pose estimation

To extract pose information, we utilized Google’s MediaPipe library (Lugaresi et al., 2019) using the BlazePose GHUM 3D model (Bazarevsky et al., 2020) for 3D pose estimation. This library is used for both the performance and accuracy of the landmark selection and its inclining usage in the recent literature for a fair comparison (Boháček & Hruží, 2022; Gueuwou et al., 2023; Uthus, et al., 2023).

5.1.1 Implementation details

We employed the parameters for initializing the Holistic pipeline as in Table 4.

Table 4. Model Parameters for Initializing MediaPipe Holistic Pipeline

Model Parameters	Value
Model Complexity	2
Min. Detection Confidence	0.3
Min. Tracking Confidence	0.3

5.1.2 Landmark selection

Following the methodology of the YouTube-ASL dataset (Uthus et al., 2023), we selected specific landmarks pertinent to sign language analysis:

- 21 Left-hand + 21 Right-hand landmarks
- 25 Body Pose Landmarks
- 35 Facial Landmarks

25 upper-body landmarks, focusing on areas actively involved in signing while lower-body landmarks were excluded due to their limited relevance in sign language and visibility in the video content. While MediaPipe provides 468 facial landmarks, we selected those most relevant to sign language expressions. 35 landmarks were selected to represent facial areas that are important for extracting information about non-manual markers (NMMs) or mouthing information required for the construction of grammatical sequences. Our selection is carried out on Uthus et al.'s (2023) paper for comparative purposes. All of this processing, including data collection, pose estimation, and sentence alignment, has been done on 8 x (Intel(R) Xeon(R) Platinum 8480+ 2.0GHz with 56 cores) on a TRUBA cluster served by TÜBİTAK ULAKBİM High Performance and Grid Computing Center.

5.1.3 Sign embedding configurations

Three configurations are employed to explore the impact of different encoding strategies on translation performance: a Linear Layer, a Multi-Layer Perceptron (MLP) layer, and a Convolutional (Conv1D) Layer.

5.1.3.1 Linear layer

The Linear layer configuration consists of a fully connected layer that maps the 255-dimensional pose input into a model-specific embedding space, or input dimension, followed by a dropout layer with a probability of 0.1 to prevent overfitting.

5.1.3.2 MLP layer

MLP configuration includes two sequential layers. The first layer maps the 255-dimensional input to the model-specific embedding space, while the second layer performs a transformation within the embedding space. Both layers utilize the GELU activation function and incorporate dropout with a 0.1 probability, and a batch normalization layer is applied during training for regularization purposes.

5.1.3.3 Conv1D layer

The Conv1D Encoder configuration leverages three 1D convolutional layers with kernel sizes of [5, 3, 3] and strides of [2, 1, 1], in respective layers. The first layer transforms the 255-dimensional input to model vocab dimensionality, while the subsequent layers maintain the input dimensionality. Each convolutional layer is followed by GeLU activation, and a feed-forward layer is added at the end of the convolutional stack.

We will utilize the model setting to compare the pre-trained model's size with the pre-trained data language.

5.1.4 Language model configuration

Our SLT model employs transformer-based architectures for comparison in terms of their parameter sizes and pretraining datasets. We have provided parametric differences in comparison in Table 5.

5.1.4.1 mT5-small

The mT5-small model, a multilingual version of T5 (Xue, 2020), is trained on the mC4 corpus, which spans 101 languages, including Turkish. Its architecture matches that of the T5-Small model, consisting of approximately 300 million parameters. This model is designed to evaluate how smaller-scale models perform in the domain of Turkish sign language translation and methodological comparison with Uthus et al. (2023).

5.1.4 mT5-large

The mT5-large model builds upon the mT5 architecture, expanding its parameter count to approximately 580 million. This larger model is also trained on the mC4 corpus, providing enhanced language modeling capabilities. We include mT5-large to explore the impact of increased model capacity on Turkish sign language translation. The addition of mT5-large allows for a direct comparison between model parameter size with the same training strategies.

5.1.4.3 TURNA (Uludoğan et al., 2024)

TURNA is a monolingual Turkish language model based on the Unified Language Learning (UL2) framework (Tay et al., 2023), designed with 1.1 billion parameters to specialize in Turkish text understanding and generation. Its architecture includes

36 encoder and decoder layers, 16 attention heads, 1024-dimensional token embeddings, and Multi-Layer Perceptron (MLP) layers with 2816 hidden dimensions using Gated GeLU activations. However, it should be noted that there is no reported performance on the translation capabilities of the model. The reason behind the addition of this model to experiments is to provide a comparison between the performance of language-specific and multilingual model selection on sign language translation.

Table 5. Hyperparameters of mT5-small, mT5-large, and TURNA Models

Hyperparameter	mT5-small	mT5-large	TURNA
Vocabulary Size	250,112	250,112	32,128
Hidden Size	512	1024	1,024
# Encoder Layers	8	24	36
# Decoder Layers	8	24	36
# Attention Heads	6	16	16
Feed-Forward Layer Size	2,048	3,072	2,816
Activation Function	ReLU	ReLU	Gated GeLU
Dropout Rate	0.1	0.1	N/A
Attention Dropout Rate	0.1	0.1	N/A
# Training Steps	1,000,000	1,000,000	N/A
Warmup Steps	10,000	10,000	N/A
# Parameters	300M	1.2B	1.1B

5.2 Dataset configuration

The dataset is divided into two splits: SC and LC. The SC split comprises shorter clips, while the LC split includes extended sequences. This approach maintains consistent lexical distributions across the training, development, and test sets, as stratified sampling divides a population into distinct categories. The following tables provide detailed statistics for each split:

5.2.1 SC dataset distribution

Table 6. Statistics of the SC Split

Metric	Train	Dev	Test
# clips	108,530	6,029	6,030
# hours	138.47	7.72	7.70
Avg. duration (sc)	2.59	2.61	2.60
Avg. text length (ch)	29.93	30.38	30.02
Avg. text length (token)	6.04	6.09	6.04
Avg. sentences	1.13	1.12	1.13

5.2.2 LC dataset distribution

Table 7. Statistics of the LC Split

Metric	Train	Dev	Test
# clips	106,665	5,926	5,926
# hours	346.96	19.38	19.41
Avg. duration (sc)	9.71	9.77	9.79
Avg. text length (ch)	114.23	114.53	115.2
Avg. text length (token)	22.28	22.37	22.5
Avg. sentences	3.99	4.02	4.04

Despite variations in clip duration, character length, and token length between the subsets, we achieved an even distribution of translation density within each split. The primary distinctions between the SC and LC subsets lie in the Avg. clip duration, character length and token length. The SC subset comprises shorter clips with fewer tokens, while the LC subset contains longer clips with more extensive textual content.

In our training process, we are using the training and development sets to control the loss and evaluation metrics for early stopping purposes, and we are running inference on the test set for final scoring.

5.3 Training configuration

The model was trained using the Adafactor optimizer (Shazeer & Stern, 2018), configured with a learning rate of $1e-5$ for the TURNA model and $1e-4$ for other models. The optimizer’s epsilon parameters were set to $(1e-30, 0.001)$ for regularization constants for the square gradient and parameter scale, respectively. A clip threshold of 1.0 was applied to the root mean square of the final gradient update, and a decay rate of -0.8 was used to compute running averages of squared gradients. The beta1 parameter was not utilized, indicating that first-moment estimation was not employed. No weight decay was applied, while the relative step was applied, with step-based scaling rather than epoch-based one to the learning rate to be scaled by the root mean square of the parameters and computed in a time-dependent manner, respectively. Warmup initialization was not used. A linear learning rate scheduler was employed to adjust the learning rate progressively during training. Experiments were conducted on the LUMI supercomputer, utilizing nodes equipped with four AMD MI250X GPUs and a 64-core AMD EPYC "Trento" CPU. To maintain an effective batch size of 512, gradient accumulation steps were adjusted based on model size: for mT5-small, a per-GPU batch size of 32 with 2 gradient accumulation steps; for mT5-large, a per-GPU batch size of 16 with 4 gradient accumulation steps; and for TURNA, a per-GPU batch size of 4 with 16 gradient accumulation steps. Mixed precision training with FP16 was also utilized.

5.4 Generation Configuration

During inference on both development and test sets, beam search with 4 beams was employed, applying a length penalty of 1.0 and preventing the repetition of any 3-gram sequences in the output. The decoder was allowed to generate up to 128 new tokens, with early stopping enabled to terminate generation upon reaching the end-of-sequence token. This generation protocol was adopted from the Uthus et al. (2023) paper.

5.5 Evaluation protocol

In this study, we evaluate the performance of our models using the BLEU (Papineni et al., 2002) score, which is widely used in natural language translation tasks to measure the n-gram overlap between machine-generated translations and human references. BLEU provides a straightforward and interpretable metric for assessing the quality of sign language translation models, particularly in terms of lexical and syntactic similarity.

However, it is important to note that BLEU has limitations, especially when evaluating translations involving languages with significant morphological and syntactic differences. More recently, metrics like BLEURT (Sellam et al., 2020) have been introduced to address these limitations by incorporating semantic representations on sentence similarity, while BLEU score is limited to n-gram overlap. Unfortunately, as of now, there is no available BLEURT model specifically fine-tuned for Turkish. This restricts our ability to replicate evaluation approaches such as those used by Uthus et al. (2023), which incorporate BLEURT for more comprehensive assessments of translation quality.

To solve this issue, we have performed the BERTScore metric (Zhang et al., 2020) using the BERTurk model, which is a pretrained encoder model for the Turkish language. Since the metric provides similar performance as the BLEU metric, we will not be reporting the output of this metric.

5.6 Tools

This research employed the Transformers (Wolf et al., 2020) and PyTorch (Paszke et al., 2019) for model and training implementations, accelerate (Gugger et al., 2022) for multi-GPU training, and SacreBLEU (Post, 2018) for standardized BLEU score evaluation.

CHAPTER 6

EXPERIMENTAL RESULTS

Our experiments were designed to evaluate the impact of various factors on model performance, including model size, the variation of sign embedding layers, and the effect of data augmentation. By testing multiple setups, we aimed to analyze how these components influence translation quality and the representation of sign language content.

Since we have multiple components, we are required to have a large experimental setup. As you can see in Figure 11, we have overall done 36 experiments in this work.

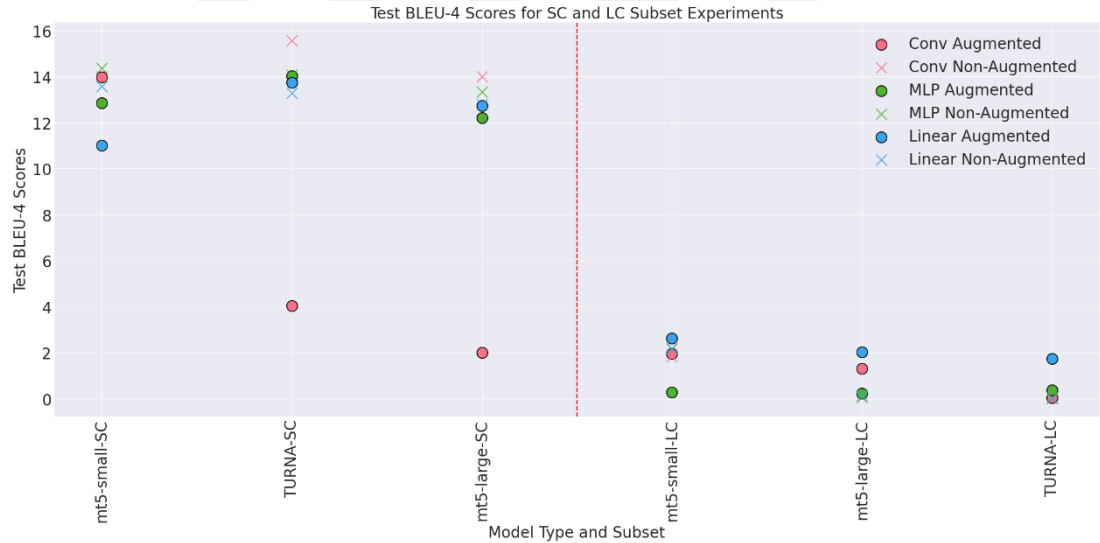


Figure 11. BLEU-4 results on test set for all experiments, with a vertical red line-separating models trained on SC or LC subset

At an initial glance, the first noticeable difference is the BLEU-4 value difference between the SC and LC datasets. While the BLEU-4 value margin is around a 10-16 period in the SC subset, this value decreases to 0.05-2.5 for the LC subset. The reason behind this is the probability of mismatch in longer contexts. As the context

length increases, the BLEU-3 and BLEU-4 match rates decrease, as seen in the overall figure, while the BLEU-1 and BLEU-2 values are closer and less prone to change across subsets (see Table 9 and Table 8).

Before we start our analysis with each component, we first need to clarify the interpretation of the BLEU metric. Our analysis distinguishes the BLEU score interpretation for each subset based on the divergence observed in SLT systems. For the SC dataset, we separate BLEU-1 and BLEU-2 scores from BLEU-3 and BLEU-4, defining two categories: "single-word SC," referring to sequences consisting mainly of 1-2 words, and "multi-word SC," which includes short sequences with more than 3 words. This categorization aligns with the grammatical structure of Turkish, where the agglutinative nature leads to a significant number of sentences comprising only 1-2 words, as discussed further in Chapter 7.

For these sequences, our analysis focuses on changes in model performance across SC subset training, while for the LC subset, the dataset predominantly includes sequences of more than 3 words. Hence we limit our analysis strictly to the BLEU-4 metric in the LC subset.

6.1 Comparison between model sizes

The effects of SC and LC datasets on model performance are distinct and align with the dataset characteristics. In the SC case, we have two findings. While the TURNA model emerges as the best-performing architecture in BLEU-3 and BLEU-4, the mT5-small model performs better in BLEU-1 and BLEU-2 metrics. These results highlight two key points about the SC subset. For sequences classified as "single-word SC" sequences, model performance decreases as parameter size increases. We argue that this is due to underfitting caused by the limited number of "single-word

SC" sequences in the training set. Smaller models, which require less data to generalize effectively, perform better on these shorter sequences. Another contributing factor could be the semantic density of these sequences. The agglutinative nature of Turkish allows for complex meanings to be condensed into single words. Predicting such sequences with larger models may demand more training data to capture this semantic richness, leading to better performance with smaller models in these cases.

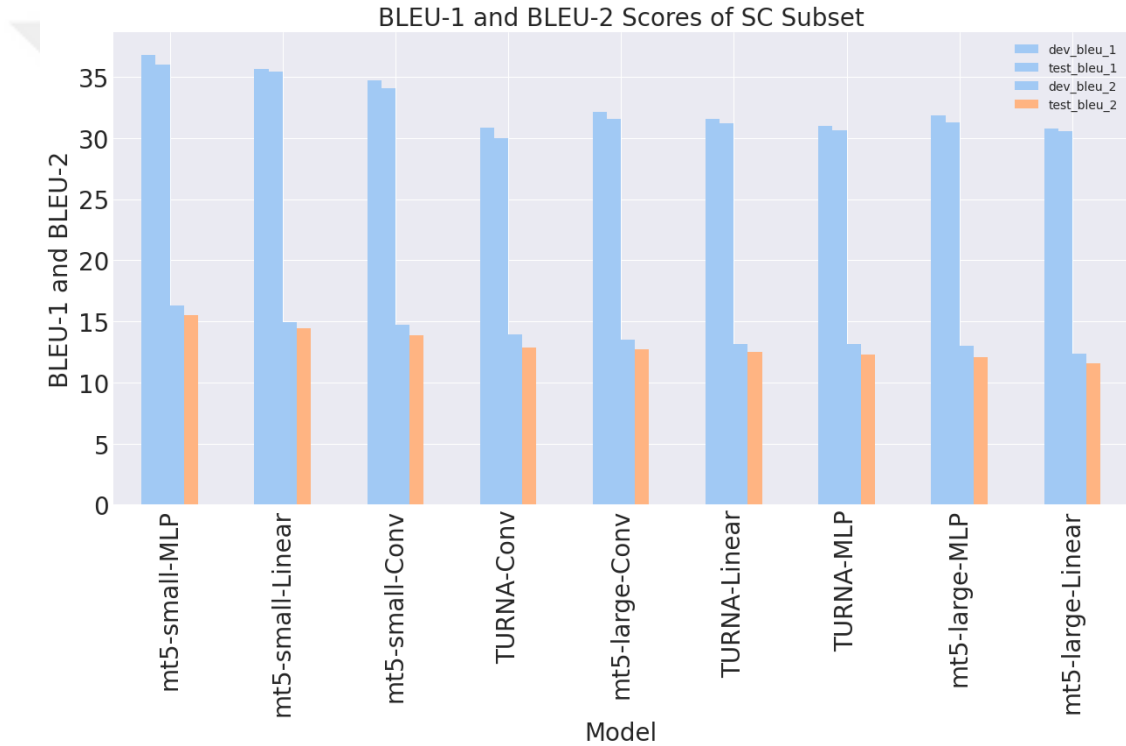


Figure 12. BLEU-1 and BLEU-2 performance comparison by model type across SC development and test sets

This result is expected, as TURNA is a language-specific model designed for Turkish, and the SC dataset provides many shorter-context instances that align well with the model's training objectives. For BLEU-3 and BLEU-4 metrics, the TURNA model, with its Conv1d encoder, demonstrates the best performance. This indicates

that for shorter sequences with sufficient training data, a language-specific encoder is more effective in capturing the nuances of Turkish.

When compared to other models, such as mT5-small and mT5-large, the results further highlight the impact of parameter size and pretraining language on translation performance. Notably, the mT5-large model underperforms compared to both mT5-small and TURNA, suggesting that the increased parameter size of mT5-large does not necessarily translate to better performance without sufficient alignment with the training data. These findings emphasize that both model architecture and pretraining on the target language significantly influence the quality of translation in SC datasets. However, the effect of parameter size has more influence on the changes when comparing the mT5-small and mT5-large models in the LC subset, as shown in Figure 14.

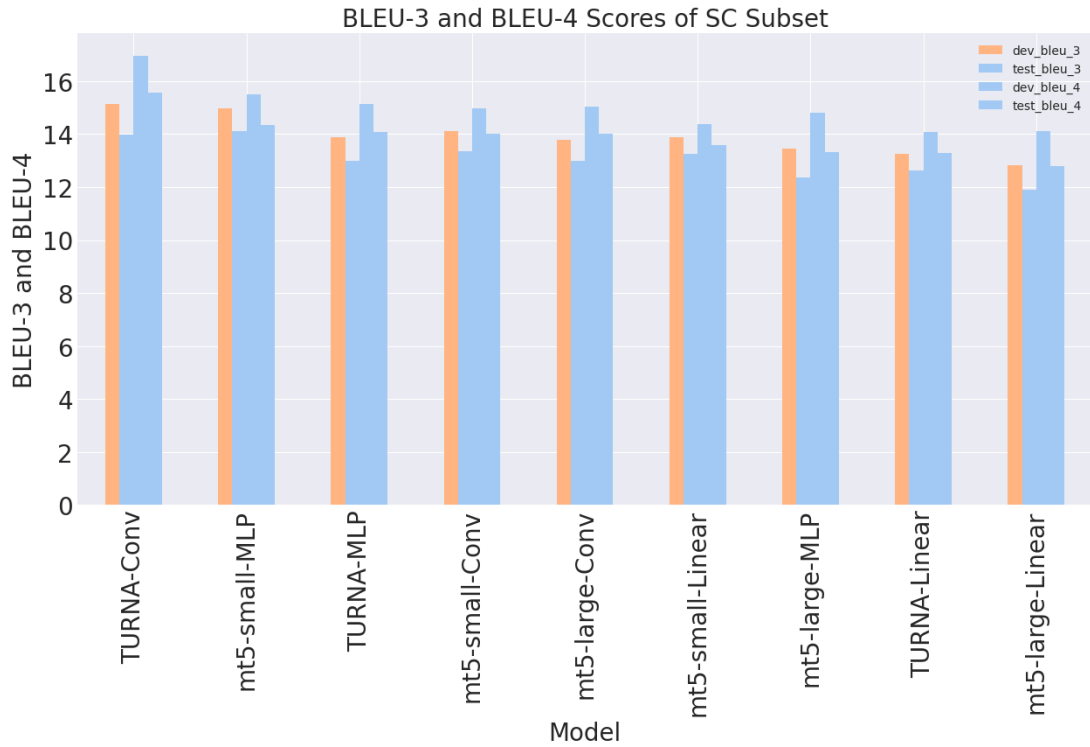


Figure 13. BLEU-3 and BLEU-4 performance comparison by model type across SC development and test sets

As Zhang et al. (2024) observed, larger models tend to overfit on smaller datasets, leading to lower BLEU scores. This pattern is reflected in the LC dataset, where mT5-small outperforms mT5-large and TURNA models across all configurations. For instance, mT5-small with a Linear encoder achieves a BLEU-4 score of 2.19 on the SC test set without augmentation, compared to 0.16 for mT5-large. The smaller parameter count of mT5-small helps it generalize better in this case, avoiding overfitting to the limited data.

The long context and fewer instances in the LC dataset pose a challenge for all models, particularly larger and more complex ones like mT5-large. These models struggle to learn the long-context dependencies effectively due to the limited training instances. For example, mT5-large consistently underperforms relative to mT5-small in BLEU-4 scores across LC configurations. This pattern highlights the difficulty of aligning model complexity with dataset constraints, particularly for longer sequences requiring a broader context.

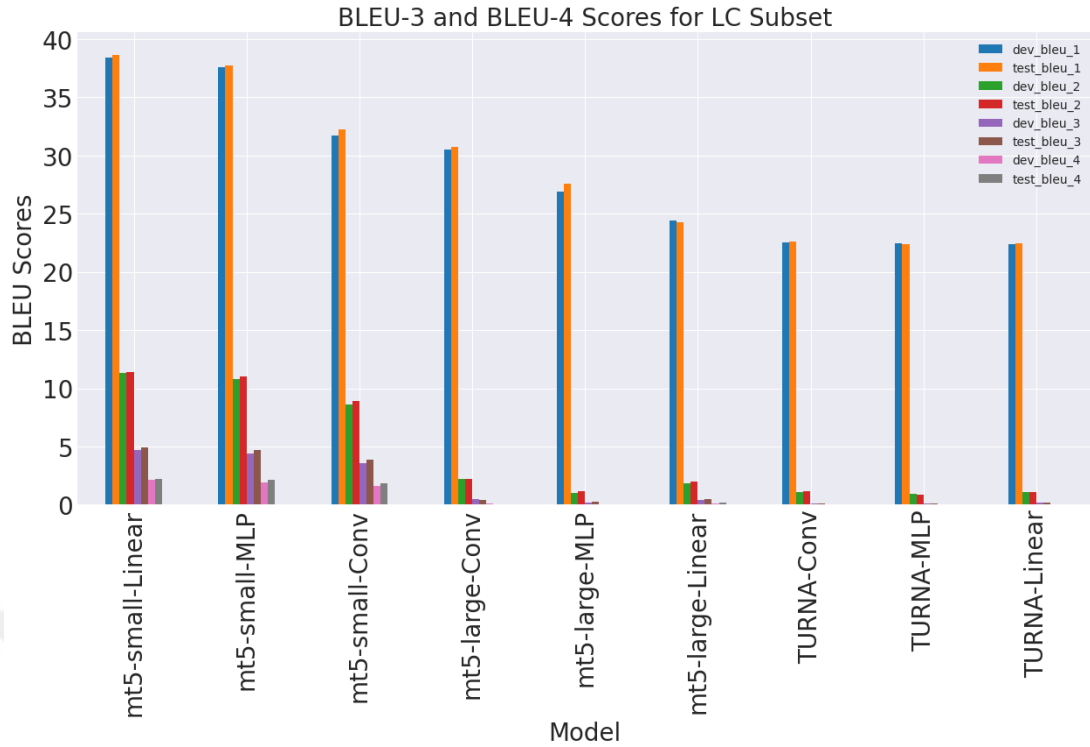


Figure 14. BLEU-4 performance comparison by model type across LC development and test sets

Overall, these findings underscore the importance of balancing model size and architecture with dataset characteristics. TURNA excels in SC due to its language-specific design and the availability of numerous short-context instances, while mT5-small demonstrates robustness in both SC and LC datasets, benefiting from its smaller parameter count. Conversely, mT5-large’s performance reflects the challenges of overfitting in SC and underfitting in LC. This shows the need for training strategies in diverse dataset contexts. That’s why we have added our augmentation strategies to our experimental methodologies.

6.2 Comparison between encoder types

Initially, it is important to note that the SC and LC datasets require distinct analyses for encoder comparisons. The LC dataset is not an ideal ground for encoder-type comparisons due to an underfitting pattern, as discussed in the model comparison

section. However, the performance increase observed in frame-level encoders, such as MLP and Linear, indicates that frame-level information plays a crucial role in extracting meaningful features. Unlike the Conv1D encoder, which incorporates temporal pooling and limits information extraction from larger sequences, frame-level encoders are better suited for datasets requiring fine-grained temporal details.

As observed in Figure 15, the LC dataset generally produces lower BLEU scores compared to the SC dataset due to differences in word distribution. The LC dataset’s longer average token distribution results in a descending BLEU score order, as higher n-gram matches become less likely. This phenomenon aligns with the dataset’s average sentence length being greater than 4 tokens, as reflected in the Table 7 and Table 8.

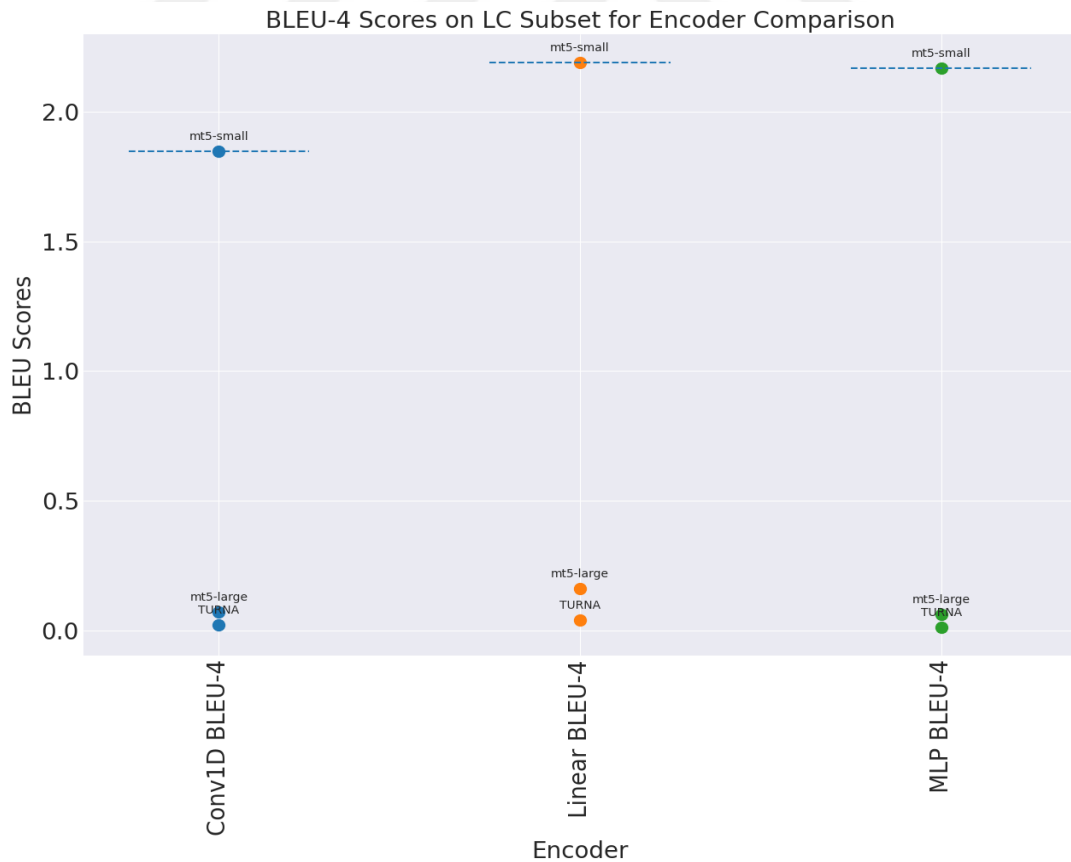


Figure 15. BLEU-4 performance comparison by encoder type across LC development and test sets

This underfitting pattern is further supported by the effect of augmentation on the LC dataset. Unlike SC, where augmentation introduces noise, it significantly improves performance on LC, indicating that the dataset benefits from the added diversity. For example, mT5-small with a Linear encoder achieves a BLEU-4 score of 2.64 on the test set with augmentation, compared to 2.19 without it. This improvement underscores how augmentation mitigates underfitting in long-context scenarios by providing richer training examples.

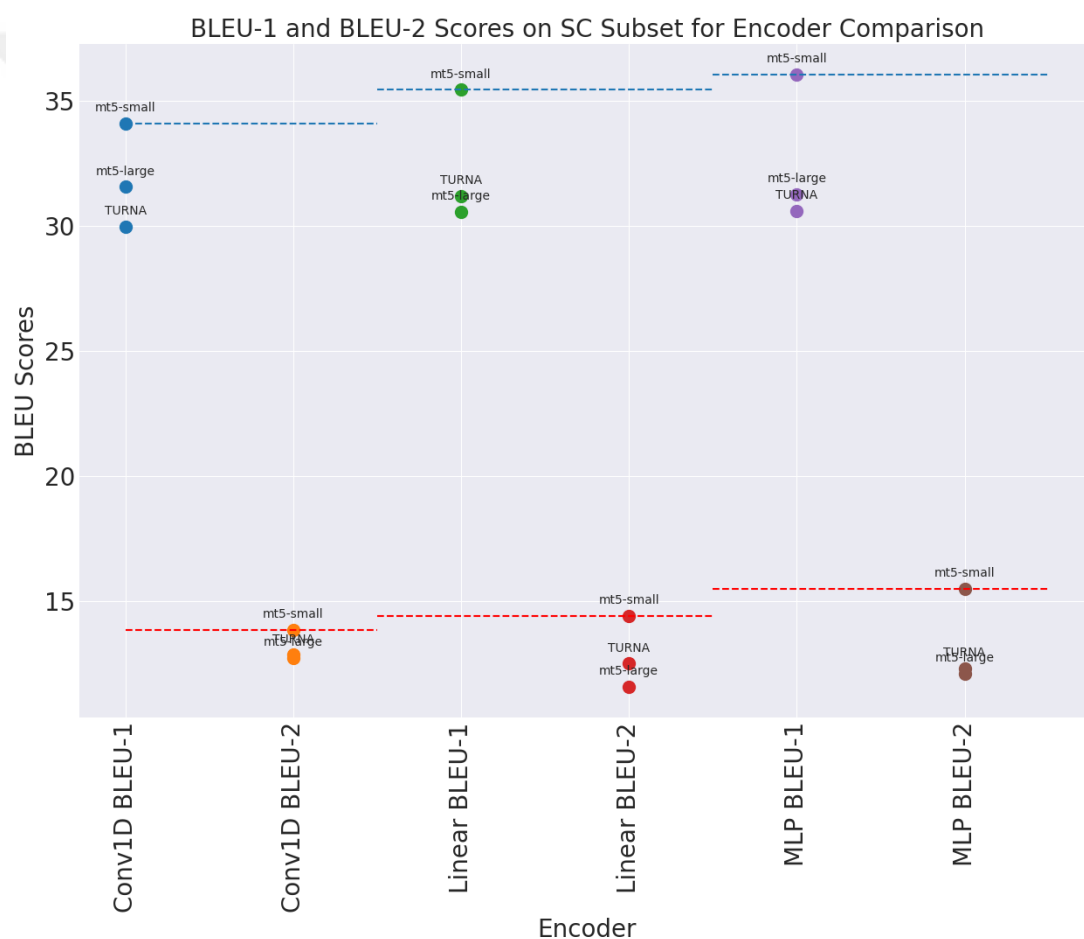


Figure 16. BLEU-1 and BLEU-2 performance comparison by encoder type across SC development and test sets

The analysis of the SC subset reveals notable differences in model performance between "single-word SC" and "multi-word SC" sequences,

highlighting the interplay between sequence length, encoder architecture, and dataset characteristics.

For "single-word SC" sequences, frame-level encoders, such as Linear and MLP, perform comparably or better than Conv1D encoders as shown in Figure 16. For instance, on the SC development set, the mT5-small model with an MLP encoder achieves a BLEU-1 score of 16.33 and a BLEU-2 score of 14.97, outperforming the Conv1D encoder, which achieves 14.69 and 14.13 for the same metrics. This trend can be attributed to the pooling mechanism in Conv1D encoders. While effective for temporal representation, the encoder limits the ability to capture such dense information in very short sequences. Another finding shows that the For "single-word SC" sequences, frame-level encoders, such as Linear and MLP, perform comparably or better than Conv1D encoders as shown in Figure 16. For instance, on the SC development set, the mT5-small model with an MLP encoder achieves a BLEU-1 score of 16.33 and a BLEU-2 score of 14.97, outperforming the Conv1D encoder, which achieves 14.69 and 14.13 for the same metrics. This trend can be attributed to the pooling mechanism in Conv1D encoders. While effective for temporal representation, the encoder limits the ability to capture such dense information in very short sequences. Another finding shows that the more complex. encoder MLP too surpasses the performance of the Linear encoder, which further highlights that the extraction of frame-level information is required in low-resource settings.

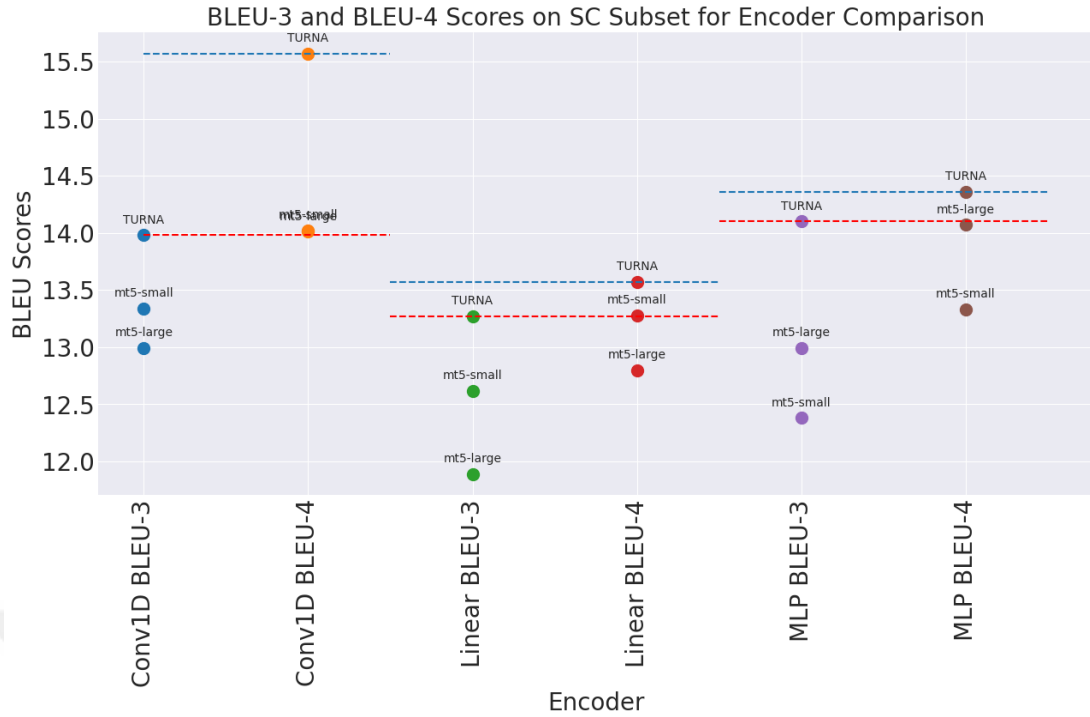


Figure 17. BLEU-3 and BLEU-4 performance comparison by encoder type across SC development and test sets

In contrast, for "multi-word SC" sequences, Conv1D encoders demonstrate superior performance, as they better represent the temporal dependencies inherent in longer sequences, as demonstrated in Figure 17. For example, the TURNA model with a Conv1D encoder achieves a BLEU-4 score of 16.95 on the SC development set, surpassing both the Linear (14.09) and MLP (15.50) setups. This suggests that when sufficient training data is available, Conv1D encoders can more effectively model sequential information, which is crucial for processing multi-word sequences.

These findings underscore the importance of the encoder architecture to the characteristics of the dataset. Frame-level encoders are well-suited for capturing the dense semantic content of shorter sequences with fewer resources, while Conv1D encoders excel at leveraging temporal relationships in longer sequences when provided with enough training data.

6.3 Addition of paraphrasing and pose augmentation in training

6.3.1 Paraphrasing

Paraphrasing introduces variability into the dataset by aligning a single sign with multiple paraphrased sequences, which can introduce noise into the training process and result in lower BLEU scores during evaluation; thus, we will not report the results of these experiments. This phenomenon reflects the inherent complexity of representing a one-to-many relationship between sign language input and its textual translations. Yet, paraphrasing is a necessary component in our translation system for calculating BLEU scores. The paraphrasing tool gives us better reference sequences that are more robust to the language variation seen in Turkish, where the change in model performance is given in Figure 18. By providing the idiosyncratic variations of possible sequences, our BLEU-3 and BLEU-4 metrics are more reliable for a language like Turkish. The discussions on paraphrasing will be provided in Section 7.2.

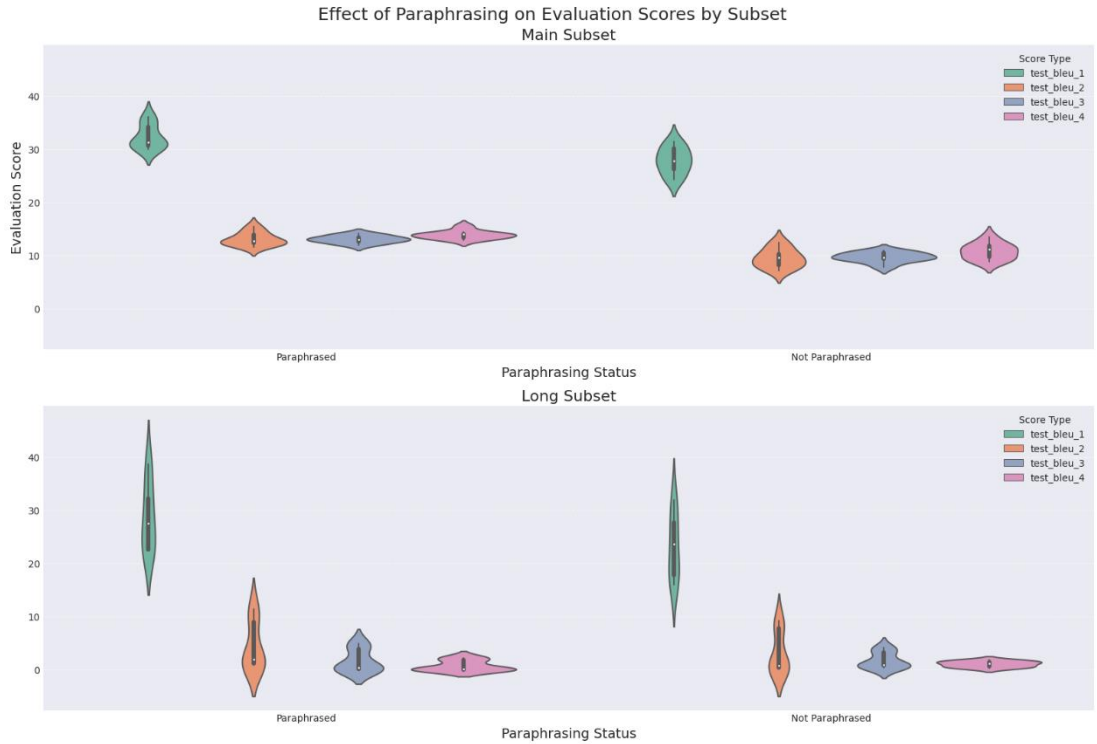


Figure 18. Comparison of BLEU-1 to BLEU-4 scores between paraphrased and non-paraphrased SC and LC test sets

In the SC subset, the differences in BLEU-1 and BLEU-2 scores are minimal, indicating that paraphrasing has a limited impact on unigram and bigram precision. However, as we move to higher n-grams (BLEU-3 and BLEU-4), the difference becomes more pronounced. This trend suggests that paraphrasing introduces variations in word order, which enhances n-gram matches and improves the model’s ability to handle syntactic flexibility and diversity. We will discuss this in detail in Section 7.2.

In contrast, the LC subset offers a less descriptive view of the impact of paraphrasing. The BLEU-4 scores remain very low, typically ranging between 2 and 0.5, regardless of the paraphrasing status. This is likely due to the LC subset’s inherent challenges, such as underfitting and the inability to effectively model long contexts, which paraphrasing alone cannot address.

Overall, while paraphrasing demonstrates benefits for both subsets by improving higher-order n-gram matching, its influence is limited in the LC subset, where low scores reflect broader challenges unrelated to paraphrasing.

6.3.2 Augmentation

Augmentation has markedly different effects on the SC and LC subsets, as reflected in the scores presented in Figures 19 and 20. In the SC subset, augmentation shows lower BLEU scores in all configurations. For example, mT5-small with a Linear encoder shows a BLEU-4 score of 2.16 on the development set without augmentation, which decreases to 2.53 when augmentation is applied. We argue that this drop can be attributed to the limited number of temporal sequences in the SC dataset. With an average of 3 seconds per video, the augmented data likely

introduces noise rather than improving performance, disrupting the sparse spatio-temporal information.

In contrast, augmentation improves performance on the LC dataset, reducing underfitting phenomena by its data distribution and modeling challenges. The consistent increase across all encoder and model variations suggests two key factors.

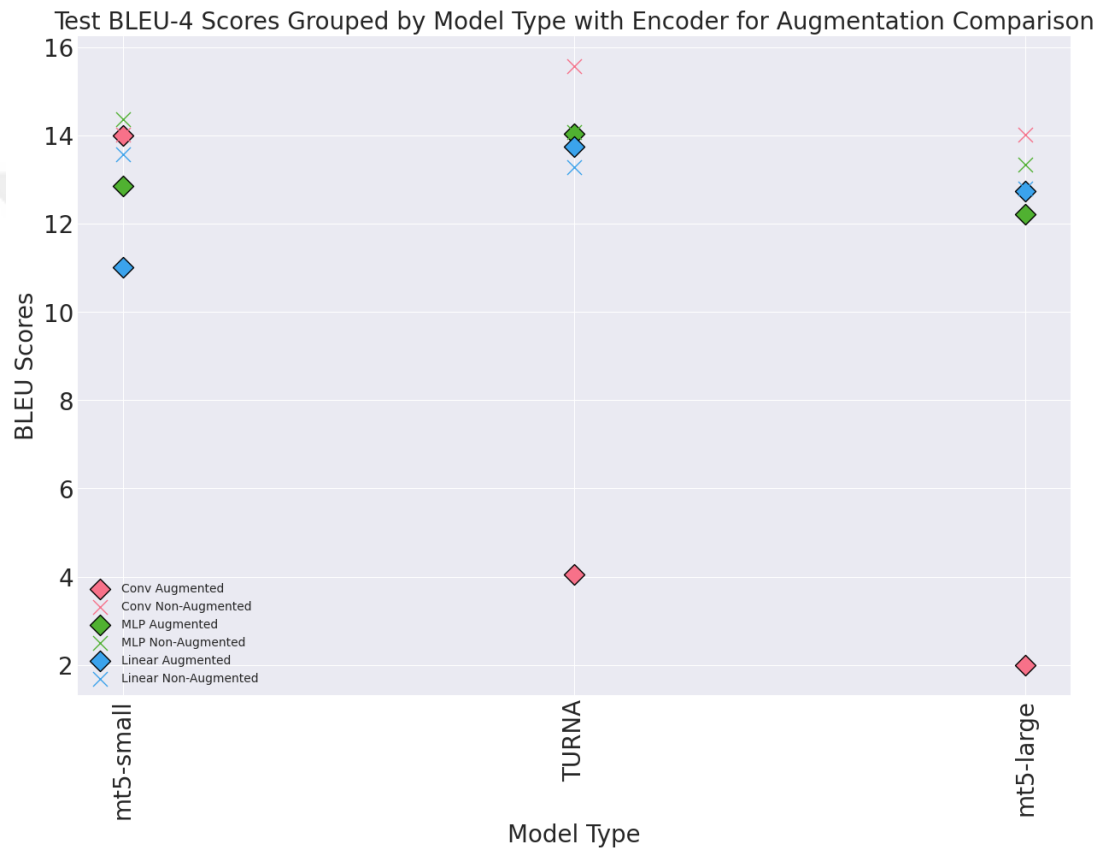


Figure 19. BLEU-4 performance comparison on effect of augmentation across model and encoder type configurations for SC subset

First, the improvement addresses an underfitting pattern in the LC data, which arises because the dataset is provided with a similar number of instances as SC, despite requiring alignment across longer contexts. For example, in the Linear encoder setup, mT5-small achieves a BLEU-4 score of 2.64 on the test set with augmentation compared to 2.19 without it. This consistent improvement underscores how

augmentation is required for long-context alignment with controlled variation for pose information.

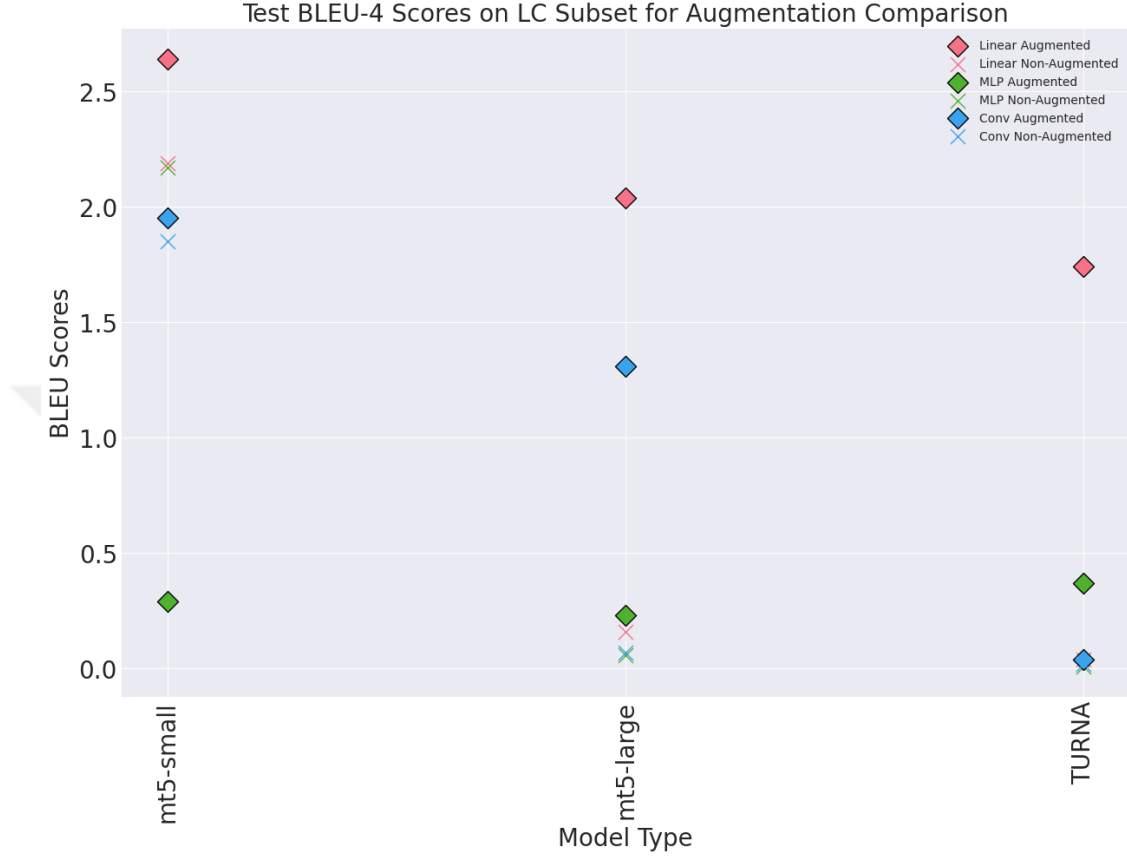


Figure 20. BLEU-4 performance comparison on effect of augmentation across model and encoder type configurations for LC subset

6.3.3 Overall experimental findings

The results of our experiments; summarized in Tables A1, A2, A3 and A4; show the overall relationship between dataset characteristics, model architectures, and augmentation strategies. Across both SC and LC subsets, several key findings emerge.

In the SC subset, TURNA consistently outperforms other models, leveraging its language-specific pretraining and its ability to handle the high frequency of shorter-context instances. This underscores the importance of language-specific

encoders for tasks requiring precise linguistic alignment. The mT5-small model also demonstrates competitive performance, often surpassing mT5-large, suggesting that smaller models are better suited for datasets with limited complexity. This aligns with prior findings, such as those by Zhang et al. (2024), reinforcing the suitability of compact models in scenarios where overfitting poses a risk. Additionally, the effect of paraphrasing in the SC subset is notable; while BLEU-1 and BLEU-2 scores show minimal change, higher n-gram scores, such as BLEU-3 and BLEU-4, benefit significantly. This improvement is likely due to paraphrasing, introducing word order variations, enhancing group matching, and model adaptability.

The LC subset, in contrast, presents unique challenges due to its longer average sequence lengths and smaller dataset size. Augmentation plays a crucial role in this context, consistently improving performance by diversifying the training data.

Despite these improvements, smaller models like mT5-small continue to outperform larger ones, with mT5-large and TURNA struggling to effectively capture long-context dependencies, a clear indication of underfitting. Furthermore, paraphrasing has a limited impact on BLEU-4 scores in the LC subset, which remains low, suggesting that data scaling is more critical for handling long-context tasks effectively.

Encoders also play a significant role in performance outcomes. Frame-level encoders, such as Linear and MLP, are better suited for shorter sequences, particularly in the "single-word SC" subset, where they excel in capturing dense semantic content. In contrast, Conv1D encoders demonstrate superior performance in the "multi-word SC" subset, as they effectively model temporal dependencies when sufficient training data is available. However, in the LC subset, where sequences are

significantly longer and instances fewer, all encoder types struggle due to underfitting and the inherent challenges of capturing long-context dependencies. Overall, the experiments highlight the necessity of aligning model architectures, dataset characteristics, and preprocessing strategies for optimal performance. Language-specific models like TURNA excel in handling short-context tasks in the SC subset, while smaller, more generalizable models like mT5-small perform robustly across both SC and LC subsets. The results also emphasize the importance of augmentation and translation length constraints, particularly for low-resource and long-context scenarios, where data diversity and architectural adaptability are key to addressing inherent challenges.

CHAPTER 7

DISCUSSION

7.1 Input modality

Our input modality for the representation of sign language sequences is the skeletal information extracted from sign language video input using the MediaPipe pose estimation model, specifically the BlazePose GNUM 3D model. This model performs 2D landmark detection with automatic 3D projection, leveraging learned representations from its training without relying on heuristic processes (Bazarevsky et al., 2020). While this approach provides a computationally efficient and structured representation of the signer's movements, several challenges arise from the variability and quality of the input data.

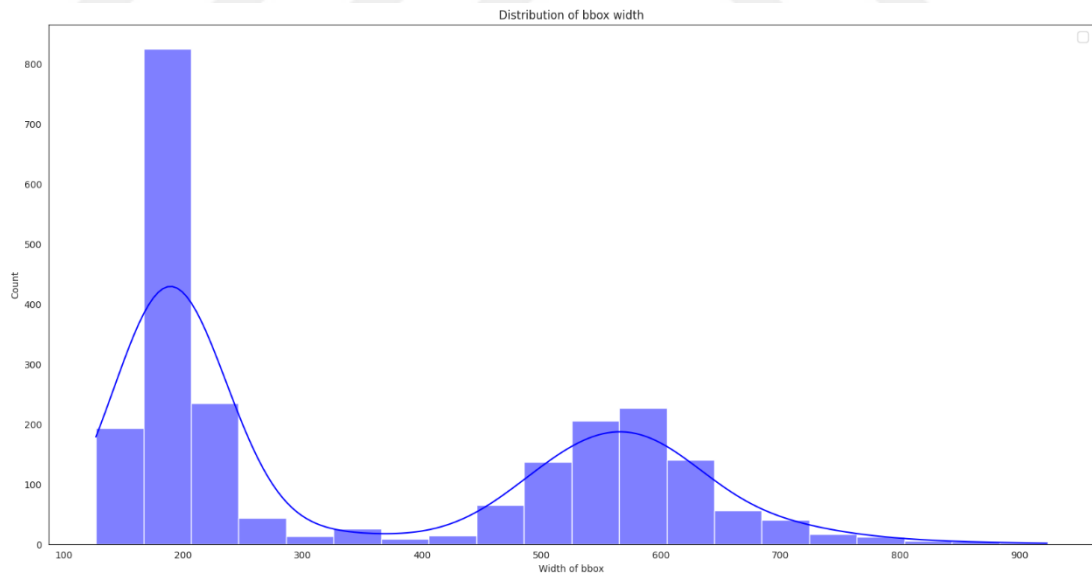


Figure 21. The width distribution of annotated bounding boxes across videos

The first challenge stems from the variability in dataset quality and visual resolution.

As discussed in Chapter 3.3.1, the bounding box corresponding to the signer's

location differs significantly across videos. This variation occurs due to the differing height and width pixel values, which depend on both the overall quality of the videos (ranging from 360p to 1080p) as seen in Figure 21 and the maximum size of the embedded sign video content. These inconsistencies affect the accuracy of pose estimation, making it difficult to maintain uniform performance across clips. Additionally, even in high-resolution contexts, the pose estimation model shows error propagation under challenging conditions, such as occlusion caused by camera angles or hand-face interactions, resulting in noisy or missing skeletal outputs (Gueuwou et al., 2024; Moryossef et al., 2021).

While it is difficult to directly determine the effect of noisy pose information on model performance, the implementation of our data augmentation techniques mitigates some of these issues by improving the robustness of the models. These augmentations, such as random frame drops and cue masking, simulate real-world noise and enhance the model's ability to generalize under suboptimal conditions.

The possibility of noisy landmarks is a well-documented issue in the SLT literature, with various solutions proposed. One common approach is to use image-based models (Albanie et al., 2021; Gueuwou et al., 2024; Rust et al., 2024) that encode spatial and contextual information or to employ multiple representation models with feature fusion (Camgoz et al., 2020a; Chen et al., 2022) to integrate skeletal information with visual features. These approaches aim to compensate for the limitations of skeletal data alone.

Finally, it is important to note that the challenges associated with skeletal information extend beyond noisy outputs from low-quality video inputs. Skeletal representations inherently lack certain critical aspects of sign language, such as non-manual markers, hand shapes, and subtle finger movements, which are essential for

accurate linguistic representation. These limitations underscore the need for further advancements in input modalities to capture the full complexity of sign language for SLT systems.

7.2 Paraphrasing and linguistic difference

In this work, we employed a paraphrasing tool to evaluate the performance of our models on paraphrased translations of the given sign sequences. This approach was motivated by the need to address discourse-level variations in Turkish, a language with significant typological differences and idiosyncratic characteristics in its utterances.

Instead of relying on a paraphrasing tool, an alternative approach, such as translating the target text into English, could be considered. This method addresses not only the challenges posed by grammatical variation in Turkish but also leverages language-specific models in English, which we argue deliver superior translation performance. However, due to the large dataset size, these translations must be performed automatically, limiting our ability to control for potential mistranslations.

A further limitation of this approach lies in handling language-specific expressions, including idiomatic phrases and sociolinguistic variations common in everyday language. While this method might improve translation quality for datasets with controlled vocabulary and domain-specific language, we hypothesize that, in our case, applying it to the target translations would still affect overall translation quality.

Therefore, we argue that paraphrasing serves as a better tool for mitigating these variations, enabling a more nuanced and comprehensive evaluation of the

model's performance. However, the usage of the Turkish language as the target language results in a challenge in BLEU metric evaluation.

The limitations of BLEU, which relies on exact word matches, are particularly evident in Turkish due to its morpho-syntactic characteristics and agglutinative properties. This often leads to low BLEU scores, even when translations are semantically accurate. Incorporating paraphrasing into our evaluation shifts the focus from surface-level matches to assessing the model's ability to capture the semantic content of Turkish Sign Language (TID) inputs.

However, automatic paraphrasing is not enough to cover the typological differences between Turkish and TID. Thus, we need to address what possible typological differences result in these mismatches. By analyzing discrepancies between machine-generated translations and human-authored references, we identify recurring phenomena that highlight challenges and inform potential improvements in translation quality. This analysis points out the need for evaluation metrics that prioritize semantic fidelity over exact lexical correspondence, especially for morphologically complex languages like Turkish, as well as the possible limitations of the sign language translation systems

7.2.1 Difference in Word-Order between TID and Turkish

An example of this is the effect of "word order" variation, which has a functional usage and makes the various usages possible in Turkish (Erguvanlı & Taylan, 1984). Also, TID enables word order variation in some cases depending on the argument structure (Arık, 2020; Gökgöz & Arık, 2011). However, there is no literature on the translation strategies reflecting variation from both languages.

- (1) a. *Gel* *benimle*
 Come with.me
 “Come with me”
- a. *Benimle* *gel*
 Come with.me
 “Come with me”

This example of "Gel benimle" and "Benimle gel," in (1) shows the effect of scrambling where metrics like BLEU yield zero matching since it relies on exact n-gram matching. Although both sentences convey the same meaning, their word order differs due to Turkish's free word order characteristics.

- (2) a. *Attım* *içeri*
 Throw.PAST.1Sg inside
 ‘I threw (it) inside’
- b. *Onu* *attım* *içeri*
 It Throw.PAST.1Sg inside
 ‘I threw it inside’

The examples "Attım içeri" (I threw [it] inside) and "Onu içeri attım" (I threw it inside) in (2) illustrate the linguistic phenomena of both scrambling and object drop in Turkish and their impact on translation evaluation. In "Attım içeri," scrambling is evident as the adverbial "içeri" (inside) follows the verb "Attım" (I threw), deviating from the more neutral order seen in "Onu içeri attım." Additionally, "Attım içeri"

demonstrates pro-drop (Öztürk, 2006), a common phenomenon in Turkish. Here, the third-person object pronoun "Onu" (it) is omitted but understood from the context.

These word-order-based variations and dropped items pose significant challenges for translation evaluation, particularly when using metrics like BLEU, which rely heavily on exact n-gram matching. Scrambling alters word order, and object drop results in differing token counts, leading to low BLEU scores despite being semantically accurate translations.

7.2.2 Difference in morphological markers between TID and Turkish

(3) a. *Duydum*

Hear.PAST.1Sg

‘I heard’

b. *Dinliyorum*

Listen.PRES.CONT.1Sg

‘I am listening’

The distinction between "duydum" (I heard) and "dinliyorum" (I am listening) highlights the linguistic intricacies of Turkish Sign Language (TID) and the challenges it presents for sign language translation (SLT). (Kubuş, 2008) notes that TID marks plural forms on nouns and adjectives, albeit infrequently. Moreover, TID employs a complex system of verbs and distinguishes between two primary aspects: completive and progressive/continuative. These distinctions are particularly relevant when analyzing verbs like "duymak" and "dinlemek," which are homophonous in TID.

(4) a. *Ne yapıyorsun?*

What do.PRES.CONT.2Sg

‘What are you doing?’

b. *Ne yaptın?*

What do.PAST.CONT.2Sg

‘What are you doing?’

The progressive/continuative aspect in TID is often indicated through the repetition of a sign, which implies an ongoing or habitual action. For instance, "dinliyorum" (I am listening) could involve a repeated gesture to signify the act of continuous listening, whereas "duydum" (I heard) might be expressed as a single, abrupt sign to denote a completed hearing event. However, this repetition varies significantly among signers, introducing variability in how the aspect is conveyed. This signer-dependent variation can lead to challenges in both comprehension and translation, as the absence of consistent repetition may cause ambiguities in determining whether the intended meaning is progressive/continuative or complete. The mismatch between the model prediction and actual translation occurs due to the morpho-syntactic difference arising from TID and Turkish, resulting in a 0 BLEU score for this example.

7.2.3 Referentiality in TID

- (5) a. *Özür* *dilerim*
What do.PRES.CONT.2Sg
'I am sorry'
- b. *Özür* *dilerim* *Esat*
What do.PAST.CONT.2Sg Esat
'I am sorry, Esat'

This error is caused by the unnecessary usage of a referentiality prominent form discussed in Keleş et al.'s (2023) work. When a new referent is introduced in the context, a signer first introduces a referent with a prominent form, for instance, a proper name or a full noun phrase, which usually also has, e.g., longer duration and an increase in signing space (Keleş, 2024). When the referent becomes old information, less prominent tools such as pronouns or zero arguments (like the dropped object discussed above) are used. Since "Esat" was used in the context before, a more referentially economical sentence should exclude it, and this will reduce the translation to "I am sorry" instead of "I am sorry, Esat". This example shows the alignment problem with the target translation and the actual articulated sentence. Since the dataset consists of interpretations of the spoken language sequence, translators' production of the sentence and caption do not match due to the nature of this dataset, affecting the BLEU score in word-level matching.

7.2.4 Idiosyncratic expressions

Below, we have provided a few idiosyncratic expressions that are semantically close but yield zero matches when evaluated with a BLEU score. This limitation arises from the inherent variability in language and translation, particularly in contexts involving complex idiosyncratic expressions. Several factors contribute to this issue, including the translator's style of translation and the discourse context of the translation.

- (6) a. *Gel* *bakalım*
Come let's-see.ASP.1Sg
'Come, let's see.'
- b. *Hadi* *bakayım*
Come-on see.1Sg
'Come, let's see.'

- (7) a. *Babam* *için*
Father.POSS.1Sg for
'For my father'
- b. *Babama*
Father.POSS.1Sg-DAT
'To my father'

(8) a. *Anne*.

Mother

‘Mother’

b. *Anacağım*

Mother.DIM.Poss.1Sg

‘My dear mother’

(9) a. *Eyvallah* *Elif*

Thank-you Elif

‘Thank you, Elif’

b. *Sağolasın* *Elif*

Live-be.3Sg Elif

‘Be well, Elif (Intended: Thank you, Elif)’

In these cases, the BLEU score, commonly used for evaluating machine translations, may not fully capture this idiosyncrasy between expressions, as it primarily measures n-gram overlap without considering context or meaning. Therefore, a comprehensive error analysis, as demonstrated above, is crucial for understanding and improving translation quality.

CHAPTER 8

FUTURE WORK

This research has culminated in the development of the first open-source, large-scale dataset for TID-to-Turkish translation, marking it as the sixth substantial contribution to SLT datasets globally. Additionally, we have devised methodologies tailored to the unique linguistic properties of both Turkish and TID. The data collection and annotation processes, while incorporating manual efforts, have predominantly depended on automatic annotation techniques. Consequently, to output representative data of the linguistic integrity of both Turkish and TID, it is imperative to integrate comprehensive manual verification and annotation. Our primary objective moving forward is to establish a meticulously curated, manually annotated corpus.

8.1 Corpus creation

The creation of a high-quality corpus necessitates active involvement from the Deaf and hard-of-hearing communities. Translations should receive validation from Deaf translators and be annotated by multiple experts to achieve a high inter-annotator agreement, ensuring reliability and consistency. Currently, the dataset comprises direct translations without intermediary linguistic information such as gloss annotations, fingerspelling, or detailed phonological and morphological markers. Incorporating these elements is essential for broadening the dataset's applicability across various linguistic and computational tasks.

8.1.1 Sign language recognition

Existing TID datasets primarily focus on isolated sign instances. However, lexical items exhibit variations in continuous signing contexts. To advance SLR technologies, it is crucial to annotate gloss-level information, capturing signs in diverse contexts. This approach will facilitate the development of applications that translate glosses into text, thereby enhancing the benchmarking of SLR systems.

8.1.2 Adding linguistic information

To create a multidisciplinary dataset, we aim to include multiple layers of linguistic annotation, enabling linguists to build extensive corpora for statistical analysis while also supporting researchers in exploring the relationship between linguistic information and the performance of computational systems in SLR/SLT.

8.1.2.1 Phonological annotation

Documenting features such as handshape, location, movement, and orientation will be an important task (Brentari, 1998). This process will be supported by our Automatic Sign Feature Extraction (AS-FE) tool (Şahin & Gökgöz, 2024), facilitating semi-automatic annotation that is subsequently refined through manual review. This methodology aims to produce a large-scale dataset within a reduced timeframe.

8.1.2.2 Morphological annotation

Integrating gloss annotations with morphological details is vital, as glossing often overlooks inflectional nuances. TID exhibits various morphological markers, including verb agreement for number and person, noun pluralization, possessive and

case markings, and classifier constructions based on semantic properties. Detailed morphological annotation will provide insights into these complex structures. In addition to refining our dataset, we aim to enhance our models by addressing key components that influence translation performance.

8.2 Improvements on architecture

8.2.1 Sign language representation

Currently, we utilize normalized pose information to represent sign language videos. However, the literature offers various alternative representations that are used in the SLT, such as CNN-based Pretrained 2D, 3D ResNet, or I3D embeddings, or more recent transformer-based pre-trained vision encoders such as Vision Transformers (ViT) and DINO. We are planning to explore our experiments further with these representations.

8.2.2 Feature encoders

Our current approach employs linear layers, multi-layer perceptrons, and 1D convolutional networks to encode pose information into token space. We plan to test the combinations of the encoders, such as the MLP layer, followed by Conv1D to further maximize the performance of sign encoders.

Another encoder we plan to use comes from the utilization of a popular architecture in SLR and SLT systems, namely graph neural networks (GNNs), which can effectively model the intricate relationships between different body parts in sign language. For instance, hierarchical spatiotemporal graph neural networks have been proposed to capture both local and global dependencies in sign language data. We

plan to experiment with GNN-based architectures to potentially enhance our feature encoding process.

8.2.3 LLM architecture

Our current framework utilizes encoder-decoder architectures, namely T5. However, decoder-only architectures that provide generation capabilities in both an auto-regressive and causal manner, such as those employed in models like LLaMA, have achieved state-of-the-art results in various natural language processing tasks. By integrating training techniques like low-rank adaptation (LoRA) and instruction tuning, we aim to improve our models' performance in sign language translation.

CHAPTER 9

CONCLUSION

This thesis presents a comprehensive study on Turkish Sign Language (TID) translation, contributing to both the creation of a novel dataset and the exploration of advanced methodologies in sign language translation (SLT). The proposed dataset is the first large-scale, open-source dataset for TID, comprising over 500 hours of video footage aligned with Turkish translations. This dataset aims to provide a resource that supports the development of models specific to the Turkish language and sign language while also being scalable and reproducible using publicly available YouTube content.

The work evaluates the performance of several pre-trained language models, including mT5-small, mT5-large, and TURNA, in the context of SLT. The TURNA model demonstrates superior performance due to its Turkish-specific pretraining, effectively capturing the information in the Turkish language for SLT. The results highlight that, for SLT tasks, the alignment between the model’s pretraining data and the target language is more critical than the size of the model. Notably, mT5-small outperforms its larger counterpart, mT5-large, reflecting that smaller models may generalize better under constrained datasets, supported by findings of Zhang et al. (2024).

Our work further investigates various architectural choices in the sign embedding layer and its possible configurations for representing skeletal information extracted via pose estimation. The study highlights the challenges posed by input modalities, such as noisy pose information, variable video quality, and the limitations of skeletal representations in capturing the optimal projection layer for sign

language. Augmentation strategies, such as random frame skipping and cue masking, are introduced to improve robustness, though the results indicate the need for further refinements.

Paraphrasing-based evaluation is carried out due to the lack of BLEURT for Turkish, which underscores the limitations in current evaluation protocols for low-resource languages, emphasizing the necessity for future advancements in this area.

This thesis also contextualizes its contributions within the broader SLT literature, tracing the evolution from early statistical machine translation techniques to the use of modern transformer-based architectures and pretraining strategies. The work validates findings in recent literature, such as the importance of large-scale data and pretraining, while offering a comparative analysis of models and encoders specific to SLT tasks.

In summary, this research represents a significant step forward for TID translation, introducing the largest dataset for Turkish SLT and the sixth largest dataset proposed in SLT literature and providing detailed analyses of model architectures, pretraining strategies, and linguistic challenges. Future directions include the creation of manually annotated corpora, the incorporation of linguistic annotations for phonological and morphological properties, and the exploration of different architecture types for improved translation performance. This work lays a robust foundation for advancing SLT technologies for Turkish Sign Language and beyond.

APPENDIX A

OVERALL EXPERIMENTAL RESULTS

Table A1. Experimental Results for Sign Language Translation on SC Dataset (Development and Test Scores)

Model Type	Sign Encoder	Augment	Dev				Test			
			BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU-1	BLEU-2	BLEU-3	BLEU-4
mT5-small	Linear	False	35.62	14.95	13.88	14.39	35.43	14.40	13.27	13.57
mT5-large	Linear	False	30.78	12.36	12.84	14.10	30.56	11.56	11.89	12.80
TURNA	Linear	False	31.53	13.17	13.24	14.09	31.20	12.50	12.62	13.28
mT5-small	Conv1D	False	34.72	14.69	14.13	14.96	34.10	13.85	13.34	14.02
mT5-large	Conv1D	False	32.11	13.53	13.80	15.04	31.55	12.72	12.99	14.01
TURNA	Conv1D	False	30.86	13.92	15.13	16.95	29.97	12.85	13.98	15.57
mT5-small	MLP	False	36.78	16.33	14.97	15.50	36.03	15.50	14.10	14.36
mT5-large	MLP	False	31.84	12.99	13.47	14.81	31.27	12.09	12.38	13.33
TURNA	MLP	False	30.96	13.17	13.87	15.13	30.60	12.31	12.99	14.07

Table A2. Experimental Results for Sign Language Translation on SC Dataset with Augmentation (Development and Test Scores)

Model Type	Sign Encoder	Augment	Dev				Test			
			BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU-1	BLEU-2	BLEU-3	BLEU-4
mT5-small	Linear	True	33.89	13.34	11.86	11.92	33.33	12.60	11.12	11.01
mT5-large	Linear	True	31.84	12.62	12.80	13.81	31.35	11.53	11.75	12.73
TURNA	Linear	True	31.25	13.11	13.54	14.68	30.90	12.43	12.86	13.74
mT5-small	Conv1D	True	32.30	13.44	13.48	14.50	32.39	13.28	13.20	13.99
mT5-large	Conv1D	True	20.08	03.04	2.61	2.53	19.76	2.69	2.18	2.00
TURNA	Conv1D	True	25.42	5.79	5.20	5.30	25.01	4.88	4.16	4.05
mT5-small	MLP	True	35.95	15.08	13.62	13.95	35.82	14.73	12.99	12.85
mT5-large	MLP	True	32.03	12.62	12.50	13.23	31.35	11.69	11.55	12.21
TURNA	MLP	True	32.42	13.84	13.94	14.97	31.62	13.09	13.20	14.03

Table A3. Experimental Results for SLT on LC Dataset (Development and Test Scores)

Model Type	Sign Encoder	Augment	Dev				Test			
			BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU-1	BLEU-2	BLEU-3	BLEU-4
mT5-small	Linear	False	38.42	11.32	4.73	2.16	38.62	11.36	4.89	2.19
mT5-large	Linear	False	24.41	1.82	0.43	0.14	24.29	1.98	0.51	0.16
TURNA	Linear	False	22.39	1.10	0.17	0.06	22.46	01.06	0.16	0.04
mT5-small	Conv1D	False	31.70	8.63	3.57	1.63	32.24	8.91	3.85	1.85
mT5-large	Conv1D	False	30.49	2.23	0.48	0.13	30.72	2.24	0.43	0.07
TURNA	Conv1D	False	22.55	1.12	0.12	0.02	22.58	1.16	0.14	0.02
mT5-small	MLP	False	37.61	10.83	4.41	1.91	37.76	10.99	4.73	2.17
mT5-large	MLP	False	26.89	0.99	0.19	0.05	27.54	1.13	0.23	0.06
TURNA	MLP	False	22.45	0.95	0.13	0.03	22.39	0.89	0.08	0.01

Table A4. Experimental Results for SLT on LC Dataset with Augmentation (Development and Test Scores)

Model Type	Sign Encoder	Augment	Dev				Test			
			BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU-1	BLEU-2	BLEU-3	BLEU-4
mT5-small	Linear	True	40.08	12.60	5.48	2.53	40.12	12.74	5.68	2.64
mT5-large	Linear	True	35.10	9.19	3.88	1.80	35.23	9.29	4.11	2.04
TURNA	Linear	True	31.58	6.56	2.80	1.52	31.94	6.93	3.10	1.74
mT5-small	Conv1D	True	36.40	9.89	04.01	1.66	36.89	10.13	4.31	1.95
mT5-large	Conv1D	True	31.94	6.87	2.66	1.22	32.33	07.03	2.85	1.31
TURNA	Conv1D	True	22.76	01.07	0.18	0.05	23.03	1.12	0.19	0.04
mT5-small	MLP	True	29.10	3.18	0.84	0.29	29.51	3.40	0.90	0.29
mT5-large	MLP	True	24.36	2.34	0.70	0.31	24.64	2.30	0.60	0.23
TURNA	MLP	True	23.89	1.78	0.64	0.43	23.99	1.65	0.54	0.37

REFERENCES

- Albanie, S., Varol, G., Momeni, L., Afouras, T., Chung, J. S., Fox, N., & Zisserman, A. (2020). Bsl-1k: Scaling up co-articulated sign language recognition using mouthing cues. *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, 35–53.
- Albanie, S., Varol, G., Momeni, L., Bull, H., Afouras, T., Chowdhury, H., Fox, N., Woll, B., Cooper, R., McParland, A., & Zisserman, A. (2021). BOBSL: BBC-Oxford British Sign Language Dataset. *arXiv*.
- Arik, E. (2020). An experimental approach to word order in turkish sign language. *Bilig*, 94, 75–96.
- Arvanitis, N., Constantinopoulos, C., & Kosmopoulos, D. (2019). Translation of sign language glosses to text using sequence-to-sequence attention models. *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 296–302.
- Athitsos, V., Neidle, C., Sclaroff, S., Nash, J., Stefan, A., Yuan, Q., & Thangali, A. (2008). The american sign language lexicon video dataset. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1–8.
- Aung, N. H., Thu, Y. K., Maung, S. S., Moe, S. Z., & New, H. M. (2020). Transfer learning based myanmar sign language recognition for myanmar consonants [*Doctoral dissertation, MERAL Portal*].
- Bahdanau, D., Cho, K., & Bengio, Y. (2016). Neural machine translation by jointly learning to align and translate. <https://arxiv.org/abs/1409.0473>
- Bauer, B., Nießen, S., & Hienz, H. (1999). Towards an automatic sign language translation system. *Proceedings of the international workshop on physicality and tangibility in interaction: towards new paradigms for interaction beyond the desktop, Siena, Italy*.
- Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). Blazepose: On-device real-time body pose tracking. <https://arxiv.org/abs/2006.10204>
- Boháček, M., & Hruží, M. (2022). Sign pose-based transformer for word-level sign language recognition. *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 182–191.

- Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., & Ringel Morris, M. (2019). Sign language recognition, generation, and translation: An interdisciplinary perspective. *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, 16–31. <https://doi.org/10.1145/3308561.3353774>
- Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., & Ringel Morris, M. (2019). Sign language recognition, generation, and translation: An interdisciplinary perspective. *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, 16–31. <https://doi.org/10.1145/3308561.3353774>
- Brentari, D. (1998). A prosodic model of sign language phonology. *A Bradford Book*.
- Brentari, D. (2010). *Sign languages*. Cambridge University Press.
- Buehler, P., Zisserman, A., & Everingham, M. (2009). Learning sign language by watching tv (using weakly aligned subtitles). *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2961–2968.
- Bull, H., Afouras, T., Varol, G., Albanie, S., Momeni, L., & Zisserman, A. (2021). Aligning subtitles in sign language videos. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11552–11561.
- Bungeroth, J., & Ney, H. (2004). Statistical sign language translation. *sign-lang@LREC 2004*, 105–108.
- Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., & Bowden, R. (2018). Neural sign language translation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7784–7793.
- Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020a). Multi-channel transformers for multi-articulatory sign language translation. In A. Bartoli & A. Fusiello (Eds.), *Computer vision – eccv 2020 workshops* (pp. 301–319). Springer International Publishing.
- Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020b). Sign language transformers: Joint end-to-end sign language recognition and translation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., & Bowden, R. (2018). Rwth-phoenix-weather 2014 t: Parallel corpus of sign language video, gloss and translation. *CVPR, Salt Lake City, UT*, 3, 6.

- Camgoz, N. C., Kındıroğlu, A. A., Karabüklü, S., Kelepir, M., Özsoy, A. S., & Akarun, L. (2016). Bosphorussign: A turkish sign language recognition corpus in health and finance domains. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 1383–1388.
- Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., & Sheikh, Y. A. (2019). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Cardoso Duarte, A., Palaskar, S., Ventura Ripol, L., Ghadiyaram, D., DeHaan, K., Metze, F., Torres Viñals, J., & Giró Nieto, X. (2021). How2sign: A large-scale multimodal dataset for continuous american sign language. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition: Virtual, 19-25 June 2021: proceedings*, 2734–2743.
- Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6299–6308.
- Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., & Yu, G. (2023). Executing your commands via motion diffusion in latent space. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18000–18010.
- Chen, Y., Zuo, R., Wei, F., Wu, Y., Liu, S., & Mak, B. (2022). Two-stream network for sign language recognition and translation. *Advances in Neural Information Processing Systems*, 35, 17043–17056.
- Chipman, H. A., George, E. I., McCulloch, R. E., & Shively, T. S. (2022). Mbart: Multidimensional monotone bart. *Bayesian Analysis*, 17(2), 515–544.
- Cihan Camgoz, N., Hadfield, S., Koller, O., & Bowden, R. (2017). Subunets: End-to-end hand shape and continuous sign language recognition. *Proceedings of the IEEE international conference on computer vision*, 3056–3065.
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 1, 886–893.
- De Coster, M., D'Oosterlinck, K., Pizurica, M., Rabaey, P., Verlinden, S., Van Herreweghe, M., & Dambre, J. (2021). Frozen pretrained transformers for neural sign language translation. *18th Biennial Machine Translation Summit (MT Summit 2021)*, 88–97.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Association for Computational Linguistics. doi:10.18653/v1/N19-1423.

- Dreuw, P., Deselaers, T., Keysers, D., & Ney, H. (2006). Modeling image variability in appearance-based gesture recognition. *ECCV workshop on statistical methods in multi-image and video processing*, 7–18.
- Erguvanli, E. E., & Taylan, E. E. (1984). *The function of word order in turkish grammar* (Vol. 106). Univ of California Press.
- Fang, G., & Gao, W. (2002). A sm/hmm system for signer-independent continuous sign language recognition. *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, 312–317.
- Fayyazsanavi, P., Anastasopoulos, A., & Košecká, J. (2024). Gloss2text: Sign language gloss translation using llms and semantically aware label smoothing. *arXiv preprint arXiv:2407.01394*.
- Fenlon, J. (2010). *Seeing sentence boundaries: The production and perception of visual markers signalling boundaries in signed languages* [Doctoral dissertation, UCL (University College London)].
- Forster, J., Schmidt, C., Hoyoux, T., Koller, O., Zelle, U., Piater, J. H., & Ney, H (2012). Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus. *LREC*, 9, 3785–3789.
- Giles, C. L., Kuhn, G. M., & Williams, R. J. (1994). Dynamic recurrent neural networks: Theory and applications. *IEEE Transactions on Neural Networks*, 5(2), 153–156.
- Gökgöz, K., & Arık, E. (2011). Distributional and syntactic characteristics of nonmanual markers in turkish sign language (turk isaret dili, tid). *MIT Working Papers in Linguistics*, 62, 63–78.
- Gong, J., Foo, L. G., He, Y., Rahmani, H., & Liu, J. (2024). Llms are good sign language translators. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18362–18372.
- Gueuwou, S., Du, X., Shakhnarovich, G., & Livescu, K. (2024). Signmusketeers: An efficient multi-stream approach for sign language translation at scale. *ArXiv preprint arXiv:2406.06907*.
- Gueuwou, S., Siake, S., Leong, C., & Müller, M. (2023, December). JWSign: A highly multilingual corpus of Bible translations for more diversity in sign language processing. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Findings of the association for computational linguistics: Emnlp 2023* (pp. 9907–9927). Association for Computational Linguistics. doi:10.18653/v1/2023.findings-emnlp.664.
- Gueuwou, S., Takyi, K., Müller, M., Nyarko, M. S., Adade, R., & Gyening, R.-M. O. M. (2023). Afrisign: Machine translation for african sign languages. *4th Workshop on African Natural Language Processing*. <https://openreview.net/forum?id=EHldk3J2xk>

- Gugger, S., Debut, L., Wolf, T., Schmid, P., Mueller, Z., Mangrulkar, S., Sun, M., & Bossan, B. (2022). Accelerate: Training and inference at scale made simple, efficient and adaptable.
- Guo, D., Wang, S., Tian, Q., & Wang, M. (2019). Dense temporal convolution network for sign language translation. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 744–750. doi:10.24963/ijcai.2019/105.
- Han, Y., Fan, X., Bhosale, R., Sundaram, R., & Liaw, J. (2021). Fusion-based spatiotemporal convolutions with constant temporal snapshots for sign language recognition. *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, 01–08.
- Hanke, T., König, S., Konrad, R., Langer, G., Barbeito Rey-Geißler, P., Blanck, D., Goldschmidt, S., Hofmann, I., Hong, S.-E., Jeziorski, O., Kleyboldt, T., König, L., Matthes, S., Nishio, R., Rathmann, C., Salden, U., Wagner, S., & Worsack, S. (2020). MEINE DGS. Öffentliches Korpus der Deutschen Gebärdensprache, 3. Release. doi:10.25592/dgs.meinedgs-3.0.
- Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). ArXiv preprint arXiv:1606.08415.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Hu, H., Zhao, W., Zhou, W., & Li, H. (2023). Signbert+: Hand-model-aware self supervised pre-training for sign language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9), 11221–11239.
- Hu, H., Zhao, W., Zhou, W., Wang, Y., & Li, H. (2021). Signbert: Pre-training of hand-model-aware representation for sign language recognition. *Proceedings of the IEEE/CVF international conference on computer vision*, 11087–11096.
- Huenerfauth, M. (2004). A multi-path architecture for machine translation of english text into american sign language animation. *Proceedings of the student research workshop at HLT-NAACL 2004*, 25–30.
- Itseez. (2015). Open source computer vision library.
- Jing, L., Song, X., Zu, X., Zheng, N., Zhao, Z., & Nie, L. (2024). Vk-g2t: Vision and context knowledge enhanced gloss2text. *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7860–7864.
- Joze, H. R. V., & Koller, O. (2018). Ms-asl: A large-scale data set and benchmark for understanding american sign language. *arXiv preprint arXiv:1812.01053*.

- Kan, J., Hu, K., Hagenbuchner, M., Tsoi, A. C., Bennamoun, M., & Wang, Z. (2022). Sign language translation with hierarchical spatio-temporal graph neural network. *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 3367–3376.
- Kelepir, M. (2020). A grammar of turkish sign language (tid). *SIGN-HUB Sign Language*.
- Keleş, O. (2024). *Discourse cohesion and phonetics in turkish sign language (tid): An experimental and computational approach* [Master's thesis, Boğaziçi University] [Unpublished master's thesis].
- Keleş, O., Atmaca, F., & Gökgöz, K. (2023). Reference tracking strategies of deaf adult signers in Turkish Sign Language [Publisher: Elsevier]. *Journal of Pragmatics*, 213, 12–35.
- Kim, S., Kim, C. J., Park, H.-M., Jeong, Y., Jang, J. Y., & Jung, H. (2020). Robust keypoint normalization method for korean sign language translation using transformer. *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 1303–1305.
- Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Ko, S.-K., Kim, C. J., Jung, H., & Cho, C. (2019). Neural sign language translation based on human keypoint estimation. *Applied sciences*, 9(13), 2683.
- Koller, O., Zargaran, S., Ney, H., & Bowden, R. (2016). Deep sign: Hybrid cnn-hmm for continuous sign language recognition. *BMVC*, 136–1.
- Kreutzer, J., Bastings, J., & Riezler, S. (2019). Joey NMT: A minimalist NMT toolkit for novices. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, 109–114. doi:10.18653/v1/D19-3019.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Kubuş, O. (2008). *An analysis of turkish sign language (tid) phonology and morphology* [Master's thesis, Middle East Technical University].
- Laptev, I., Marszalek, M., Schmid, C., & Rozenfeld, B. (2008). Learning realistic human actions from movies. *2008 IEEE conference on computer vision and pattern recognition*, 1–8.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436–444.

- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Li, D., Rodriguez, C., Yu, X., & Li, H. (2020). Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. *The IEEE Winter Conference on Applications of Computer Vision*, 1459-1469.
- Liang, Z., Li, H., & Chai, J. (2023). Sign language translation: A survey of approaches and techniques. *Electronics*, 12(12). doi:10.3390/electronics12122678.
- Lison, P., Tiedemann, J., & Kouylekov, M. (2018, May). OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, & T. Tokunaga (Eds.), *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*. European Language Resources Association (ELRA). <https://aclanthology.org/L18-1275>
- Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual instruction tuning. NeurIPS. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., et al. (2019). Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.
- Luong, M.-T. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Makaroğlu, B. (2021). Blend formation in turkish sign language: Are we missing the big picture? *Journal of Language and Linguistic Studies*, 17(1), 139–157.
- Makaroğlu, B., & Dikyuva, H. (2017, January). *Güncel Türk İşaret Dili Sözlüğü*. Ankara: Aile ve Sosyal Hizmetler Bakanlığı, <https://tidsozluk.aile.gov.tr>.
- Marshall, I., & Sáfár, É. (2002). Sign language generation using hpsg. *Proceedings of the 9th Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages: Papers*.
- Matthes, S., Hanke, T., Regen, A., Storz, J., Wörseck, S., Efthimiou, E., Dimou, A.L., Braffort, A., Glauert, J., & Safar, E. (2012). Dicta-sign-building a multilingual sign language corpus. *5th Workshop on the Representation and Processing of Sign languages: Interactions between Corpus and Lexicon. Satellite Workshop to the eighth International Conference on language Resources and Evaluation (LREC-2012)*.
- Moe, S. Z., Thu, Y. K., Thant, H. A., Min, N. W., & Supnithi, T. (2020). *Unsupervised neural machine translation between myanmar sign language and myanmar language* [Doctoral dissertation, MERAL Portal].

- Morrissey, S., Somers, H., Smith, R., Gilchrist, S., & Dandapat, S. (2010). Building a sign language corpus for use in machine translation.
- Moryossef, A., Tsochantaridis, I., Dinn, J., Camgoz, N. C., Bowden, R., Jiang, T., Rios, A., Muller, M., & Ebling, S. (2021). Evaluating the immediate applicability of pose estimation for sign language recognition. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3434–3440.
- Moryossef, A., Yin, K., Neubig, G., & Goldberg, Y. (2021). Data augmentation for sign language gloss translation. *arXiv preprint arXiv:2105.07476*.
- Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2023). A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.
- OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Zoph, B. (2024). Gpt-4 technical report. <https://arxiv.org/abs/2303.08774>
- Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Bojanowski, P. (2024). Dinov2: Learning robust visual features without supervision. <https://arxiv.org/abs/2304.07193>
- Orbay, A., & Akarun, L. (2020). Neural sign language translation by learning tokenization. *2020 15th IEEE international conference on automatic face and gesture recognition (FG 2020)*, 222–228.
- Othman, A., & Jemni, M. (2012). English-asl gloss parallel corpus 2012: Aslg-pc12. *Sign-lang@ LREC 2012*, 151–154.
- Özdemir, O., Baytaş, İ. M., & Akarun, L. (2023). Multi-cue temporal modeling for skeleton-based sign language recognition. *Frontiers in Neuroscience*, 17, 1148191.
- Özdemir, O., Kındıroğlu, A. A., Cihan Camgoz, N., & Akarun, L. (2020). BosphorusSign22k Sign Language Recognition Dataset. *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*.
- Öztürk, B. (2006). Null arguments and case-driven agree in turkish. *Minimalist essays*, 268–287.
- Öztürk, Ş., & Keles, H. Y. (2024). E-tsl: A continuous educational turkish sign language dataset with baseline methods. *2024 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 1–7. doi:10.1109/HORA61326.2024.10550648.

- Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002, July). Bleu: A method for automatic evaluation of machine translation. In P. Isabelle, E. Charniak, & D. Lin (Eds.), *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318). Association for Computational Linguistics. doi:10.3115/1073083.1073135.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-highperformance-eep-learning-library.pdf>
- Post, M. (2018). A call for clarity in reporting BLEU scores. *Proceedings of the Third Conference on Machine Translation: Research Papers*, 186–191. <https://www.aclweb.org/anthology/W18-6319>
- Prillwitz, S., Hanke, T., König, S., Konrad, R., Langer, G., & Schwarz, A. (2008a). Dgs corpus project–development of a corpus based electronic dictionary german sign language/german. *sign-lang@ LREC 2008*, 159–164.
- Prillwitz, S., Hanke, T., König, S., Konrad, R., Langer, G., & Schwarz, A. (2008b, June). DGS Corpus project – development of a corpus based electronic dictionary German Sign Language / German. In O. Crasborn, E. Efthimiou, T. Hanke, E. D. Thoutenhoofd, & I. Zwitserlood (Eds.), *6th international conference on language resources and evaluation (LREC 2008). Proceedings of the LREC2008 3rd workshop on the representation and processing of sign languages: Construction and exploitation of sign language corpora* (pp. 159–164). European Language Resources Association (ELRA). <https://www.sign-lang.uni-hamburg.de/lrec/pub/08018.pdf>
- Radford, A. (2018). Improving language understanding by generative pre-training.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. <https://arxiv.org/abs/2103.00020>
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140), 1–67.
- Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. *Expert Systems with Applications*, 164, 113794. doi:<https://doi.org/10.1016/j.eswa.2020.113794>.

- Rust, P., Shi, B., Wang, S., Camgoz, N. C., & Maillard, J. (2024, August). Towards privacy-aware sign language translation at scale. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 8624–8641). Association for Computational Linguistics. doi:10.18653/v1/2024.acl-long.467.
- Ryali, C., Hu, Y.-T., Bolya, D., Wei, C., Fan, H., Huang, P.-Y., Aggarwal, V., Chowdhury, A., Poursaeed, O., Hoffman, J., et al. (2023). Hiera: A hierarchical vision transformer without the bells-and-whistles. *International Conference on Machine Learning*, 29441–29454.
- Şahin, K., & Gökgöz, K. (2024, May). Decoding sign languages: The SL-FE framework for phonological analysis and automated annotation. In E. Efthimiou, S.-E. Fotinea, T. Hanke, J. A. Hochgesang, J. Mesch, & M. Schulder (Eds.), *Proceedings of the Irec-coling 2024 11th workshop on the representation and processing of sign languages: Evaluation of sign language resources* (pp. 335–342). ELRA; ICCL. <https://aclanthology.org/2024.signlang1.37>
- Schembri, A., Fenlon, J., Rentelis, R., Reynolds, S., & Cormier, K. (2013). Building the british sign language corpus.
- Sellam, T., Das, D., & Parikh, A. P. (2020). Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.
- Sennrich, R., & Haddow, B. (2016). Linguistic input features improve neural machine translation. *arXiv preprint arXiv:1606.02892*.
- Shazeer, N., & Stern, M. (2018). Adafactor: Adaptive learning rates with sublinear memory cost. *International Conference on Machine Learning*, 4596–4604.
- Shi, B., Brentari, D., Shakhnarovich, G., & Livescu, K. (2022). Open-domain sign language translation learned from online video. *arXiv preprint arXiv:2205.12870*.
- Sincan, O. M., & Keles, H. Y. (2020). Autsl: A large scale multi-modal turkish sign language dataset and baseline methods. *IEEE access*, 8, 181340–181355.
- Starner, T., Weaver, J., & Pentland, A. (1998). Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12), 1371–1375. doi:10.1109/34.735811.
- Stokoe, W. C. (1980). Sign language structure. *Annual review of anthropology*, 365–390.
- Sutton, V. (2000). Deaf action committee for sign writing. *Sign writing*.
- Tanzer, G., & Zhang, B. (2024). Youtube-sl-25: A large-scale, open-domain multilingual sign language parallel corpus. <https://arxiv.org/abs/2407.11144>

- Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., Chung, H. W., Shakeri, S., Bahri, D., Schuster, T., Zheng, H. S., Zhou, D., Houlsby, N., & Metzler, D. (2023). U12: Unifying language learning paradigms. <https://arxiv.org/abs/2205.05131>
- The Ministry of Family and Social Services. (2023, April). Engelli ve yaşlı hizmetleri genel müdürlüğü istatistik bülteni nisan 2023 [Accessed:2024-12-09]. https://aile.gov.tr/media/135432/eyhgm_istatistik_bulteni_nisan_23.pdf
- Tillmann, C., & Ney, H. (2003). Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1), 97–133. doi:10.1162/089120103321337458.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., . . . Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. <https://arxiv.org/abs/2307.09288>
- Uludoğan, G., Balal, Z., Akkurt, F., Turker, M., Gungor, O., & Üsküdarlı, S. (2024, August). TURNA: A Turkish encoder-decoder language model for enhanced understanding and generation. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Findings of the association for computational linguistics: Acl 2024* (pp. 10103–10117). Association for Computational Linguistics. doi:10.18653/v1/2024.findings-acl.600.
- Uthus, D., Tanzer, G., & Georg, M. (2023). Youtube-asl: A large-scale, open-domain american sign language-english parallel corpus. <https://arxiv.org/abs/2306.15162>
- Van Zijl, L., & Barker, D. (2003). South african sign language machine translation system. *Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa*, 49–52.
- Varol, G., Laptev, I., & Schmid, C. (2017). Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6), 1510–1517.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems (Vol. 30)*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- Vogler, C., & Metaxas, D. (1999). Parallel hidden markov models for american sign language recognition. *Proceedings of the seventh IEEE international conference on computer vision*, 1, 116–122.

- von Agris, U., & Kraiss, K.-F. (2010). Signum database: Video corpus for signer-independent continuous sign language recognition. *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 243–246.
- Wang, H., & Schmid, C. (2013). Action recognition with improved trajectories. *Proceedings of the IEEE international conference on computer vision*, 3551–3558.
- Wilbur, R. B. (2013). Phonological and prosodic layering of nonmanuals in american sign language. In *The signs of language revisited* (pp. 196–220). Psychology Press.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., . . . Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. <https://www.aclweb.org/anthology/2020.emnlp-demos.6>
- Wong, R., Camgoz, N. C., & Bowden, R. (2024). Sign2gpt: Leveraging large language models for gloss-free sign language translation. <https://arxiv.org/abs/2405.04164>
- World Federation of the Deaf. (2024). Our work [Accessed: 2024-12-09]. <http://wfdeaf.org/our-work/>
- Xue, L. (2020). MT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2021). MT5: A massively multilingual pre-trained text-to-text transformer. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 483–498. doi:10.18653/v1/2021.naacl-main.41.
- Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *Proceedings of the AAAI conference on artificial intelligence*, 32(1).
- Yang, H.-D., Sclaroff, S., & Lee, S.-W. (2008). Sign language spotting with a threshold model based on conditional random fields. *IEEE transactions on pattern analysis and machine intelligence*, 31(7), 1264–1277.
- Zhang, B., Tanzer, G., & Firat, O. (2024). Scaling sign language translation. <https://arxiv.org/abs/2407.11855>
- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). Bertscore: Evaluating text generation with bert. *International Conference on Learning Representations*. <https://openreview.net/forum?id=SkeHuCVFDr>

- Zhang, X., & Duh, K. (2021). Approaching sign language gloss translation as a low-resource machine translation task. *Proceedings of the 1st International 106 Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL)*, 60–70.
- Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2), 4–10.
- Zhao, W., Hu, H., Zhou, W., Shi, J., & Li, H. (2023a). Best: Bert pre-training for sign language recognition with coupling tokenization. <https://arxiv.org/abs/2302.05075>
- Zhao, W., Hu, H., Zhou, W., Shi, J., & Li, H. (2023b). Best: Bert pre-training for sign language recognition with coupling tokenization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(3), 3597–3605.
- Zhou, B., Chen, Z., Clapés, A., Wan, J., Liang, Y., Escalera, S., Lei, Z., & Zhang, D. (2023). Gloss-free sign language translation: Improving from visual-language pretraining. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 20871–20881.
- Zhou, H., Zhou, W., Qi, W., Pu, J., & Li, H. (2021a). Improving sign language translation with monolingual data by sign back-translation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1316–1325.
- Zhou, H., Zhou, W., Qi, W., Pu, J., & Li, H. (2021b). Improving sign language translation with monolingual data by sign back-translation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1316–1325.
- Zimmermann, C., & Brox, T. (2017). Learning to estimate 3d hand pose from single rgb images. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.