NEURAL SIGN LANGUAGE TRANSLATION BY LEARNING TOKENIZATION

by

Alptekin Orbay

B.S., Computer Engineering, Bogazici University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2020

ACKNOWLEDGEMENTS

First, and most of all, I would like to thank my thesis advisor Prof. Lale Akarun for her expertise, guidance, assistance and patience during my masters education. I would also like to thank Prof. Murat Saraçlar and Assistant Prof. Furkan Kıraç for participating in my thesis jury and their valuable comments and suggestions on my thesis.

I would like to express my deepest gratitude to my family for their endless support and encouragement throughout my life. Without them, I wouldn't be the person who I am today.

I would like to thank my friends and colleagues in Boğaziçi University and Perceptual Intelligence Laboratory for their endless support and valuable friendship, namely, Ahmet Alp Kındıroğlu, Doğa Siyli, Gizem Esra Ünlü, Mehmet Burak Kurutmaz, Oğulcan Özdemir. I would also like to extend my thanks to Umut Başer, Ecem Alantuğ and Burcu Topçcu for relaxing coffee breaks in our busy schedules.

This thesis has been supported by the Scientific and Technological Research Council of Turkey (TUBITAK) Project #117E059.

ABSTRACT

NEURAL SIGN LANGUAGE TRANSLATION BY LEARNING TOKENIZATION

In this thesis, we propose a multitask learning based method to improve Neural Sign Language Translation (NSLT) consisting of two parts, a tokenization layer and Neural Machine Translation (NMT). The tokenization part focuses on how Sign Language (SL) videos should be represented to be fed into the other part. It has not been studied elaborately whereas NMT research has attracted several researchers contributing enormous advancements. Up to now, there are two main input tokenization levels, namely frame-level and gloss-level tokenization. Glosses are world-like intermediate presentation and unique to SLs. Therefore, we aim to develop a generic sign-level tokenization layer so that it is applicable to other domains without further effort.

We begin with investigating current tokenization approaches and explain their weaknesses with several experiments. To provide a solution, we adapt Transfer Learning, Multitask Learning and Unsupervised Domain Adaptation into this research to leverage additional supervision. We succeed in enabling knowledge transfer between SLs and improve translation quality by 5 points in BLEU-4 and 8 points in ROUGE scores. Secondly, we show the effects of body parts by extensive experiments in all the tokenization approaches. Apart from these, we adopt 3D-CNNs to improve efficiency in terms of time and space. Lastly, we discuss the advantages of sign-level tokenization over gloss-level tokenization. To sum up, our proposed method eliminates the need for gloss level annotation to obtain higher scores by providing additional supervision by utilizing weak supervision sources.

ÖZET

SİMGELEME ÖĞRENEREK SİNİRSEL AĞLARLA İŞARET DİLİ ÇEVİRİSİ

Bu tezde, Çoklu Görev Öğrenmesi tabanlı bir yöntem Sinirsel Ağlarla İşaret Dili Çevirisini geliştirmek için önerilmiştir. Sinirsel Ağlarla İşaret Dili Çevirisi, Sinirsel Ağlarla Makine Çevirisi ve Simgeleme olarak iki parçadan oluşmaktadır. Simgeleme tabakası ana odağına işaret dili videolarının nasıl bir gösterim ile diğer katmana beslenmesi gerektiğini ele alır. Simgeleme pek fazla keşfedilmiş bir alan değildir. Halbuki, Sinirsel Ağlarla Makine Çevirisi pek çok farklı disiplinden araştırmacının ilgisini çekmiştir ve bu sayede alanda sürekli ilerleme kaydedilmektedir. Simgeleme bugüne kadar çerçeve ve gloss olmak üzere iki seviyede incelenmiştir. Glosslar kelime düzeyinde anlam taşıyan ara gösterimlerdir ve her işaret diline özel anlam taşırlar. Ayrıca, gloss bazında etiketleme çok emekte gerektiren bir süreçtir. Bu yüzden, çerçeve seviyesinde Simgeleme ile herhangi bir işaret diline kolay uygulanabilir yöntem geliştirmeyi hedefledik.

Öncelikle bugüne kadar sunulan yöntemlerin eksik yanlarını pek çok deney ile açığa çıkarttık. Elde ettiğimiz kazanımlar ile Öğrenme Aktarması, Çoklu Görev Öğrenme ve Gözetimsiz Alan Uyarlaması tekniklerini bu probleme uyarladık. Bu sayede etiketlenip anlamlandırılmış veri eksikliğini gidermeye çalıştık. İşaret dilleri arasında bilgi aktarımını sağlayarak BLUE-4 metriğinde 5 ve ROUGE skorunda 8 puanlık bir iyileştirme sağladık. İkinci olarak, insan vücut bölümlerinin anlama etkilerini her deneyimizde ayrıca inceledik. 3D-CNN yapısını zamansal ve mekansal bakımdan daha etkili bir sistem oluşturmak için uyguladık. Son olarak, işaret seviyesinde simgelemenin glossa göre avantajlarını açıkladık. Özetlemek gerekirse, sunduğumuz yöntem zayıf gözetim kaynaklarını kullanarak gloss bazında etiketlemeye olan bağlılığı azaltıyor.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xi
LIST OF ACRONYMS/ABBREVIATIONS . . . xii
1. INTRODUCTION . . . 1
1.1. Motivation . . . 5
1.1.1. Input Tokenization in NSLT . . . 6
1.2. Research Overview and Contributions . . . 9
1.3. Thesis Outline . . . 11
2. RELATED WORK . . . 13
2.1. Sign Language Recognition . . . 13
2.2. Sign Language Recognition Datasets . . . 16
2.3. Neural Machine Translation . . . 19
2.4. Neural Sign Language Translation Datasets . . . 23
3. LEARNING TOKENIZATION . . . 25
3.1. Neural Sign Language Translation . . . 25
3.1.1. Sign Embeddings . . . 25
3.1.2. Attentional NMT . . . 27
3.2. Tokenization with 2D-CNN Trained on Hand Shape Classification . . . 30
3.2.1. Multitask Learning . . . 32
3.2.2. Domain Adaptation . . . 34
3.3. Tokenization with 3D-CNN Trained on Action Recognition . . . 36
4. EXPERIMENTS & RESULTS . . . 37
4.1. Input Analysis of Implicit Tokenization Layer . . . 37
4.2. Tokenization with a Frozen CNN . . . 39

4.2.1. Attention Search 39
4.2.2. Transfer Learning From ImageNet 42
4.2.3. Human Keypoints 43
4.2.4. Hand Shape Learning 44
4.3. Transfer Learning from Kinetics 50
4.4. Tokenization Learning on Target Domain 52
4.5. Analysis of Results 53
4.6. Comparison of Frame Levels & Gloss Level Translation 55
5. CONCLUSION 57
5.1. Remarks 57
5.2. Future Work 58
REFERENCES 59
APPENDIX A: ATTENTIONAL NMT MODEL 69
A.1. Implicit Tokenization 69
A.2. Bahdanau & Luong Model Details 69
A.3. Transformer Model Details 70
APPENDIX B: CNN TRAINING 71
B.1. Multitask Learning 71
B.2. Domain Adaptation 71

LIST OF FIGURES

Figure 1.1. Important elements and steps in Sign Language research. 2

Figure 1.2. An illustration of glosses and signs in a German Sign Language sentence. 3

Figure 1.3. An example of translation between three different spoken languages. 4

Figure 1.4. An example of Sign Language translation. 5

Figure 1.5. An illustration of word-level tokenization in NMT. 7

Figure 1.6. An illustration of frame to gloss tokenization in NMT. 8

Figure 1.7. An illustration of frame level tokenization in NMT. 9

Figure 2.1. Three sample images from each data source. Top to bottom ph, danish, nz, TSL. 17

Figure 2.2. Three sample images from right hands, both hands and full frame from top to bottom. 24

Figure 3.1. Overall schema of Neural Sign Language Translation. 26

Figure 3.2. Illustration of Bahdanau and Luong attention mechanisms. 30

Figure 3.3. Tokenization with a 2D-CNN trained for hand shape classification. 31

Figure 3.4. Multitask learning framework. 33

Figure 3.5. Illustration of domain adaptation with Gradient Reversal Layer. . 35

Figure 3.6. Illustration of Tokenization with 3D-CNN. 36

Figure 4.1. Illustration of Tokenization with frozen 2D-CNN. 40

Figure 4.2. Convergence graph of the models. 41

Figure 4.3. BLEU-4 evaluation score graph of the models. 42

Figure 4.4. Samples from OpenPose output coordinates. 45

Figure 4.5. Learning schema of hand shapes with a 3D-CNN. 47

Figure 4.6. Combination of right hands and full frames in Tokenization layer. 49

Figure 4.7. Processing speed graph of NSLT with 3D-CNN and 2D-CNN. . . . 51

LIST OF TABLES

Table 2.1. Corpus statistics of hand shape datasets. 16

Table 2.2. Datasets used in the literature of Sign Language Recognition. . . . 18

Table 4.1. Test results of implicit Tokenization. 38

Table 4.2. Comparison of attention mechanisms. 41

Table 4.3. Test results of different input settings. 43

Table 4.4. Test results of Tokenization with keypoints. 44

Table 4.5. Hand shape classification results on the the target domain. 47

Table 4.6. Test results of Tokenization with hand shape representation. 48

Table 4.7. Test results of Tokenization with I3D. 51

Table 4.8. Test results with Tokenization trained on the target domain. . . . 52

Table 4.9. Paired bootstrap sampling results on different translation comparisons. Delta refers to the differences in BLEU-4 Score and 250, 400 and 600 are the number of the sentences drawn for a sample. The table displays the confidence of the claims under different sentence sizes. 54

Table 4.10. The table displays unique word numbers under varying frequencies in the translation systems. 55

LIST OF SYMBOLS

a_t	Attention weights at step t
c_t	Context vector at step t
$e_{1:Z}$	Distance scores of all encoder outputs
f	Encoder Function
g	Decoder Function
$f \circ g$	Decoder Function
$h_{1:S}$	Decoder outputs from step 1 to S
h_f	Encoder hidden states
L	Loss Term
$o_{1:Z}$	Encoder outputs from step 1 to step Z
$t_{1:Z}$	Tokens from step 1 to step Z
T_x	Normalized x coordinate values of hand keypoints
T_y	Normalized y coordinate calues of hand keypoints
V_x	x coordinate values of hand keypoints
V_y	y coordinate values of hand keypoints
$w_{1:S}$	Target words from step 1 to step S
$x_{1:N}$	Frames from step 1 to step N
α	Distance function between output vectors
$\beta(t)$	Loss term weight at time step t
θ	Dense layer
$\lambda(t)$	Coefficient in Gradient Reversal Layer at iteration t
σ	Standard Deviation
ψ	Tokenization Function

LIST OF ACRONYMS/ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
2D CNN	Two Dimensional Convolutional Neural Network
3D CNN	Three Dimensional Convolutional Neural Network
ASL	American Sign Language
BSL	British Sign Language
CSL	Chinese Sign Language
CSLR	Continuous Sign Language Recognition
CNN	Convolutional Neural Network
CTC	Connectionist TemporalCclassification
ED	Encoder-Decoder
DTW	Dynamical Time Warping
HOF	Histogram of Oriented Gradients
HOF	Histogram of Optical Flow
HMM	Hidden Markov Model
GSL	German Sign Language
GRL	Gradient Reversal Layer
GRU	Gated Recurrent Unit
IDT	Improved Dense Trajectories
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
IOHMM	Input-Output Hidden Markov Model
LSTM	Long-short Term Memory
NMT	Neural Machine Translation
NSLT	Neural Sign Language Translation
PH	RWTHPHOENIX-Weather 2014
RNN	Recurrent Neural Network
SL	Sign Language

SLR	Sign Language Recognition
TSL	Turkish Sign Language

1. INTRODUCTION

Sign Languages (SL) are the main way of communication for hearing impaired people. Deaf people in a region develop a unique sign language independent of the region's spoken languages. In addition to that, there is no strong relation between different sign languages. Therefore, SLs are considered as natural languages like spoken and written languages. This means SLs have their own linguistic properties as well as visual properties. Hence, it is required to identify their complex structures and treat them as separate languages.

There have been several studies to bridge the communication gap between spoken and sign languages. Different concepts have been developed to analyze SLs in detail as seen in Figure 1.1. First of all, signs are semantic parts in SL videos. In a SL video, all frames are usually not meaningful. In general, some frames are transition frames whereas some are keyframes where signers perform special hand shapes. The other key concept is gloss. Glosses are word-like representations corresponding to signs in terms of meaning. Glosses are in text form, but they do not construct a full sentence if they are ordered in the same manner as signs. As seen in Figure 1.2, glosses in German Sign Language (GSL) are illustrated from the example frame sequence. "ICH" means "I" in English. In this example, it means "we", but corresponding signs may have different meanings like "my", "I", "me" in a different video. This approach generates meaningful interpretations; however, linguistic properties are not ignored. Furthermore, the hand shapes in SL videos require special attention as they do not have the same meanings as daily life gestures. Hence, hand shapes should be studied very carefully. Another challenge is that subtle changes in SL hand shapes may lead to dramatic differences in meaning.

Sign Language Recognition refers to a number of recognition tasks. In the isolated setting, it may refer to identifying the labels of a sign; which are often glosses. In the continuous setting, it refers to a broad range of tasks, from sign spotting to sign sentence

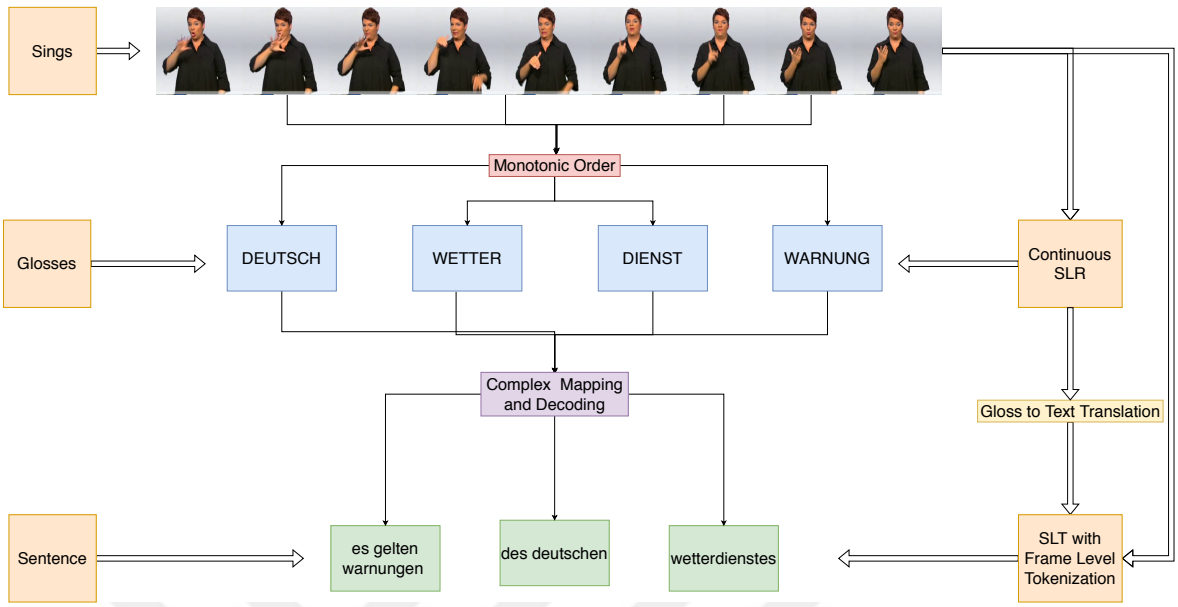

Figure 1.1. Important elements and steps in Sign Language research.

identification. As stated earlier, hand shape recognition is crucial, but hand shapes differ from one sign language to another semantically and syntactically. In addition to that, the conveyed information is not limited to hand shapes, but also general frame contexts play an important role. The second challenge is that the meaning of signs is dependent on the overall context. This is to say that a proper approach should cover the relationship between different signs despite the fact that the order of glosses and signs is monotonic in a sentence. In short, a good approach should be holistic to attain successful results for SLR.

Recently, another research field, Neural Sign Language Translation (NSLT), has emerged to provide translations from sign language to spoken languages just like translation from a spoken language to another. Sign Language Translation (SLT) shares the same basics with other translation tasks as seen as seen in Figure 1.3 and Figure 1.4. In addition to that, SLT requires further effort to tackle with its visual properties. An example translation between three different languages is shown in Figure 1.3. The three sentences have the same meaning, but there are certain differences such as the lengths of sentences and the order of the words. Therefore, word by word translation

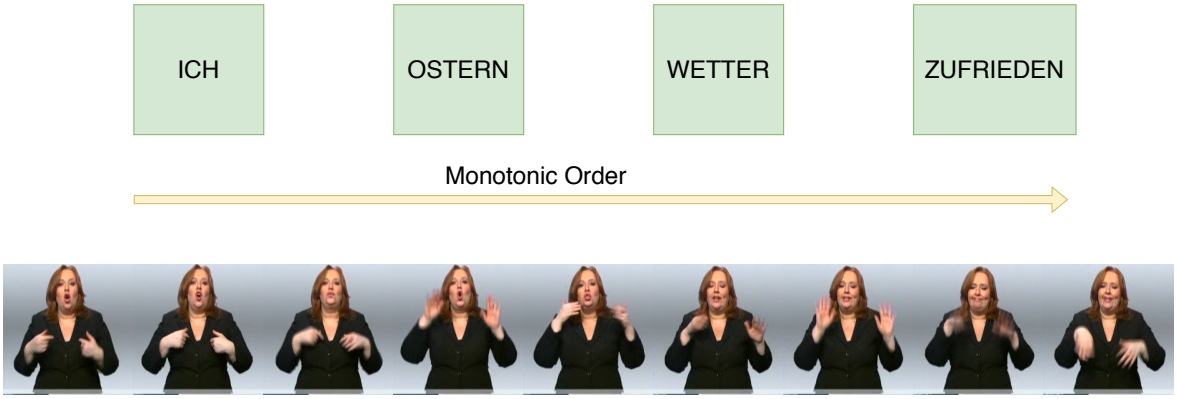

Figure 1.2. An illustration of glosses and signs in a German Sign Language sentence.

is not the proper approach. Furthermore, word meaning is dependent on the whole sentence. For example, "to" in an English sentence may have several meanings. We can choose the correct one in this sentence by examining the words placed before and after it. Another important point is that the output words must not violate the grammar of the target language. To conclude, the solution should cover all the concerns and those rules are also prevailing for SLT.

NSLT aims to enable continuous dialog between hearing-impaired individuals and people who do not know SLs. Neural Machine Translation (NMT) has achieved several advancements to sustain human-level translations. Therefore, NSLT may use the same methods and findings, but the visual properties require new approaches. NMT formulation may be adapted by NSLT. Firstly, the source and target sentences are represented with tokens. Tokens are considered as atomic and meaningful blocks in sentences. Tokenization is the name of the process to find the most appropriate tokens for the translation. Splitting words in a sentence is the most straightforward and effective approach. However, agglutinative languages are not suitable for word-level tokenization so n-gram chars are widely used for this type of languages. This implies that tokenization strategy must be tailored for each target domain. After choosing a proper tokenization method, we can continue to formulate the problem as follows. There are $(x_1, x_2, \ldots, x_n)$ symbols as input tokens, $\mathbf{x}$. The symbols and their order represent a context, $c_{\mathbf{x}}$. The target symbols are $(y_1, y_2, \ldots, y_n)$ as output tokens,

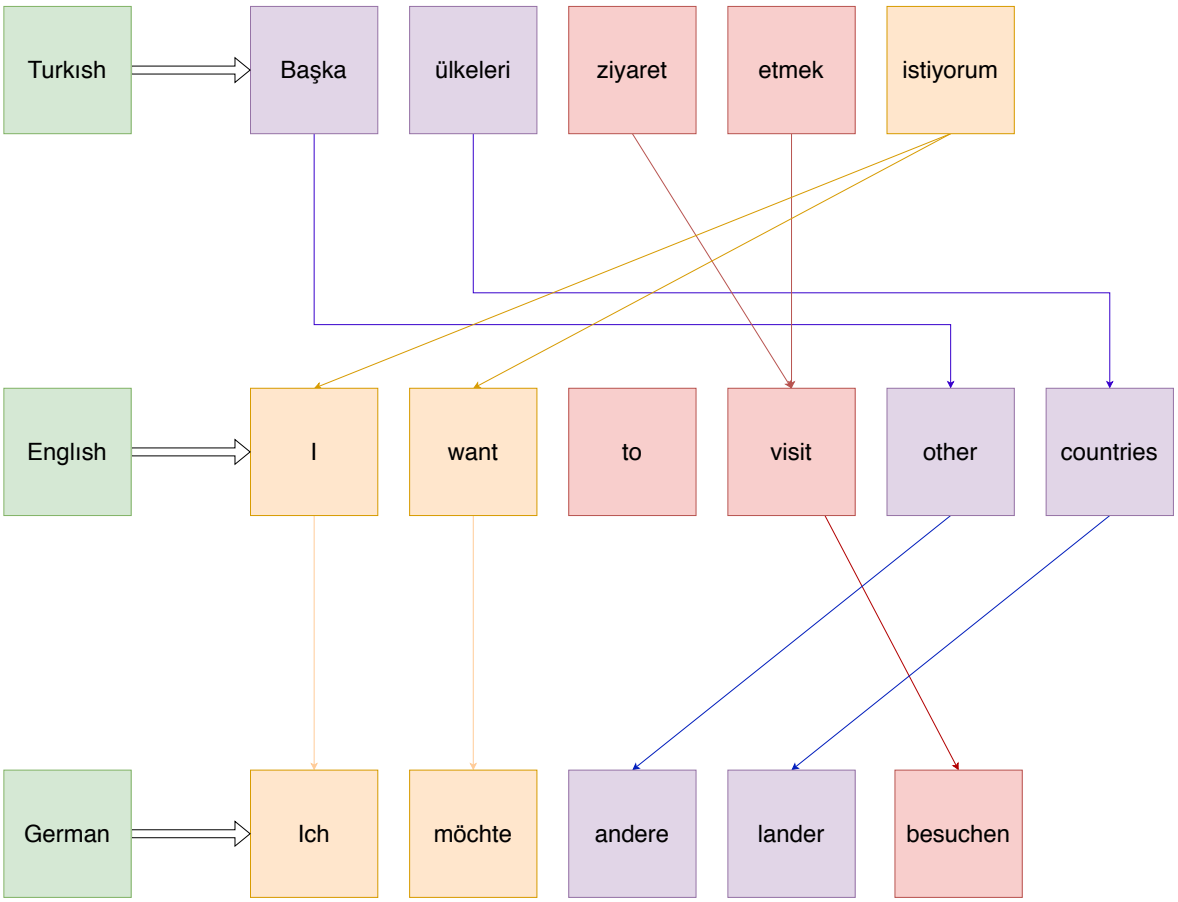

Figure 1.3. An example of translation between three different spoken languages.

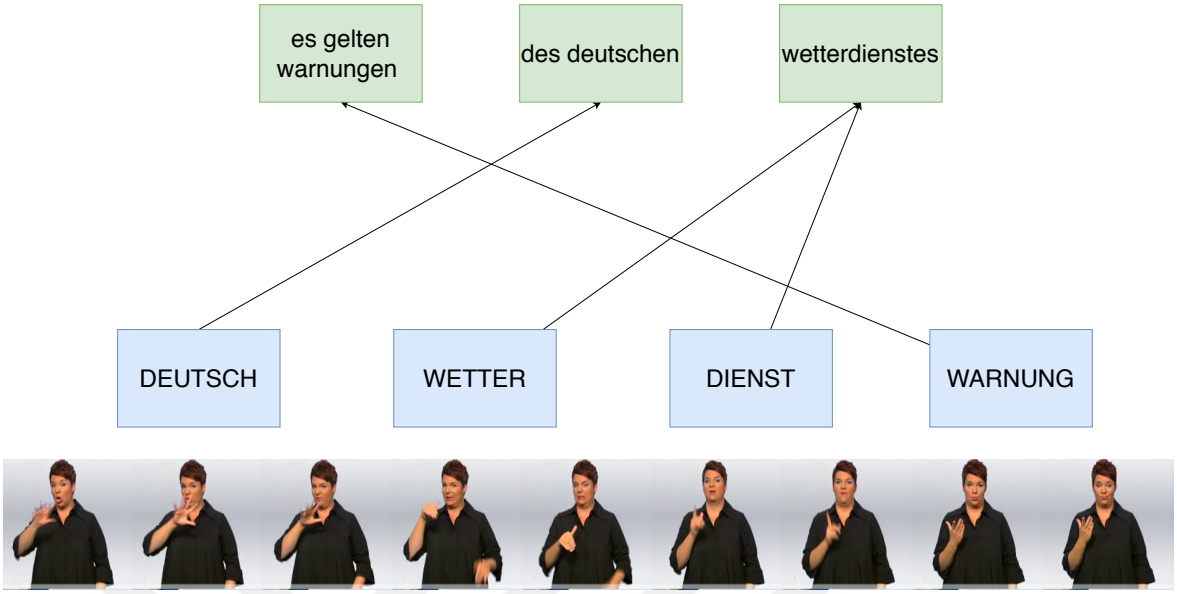

Figure 1.4. An example of Sign Language translation.

$\mathbf{y}$. Eventually, a complex function $f((x_1, x_2, \ldots, x_n), c_{\mathbf{x}}) \rightarrow (y_1, y_2, \ldots, y_n)$ should be found. NMT researchers have proposed several architectures and methods to find the most optimal mapping between the tokens. However, in SLT research, we need a special tokenization approach to apply NMT architectures successfully.

1.1. Motivation

In this section, we mention different tokenization techniques for SLT and explain our perspective on the problem. We mentioned about the basics of SLT and NMT. From our research perspective, NMT methods can provide successful results if we have good tokens from SL vides. Therefore, tokenization is seen as the most crucial part of this research. Firstly, the visual properties are involved in the tokenization part. Secondly, there is no a generic approach to obtain strong tokens for all the SLs. In addition to that, it is not clear that discrete tokens should be obtained for better translation quality. For this reason, we extend the meaning of tokenization for NSLT and it covers the overall process to prepare the frames for the NMT module.

For spoken to spoken languages, we generally use words as tokens to feed the NMT module. The current state-of-the-art method converts those tokens to continuous embeddings to reach a semantic representation. While learning translation, the word embedding is also trained to learn the relationship between words. Eventually, a meaningful embedding is obtained before the NMT module as seen in Figure 1.5. Based on this, it may be a good idea to learn a good representation of signs to replace with word embeddings to achieve the same advancements in NSLT as NMT has done. This representation is cross-lingual; but learning it is an open problem. Our research is mainly focused on this problem. Before introducing our approach, we discuss the existing three tokenization approaches in the following subsection.

1.1.1. Input Tokenization in NSLT

The first approach is using glosses as tokens. Glosses are intermediate word-like representations between signs and words in sentences. Therefore, they can be directly applicable to the NMT framework without any further effort. However, there are certain shortcomings in this method. Firstly, glosses rarely exist in real life. Gloss annotation requires a laborious process and special expertise. Secondly, glosses are unique to SLs. Therefore, each SL requires special effort to obtain glosses whereas sentences are commonly available. The last drawback is that a mistake in the gloss level can produce dramatic meaning differences in translation, since glosses are high level annotations, similar to words.

The second approach is the same as the first one in terms of tokens. On top of that, this approach learns to extract glosses from frames. In other words, this method uses glosses as explicit intermediate representations as seen in Figure 1.6. It eliminates the further search for tokenization, but it needs a special network for frame to gloss conversion. There are two main concerns. The first one is that a network for frame to gloss conversion is still dependent on gloss annotations. The second is that it is not clear that glosses are the upper bound for SLT as there is not sufficient evidence. The

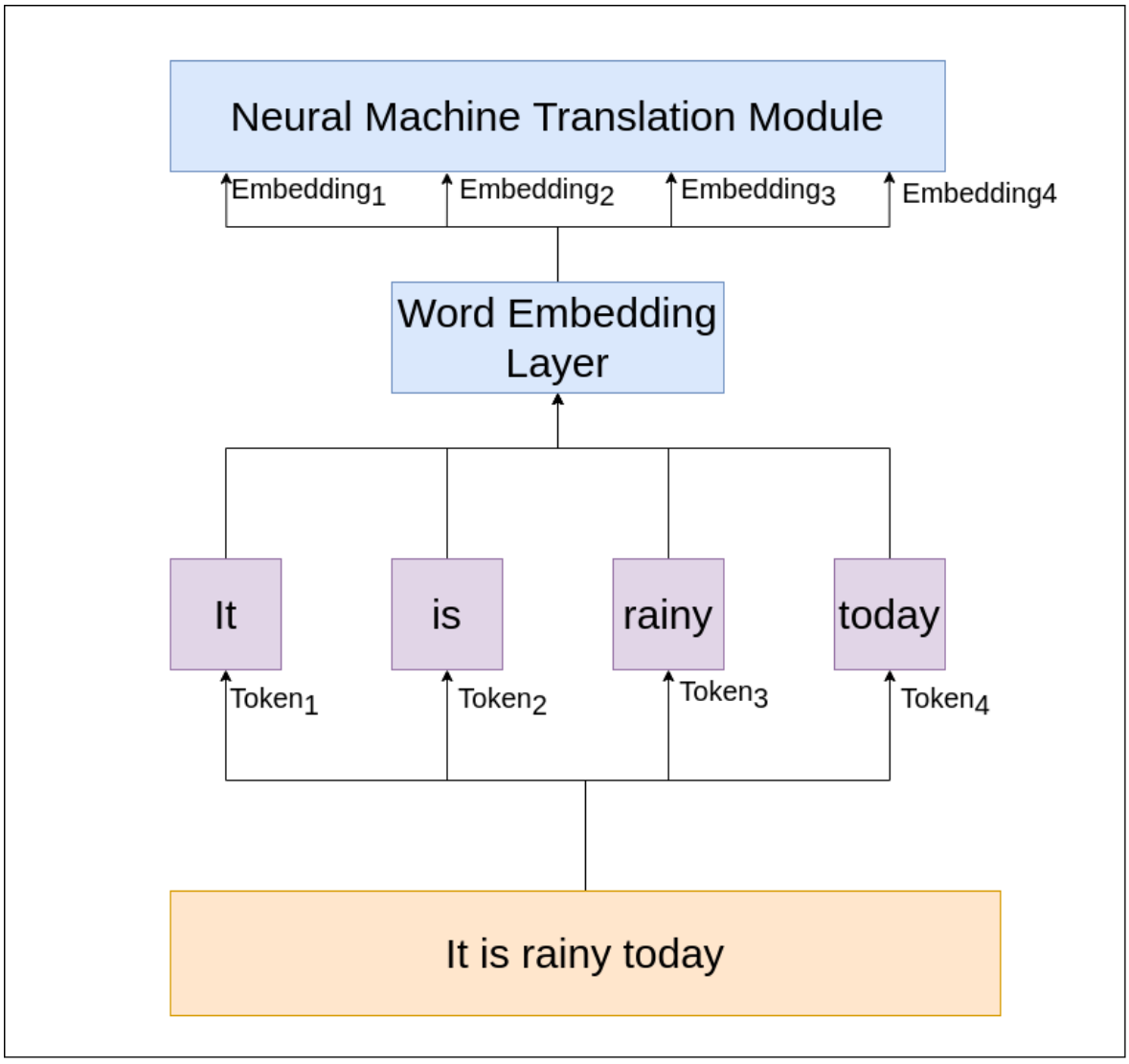

Figure 1.5. An illustration of word-level tokenization in NMT.

problem is immature and the result in [1] provides clues about whether glosses may restrict translation quality.

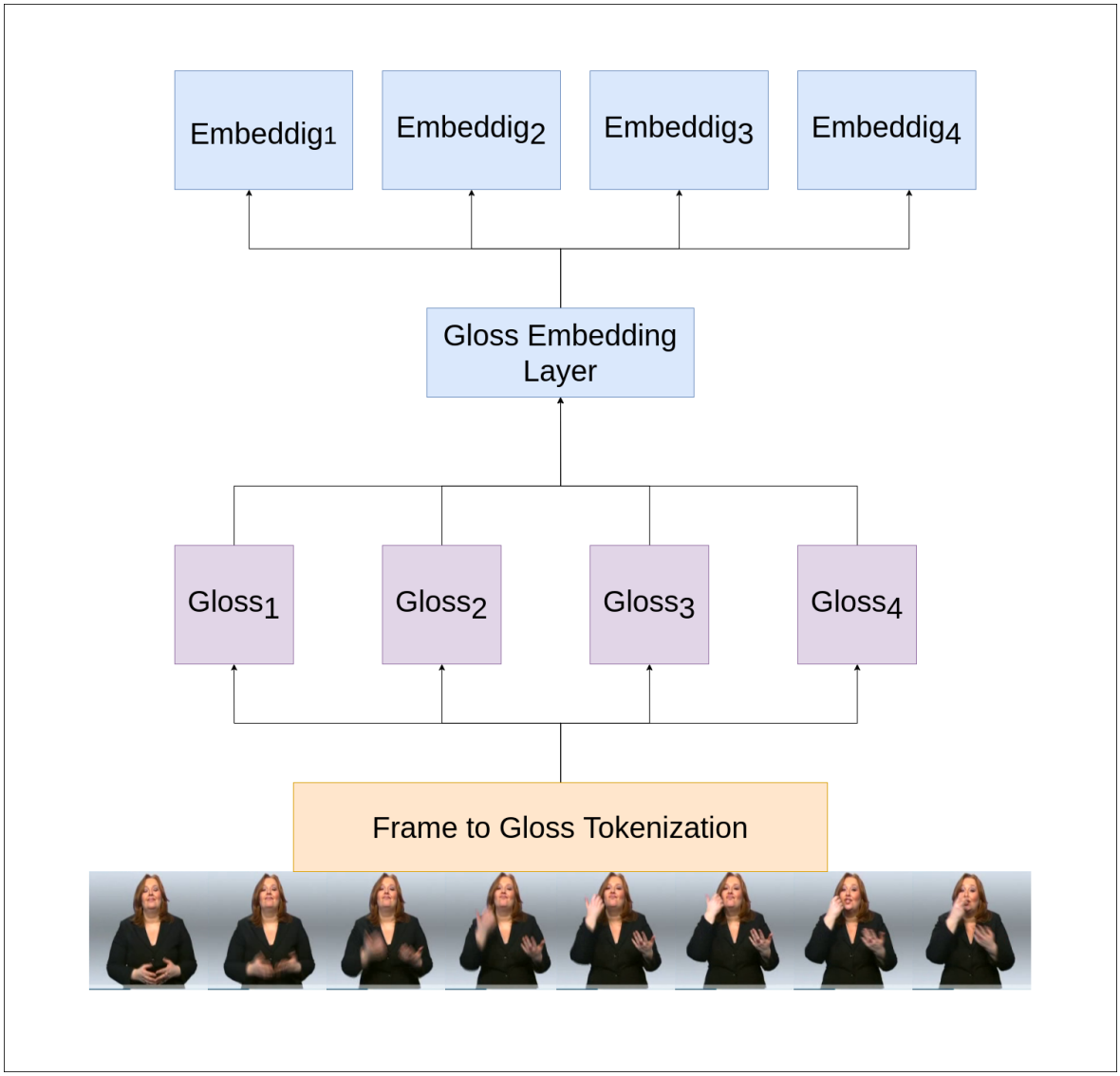

Figure 1.6. An illustration of frame to gloss tokenization in NMT.

The third approach is called frame-level tokenization. This approach does not establish any explicit intermediate representation as seen in Figure 1.7. It aims to learn good sign embeddings to replace with word-embeddings. However, there is no golden way to represent signs with embeddings to feed into the NMT module. Furthermore, it is not clear what the length of the embedding should be. Embeddings can be obtained from each frame or extracted from inner short clips in the video. In addition to that,

the representation can be learned with sentence-video pairs or trained outside the NSLT system. There are several ways for frame-level tokenization. However, the main difference from the gloss level tokenization is that discrete representation can be eliminated. If we find a proper one, there would be several advantages. The first one is that the resulting framework can be applied to any SL translation task without requiring annotation. The second advantage is the opportunity to inject additional supervision. The representations would be trained on different tasks and different datasets whereas gloss level tokenization cannot cover different SLs. The third one is that the token length can be adjusted. To boost translation speed, the number of tokens can be reduced to a pre-determined number.

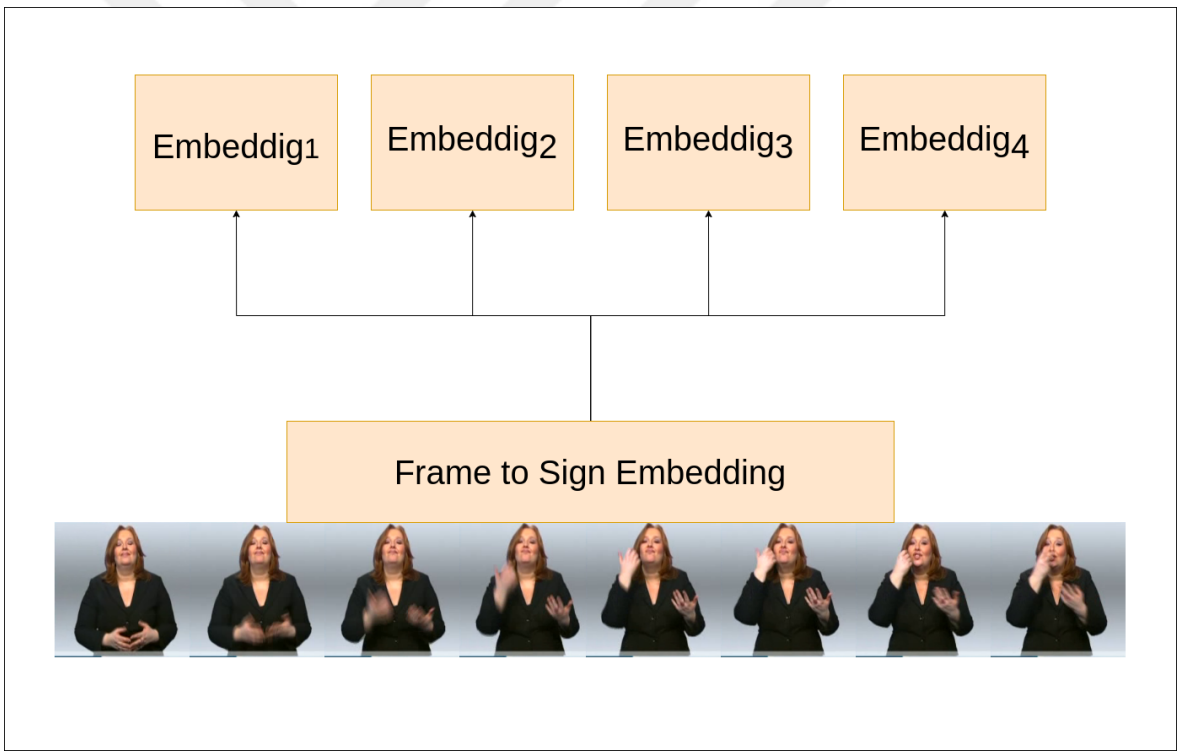

Figure 1.7. An illustration of frame level tokenization in NMT.

1.2. Research Overview and Contributions

The study focused mainly on searching alternative tokenization methods. We have three main motivations. The first one to leverage additional supervision to SLT.

We suspect that the data amount is limited. Generally, SL data is very scarce and we would like to benefit from all sorts of data with maximum efficiency. The second one is that our proposed approach would enable a generic tokenization layer. In this manner, the method can be adopted by any SL. The third one is providing flexibility for real-time applications. The current approaches require a vast amount of computational power. Therefore, we search for ways to increase efficiency and analyze the effects on translation quality.

Our approach is well suited to benefit from different datasets and different sign languages to leverage additional supervision. We claim that general purpose datasets are sufficient to provide better translations. As a result, the need for annotation from domain experts is eliminated. In this manner, the first and second are addressed. We also show a trade-off between time efficiency and translation quality. The overall contributions of this study can be summarized as:

- It illustrates that end-to-end learning is improved by learning tokenization explicitly.
- It proposes a semi-supervised tokenization method utilizing Multitask Learning, Domain Adaptation and Transfer Learning to leverage additional supervision.
- It shows the effectiveness of 3D-CNN in Sign Language Translation.
- It proves the effectiveness of the proposed method by comparing other tokenization methods trained on target domain data.

First, we investigate end-to-end learning scheme proposed by Camgoz et al. [2] which formalized Neural Sign Language Translation (NSLT) for the first time. The formalized system mainly consists of two parts, tokenization and NMT. This study reveals the shortcomings of the newly proposed NSLT system under different input settings. The results of the initial experiments indicate that it has two drawbacks. The first one is that it cannot attach enough attention to the body parts. The data amount in training is not sufficient to make the model fully functional.

To deal with the first drawback, this study addresses the tokenization part of the system. Along with the tokenization approach with human keypoint proposed by Ko et al. [3], it conducts several experiments with new tokenization methods. As a result, it illustrates the effectiveness of alternative tokenization methods over the end-to-end learning scheme.

It is known that annotation in SL requires additional effort and expertise, Hence, the study utilizes Multitask Learning, Domain Adaptation and Transfer Learning methods to develop an efficient tokenization approach in the absence of insufficient amount of data. In other words, the main contribution of this study is that it proposes a state-of-the-art tokenization approach that is independent of the target domain.

Moreover, the study tests the proposed method in different experimental setups. As the data amount for translation is very limited and NMT models are large enough to overfit training datasets easily, it should be shown that the tokenization part is generic to function in other SL datasets. Therefore, the study divides experiments into two parts. One is tokenization learning involving target domain data and the other is without target domain knowledge. Eventually, the study concludes that the proposed method can close the gap between frame level and gloss level tokenization approaches even if target Sign Languages are not seen in learning tokenization.

1.3. Thesis Outline

The rest of the thesis is organized as follows: In Chapter 2, we review the literature on both sign language recognition and sign language translation. We also investigate the popular datasets for both domains in the same chapter. In Chapter 3, we explain the background of NSLT, Multitask Learning and Domain Adaptation. After we explain their background information, we introduce our experimental setups and explain the usage of the proposed methods along with other alternatives in the literature in Chapter 4. The chapter also presents experiment results and their im-

plications. Lastly, we share our final thoughts about our work and give some future directions in Chapter 5.

2. RELATED WORK

2.1. Sign Language Recognition

Sign language research is a special area that requires a multi-disciplinary approach to conduct detailed studies. The development of this research is highly dependent on data collection as its annotation process is very laborious and requires special knowledge. However, there are some dictionaries [4], [5], [6] for educational purposes. These types of sources are not useful unless they include an automatic labeling strategy.

Some attempts are performed to collect data with wearable sensors. Those provide very informative measurements like angles between fingers, coordinates of key points and velocities. In [7], PowerGlove is used to collect data and hand-crafted features from those measurements are fed to decision trees. Magnetic sensors are used to collect American Sign Language (ASL) data in [8]. These approaches are important to reveal dynamics of Sign Languages, but not applicable in the wild.

As a result, researchers have begun to collect datasets [9], [10], [11] in controlled environment with RGB cameras. They are beneficial for identifying signs but they are very limited in terms of variation and context. They have paved the way for developing bigger datasets with their supervision.

Researchers have continued to search for ways to construct larger datasets without additional efforts. The studies in [12], [13] , [14] have utilized weak labels in public broadcasts for hearing impaired people. This type of data can be aligned with subtitles or lip reading. Thanks to a huge amount of data, those approaches have been accomplished to create big datasets by utilizing those side information. Forster et al. released RWTH-PHOENIX-Weather 2012 [15] and later its extended version RWTH-PHOENIX-Weather 2014 [16] from TV broadcasts for hearing impaired peo-

ple. Eventually, a dataset is publicly available to researchers with continuous gloss and sentence annotations.

In previous studies, hand-crafted features are used to represent signs in images. Dalal *et al.* [17] proposed Histogram of Oriented Gradients (HOG) to detect humans in images. Liwicki *et al.* [18] modifies this hand shape descriptors to apply on British Sign Language (BSL) alphabets. Lim *et al.* [19] uses Histogram of Optical Flow (HOF) [20]. Ozdemir *et al.* [21] applies Improved Dense Trajectories (IDT) [22] on Isolated Turkish Sign Language (TSL) data. They also extend their work in [23] as IDT is computationally expensive. HOG is calculated on blocks in images with orientation gradients to represent appearances and shapes of objects. HOF is similar to HOG, but extracted from optical flows in consecutive frames. Those descriptors are very robust to different environmental conditions.

Dynamical Time Warping (DTW) [24] is a technique that finds patterns in sequences with dynamic programming. Hidden Markov Models (HMM) [25] is a graph-based probabilistic for temporal modeling. HMMs are generative models, but Bengio *et al.* [26] proposed Input-Output HMM which has discriminant properties. Vogler *et al.* [8] adapts HMMs to recognize ASL. Note that every sign must be modeled with a different HMM so pruning is needed to classify with lots of unique signs. Keskin *et al.* [27] uses DTW for unsupervised clustering of sequences and IOHMMs to predict hand gestures on the clustered data. DTW generates a similarity score between sequences and IOHMM is learned to identify signs by their discriminative power.

While the prior techniques boost sign language research in terms of data and domain knowledge, Deep Learning (DL) opened a new era for Computer Vision research. Krizhevsky *et al.* [28] made a breakthrough in image recognition on Imagenet-LSVRC contest. The success of their approach is based on the advanced capabilities of CNNs to represent images compared to hand-crafted features. Moreover, sequence modeling with Long Short Term Memory (LSTM) [29] has become one of the most important ad-

vances in DL. Several achievements are accomplished in speech recognition [30], speech translation [31] and image captioning [32].

With new datasets and DL techniques, Continuous Sign Language Recognition (CSLR) research has been boosted dramatically. CSLR maps input frames to glosses while SLR generally deals with isolated datasets. Koller *et al.* [33] uses CNNs as frame-level feature extractor and HMMs so as to model of transition between frames. The models are trained jointly with Expectation-Maximization method. The study accomplishes to train on over one million images. Another important result is that the features extracted from the CNN outperform hand-crafted features (HOG-3D) dramatically in CLSR. In the study of [34], CNN-HMM architectures are directly trained on continuous gloss recognition. HMMs maximize the probability of gloss alignment and CNNs learn representations of images to construct hidden states of HMMs. Later, Bidirectional LSTMs (bi-LSTM) [35] are deployed between CNNs and HMMs to encapsulate contextual information in [36]. LSTM outputs are fed as hidden states of HMMs. The emergence of Connectionist Temporal Classification (CTC) proposed by Graves *et al.* [37] enables optimize deep architectures in an end-to-end manner. CTC is a loss layer that calculates a penalty between two sequences following a monotonic order. While previous studies use Bayesian optimization with HMMs, Camgoz *et al.* replace HMMs with LSTMs for continuous hand shape recognition and outperforms DeepHand (CNN-HMM) [33] on weakly annotated datasets.

Moreover, Carreira *et al.* [38] achieves successful results on Kinetics dataset with 3D-CNNs architectures. Compared to CNN-LSTM models, 3D-CNNs can produce spatio-temporal embeddings with additional convolution operations in time. 3D-CNNs are adapted to gesture recognition tasks in [39], [40]. Yang1 *et al.* [41] propose a architecture utilizing 2D and 3D convolutions to improve image representation quality for SLR.

2.2. Sign Language Recognition Datasets

We are interested in the two datasets. The first one includes over one million images proposed by Koller *et al.* [33]. Its labeling strategy is determined by linguistic research. They claim that the proposed hand shapes cover all of the possible signs in different sign languages. The dataset consists of three different sign languages, namely German (ph), New Zealand (nz) and Danish SL. The major challenge in this dataset is that labeling involves a high level of noise.

The German dataset consists of the training set of RWTH-PHOENIX-Weather 2014 [16]. Danish [4] and New Zealand [5] datasets are fetched from online dictionaries. Those are labeled for 61 classes with one junk class. As seen in Figure 2.1, RWTH-PHOENIX-Weather 2014 is so larger than others and involves many more signs and variations. Note that most of the frames are automatically labeled as junk class.

Table 2.1. Corpus statistics of hand shape datasets.

	danish	nz	ph	TSL
frames	145,720	288,593	799,006	56110
hand shape frame	65,088	153,298	786,750	56110
garbage frames	80,632	135,295	12,256	0
signs	2,149	4,155	65,227	30
signers	6	8	9	7
unqiue hand shapes	61	61	61	45

The second main dataset is Turkish Sign Language (TSL) introduced by Siyli *et al.* [42]. This dataset is too small compared to other datasets. However, the dataset is labeled in an unsupervised manner and hand shapes are determined in a data-driven fashion instead of linguistic research. Moreover, this dataset labels convey semantic meaning and more information compared to others. In other words, labels do not only mention hand shape, but also interactions of hands with other body parts. Some samples from the four datasets are seen in Figure 2.1.

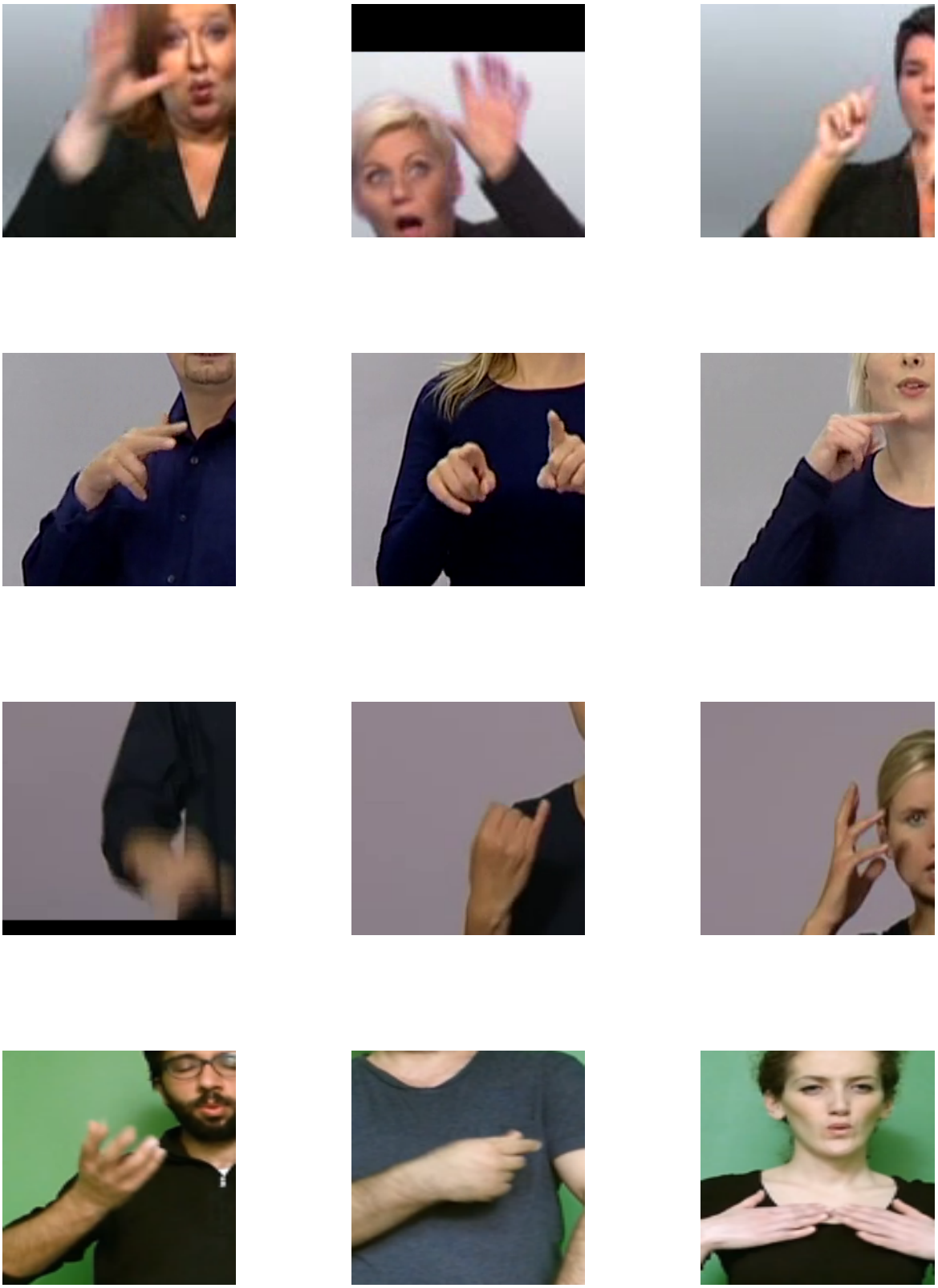

Figure 2.1. Three sample images from each data source. Top to bottom ph, danish, nz, TSL.

Table 2.2. Datasets used in the literature of Sign Language Recognition.

Dataset Name	Year	Signs	Samples	Users	Modality	Environment
SIGNUM [43]	2010	450	33,210 utterances	25	RGB	Controlled
RWTH-PHOENIX-Weather 2014 [16]	2014	1231	6931 Sentences	9	RGB	Wild
DEVISIGN [44]	2015	2000	24000	8	RGB-D	Controlled
ChaLearn LAP Gesture [45]	2016	249	47933	21	RGB-D	Controlled
BosphorusSign [10]	2016	636	24161	6	RGB-D	Controlled
Video-Based CSL [46]	2018	178	25000	50	RGB	Controlled
MS-ASL 2019 [47]	2019	1000	25513	222	RGB	Wild

Apart from these, there are other SLR datasets collected with different methods. The datasets are listed in Table 2.2. SIGNUM [43] is a German Sign Language Dataset involving isolated and continuous videos. It covers 450 signs and 780 sentences. DEVISIGN is a Chinese Sign Language (CSL) dataset consisting of 2000 isolated signs in 24000 videos. In the work of [46], there are 100 continuous sentences which are performed 5 times with 50 signers. This is also collected in a laboratory environment. ChaLearn LAP Gesture dataset [45] is a general gesture recognition dataset. There are 259 gestures performed by 21 different individuals. BosphorusSign [10] is collected with Microsoft Kinect v2 depth sensor that provides RGB along with depth information. It includes signs from health, finances and most common expressions in Turkish. The dataset covers 636 isolated signs with 24161 videos. MS-ASL [47] is a American Sign Language (ASL) dataset. The labeling is done at gloss-level. The videos are captured in several background conditions and there are 222 unique signers are involved in 25513 videos.

2.3. Neural Machine Translation

Neural Machine Translation aims to generate a meaningful sentence in target languages when a sentence in source languages is given. It learns the linguistic properties of source and target languages, word similarities at the same time. Mikolov *et al.* [48] shows the effectiveness of word embeddings in an unsupervised manner. So, NMT models have begun to adapt word embeddings to obtain more informative word representations for the source and target instead of one-hot vectors. However, prediction whole sequences from the beginning sometimes are cumbersome. The reason is that initial predictions of target words are generally noisy. This harms the learning procedure as the models generate word sequences depending on previously predicted words. Bengio *et al.* [49] proposes a procedure that feeds golden labels as predictions to stabilize the learning in the first steps of learning.

It is a very challenging problem to evaluate human translations. In fact, there may be several perfect translations for a given sentence. Also, spoken language grammars

allow sentences to be in different forms with the same words. It is more complicated to evaluate machine translations as they are applied to huge datasets consisting of millions of sentences. There are three popular evaluation metrics for automatic language translation in NMT. The first one is METEOR proposed by Banerjee *et al.* [50]. The formulation is as follows:

$$P = \frac{m}{w_t}, \tag{2.1}$$

$$R = \frac{m}{w_r}, \tag{2.2}$$

$$F = \frac{10PR}{R + 9P}, \tag{2.3}$$

$$p = 0.5(c/u_m)^3, \tag{2.4}$$

$$M = F(1 - p), \tag{2.5}$$

where m_t is the number of matched unigrams between candidate and reference sentences. w_t and w_r are the number of unigrams in candidate and reference sentences. c is the fewest possible unigram chunks adjacent in the reference and candidate. u_m is the number of unigrams in the matched chunks. M score converges 1 if the perfect matching occurs with an infinite number of words. For a perfect match with a three-word sentence, c is 1 and u_m is 3 where M is 0.96. The second method is Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [51]. ROUGE-n means the ratio of overlapped n-grams over total n-grams between reference and candidate translations. The most popular one is Bilingual Evaluation Understudy (BLEU) proposed by Papineni *et al.* [52]. BLEU modifies precision with clipped counts compared to the equations above. If an n-gram is matched several times with n-grams in references, the number of matches is restricted with the maximum number of counts in the references. For example, "a" is a very common stop word in English. If a candidate sentence consists of only this stop word, every n-gram in a candidate can be matched. Therefore, it reduces the score of junk sentences. Moreover, BLEU-n calculates the precision of all n-grams in the candidates and takes a geometric average of them. To enforce a penalty to shorter candidates than references, it multiplies the geometric mean with $\exp(1 - r/c)$ where r is the reference length and c is the candidate length. However, those attempts

are not perfect to evaluate translations for two reasons. First of all, every word has many synonyms that implies the same meanings. Secondly, these approaches do not have a holistic grammatical perspective. A candidate sentence can have a high score in the metrics but does not have a meaning. Hence, two or more distinct metrics are indicated to be more confident about the quality.

While language models are improved by Natural Language Processing (NLP) methods, DL made a huge impact on sequence-to-sequence models. After LSTMs are shown to be effective for language modeling in [53], Kalchbrenner *et al.* [54] led the way for the development of Encoder-Decoder architectures for mapping one sequence to another. This also has brought several advancements in Neural Machine Translation. Encoder-Decoder architectures encode a sequence into a fixed-sized dimensional space. It may be called latent space and every latent representation is decoded into a sequence. In addition to that, Sutskever el al. [55] proposed to use separate RNNs for encoders and decoders. As a result, encoders are specialized in source languages and decoders learn to construct sentences in target languages. This framework has become the state-of-the-art method for translation tasks. Later, multisource NMT is proposed by [56]. A separate encoder is assigned to learn each source language and it is shown that Encoder-Decoder networks can be extended by improving the state-of-the-art results.

DL approaches enable us to train models large models with automatic differentiation. It is an open question of what the machines really learn. These models sometimes cannot focus on important parts of inputs. Mnih *et al.* [57] imposes an attention model in RNNs inspired by human eyes. This method tries to sum numbers shown in images by focusing on the different parts of them at each step. It is shown that the model learns to eliminate unnecessary parts of images thanks to the additional hard attention mechanism. Later, this approach is adopted by image captioning research. Xu *et al.* [32] compares hard attention mechanisms with soft attention ones by revealing where the models focus on in describing an image. Soft attention creates a context from different parts while hard attention uses only one part at a time. Later,

attention-based models are also adapted into video captioning and successful results are obtained [58].

Later, those attention-based models attract NMT researchers as a fixed-sized latent dimension is not enough to encapsulate all of the information in sentences. This leads to the development of Attentional RNN Encoder-Decoder architectures. Bahdanau *et al.* proposes an attentional sequence-to-sequence model that generates a different context vector from a sentence at each time. Simply, the architecture is designed to focus on different input words while generating each target word. This is similar to the human translation approach. After extracting the context of the sentence, we can decode words considering relevant parts in the source sentence. In DL perspective, it provides relaxation to the network that improves learning by refining irrelevant parts in sentences. Later, the work in [59] improves Bahdanau Attention mechanism and proposes a local attention method in addition to the global one. The network takes all encoder outputs into account in the global attention while the local attention chooses a position and extracts the context vector from a window near the position. It is very similar to the hard attention proposed by [57], but this method is fully differential. Therefore, it does not require further Bayesian optimization techniques.

Contrary to the previous approaches, Vaswani *et al.* [60] proposes Transformer constructed with self-attention units instead of RNN units. They preserve Encoder-Decoder architectures and self-attention units reveal the relation between input words to understand the word's real meaning in the given sentence. This approach uses feed-forward networks therefore an extra positional indicator is needed. Otherwise, the network has no notion of the order in the sequence. We know that the order of words plays a crucial role in language modeling. Therefore, they feed the position of the words with sine and cosine functions. This architecture has become state-of-the-art in NMT as well as many NLP applications [61].

Based on those advancements, Neural Sign Language Translation research has emerged. Camgoz *et al.* [2] formalizes the first architecture with the attention-based

models. The architecture requires an additional effort for obtaining tokens. In NMT, tokens are produced with words or characters, but it is not clear to obtain tokens for sign language translation. The work introduced three different tokenization variants. The first one is gloss to sentence translation which is very similar to spoken language translation. The Second is sign to gloss to sentence translation. Sign to gloss is performed by another network. The network is needed to be trained separately on sign to gloss alignment. Therefore, they use the network proposed by Koller *et al.* [36], but it is trained on the same dataset with the translation dataset. The last one is sign to sentence translation. Contrary to the first two variants, this converts frames to sentences without any explicit intermediate representation like glosses. It is formalized as two parts sequence-to-sequence model and tokenization part. The sequence-to-sequence model is adapted from NMT. Tokenization is the tricky part and requires further research. In the paper, a pre-trained CNN and NMT model is trained in tandem. the CNN extracts features from frames and features are concatenated in time. Naturally, the resulting model has longer sequences compared to the gloss-to-sentence model. But, the attention mechanism seems to handle the long sequences.

In addition to that, Ko *et al.* [62] proposes a new tokenization method that uses human keypoint coordinates instead of CNN last layer features. Later, Sign Language Transformers are proposed by [1]. In this study, glosses and sentences are learned jointly, yet requires tedious gloss annotations. It uses common representations of images as tokens. The result exceeds golden gloss to sentence translation. It is inferred that glosses are not perfect tokens for SLT. More relaxed representations can attain higher scores instead of hard representations like glosses.

2.4. Neural Sign Language Translation Datasets

We use RWTHPHOENIXWeather 2014T introduced by Camgoz *et al.* [2]. It is an extended version of RWTHPHOENIX-Weather 2014. This is the most challenging SLT dataset publicly available to the best of our knowledge. They published this dataset with full-frames that involve faces, hands and upper-body parts of signers. We add

two other settings. We identify body parts with OpenPose [63], [64], [65]. For the first settings, we use only right hand crops, concatenation of left and right hands are prepared for the other. The samples is seen in Figure 2.2.

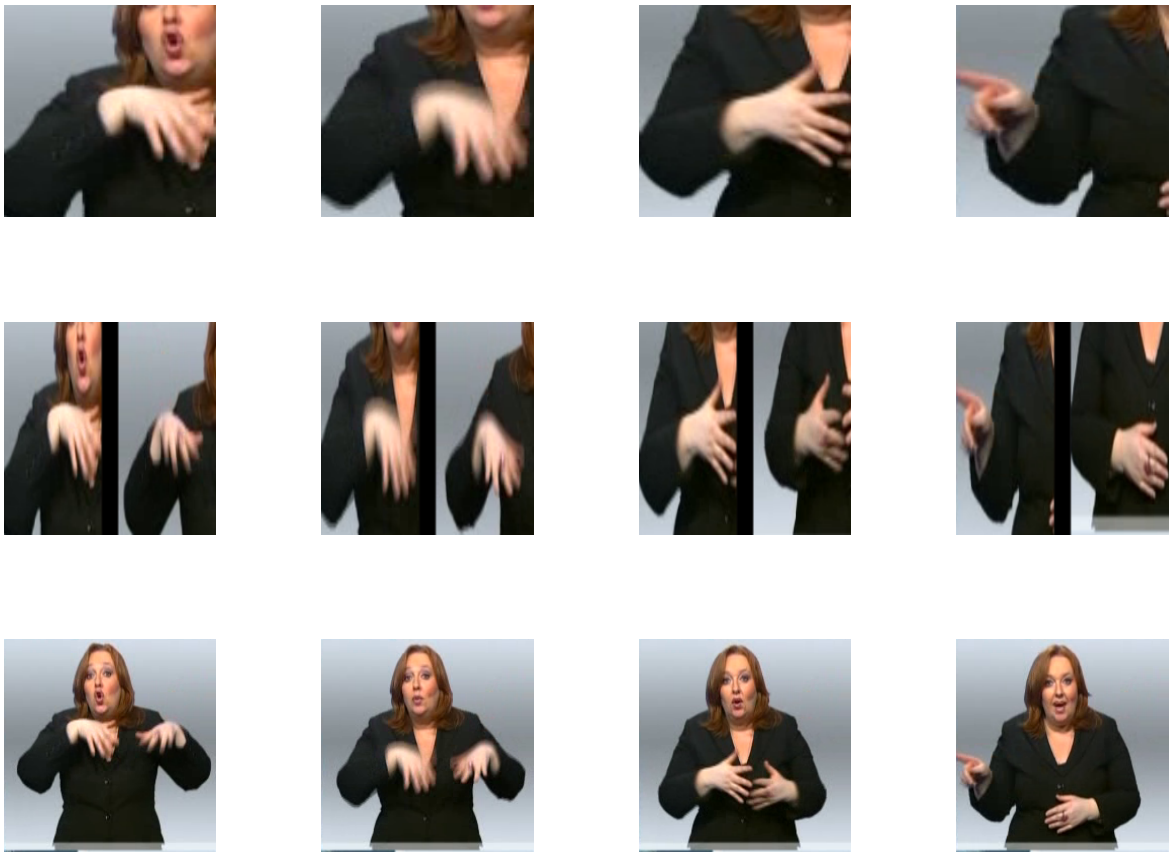

Figure 2.2. Three sample images from right hands, both hands and full frame from top to bottom.

The dataset is divided into train, test and validation. The train consists of 7096 video and sentence pairs. Validation and test data sets consist of 519 and 642 samples, respectively. The unique word size is 2891 and the unique gloss size is 1235.

3. LEARNING TOKENIZATION

In this section, we present the overall architecture of Neural Sign Language Translation. We propose different tokenization methods to obtain string sign embeddings. First, we use a 2D-CNN trained for a task relevant to SLR. As a prior knowledge, we know that hand shapes are very important to understand what signers say. The literature suggests that possible hand shapes are similar in all sign languages, but their meanings differ from language to language. Therefore, learning hand shapes may be beneficial to obtain a generic tokenization layer. Secondly, 3D-CNNs have become very successful in action recognition tasks. We adapt a 3D-CNN into the tokenization layer to enable knowledge transfer from action recognition. The major advantage is that sign language videos involve so many redundant frames that temporal convolution may decrease training time by eliminating unnecessary information.

3.1. Neural Sign Language Translation

In this section, we introduce overall NSLT schema which mainly consists of two parts, tokenization and Neural Machine Translation (NMT). NMT handles sequence to sequence modelling by mapping input tokens to output words. When the sequence length is very long, attention mechanisms tackle long term dependencies. There are three major choices for attention methods, Bahdanau [66], Luong [59] Attention and Self-Attention [60]. The first two are designed to function on LSTMs, GRUs [67] or Vanilla RNNs and the other requires a novel architecture called Transformers. The parts of NLST is seen in Figure 3.1.

3.1.1. Sign Embeddings

We think the tokenization part is essential to adopt new NMT techniques into SLT. If we find appropriate representations of videos, the developed methods in NMT are effectively applied. Different from spoken language translation, the tokenization

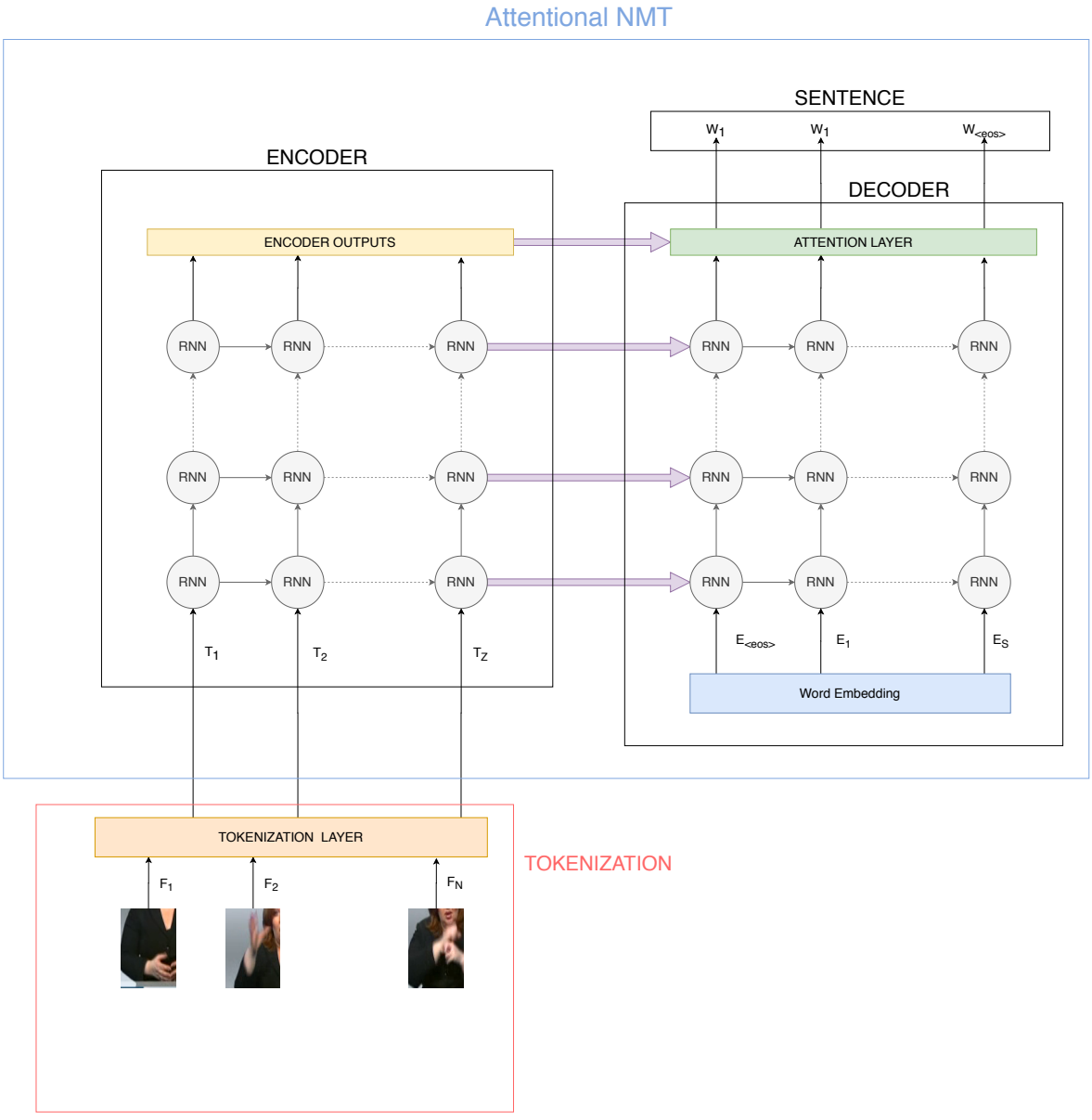

Figure 3.1. Overall schema of Neural Sign Language Translation.

part in NSLT also deals with the visual properties of SLs. Therefore, the tokenization part does not have to produce discrete tokens. In NMT, sentences are represented with discrete words or n-grams. Those tokens are converted to continuous space to provide more semantic information. However, in NSLT, the raw input is already continuous. The frame-to-gloss tokenization approach extracts discrete tokens, glosses, from videos to mimic successful NMT systems. We claim that this process can cause information loss. A transformation from continuous to discrete space is not required as they are again transformed into continuous space eventually. Also, small perturbations in discrete domains may amplify errors in sentence levels.

Our approach is based on strong sign embeddings. A sign can be represented with a single vector or concatenation of several frame representations. Additionally, the last layer of 3D-CNNs is also a good candidate as signs consist of consecutive frames. It is important that our tokenization approach does not require to represent strictly signs as the NMT module can combine distributed information implicitly. Therefore, we extend the scope of tokenization and our frame-level tokenization produces sign level embeddings to be replaced with word embeddings in NMT. Conversely, sign embeddings can be learned from other data sources. The word embedding is a single dense layer, but tokenization to sign embeddings should involve CNN like architectures. As they are very deep, they cannot be learned easily with sentence-video pairs. We know that SL research generally suffers from limited data. Considering all the issues, we claim that frame-level tokenization is a more proper approach and enables additional supervision, unlike gloss level tokenization. In the following subsections, we propose different frame-level tokenization approaches and compare them.

3.1.2. Attentional NMT

Assume that we have an optimal tokenization mechanism that produces Z tokens, $t_{1:Z}$, for a given N length frame sequence, $x_{1:N}$. Note that sequences can be reversed after the tokenization, ψ, as suggested in [55] to shorten the average distance between input tokens and decoded words. This helps to decrease long term dependencies while

decoding. The problem is reduced to learning a sequence to sequence model between $t_{1:Z} \rightarrow w_{1:S}$ where $w_{1:S}$ is the sequence of target words. Encoder-Decoder, $g \circ f$, architecture is used to model this mapping between input and target. The encoder, f, produces outputs, $o_{1:Z}$ and hidden states, h_f. The outputs are used to generate more meaningful contexts. The hidden states are the only way to bridge the encoder and decoder in the absence of an attention mechanism. The encoder produces the same number of outputs as the number of tokens. Also, the last hidden states are used to initialize the hidden states of the decoder. In this manner, the context can be conveyed to the decoder. In short, the encoder produces Z outputs and the same number of hidden states as the encoder layer size.

The resulting $o_{1:Z}$ is used in attention mechanisms. The decoder, g, generates its own outputs, $h_{1:S}$ and produces the target words, $w_{1:S}$, by using attention mechanisms. The attention mechanism develops a similarity notion between the encoder outputs, $o_{1:Z}$ and the decoder outputs, $h_{1:S}$. The decoder is designed to provide the most likely target word sequence if a token sequence is given. To construct a proper sentence, the decoder uses the encoder outputs, $h_{1:t}$, the generated words, $w_{1:t}$ and all the encoder outputs, $o_{1:Z}$ to generate the word at step $t+1$. Eventually, the network decomposes the conditional probability of a word at a time. In the following equations, the Encoder-Decoder formulation is stated:

$$t_{1:Z} = \psi(x_{1:N}), \tag{3.1}$$

$$o_{1:Z}, h_f = f(t_{1:Z}), \tag{3.2}$$

$$w_{1:S}, h_{1:S} = g(o_{1:Z}, h_f), \tag{3.3}$$

$$p(w_i|w_{<i}, t_{1:Z}) = g \circ f(t_{1:Z}), \tag{3.4}$$

$$\log p(w_{1:S}|t_{1:Z}) = \sum_{i=1}^{Z} \log p(w_i|w_{<i}, o_{1:Z}). \tag{3.5}$$

There are two main attention mechanisms when the Encoder-Decoder architecture is constructed with recurrent neural networks. The first one is Bahdanau and

the second is Luong attention mechanism. Figure 3.2 displays the overall architectures of the two attention mechanisms. Bahdanau attention creates context representations and generates the words as follows:

$$e_{1:z} = \alpha(h_{t-1}, o_{1:z}), \tag{3.6}$$

$$a_i = \frac{\exp(e_i)}{\sum_{k=1}^{z} \exp(e_k)}, \tag{3.7}$$

$$c_t = \sum_{i=1}^{z} a_i o_i, \tag{3.8}$$

where c_t is fed into the last layer of RNN cell. α is an arbitrary distance function between two vectors.

Luong Attention mechanism obeys the following equations. α is an arbitrary distance function between two vectors. θ is a dense layer with softmax. y_t is fed into the first layer of the decoder with the word embedding such that

$$e_{1:z} = \alpha(h_t, o_{1:z}), \tag{3.9}$$

$$a_i = \frac{\exp(e_i)}{\sum_{k=1}^{z} \exp(e_k)}, \tag{3.10}$$

$$c_t = \sum_{i=1}^{z} a_i o_i, \tag{3.11}$$

$$y_t = \theta(c_t, h_t). \tag{3.12}$$

The main difference between them is that Bahdanau uses the decoder output of the previous time step and the other uses the current decoder output. The second difference is that Luong feeds its decision over attention, y_t into the first layer.

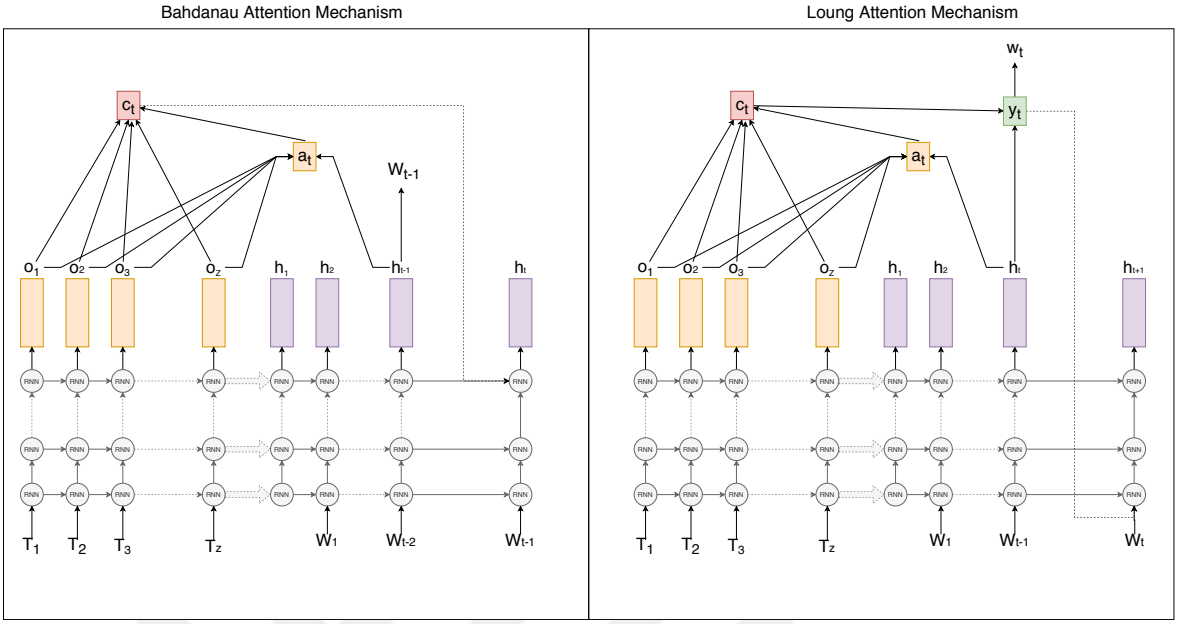

Figure 3.2. Illustration of Bahdanau and Luong attention mechanisms.

3.2. Tokenization with 2D-CNN Trained on Hand Shape Classification

Hand shapes play a crucial role in interpreting the meaning of sign languages. We claim that a tokenization layer can benefit from this prior knowledge. Firstly, hand crops should be extracted to be processed for spatio-embeddings. There are several options for this challenge. A hand tracker or a hand detection network may be deployed. Openpose [65], [64], [63] is a strong human keypoint detector functioning in different conditions. Also, Ko *et al.* [62] shows it is very successful as a tokenization layer with a different formulation. OpenPose naturally eliminates some frames as it is not able to detect hands in blurry images. This effect may decrease the noise level in training but can result in ignoring some important frames. We consider this very unlikely as the training set of OpenPose includes high variation in terms of light conditions and camera views. The overall architecture is seen in Figure 3.3.

Koller *et al.* [33] introduces a big dataset and a method dealing with weak labeling. Most of the dataset is common with the translation dataset. We suspect that a network trained on this method and dataset may not be robust for datasets from other domains.

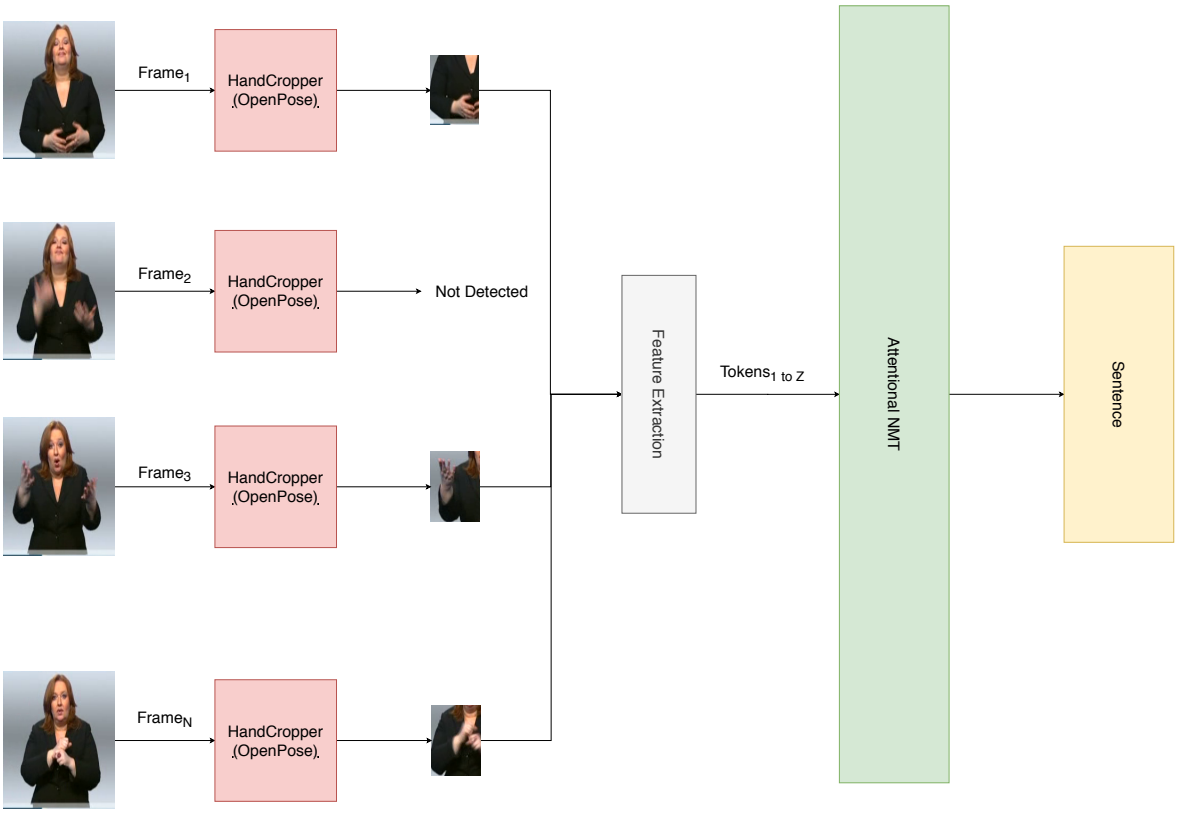

Figure 3.3. Tokenization with a 2D-CNN trained for hand shape classification.

Therefore, we propose two other solutions to deal with that while maintaining domain independence of the network. The first one is training a network on multiple tasks and datasets. It may force the tokenization layer to learn a more general feature space. The second is using domain adaptation techniques explicitly. If there is some data from the translation dataset, this technique may enable the feature extractor network to learn domain-independent features. In this manner, we may have an improved tokenization layer for unknown sign language data.

3.2.1. Multitask Learning

The aim of using Multitask Learning is to deal with two main challenges in SLR. The first one is that data is weakly labeled. It leads the feature extractor network to learn a bad latent representation space. The second is that the environmental conditions where data is collected make the network very sensitive to perturbations in input space. Small changes in the background or camera angles can lead the network to be saturated. The network cannot produce robust features due to this. As seen in Figure 3.4, the architecture is designed to obtain a common latent representation for different datasets.

As mentioned in Section 2.2, we have different data sources and different labeling approaches. Furthermore, the datasets vary in terms of the noise level. They may not be seen as sufficiently strong signals to train a good tokenization layer if used alone. However, we claim that multitask learning is a proper way to combine those weak signals to train a strong network. Therefore, the TSL dataset may be used as a regularization factor on training danish and NZ datasets since the TSL dataset is very small. This may provide two different advantages. Firstly, it may reduce the negative effect of incorrect labels in the big dataset as the TSL dataset is more reliable in terms of annotation. Therefore, the TSL dataset imposes a regularization effect on the network which prevents it from overfitting to the noisy labels. The second advantage is that its labels involve contextual information meanwhile labeling the big dataset

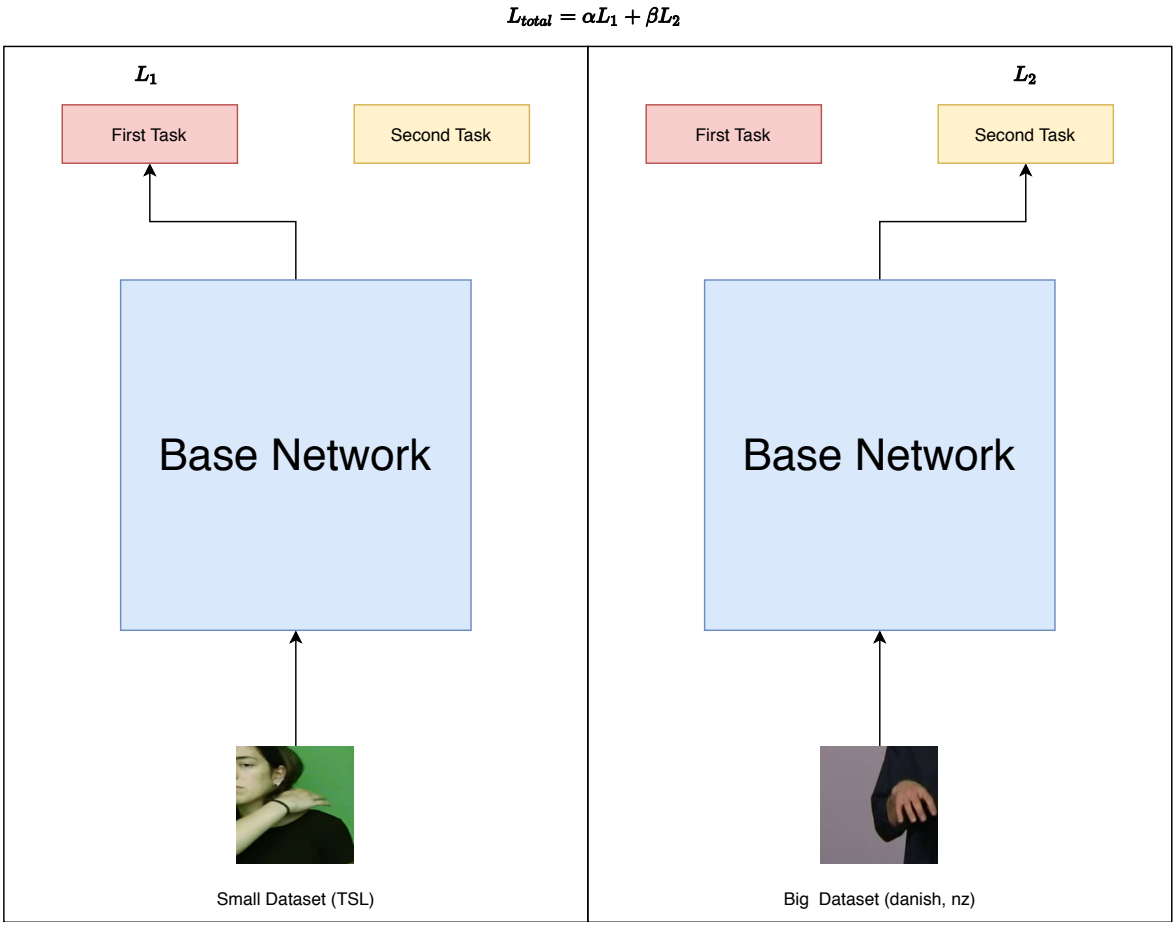

Figure 3.4. Multitask learning framework.

only concern about context-independent hand shapes. In other words, the small data involves syntactic and semantic meaning contrary to the big.

Another major benefit is that combining different datasets naturally makes the network robust to different environment conditions since they increase the diversity of signers, background and camera settings. We expect that this approach attains better results without any information from the target domain.

However, multitask learning requires additional effort to attain a balanced learning schedule between tasks. Assume that we have two different tasks and we update weights after one batch from two tasks are iterated. The loss function is

$$L_{total} = L_1 + L_2. \tag{3.13}$$

This means that both tasks are equally weighted. Generally, two tasks are unlikely to be learned at the same rate from a DL perspective. Moreover, sometimes loss function scales of separate tasks differ dramatically. Even if the loss function is the same, the number of classes can be different and that may generate a huge imbalance in the loss function. Therefore, the loss function may be arranged as

$$L_{total} = L_1 + \beta(t)L_2. \tag{3.14}$$

The coefficient of the second function may be dependent on the iteration number t or a constant value. The first loss term's coefficient is always 1 as it can be arranged with the learning rate.

3.2.2. Domain Adaptation

We know that every sign language dataset involves nearly the same hand shapes, but some domain shifts occur between datasets. It can be fixed with domain adaptation techniques, however, some annotation or incorporation of synthetic data is needed.

Ganin *et al.* [68] proposed an unsupervised method that can be used on deep architectures. It may be useful to apply this method while training the feature extractor network. This enables the network to learn domain invariant features while gaining the capability to distinguish the classes. We expect this approach to decrease classification error in the target domain. This may contribute to an increase in translation quality. The overall architecture is illustrated in Figure 3.5.

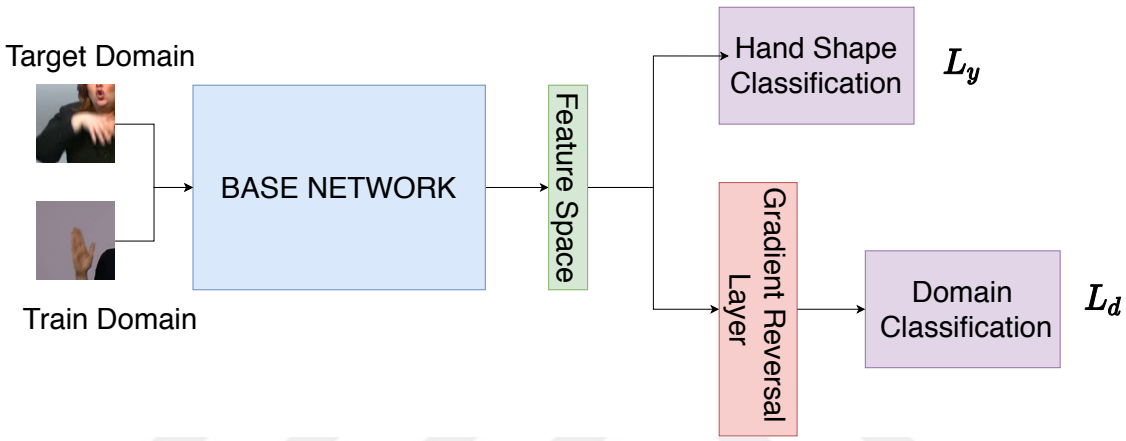

Figure 3.5. Illustration of domain adaptation with Gradient Reversal Layer.

The gradient reversal layer (GRL) and domain classification layer are the modifications to the normal architecture. Labels of translation dataset frames are unknown, but their domain label is different than training data. Hence, the method may approximate the feature space distribution of the training data to the translation data. The gradient reversal layer is identity function in the forward pass. But, in the backward pass, it reverses the gradient with an amplifying factor such that

$$GRL(x) = x, \tag{3.15}$$

$$\frac{\delta GRL(x)}{\delta x} = -\lambda(t)\cdot \tag{3.16}$$

This arrangement means that the learned features to distinguish the domain are diminished with a gradient reversal trick. The λ is gradually increased from 0 to 1 as

learning at the initial steps may be harmed by noise caused by the gradient reversal layer.

3.3. Tokenization with 3D-CNN Trained on Action Recognition

Instead of spatio-embeddings with 2D-CNNs, 3D-CNNs can produce spatio-temporal embeddings as a tokenization layer. Carreira *et al.* [38] proposes a novel 3D-CNN architecture providing promising results on human action recognition. Also, the studies in [69], [70] show that 3D-CNNs are suitable for videos in sign languages.

We know that consecutive frames are very similar to each other. Therefore, temporal convolution may prevent sequence-to-sequence models from handling very long sequences. Furthermore, 3D-CNNs may enable information transfer from human action recognition. In this thesis, we emphasize that hand shapes are very important for sign languages. This method is able to take human action into account along with hand shapes. The overall mechanism can be seen in Figure 3.6.

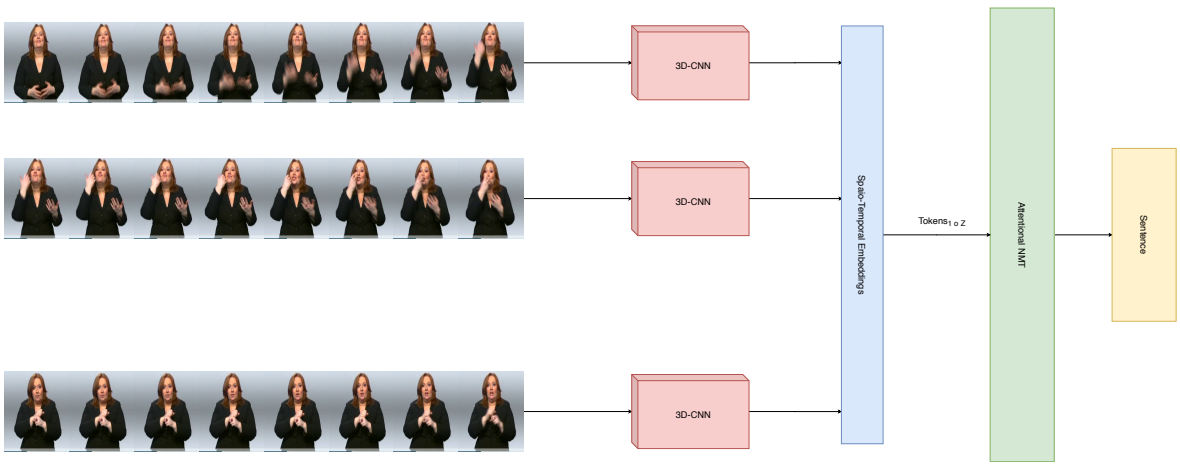

Figure 3.6. Illustration of Tokenization with 3D-CNN.

4. EXPERIMENTS & RESULTS

For the translation dataset, we use the dataset introduced in Section 2.4. Tensorflow [71] version of OpenNMT [72] library is used for sequence-to-sequence modeling. PyTorch [73] is utilized to import pre-trained models and construct customized networks. Also, end-to-end learning experiments are performed with the code published by Camgoz *et al.* [2]. To evaluate our translation results, we used the same metrics in [2], BLEU (BLEU-1, BLEU-2, BLEU-3, BLEU-4) [52] and ROUGE (ROUGE-L F1-Score) [51], to enable consistent comparisons.

Each model is trained up to 30.000 iterations. Models parameters are saved every 1000 iterations and resulting parameters are average of the last five saved parameters. This reduces biases in selecting the best models. Model hyperparameters are chosen to make the models as simple as possible to make the experiments generic. Concerning the computational effort to train a translation model, it is not possible to perform an extensive grid search on hyper-parameters. The hyper-parameters and other details are listed in the Appendix.

4.1. Input Analysis of Implicit Tokenization Layer

We claim that the end-to-end learning scheme proposed by Camgoz *et al.* [2] is not effective due to small data size. The unique word size is 2891 and the unique gloss size is 1235 whereas there are a total of 7096 sentence-video pairs. Despite the fact that RWTH-PHOENIX-Weather 2014 is seen to be sufficiently large for SLR, SLT requires more sentence-video pairs. As a prior knowledge, we know that hand parts of frames involve the most crucial information to understand signers. However, sign languages are very context-dependent languages so general frame information is also important.

We prepare three different versions of the translation data as illustrated in Figure 2.2. The first one is full frames, the second one is only right hand crops and the last one

is the concatenation of both hands. The reaction of implicit tokenization to the different input types provides clues about the effectiveness of this method. If the method attains the best results with full frames, it is claimed that the implicit tokenization layer is able to obtain strong feedback signals from the sequence-to-sequence model. It means that the tokenization layer can infer the information both from hands and the context of the frame.

AlexNet is used to extract spatio-embeddings. To test the claims above, we carry out six experiments. In the first three, the experiments are conducted with the end-to-end learning scheme, which means that AlexNet is fed with gradients coming from the sequence-to-sequence model. In the others, AlexNet is not tuned further with the NMT model.

Table 4.1. Test results of implicit Tokenization.

	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Full-Frame & End-to-End	30.70	29.86	17.52	11.96	9.00
Both Hands & End-to-End	**34.53**	**33.65**	**21.01**	**15.19**	**11.66**
Right Hand & End-to-End	31.89	30.57	18.67	13.19	10.25
Full-Frame & Only NMT	29.25	30.91	17.42	11.80	8.85
Both-Hands & Only NMT	30.17	30.53	17.83	12.32	9.53
Right-Hand & Only NMT	**31.77**	**32.67**	**19.76**	**13.79**	**10.50**

The results raise doubts about the effectiveness of implicit tokenization. Full frames give the worst result while using both hand crops outperform the others drastically as seen in Figure 4.1. It is interesting that tuning the tokenization layer has nearly no effect on full frames and right hands. Despite that, the tokenization layer is improved with both hands where BLEU-4 score increases from 9.53 to 11.66. The reason is that both hands have an unnatural discontinuity which is very distinct from images in ImageNet. This effect prevents the tokenization from being stuck in a local optimum.

This result suggests that feedback signals coming from the sequence-to-sequence model are not sufficient to train a strong tokenization layer. However, the tokenization layer is suitable to be improved with external information. Moreover, the end-to-end learning needs a huge amount of computational power while sustaining no positive effect except both hands setting. The end-to-end training requires 50 times more resources to converge within the same duration as the frozen one. In the right hand setting, it is also inferred that the tokenization layer cannot be specialized on hand shapes with the joint learning approach.

4.2. Tokenization with a Frozen CNN

In this section, we aim to compare different frame-level tokenization approaches. First, we find the most successful attention mechanism to conduct the rest of the experiments. Later, we begin with a frozen 2D-CNN trained on ImageNet. We choose a more recent and efficient architecture instead of AlexNet. These experiments are considered as baseline. We expect that the success of the tokenization layer increases as we utilize various prior knowledge about sign languages. We test human keypoint features on our dataset with different human body parts. Furthermore, we check the validity of our approaches. Lastly, we test if human action recognition is related to sign language translation by using a 3D-CNN as a tokenization layer. Illustration of the overall schema is seen in Figure 4.1.

4.2.1. Attention Search

We conduct experiments to find the most appropriate attention mechanism. There are three options, Bahdanau, Luong and Self-Attention. We choose a 2D-CNN trained on ImageNet as it is the most basic form for a tokenization layer. If a more sophisticated tokenization layer is used, pure contributions by attention mechanisms may not be observed. Inception v3 [74] is used without the classification layer as the base network. Additionally, full-frame input setting is used as it involves all human body parts compared to other settings, right hands and both hands. For each of them,

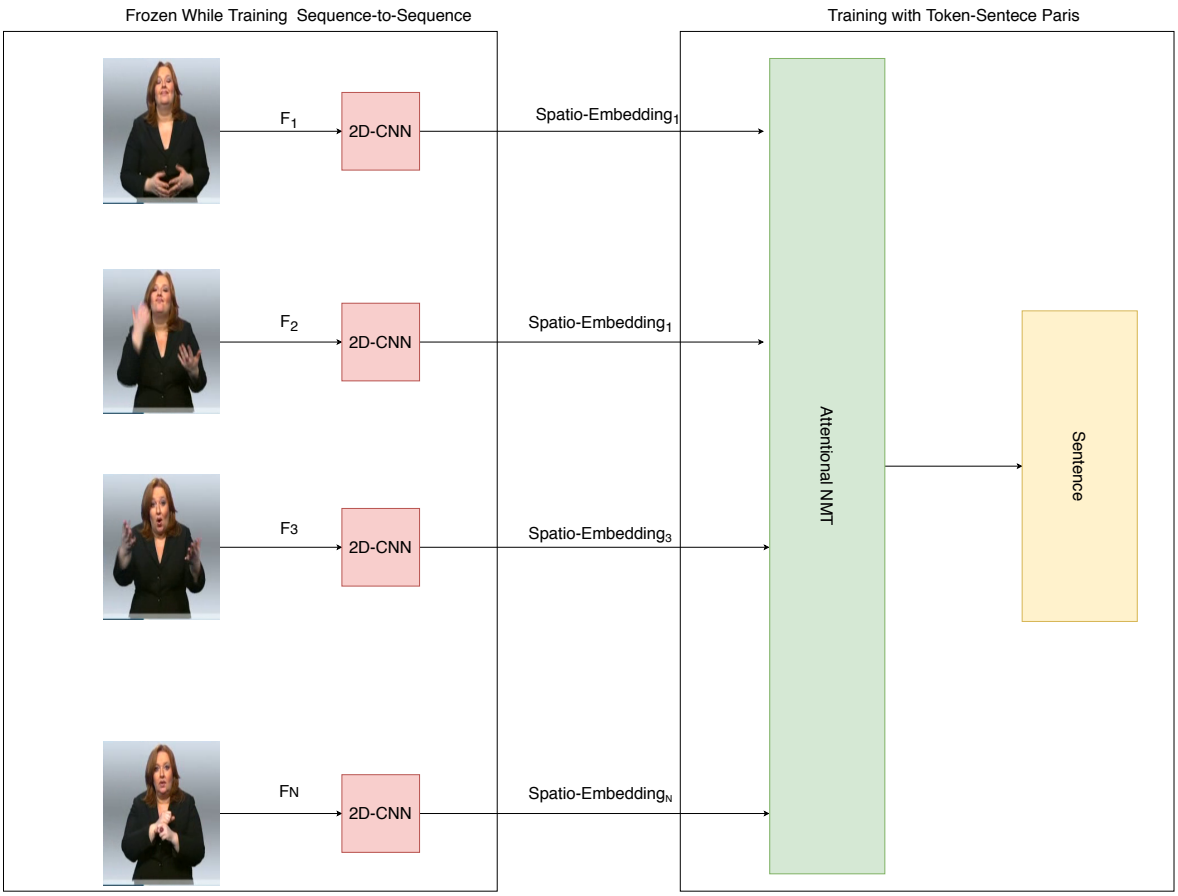

Figure 4.1. Illustration of Tokenization with frozen 2D-CNN.

the most accurate hyper-parameters are searched. For the first two, the search is initiated with the parameters suggested by [2]. Self-attention is applied for the first time in this dataset. Therefore, we start the search with the parameters in [60]. The schema of this method can be seen in Figure 4.1.

Table 4.2. Comparison of attention mechanisms.

Model	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Bahdanau	**29.41**	**30.46**	**17.79**	**12.36**	**9.40**
Transformer	28.20	28.66	16.33	11.15	8.43
Luong	26.94	27.46	15.12	10.38	7.93

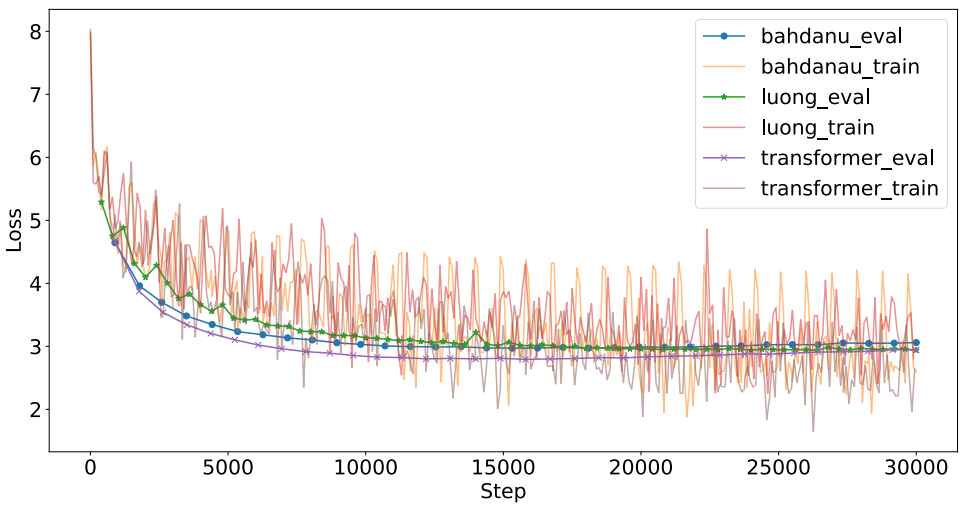

Figure 4.2. Convergence graph of the models.

All of the models are converged and provide meaningful translations. Evaluation and training losses are depicted in Figure in 4.2. The results indicate that Bahdanau outperforms others in all metrics as seen in Table 4.2. The most important indicator is the BLEU-4 score. Evaluation set BLUE-4 scores are seen in Figure 4.3. These results imply the same conclusion with test results. This result is consistent with our claim

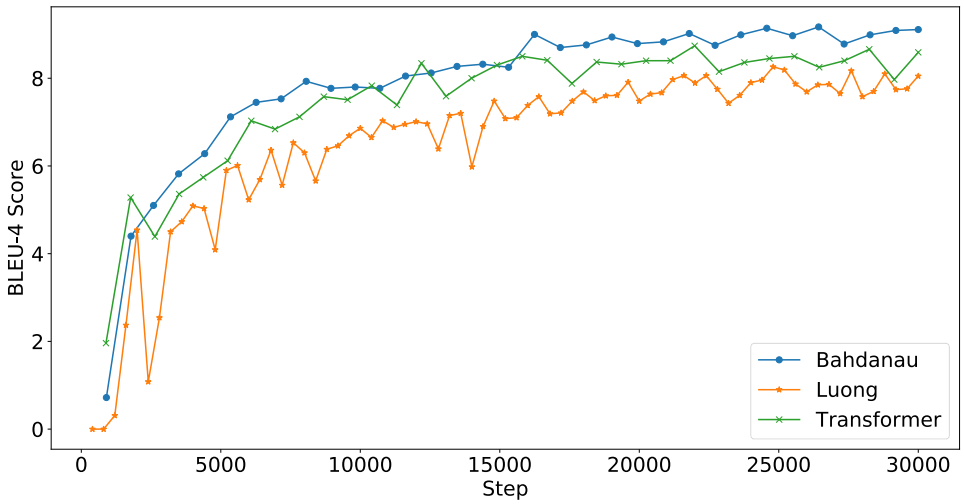

Figure 4.3. BLEU-4 evaluation score graph of the models.

that large models can memorize the dataset easily. NMT with Bahdanau consists of fewer parameters compared to Luong and Self-Attention model.

4.2.2. Transfer Learning From ImageNet

Inception v3 is our base feature extractor network. Without further training and any modification, spatio-embeddings before the classification layer is used as the tokenization layer. These experiments are aimed to determine the baseline of tokenization for the rest of the experiments. Imagenet is a very diverse dataset consisting of 1000 classes and three million images. Most of the applications in Computer Vision networks are initialized with parameters inferred from the ImageNet dataset. The result may help us to understand which parts of frames are important since it is considered as the most general feature extractor.

Contrary to the first three rows of Table 4.1, the result in in Table 4.3 indicates that inputting full frames is the most effective approach. However, there are no big

Table 4.3. Test results of different input settings.

	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Full Frames	29.41	**30.46**	**17.79**	**12.36**	**9.40**
Both Hands	**29.62**	30.24	17.46	12.14	9.33
Right Hands	29.09	29.51	17.03	11.79	9.06

differences between the approaches as illustrated in Table 4.3. 9.40 BLEU-4 score is attained without further effort compared to the result in [2] (9.58 BLEU-4)

The results are in contradiction with the last three rows of Table 4.1: The right hands are the most successful with AlexNet whereas full frames attain the best scores with Inception v3. AlexNet has 4096 feature size and Inception has 2048. Inception architecture is designed with parallel filters that combine different information from varying windows sizes. AlexNet architecture has three fully connected layers that make overfitting very likely. Hence, Inception can utilize all information from full frames and exceeds AlexNet scores (8.85) in a full-frame setting.

4.2.3. Human Keypoints

OpenPose is a very robust keypoint extractor from both human bodies, faces and hands. Human keypoints may be converted into tokens. Ko *et al.* [62] suggests a pre-processing method instead of using keypoints directly. The method is as follows

$$T_x = \frac{V_x - \hat{V_x}}{\sigma(V_x)} \tag{4.1}$$

$$T_y = \frac{V_y - \hat{V_y}}{\sigma(V_y)} \tag{4.2}$$

where V is the keypoint vector. The process describes the separate standard normalization of x and y coordinates.

The upper body, right and left-hand keypoints are extracted from the translation dataset. To mimic the full frames setting, all of the keypoints are normalized within the body parts according to the method. This tokenization approach may have two advantages. The first one is that the training set of OpenPose is very challenging; so the method is applicable under a wide range of conditions. It automatically eliminates backgrounds and irrelevant parts of frames. Moreover, the interaction of the body parts is explicitly imposed. However, we know that subtle changes in hand shapes may cause big interpretation differences in sign languages. Therefore, this method may not be capable of identifying hand shapes since the outputs of finger positions are generally very noisy as seen in Figure 4.4.

Table 4.4. Test results of Tokenization with keypoints.

	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Body & Both Hands	**32.85**	**33.18**	**20.39**	**14.26**	**10.92**
Both Hands	31.47	31.51	19.08	13.26	10.23
Right Hands	30.65	30.69	18.49	12.89	9.91

This method improves the baseline in every setting as seen in Figure 4.2.3. Also, it is consistent with early findings that hands are the most important factor, yet other body parts are also helpful to understand signers. The third row shows that this method increases BLEU-4 by nearly 1 point compared to the last row of Table 4.3. Therefore, keypoints are more accurate than the pre-trained Inception network to represent hand shapes. These findings emphasize that hand shapes should require special attention. A simple representation of hand shapes with coordinates gives promising results even if the OpenPose outputs involve considerable noise.

4.2.4. Hand Shape Learning

In this section, we want to show the effectiveness of hand shapes. The previous results show that any prior information about hand shapes improves the quality of translation. We adapt the proposed techniques in Chapter 3.2.1 to deal with weak

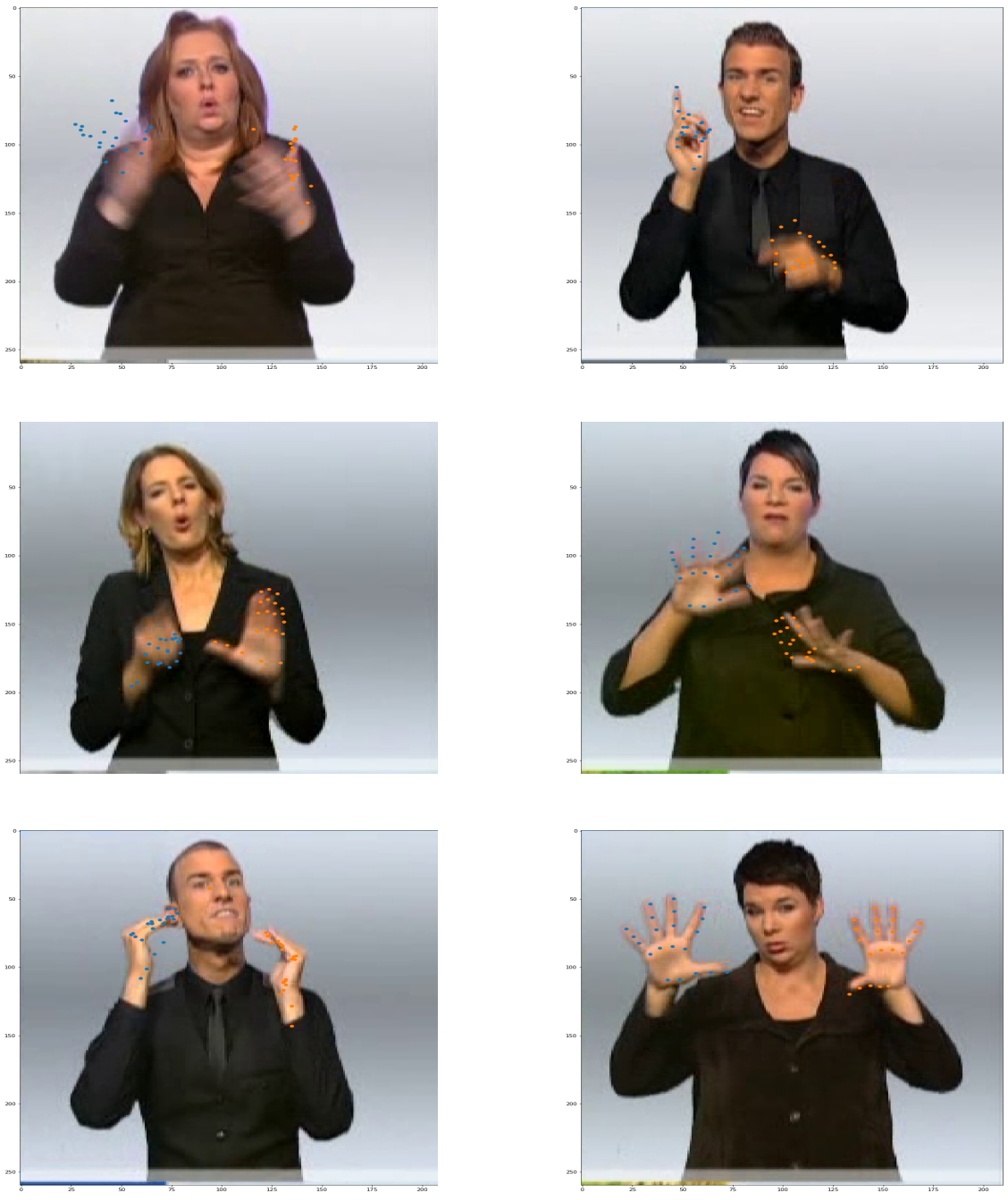

Figure 4.4. Samples from OpenPose output coordinates.

annotation and domain shifts. We know that the big datasets are weakly labeled and small datasets do not cover most of the hand shape variations. Therefore, we train the network in a multitask manner. We check whether it enables the network to have a stronger feature space. First, we test this network on a small portion of the translation dataset. The test dataset is labeled manually by Koller *et al.* [33]. Later, it is deployed as the feature extractor in Figure 3.3.

We also train a network with the unsupervised domain adaptation method explained in 3.2.2. The network is tested on the small test set to check if the technique is applied successfully. Later, the experiments reveal whether mitigating domains shifts has an important effect on translation.

Lastly, we aim to find a way to reduce token sequences in length. Instead of sampling from frames, 3D-CNNs are applied to hand shape classification as seen in Figure 4.5. As the data is coming from videos, we take a window around the desired hand shapes as inputs for 3D-CNNs. This technique may identify the desired hand shapes from sequences. We know that some frames are meaningful, but some are blurry and irrelevant in frame sequences. In these experiments, we use the big dataset with the architecture proposed by Tran *et al.* [75]. The network is initialized with pre-trained weights on Kinetics.

The datasets (New Zealand, Danish) in Section 2.2 are used to apply the techniques, expect German Sign Language, ph. The reason for the omission is that the translation dataset is the modified version of ph. Therefore, it may harm our proposed approaches as training on those frames can cause memorization. We set up our experiment as if we do not have access to the target domain. We combine Danish and New Zealand datasets as the big dataset. The dataset from Turkish Sign Language is considered as the small dataset. Our baseline network is trained on only the big dataset to observe the effects of the proposed techniques.

The experiments are performed with only right hand crops in this section. We want to obverse the effectiveness of hand shape learning on tokenization layers. For simplicity, We ignore left hands for translation. The results reveal which methods provide additional supervision to the tokenization layer. Search for combining hand representations is left as future work.

For consistency, the base CNN network is again Inception v3. For all tasks, the network is initiated with parameters trained on ImageNet to accelerate the training process. All the datasets are collected from sign language videos. Therefore, the distribution of hand shapes is highly imbalanced. We deploy a mini-batch sampler to overcome this challenge.

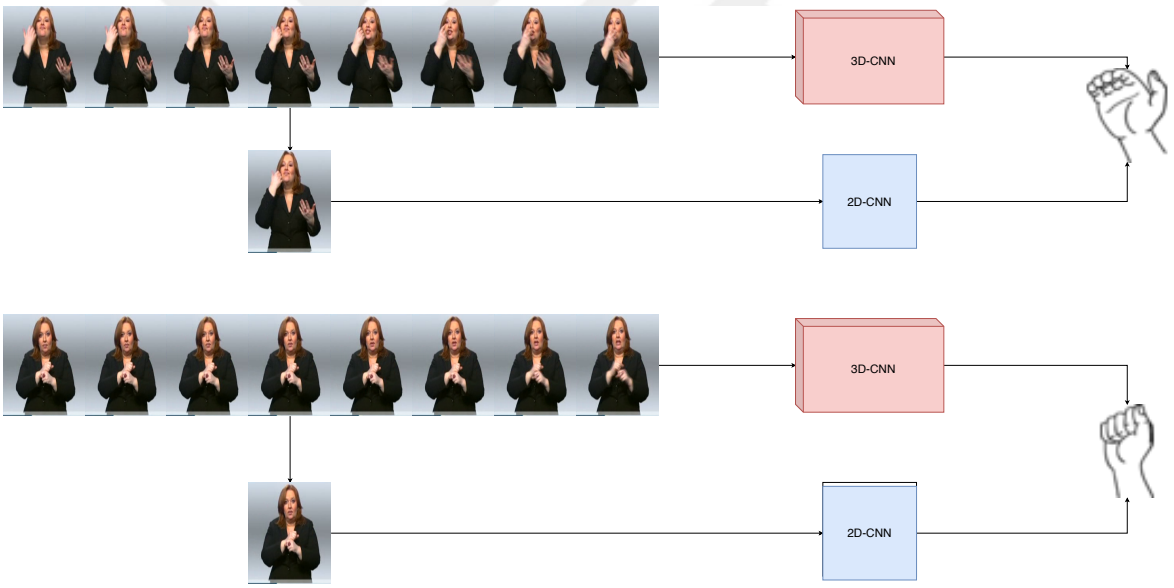

Figure 4.5. Learning schema of hand shapes with a 3D-CNN.

Table 4.5. Hand shape classification results on the the target domain.

	Top-1	Top-5
Baseline	65.49	89.49
Data Augmentation	67.07	90.65
Multitask	76.77	92.88
Domain Adaptation	**78.74**	**94.73**

Data augmentation in Table 4.5 refers to the baseline method with random color jitter. The results indicate that Multitask Learning and Unsupervised Domain Adaptation increase top-1 accuracy to 11.28 and 13.25 points. It is indicated that there is a domain shift between the training domain and the translation domain. Also, Multitask learning enables knowledge transfer between the big and small datasets. It is clear that those approaches are adapted successfully.

Table 4.6. Test results of Tokenization with hand shape representation.

	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Big Dataset 2D-CNN	34.59	35.52	22.37	15.80	12.17
Big Dataset 3D-CNN	31.14	32.33	19.38	13.30	10.06
Small Dataset	31.98	33.40	20.53	14.50	11.15
Domain Adaptation	34.41	34.84	22.07	15.75	12.21
Multitask	**36.28**	**37.22**	**23.88**	**17.08**	**13.25**
Multitask + Full Frames	**38.05**	**38.62**	**25.26**	**18.37**	**14.42**

Small dataset in Table 4.6 refers to the network trained on the small dataset. The second row indicates that a pre-trained network fine-tuned on a small dataset leads to a considerable increase in all metrics. Compared to the third rows of Table 4.3, the small dataset brings additional supervision which leads to about 2.09 points increase in BLEU-4 score. The big dataset improves the scores by about 3.11 points in BLEU-4 score. Domain adaptation seems to be ineffective as it attains nearly the same results with Big Dataset though it is the most successful method in classification. It means that the sequence-to-sequence model can handle the domain shift problem implicitly.

The first two rows of Table 4.6 presents the results of 2D-CNN and 3D-CNN based approaches trained on the big dataset. The 3D-CNN setting cannot attain the scores as the 2D-CNN does. There is a significant difference, 2.11 points in BLEU-4 and 2.45 points in ROUGE. A major reason may be that 3D-CNNs cannot focus on the desired hand shapes in a window of consecutive frames. In a certain context, irrelevant hand shapes before and after the desired one may be confused with other annotated

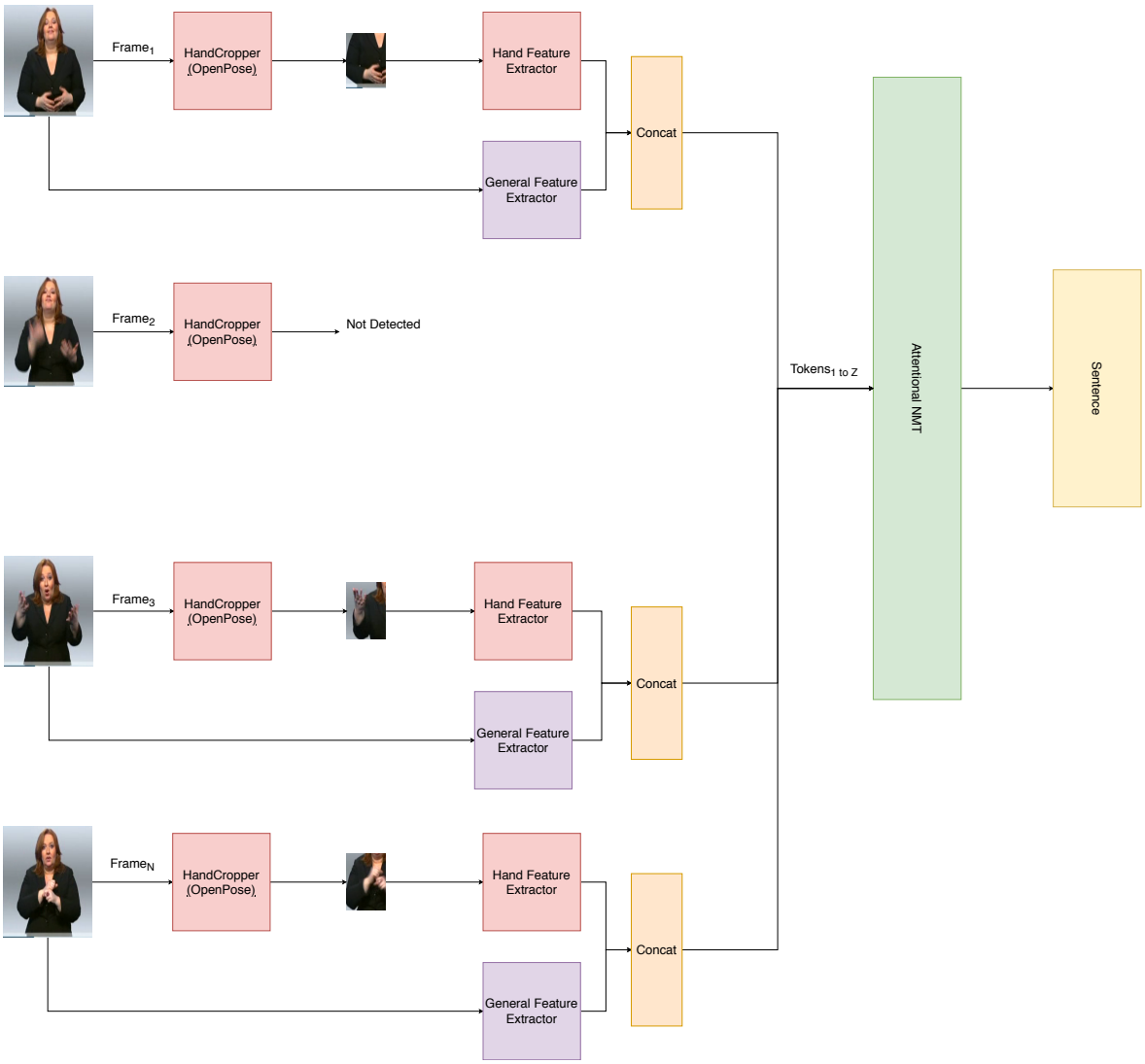

Figure 4.6. Combination of right hands and full frames in Tokenization layer.

hand shapes. Therefore, 3D-CNNs are not very suitable to be trained with weakly labeled hand shape images to leverage additional supervision to the tokenization layer.

Multitask learning is the most successful method with 13.25 BLEU-4 score. In comparison with the third row of Table 4.1, Multitask learning improves the results by 4.39 points in ROUGE and 3.00 points in BLEU-4 score. It is shown that if there is not enough data, Multitask learning is a proper way to combine weak signals to obtain a strong one.

Another key conclusion is that adding small supervision over hand shape learning attains more accurate results than human keypoints. Every result in Table 4.6 is better than the third row of Table 4.2.3. This means that OpenPose outputs are not reliable enough for hand shape representation in SL.

In addition to that, we add another feature source in our tokenization layer. Up to now, it is clear that learning hand shapes is the key factor to understand signers. However, experiments in Section 4.2.2 also indicate that the general frame context may improve translation quality. Therefore, we modified our tokenization approach as seen in Figure 4.6. The hand feature extractor is Multitask and we use Resnet-34 (512) [76] as a general feature extractor. The reason why it is chosen instead of Inception v3 (2048) is the spatio-embedding size. We want to keep the feature size as the smallest for computational concerns. The last two rows of Table 4.6 indicate that the general context improves scores by 1.77 points in ROUGE and 1.17 points in BLEU-4 scores.

4.3. Transfer Learning from Kinetics

In this section, we conduct experiments to seek answers for two questions. Besides stationary hand shapes, action recognition data may be useful as an additional data source. Recent 3D-CNN architectures are capable of operating both in the spatial and temporal domains. In other words, 3D-CNNs can both model action and identify hand

shapes. We test this assumption with I3D proposed by [38]. This is an adapted version of Inception [77] trained on ImageNet.

The frame sequences are divided into eight-frame-long splits with the same order as in Figure 3.6. The number of resulting spatio-temporal embeddings are about one-eighth of frame number. Therefore, we expect that the sequence-to-sequence model does not suffer from long term dependencies. It also helps to reduce the required computational burden for training. For real-time applications, this tokenization approach is seen to be the most promising one.

Table 4.7. Test results of Tokenization with I3D.

	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Full Frames	**29.74**	**29.52**	**17.09**	**11.64**	**8.76**
Both Hands	28.64	28.27	16.14	10.99	8.26
Right Hands	28.00	28.01	15.78	10.73	8.09

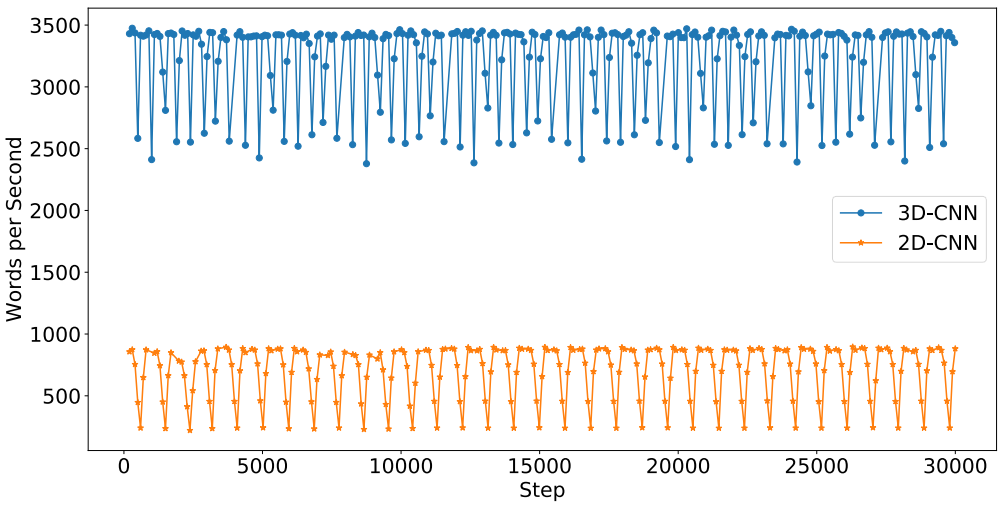

Figure 4.7. Processing speed graph of NSLT with 3D-CNN and 2D-CNN.

This approach is comparable with the result in Table 4.3. There is no certain precedence between 2D-CNNs and 3D-CNNs. They both attain the best results with

full frames. 3D-CNNs are better at Rouge score (29.74) while 2D-CNNs outperform in terms of BLEU-4 score (9.4). It is clear that action recognition can be used as a related task for sign languages. Furthermore, 3D-CNNs are more efficient in terms of memory and time. As seen in Figure 4.7, using 3D-CNN tokenization instead of 2D-CNN tokenization makes the overall model seven times faster.

4.4. Tokenization Learning on Target Domain

In this section, we include frames from RWTH-PHOENIX-Weather 2014 to train tokenization layers. We evaluate additional supervision leveraged by our proposed approaches without target information. The big dataset now involves all of the data, German (ph), Danish and New Zealand Sign Languages. We train a baseline network on the big dataset, multitask network on the big and small dataset. The last network is DeepHand proposed by Koller *et al.* [33]. This network is trained with a Hidden Markov Model that models the transition between consecutive frames to refine weak labels.

Table 4.8. Test results with Tokenization trained on the target domain.

	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
DeepHand	**38.05**	**38.50**	**25.64**	**18.59**	**14.56**
Multitask	36.35	37.11	24.10	17.46	13.50
Baseline	35.22	35.97	23.10	16.59	12.89

As expected, DeepHand outperforms other networks as it uses all of the available information. Multitask learning is nearly the same as Multitask in Table 4.6. Despite the fact that ph in Table 2.1 is two times bigger than the combination of others, it brings little improvement in all metrics. This implies that Multitask learning can reach its maximum capacity with less amount of data. Another important finding is that Baseline scores in Table 4.8 are outperformed by Multitask scores in Table 4.6. This

means that the annotation approach of the small dataset provides much more crucial abilities to the network than a huge amount of data labeled with one regime.

4.5. Analysis of Results

In this section, we discuss whether our approach really improves the results. We compare our Multitask Learning approach with the baseline methods and DeepHand. DeepHand is a special network trained on over one million hand images involving samples from the training set of the translation dataset, RWTHPHOENIXWeather 2014T. The hand shape dataset is automatically annotated so the labels are very noisy. In [33], they propose a method to refine labels with sequence information. Therefore, DeepHand is not a good candidate for a generic tokenization layer as it benefits from supervision from the target domain along with sequence knowledge. Therefore, we consider this method as an upper bound for hand shape classification and frame-level tokenization with right hands. Besides that, we provide insights into how our approach increases translation quality.

The statistical significance in NMT is discussed in the work of [78]. It proposes methods to conclude a confidence score if a translation system is better than another. We use paired bootstrap resampling to validate our contributions. We use statistical validation methods as repeating experiments many times is not feasible in NSLT research. When the sample size is very limited, we use bootstrap sampling to determine a confidence interval. Paired bootstrap resampling is a method to assign confidences to our claims. NMT adopts this method to test whether a translation system is statistically significantly better than another. For a large number of times, we sample sentences with replacement to construct test sets. We evaluate the systems on these test sets and compare the results. If the first system is better than the second one in %95 of the trials, we conclude that our claim is true with %95 confidence. We have only 642 test sentences. Therefore, we generate three different 1000-sized subsets in which a sample consists of 250,400 and 600 sentences sampled with replacement.

Eventually, we check our claims listed in Table 4.9 by this method to provide a more reliable conclusion.

Table 4.9 displays the confidence of the claims in the system comparison column. Baseline refers to the 2D-CNN tokenization trained on the Big Dataset without the target annotation. Also, no additional technique is applied, such as Multitask learning or Domain Adaptation. The table shows that the Multitask approach clearly outperforms the baseline, but does not attain DeepHand level results. This is to say that the representation of hand shapes can be improved without accessing target domain supervision. The target supervision is more useful if it exists. Moreover, our approach gets stronger when it is enhanced with the frame context. The last row of the table indicates that DeepHand is not better than Multitask-FullFrames setting with high confidence. %58.80 confidence means that the results are comparable.

Table 4.9. Paired bootstrap sampling results on different translation comparisons. Delta refers to the differences in BLEU-4 Score and 250, 400 and 600 are the number of the sentences drawn for a sample. The table displays the confidence of the claims under different sentence sizes.

System Comparison	Delta	250	400	600
Multitask better than Baseline	1.08	%94.70	%97.0	%99.30
Multitask+Full-Frames better than Baseline	2.24	%99.60	%100	%100
Multitask+Full-Frames better than Multitask	1.16	%93.50	%96.30	%99.10
DeepHand better than Baseline	2.39	%99.70	%100	%100
DeepHand better than Multitask	1.31	%93.80	%96.90	%99.10
DeepHand better than Multitask-FullFrames	0.15	%55.10	%58.10	%58.80

It is shown that our approach can attain the upper bound results and outperforms the baseline with high confidence. For SLT, this can be enabled by proving better sign embeddings as better word embeddings improve in NMT. We mentioned that Multitask learning increases top-1 and top-5 accuracy in hand shape classification dramatically.

In NSLT, identifying more hand shapes is expected to lead to an increase in word and gloss diversity in generated translations. However, noisy embeddings may also increase the diversity where the resulting sentences become meaningless. Table 4.10 is designed to show the unique number of words whose frequency are greater than the threshold. We ignore the stop words of German while counting the words. Golden refers to the reference translation used in BLEU-4 evaluations. Multitask learning uses fewer unique words, but provides higher BLUE and ROUGE scores compared to Baseline. As expected, Baseline produces more noisy sign embeddings as it is only trained on the weakly labeled dataset. In other words, diversity is higher, but the choice of words and the placement of words are not appropriate. Multitask learning can obtain higher results with less diversity as its embeddings are very robust. When we combine its embeddings with frame contexts, the BLEU-4 scores and the unique number of words increase. To conclude, Multitask learning refines the wrong labels while training. Furthermore, the hand shape classification results show that it covers more hand shapes compared to Baseline.

Table 4.10. The table displays unique word numbers under varying frequencies in the translation systems.

BLUE-4	System	0	1	2	3	5	10
12.17	Baseline	505	368	305	264	200	116
13.25	Multitask	498	350	272	234	184	121
14.42	Multitask+Full-Frames	563	420	337	272	201	121
100	Golden	989	571	412	331	237	128

4.6. Comparison of Frame Levels & Gloss Level Translation

Glosses are perfect tokens for sign language translation. However, if the ultimate goal is to construct sentences instead of gloss recognition, learning glosses restricts the model. Camgoz *et al.* [1] suggests that tokenization learning from glosses recognition outperforms inputting perfect glosses as tokens. The findings of our study also sup-

port this claim as additional supervision in tokenization leads to dramatic increases in translation quality.

Translation from golden glosses on our dataset attains 19.26 BLEU-4 score and our domain-independent tokenization approach acquires 13.25 BLEU score with only right hands. We showed that full frames are better than only right hand crops in all of the experiments. We claim that our tokenization approach can be improved much further by combining other body parts like left hands or faces. In addition to that, we show that Multitask learning is a suitable candidate to outperform the gloss level tokenization. The most important finding of this study is that we can eliminate gloss annotation with different information sources while our method is applicable to other datasets.

5. CONCLUSION

5.1. Remarks

Firstly, we show that the joint learning of tokenization and sequence-to-sequence models has certain shortcomings. As collecting a huge amount of translation samples is a very laborious task, the required additional supervision is leveraged by Sign Language Recognition tasks with the proposed tokenization learning approaches. We prepare experiments to find the most appropriate attention mechanism to deal with long sequences in frame-level tokenization. To conduct a detailed analysis, we generate two different versions of RWTH-PHOENIX-Weather 2014T with hand crops. We propose two different tokenization learning methods and compare them with existing tokenization methods. 3D-CNNs are adapted into Sign Language Translation for the first time.

The experiments are divided into two main categories, with or without access to the target domain. This division is aimed to test the effectiveness of our proposed approaches as a generic tokenization layer. Without target domain knowledge, we illustrate the positive effects of related tasks to Sign Language Translation. First, we reveal that 3D-CNNs are also good candidates since they are more efficient in terms of required computational effort and memory amount. Secondly, our proposed approach accomplishes to learn from weak signals to obtain a robust tokenization layer by Multitask learning. Moreover, domain adaptation improves classification results by mitigating domains shifts, yet does not have any effect on translation task. We infer that sequence-to-sequence models can handle domain shifts implicitly.

We increase the upper-bound of frame-level tokenization. Without target knowledge, we approximate the upper-bound with our proposed approach. Our study involves four different sign languages. One of the key findings in the study emphasizes that adding a new data source may result in advancements even if they are very limited.

In comparison with gloss-level tokenization, frame-level tokenization is more promising for further improvement. We close the gap between two different tokenization approaches by nearly 50% improvement in BLEU-4 score. The development of gloss-level tokenization is restricted to tedious annotation whereas frame-level tokenization may be improved with data from other research fields.

5.2. Future Work

We attain the best results with only right-hands while the experiments illustrate that other body parts may also be taken into account. Therefore, new studies may be conducted on how to fuse body parts in frame-level tokenization. We believe that this may enable further improvement in SLT.

Another research direction may be on modifying 3D-CNNs. They are proved to be good tokenization layers, but additional supervision may be leveraged through them. In addition to action recognition, new tasks may be established to train 3D-CNNs. Considering their ability of temporal operation, they may be even replaced with Encoder in Encoder-Decoder (ED) architecture.

REFERENCES

1. Camgöz, N. C., O. Koller, S. Hadfield *et al.*, "Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation", *ArXiv*, Vol. abs/2003.13830, 2020.

2. Camgoz, N. C., S. Hadfield, O. Koller *et al.*, "Neural Sign Language Translation", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

3. Ko, S., C. J. Kim, H. Jung *et al.*, "Neural Sign Language Translation based on Human Keypoint Estimation", *arXiv*, Vol. abs/1811.11436, 2018.

4. Kristoffersen, J. H., T. Troelsgard, A. S. Hardell *et al.*, *Ordbog over Dansk Tegnsprog*, 2008-2016, `http://www.tegnsprog.dk`, accessed in August 2020.

5. McKee, D., R. McKee, S. P. Alexander *et al.*, *The Online Dictionary of New Zealand Sign Language*, 2015, `http://nzsl.vuw.ac.nz/`, accessed in August 2020.

6. Makaroğlu, B. and H. Dikyuva, *Güncel Türk İşaret Dili Sözlüğü.*, 2017, `http://tidsozluk.net`, accessed in August 2020.

7. Kadous, M. W. and C. S. Engineering, "Machine Recognition of Auslan Signs Using PowerGloves: Towards Large-Lexicon Recognition of Sign Language", *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*, pp. 165–174, 1996.

8. Vogler, C. and D. Metaxas, "Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods", *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, Vol. 1, pp. 156–161 vol.1, 1997.

9. Yin, F., X. Chai and X. Chen, "Iterative Reference Driven Metric Learning for Signer Independent Isolated Sign Language Recognition", B. Leibe, J. Matas, N. Sebe and M. Welling (Editors), *Computer Vision – ECCV 2016*, pp. 434–450, Springer International Publishing, Cham, 2016.

10. Camgöz, N. C., A. A. Kındıroğlu, S. Karabüklü *et al.*, "BosphorusSign: A Turkish Sign Language Recognition Corpus in Health and Finance Domains", *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, European Language Resources Association (ELRA), Paris, France, may 2016.

11. Ebling, S., N. C. Camgöz, P. Boyes Braem *et al.*, "SMILE Swiss German Sign Language Dataset", *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), Miyazaki, Japan, May 2018.

12. Pfister, T., J. Charles and A. Zisserman, "Large-scale Learning of Sign Language by Watching TV (Using Co-occurrences)", *British Machine Vision Conference*, 2013.

13. Buehler, P., M. Everingham and A. Zisserman, "Learning Sign Language by Watching TV (using Weakly Aligned Subtitles)", *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

14. Cooper, H. and R. Bowden, "Sign Language Recognition: Working with Limited Corpora", C. Stephanidis (Editor), *Universal Access in Human-Computer Interaction. Applications and Services*, pp. 472–481, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

15. Forster, J., C. Schmidt, T. Hoyoux *et al.*, "RWTH-PHOENIX-Weather: A Large Vocabulary Sign Language Recognition and Translation Corpus", *Proceedings of the Eighth International Conference on Language Resources and Evalu-*

ation (LREC-2012), pp. 3785–3789, European Languages Resources Association (ELRA), Istanbul, Turkey, May 2012.

16. Forster, J., C. Schmidt, O. Koller *et al.*, "Extensions of the Sign Language Recognition and Translation Corpus RWTH-PHOENIX-Weather", *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pp. 1911–1916, European Languages Resources Association (ELRA), Reykjavik, Iceland, May 2014.

17. Dalal, N. and B. Triggs, "Histograms of oriented gradients for human detection", *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1, pp. 886–893 vol. 1, 2005.

18. Liwicki, S. and M. Everingham, "Automatic recognition of fingerspelled words in British Sign Language", *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 50–57, 2009.

19. Lim, K. M., A. W. Tan and S. C. Tan, "Block-Based Histogram of Optical Flow for Isolated Sign Language Recognition", *J. Vis. Comun. Image Represent.*, Vol. 40, No. PB, p. 538–545, Oct. 2016.

20. Laptev, I., M. Marszalek, C. Schmid *et al.*, "Learning realistic human actions from movies", *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

21. Özdemir, O., N. C. Camgöz and L. Akarun, "Isolated sign language recognition using Improved Dense Trajectories", *2016 24th Signal Processing and Communication Application Conference (SIU)*, pp. 1961–1964, 2016.

22. Wang, H. and C. Schmid, "Action Recognition with Improved Trajectories", *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013.

23. Özdemir, O., A. A. Kindiroglu and L. Akarun, "Isolated sign language recognition with fast hand descriptors", *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2018.

24. Bemdt, D. J. and J. Clifford, *Using Dynamic Time Warping to Find Patterns in Time Series*, 1994.

25. Rabiner, L. and B. Juang, "An introduction to hidden Markov models", *IEEE ASSP Magazine*, Vol. 3, No. 1, pp. 4–16, 1986.

26. Bengio, Y. and P. Frasconi, "Input-output HMMs for sequence processing", *IEEE Transactions on Neural Networks*, Vol. 7, No. 5, pp. 1231–1249, 1996.

27. Keskin, C., A. T. Cemgil and L. Akarun, "DTW Based Clustering to Improve Hand Gesture Recognition", A. A. Salah and B. Lepri (Editors), *Human Behavior Understanding*, pp. 72–81, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

28. Krizhevsky, A., I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger (Editors), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105, Curran Associates, Inc., 2012.

29. Hochreiter, S. and J. Schmidhuber, "Long short-term memory", *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

30. Amodei, D., S. Ananthanarayanan, R. Anubhai *et al.*, "Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin", M. F. Balcan and K. Q. Weinberger (Editors), *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48 of *Proceedings of Machine Learning Research*, pp. 173–182, PMLR, New York, New York, USA, 20–22 Jun 2016.

31. Chan, W., N. Jaitly, Q. V. Le *et al.*, "Listen, Attend and Spell", *CoRR*, Vol. abs/1508.01211, 2015.

32. Xu, K., J. Ba, R. Kiros *et al.*, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", *CoRR*, Vol. abs/1502.03044, 2015.

33. Koller, O., H. Ney and R. Bowden, "Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data Is Continuous and Weakly Labelled", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

34. Koller, O., S. Zargaran, H. Ney *et al.*, "Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition", *BMVC*, 2016.

35. Schuster, M. and K. Paliwal, "Bidirectional Recurrent Neural Networks", *Trans. Sig. Proc.*, Vol. 45, No. 11, p. 2673–2681, Nov. 1997.

36. Koller, O., S. Zargaran and H. Ney, "Re-Sign: Re-Aligned End-to-End Sequence Modelling with Deep Recurrent CNN-HMMs", *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3416–3424, 2017.

37. Graves, A., S. Fernández, F. Gomez *et al.*, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks", *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 369–376, ACM, New York, NY, USA, 2006.

38. Carreira, J. and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset", *CoRR*, Vol. abs/1705.07750, 2017.

39. Molchanov, P., S. Gupta, K. Kim *et al.*, "Hand gesture recognition with 3D convolutional neural networks", *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1–7, 2015.

40. Camgoz, N. C., S. Hadfield, O. Koller *et al.*, "Using Convolutional 3D Neural Networks for User-independent continuous gesture recognition", *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 49–54, 2016.

41. Yang, Z., Z. Shi, X. Shen *et al.*, "SF-Net: Structured Feature Network for Continuous Sign Language Recognition", *CoRR*, 2019.

42. Siyli, R. D., *HospiSign: A Framewise Annotated Isolated Turkish Sign Language Dataset*, `http://dogasiyli.com/hospisign/`, accessed in August 2020.

43. von Agris, U., M. Knorr and K. Kraiss, "The significance of facial features for automatic sign language recognition", *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, pp. 1–6, 2008.

44. Chai, X., H. Wanga, M. Zhoub *et al.*, *DEVISIGN: Dataset and Evaluation for 3D Sign Language Recognition*, Tech. rep., Beijing, 2015.

45. Escalante, H. J., V. Ponce-López, J. Wan *et al.*, "ChaLearn Joint Contest on Multimedia Challenges Beyond Visual Analysis: An overview", *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 67–73, 2016.

46. Huang, J., W. Zhou, Q. Zhang *et al.*, "Video-based Sign Language Recognition without Temporal Segmentation", *CoRR*, Vol. abs/1801.10111, 2018.

47. Vaezi Joze, H. and O. Koller, "MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language", *The British Machine Vision Conference (BMVC)*, September 2019.

48. Mikolov, T., I. Sutskever, K. Chen *et al.*, "Distributed Representations of Words and Phrases and Their Compositionality", *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, p. 3111–3119, Curran Associates Inc., Red Hook, NY, USA, 2013.

49. Bengio, S., O. Vinyals, N. Jaitly *et al.*, "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks", *CoRR*, Vol. abs/1506.03099, 2015.

50. Banerjee, S. and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments", *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Association for Computational Linguistics, Ann Arbor, Michigan, Jun. 2005.

51. Lin, C.-Y., "ROUGE: A Package for Automatic Evaluation of Summaries", *Text Summarization Branches Out*, pp. 74–81, Association for Computational Linguistics, Barcelona, Spain, Jul. 2004.

52. Papineni, K., S. Roukos, T. Ward *et al.*, "Bleu: a Method for Automatic Evaluation of Machine Translation", *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, Jul. 2002.

53. Sundermeyer, M., R. Schlüter and H. Ney, "LSTM Neural Networks for Language Modeling", *INTERSPEECH*, pp. 194–197, 2010.

54. Kalchbrenner, N. and P. Blunsom, "Recurrent Continuous Translation Models", *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709, Association for Computational Linguistics, Seattle, Washington, USA, Oct. 2013.

55. Sutskever, I., O. Vinyals and Q. V. Le, "Sequence to Sequence Learning with Neural Networks", Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence and K. Q. Weinberger (Editors), *Advances in Neural Information Processing Systems 27*, pp. 3104–3112, Curran Associates, Inc., 2014.

56. Zoph, B. and K. Knight, "Multi-Source Neural Translation", *CoRR*, Vol. abs/1601.00710, 2016.

57. Mnih, V., N. Heess, A. Graves *et al.*, "Recurrent Models of Visual Attention", Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence and K. Q. Weinberger (Editors), *Advances in Neural Information Processing Systems 27*, pp. 2204–2212, Curran Associates, Inc., 2014.

58. Gao, L., Z. Guo, H. Zhang *et al.*, "Video Captioning With Attention-Based LSTM and Semantic Consistency", *IEEE Transactions on Multimedia*, Vol. 19, No. 9, pp. 2045–2055, 2017.

59. Luong, T., H. Pham and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation", *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Association for Computational Linguistics, Lisbon, Portugal, Sep. 2015.

60. Vaswani, A., N. Shazeer, N. Parmar *et al.*, "Attention is All you Need", I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett (Editors), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008, Curran Associates, Inc., 2017.

61. Devlin, J., M. Chang, K. Lee *et al.*, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *CoRR*, Vol. abs/1810.04805, 2018.

62. Ko, S., C. J. Kim, H. Jung *et al.*, "Neural Sign Language Translation based on Human Keypoint Estimation", *CoRR*, Vol. abs/1811.11436, 2018.

63. Cao, Z., G. Hidalgo, T. Simon *et al.*, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields", *arXiv preprint arXiv:1812.08008*, 2018.

64. Cao, Z., T. Simon, S.-E. Wei *et al.*, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", *CVPR*, 2017.

65. Simon, T., H. Joo, I. Matthews *et al.*, "Hand Keypoint Detection in Single Images using Multiview Bootstrapping", *CVPR*, 2017.

66. Bahdanau, D., K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate", *arXiv*, 2014.

67. Cho, K., B. van Merrienboer, D. Bahdanau *et al.*, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches", *arXiv*, Vol. abs/1409.1259, 2014.

68. Ganin, Y., E. Ustinova, H. Ajakan *et al.*, "Domain-Adversarial Training of Neural Networks", *Journal of Machine Learning Research*, Vol. 17, No. 59, pp. 1–35, 2016.

69. Joze, H. R. V. and O. Koller, "MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language", *CoRR*, Vol. abs/1812.01053, 2018.

70. Bilge, Y. C., N. Ikizler-Cinbis and R. G. Cinbis, "Zero-Shot Sign Language Recognition: Can Textual Data Uncover Sign Languages?", *CoRR*, Vol. abs/1907.10292, 2019.

71. Abadi, M., P. Barham, J. Chen *et al.*, "TensorFlow: A system for large-scale machine learning", *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.

72. Klein, G., Y. Kim, Y. Deng *et al.*, "OpenNMT: Open-Source Toolkit for Neural Machine Translation", *ArXiv e-prints*.

73. Paszke, A., S. Gross, F. Massa *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library", H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox and R. Garnett (Editors), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

74. Szegedy, C., V. Vanhoucke, S. Ioffe *et al.*, "Rethinking the Inception Architecture for Computer Vision", *CoRR*, Vol. abs/1512.00567, 2015.

75. Tran, D., H. Wang, L. Torresani *et al.*, "A Closer Look at Spatiotemporal Convolutions for Action Recognition", *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

76. He, K., X. Zhang, S. Ren *et al.*, "Deep Residual Learning for Image Recognition", *CoRR*, Vol. abs/1512.03385, 2015.

77. Szegedy, C., W. Liu, Y. Jia *et al.*, "Going Deeper with Convolutions", *CoRR*, Vol. abs/1409.4842, 2014.

78. Koehn, P., "Statistical Significance Tests for Machine Translation Evaluation", *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 388–395, Association for Computational Linguistics, Barcelona, Spain, Jul. 2004.

APPENDIX A: ATTENTIONAL NMT MODEL

A.1. Implicit Tokenization

- feature size: 4096
- RNN type: GRU
- Attention Type: Luong
- Layer Size: 4
- Hidden Unit Size: 1024
- Dropout Rate: 0.2
- Residual: True
- Learning Rate: 1e-5
- Optimizer: Adam
- Batch Size: 1
- Word Embedding Size: 256

A.2. Bahdanau & Luong Model Details

- feature size: 2048
- RNN type: GRU
- Layer Size: 4
- Hidden Unit Size: 512
- Dropout Rate: 0.2
- Residual: True
- Initial Parameter Std: 0.02
- Learning Rate: 1e-4
- Optimizer: Adam
- Batch Size: 16
- Word Embedding Size: 256

A.3. Transformer Model Details

- Layer Size: 6
- Head Number: 8
- Hidden Unit Size: 1024
- Dropout Rate: 0.2
- Attentional Dropout: 0.1
- Relu Dropout: 0.1
- Residual: True
- Initial Parameter Std: 0.02
- Learning Rate: 1e-4
- Optimizer: Adam
- Batch Size: 16
- Word Embedding Size: 128

APPENDIX B: CNN TRAINING

B.1. Multitask Learning

- $L_{\texttt{total}} = L_1 + \beta(t)L_2$
- $\beta(t) = 0.1$
- Learning Rate: 1e-4
- Optimizer: Adam

B.2. Domain Adaptation

- γ: 2.5 (equation 14 in [68])
- Domain Classifier Layer Size: 2
- Doman Classifier Hidden Unit Size: 2048
- Learning Rate: 5e-3
- Optimizer: SGD (Not Converged with Adam)