Hybrid Translation System from Turkish Spoken Language to Turkish Sign Language

by

Dilek Kayahan

B.S., Computer Engineering, Yıldız Technical University, 2012

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 2019 Hybrid Translation System from Turkish Spoken Language to Turkish Sign Language

APPROVED BY:

Prof. Dr. Tunga Güngör	
(Thesis Supervisor)	
Prof. Dr. Fikret Gürgen	
Prof. Dr. Banu Diri	

DATE OF APPROVAL: 14.01.2019

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to Prof. Dr. Tunga Güngör for the continuous support of my master study and research. Besides my advisor, I would like to thank Assist. Prof. Dr. Kadir Gökgöz and Assoc. Prof. Dr. Meltem Kelepir for their guidance about the sign languages.

ABSTRACT

Hybrid Translation System from Turkish Spoken Language to Turkish Sign Language

Sign Language is the primary tool of communication for deaf and mute people. It employs hand gestures, facial expressions, and body movements to state a word or a phrase. Like spoken languages, sign languages also vary among the regions and the cultures. Each sign language has its own word order, lexicon, grammatical rules, and dialects. According to these features, a sign language also differs from the spoken language that it represents.

The aim of the study is to implement a machine translation system in order to convert Turkish spoken language into Turkish Sign Language (TİD). The advantages of the rule-based and the statistical machine translation techniques are combined into the hybrid translation system.

The proposed system is evaluated with Bilingual Evaluation Understudy (BLEU) scoring metric and it is proved that the hybrid translation approach performs better than rule-based and statistical approaches.

ÖZET

Türkçe Konuşma Dilinden Türkçe İşaret Diline Hibrit Çeviri Sistemi

İşaret dilleri işitme engelliler tarafından kullanılan görsel bir iletişim aracıdır. Bu diller de diğer doğal diller gibi ülkeye ve kültüre göre farklılıklar göstermekte olup, kendine özgü dilbilgisi kuralları ve lehçeleri bulunmaktadır.

Bu çalışmada, işitme engellilerin hayatını kolaylaştırmak amacıyla Türkçe metinleri Türkçe İşaret Diline otomatik çeviren bir sistem tasarlanmıştır. Bu sistem, kural tabanlı ve istatistiksel çeviri yöntemlerini birleştirerek daha iyi performans sağlayan hibrit çeviri yöntemini gerçekleştirmektedir.

Bu çalışmada, Aile ve Sosyal Politikalar Bakanlığı tarafından yayınlanan işaret dili sözlüğündeki örnek cümleler veri kümesi olarak kullanılmıştır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS i	ii
ABSTRACT	v
ÖZET	v
LIST OF FIGURES	ii
LIST OF TABLES	х
LIST OF SYMBOLS	ci
LIST OF ACRONYMS/ABBREVIATIONS	ii
1. INTRODUCTION	1
1.1. Motivation	2
1.2. Contributions of the Thesis	2
2. BACKGROUND THEORY	3
3. DATASET	1
3.1. Online TID Dictionary Dataset	1
4. METHODOLOGY 1	4
4.1. Rule-Based Translation Component	5
4.1.1. Turkish Natural Language Analyzer	5
4.1.2. Turkish to Turkish Sign Language Transformation Rules 1	6
4.1.2.1. Infinitive Verb Inflection	7
4.1.2.2. Punctuation Marks	9
4.1.2.3. Conjunctions \ldots \ldots \ldots \ldots \ldots \ldots \ldots 1	9
4.1.2.4. Person Agreement	0
4.1.2.5. Present Tense Rule	0
4.1.2.6. Past Tense Rule	1
4.1.2.7. Future Tense Rule	1
4.1.2.8. Necessity Rule $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 2$	1
4.1.2.9. Negation Rule $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 2$	2
4.1.2.10. Possessive Rule	3
4.1.2.11. Locative Rule	3
4.1.2.12. Ablative Rule	4

			4.1.2.13. Proper Nouns	24
		4.1.3.	Rule-Based Translator	25
	4.2.	Prepro	cessor	26
		4.2.1.	Custom Turkish Preprocessor	27
		4.2.2.	Custom TID Preprocessor	28
	4.3.	Statist	ical Translation Component	28
		4.3.1.	Language Model	29
		4.3.2.	Training Pipeline	30
			4.3.2.1. Corpus Preparation	30
			4.3.2.2. Word Alignment	31
			4.3.2.3. Lexical Translation Table	32
			4.3.2.4. Phrase Table	32
			4.3.2.5. Reordering Model	33
		4.3.3.	Tuning	35
		4.3.4.	Decoder	36
5.	EXP	ERIME	ENTS AND RESULTS	39
	5.1.	Rule-E	Based Translation Component Performance	40
	5.2.	Statist	ical Translation Component Performance	42
	5.3.	Hybrid	l Translation System Performance	44
	5.4.	Hybrid	l Translation System and Literature Comparison	47
	5.5.	Effects	s of the Translation Rules on Hybrid Translation System	48
6.	CON	ICLUSI	ON AND FUTURE WORK	50
RF	EFER	ENCES	8	52
AI	PEN	DIX A:	APPLICATION	54

LIST OF FIGURES

Figure 2.1.	$\dot{\text{TID}}$ manual alphabet [1] $\hdots \dots	3
Figure 2.2.	Hamburg Notation System's parts [2]	4
Figure 2.3.	TID translation of the Turkish word "Defans" and its HamNoSys notation	5
Figure 2.4.	Some of the symbols in SignWriting notation	5
Figure 2.5.	English to sign language translation by DRS [3] $\ldots \ldots \ldots$	6
Figure 2.6.	HamNoSys notation of the word "Deaf" in BSL and the correspond- ing SiGML representation [4]	7
Figure 2.7.	Virtual avatar for the sign "Deaf" in BSL	7
Figure 2.8.	$\dot{\mathrm{TID}}$ translation of the Turkish word "Defans" in eSIGN editor $% \mathcal{T}$.	8
Figure 2.9.	Sign language video annotation in ELAN	9
Figure 3.1.	"Anlamak" in online TID dictionary	12
Figure 3.2.	A part of the website crawler's output for letter "A"	13
Figure 4.1.	Hybrid Translation System From Turkish Spoken Language to Turk- ish Sign Language Architecture	14
Figure 4.2.	The Boun Morphological Parser Output	16

Figure 4.3.	The Boun Morphological Disambiguator Output	17
Figure 4.4.	Part of a sample language model	30
Figure 4.5.	Monotone, swap, and discontinuous orientation classes $[5]$	34
Figure 4.6.	Sample of the Moses configuration file	37
Figure 4.7.	Sample of the Mert Optimized Moses configuration file	38

LIST OF TABLES

Table 2.1.	Comparison of the notation systems	6
Table 4.1.	Preprocessor results of the Turkish sentences	27
Table 4.2.	Preprocessor results of the TİD sentences $\ldots \ldots \ldots \ldots \ldots$	29
Table 4.3.	A part of the preprocessed train corpus	31
Table 5.1.	Comparision of the rule-based translation results and the original TID translations	41
Table 5.2.	Comparision of the statistical translation results and the original TID translations	43
Table 5.3.	Comparision of the hybrid translation results and the original pre- processed TİD translations	45
Table 5.4.	Dataset comparison of the systems	48

LIST OF SYMBOLS

 λ_i

The weight of the BLEU-i in the cumulative score

LIST OF ACRONYMS/ABBREVIATIONS

TİD	Turkish Sign Language
BLEU	Bilingual Evaluation Understudy
URL	Uniform Resource Locator
JSON	Javascript Notation Format
ASL	American Sign Language
eSIGN	Essential Sign Language Information on Government Net-
VisiCAST	works Virtual Signing: Capture, Animation, Storage, and Transmis-
EU	sion European Union
WWW	World Wide Web
DRS	Discourse Representation Structure
BSL	British Sign Language
HamNoSys	Hamburg Notation System
SiGML	Signing Gesture Markup Language
XML	Extensive Markup Language
ELAN	EUDICO Linguistic Annotator
STAG	Synchronous Tree Adjoining Grammar
TAG	Tree-Adjoining Grammars
LSE	Spanish Sign Language
NMT	Neural Machine Translation
DGS	German Sign Language
stid	System TID
SMT	Statistical Machine Translation
MERT	Minimum Error Rate Training

1. INTRODUCTION

Sign languages are emerged naturally as a visual communication medium by hearing impaired people. Since sign languages are developed naturally, they are categorized as natural languages like spoken languages.

Turkish Sign Language is used by deaf communities in Turkey and the Turkish Republic of Northern Cyprus with some dialect variations especially in the lexicon ¹. But deaf people in Turkey, report that they can communicate quite easily with other deaf people from different regions of Turkey. On the other hand, they have difficulties to communicate with deaf people from other countries, such as Germans or Americans. Turkish Sign Language has no relation with European Sign Languages neither lexical nor grammatical aspects [6]. It is said that TİD is originated from Ottoman Sign Language which means that it has at least 120 years of history. But this is still not proven.

In this work, a hybrid translation system to translate Turkish spoken language into TİD is proposed. This system comprises of rule-based and statistical translation components. Turkish text is first fed into rule-based translation component which applies predefined Turkish to TİD grammatical rules. Then intermediate translation results are processed by statistical translation component and the final TİD translation is generated. Gloss representation is used to typify the TİD.

The main obstacle of the proposed translation system is the lack of information about TİD since it is still under development. There is also no written form of the sign languages which makes it more difficult to analyze. In order to create Turkish to TİD bilingual dataset, online dictionary which is published by The Ministry of Family and Social Policies is parsed, and 3561 sentence pairs are extracted.

 $^{^{1} \}rm https://tr.wikipedia.org/wiki/Türk_\dot{I}saret_Dili$

1.1. Motivation

According to the data published by Turkish Statistical Institute² in 2000, 89.000 people have a hearing disability and 55.000 have a speaking disability in Turkey. Unfortunately, these people encounter troubles to adapt to society and they fall behind in the educational system. I strongly believe that overcoming disabilities is not only their problem but also the responsibility of the community which they live in.

The motivation of the thesis is to facilitate deaf and mute people's life by providing a communication bridge between Turkish spoken language and TİD. The starting point of the study is to provide an embedded translator to televisions in order to convert Turkish subtitle into TİD virtual avatar in real time. The major part which is the Turkish to TİD translator is implemented in this study. The aim of the study is to convert Turkish text into TİD gloss sequence with high accuracy.

1.2. Contributions of the Thesis

There are two main approaches in the literature for text to sign language translation systems; rule-based and statistical methods. Rule-based studies mostly have domain constraint since it is very hard to define translation rules to cover all cases. While statistical methods require large parallel corpus for higher translation accuracy. The main contribution of the thesis is to combine these two approaches in order to decrease the drawbacks of each technique.

As stated before, major obstacles of the study are; limited parallel corpus and uncertainty of the grammatical rules in TİD. In order to define translation rules, I have attended linguistic classes and spend a lot of time to find out precise translation rules from Turkish to TİD. 13 translation rules are defined, that is the major contribution of the thesis. In addition to it, in the scope of this study, Turkish to TİD parallel corpus containing 3561 sentence pairs, is collected.

²http://www.tuik.gov.tr

2. BACKGROUND THEORY

Sign language is a natural language type, that is emerged to communicate visually. Contrary to popular opinion, sign languages are not derived from spoken languages. Each country or region has its own sign language and embodies different grammatical rules and lexicons. In this chapter, general sign language concepts, terms and tools are explained first. Then, several studies in different sign languages are investigated.



Figure 2.1. TID manual alphabet [1]

Sign languages have four main components and additional non-manual markers to articulate a sign [7]. These are hand-shape, orientation, location, and movement. Hand-shape is the form of the hand, while orientation is the direction of the palm. Location is the signing position referenced to the body, such as chest or shoulders and movement is the action of the hand-shapes such as circling or touching. Non-manual markers are extra expressions such as eye gaze, head tilting and shoulder raising that are used to support hand sign. In order to sign the words which have special meaning in the spoken language but lack a sign in the sign language, finger spelling is used. It simply expresses the word by signing the letters of the word individually. Each sign language has its own manual alphabet. TİD manual alphabet is shown in figure 2.1.

In order to typify sign languages, several notation systems are introduced such as Stoke Notation, HamNoSys, SignWriting and Gloss representation.

Stoke notation [8] is the first notation system proposed by William Stoke for American Sign Language (ASL) representation in 1960. Most of the notation systems are based upon Stokoe notation. It approximately comprises 55 symbols and a sign has movement (sig), hand-shape (dez), and location (tab) parts according to the Stokoe notation system.

HamNoSys [9] is a common notation system for all sign languages. It contains approximately 210 symbols, by the combination of these symbols it is possible to model any visual sign. It divides a sign into 4 main parts as shown in figure 2.2; hand-shape, hand position, location, and movement. Each part in the HamNoSys notation represents the relevant part of the visual sign. For example, hand is positioned according to the "hand position" part in HamNoSys notation. Gesture realization tools interpret HamNoSys notation and visualize correspondent gesture with avatars.



Figure 2.2. Hamburg Notation System's parts [2]



Figure 2.3. TID translation of the Turkish word "Defans" and its HamNoSys notation

Figure 2.3 explains how to sign the Turkish word "Defans"³ in TID and the corresponding HamNoSys notation. It is possible to write all signs with HamNoSys notation however it is not a practical language for daily use, it is more suitable for academic purposes.



Figure 2.4. Some of the symbols in SignWriting notation

SignWriting⁴ is proposed by Valerie Sutton in 1974. Contrary to Stokoe and HamNoSys notation systems, SignWriting is much more practical with its iconic symbols. SignWriting is applicable to all sign languages and used for daily communication purposes rather than academical studies. Its symbols are written in the vertical direction. A sign is represented in terms of hand-shapes, orientation, movements, body locations, contacts, and facial expressions. The contacts in the SignWriting are used

³http://www.tdk.org.tr/images/D.pdf

⁴http://www.signwriting.org

to denote the contact between hand and body location. Some of the symbols that are used in SignWriting notation system are shown in figure 2.4.

Comparison of the aforementioned notation systems is shown in table 2.1.

Notation System	Sign Language Support	Non-manual Support	Usage
Stokoe	ASL only	No	Academic
HamNoSys	All	Partly	Academic
SignWriting	All	Yes	Daily
Table 2.1. Comparison of the notation systems			

Unlike the notation systems described above, gloss representation does not involve any hint about the gesture of the signs. Simply, they work as labels for the signs and they are the capitalized forms of correspondent word translation of a sign. For example "Defans" Turkish word in figure 2.3 is represented with "DEFANS" gloss.

Essential Sign Language Information on Government Networks (eSIGN) [10] and Virtual Signing: Capture, Animation, Storage, and Transmission (VisiCAST) projects are developed to visualize sign languages by virtual humans.

ViSiCAST [11] is a European Union (EU) funded project which aims to facilitate the daily life of deaf people in Europe by providing accessibility to the public services such as transportation, learning, television broadcasts and World Wide Web (WWW). The project first converts the English text into intermediate discourse representation structure (DRS). Then, targeted sign language such as British Sign Language (BSL) is generated from these DRSs as illustrated in figure 2.5.



Figure 2.5. English to sign language translation by DRS [3]

The signs in the translated sign language, are written in HamNoSys notation. After the translation is completed, they are fed into virtual avatar component in Signing Gesture Markup Language (SiGML) format. SiGML is the representation of Ham-NoSys symbols in the Extensive Markup Language (XML) format. A sample SiGML document is shown in figure 2.6. Finally, SiGML representation of the sign is fed into SiGML player to realize the sign by virtual avatars as shown in figure 2.7.



Figure 2.6. HamNoSys notation of the word "Deaf" in BSL and the corresponding



Figure 2.7. Virtual avatar for the sign "Deaf" in BSL

The VisiCAST project⁵ is completed by 2002 and as a continuation of it, eSIGN project is initiated. The eSIGN project aims to provide sign language support to the websites.

The eSIGN editor provides a visual interface and a HamNoSys keyboard to write the signs in this notation. It works with the SiGML players to realize the signs by using the virtual avatars. The eSIGN Editor contains predefined BSL signs and provides an interface to form sentences with the help of these signs. Figure 2.8 shows the HamNoSys notation of the word "Defans" in eSIGN editor.

SIGN Editor	JA SIGML Player
File Edit Maintenance	File Avatar Ambient
	Figure Control Speed-Up (log scale)
	Play Play
	Stop
🕌 defans.xml	Cyclic-Play 1/32 1/8 1/2 1 2 4
+ - Join 🥰 defans.xml Sentence 1	Single-Sign
derans	
defans Gloss Mouth SRC GOL	
🖕 Edit HamNoSys 💿 🗉 🔀	
<pre></pre>	
	Status Player stopped: frame=32_cion=1
	hayer stopped, iranie=52, sign=1.

Figure 2.8. TID translation of the Turkish word "Defans" in eSIGN editor

EUDICO Linguistic Annotator $(ELAN)^6$ is another useful tool for sign language studies. It is a multi-layer annotation tool for video and audio contents. This tool is used to add transcriptions to sign language videos. Figure 2.9 shows an annotated sign language video with "Türkçe" and "TİD" tiers.

⁵http://www.visicast.cmp.uea.ac.uk/

 $^{^{6}} http://sign-lang.ruhosting.nl/echo/ELAN/ELAN_intro.html$



Figure 2.9. Sign language video annotation in ELAN

Zhao et al. [12] use Synchronous Tree Adjoining Grammar (STAG) to translate English text into ASL glosses. It maps English sentence to ASL by building elementary trees with lexical items such as verb, noun as nodes. These trees are joined together with substitution or adjunction events. Non-manual markers convey the meaning of the morphological markers such as tense. They are embedded into glosses of the target language. During the translation, while the input sentence is being parsed, the target language tree is generated by using the Tree-Adjoining Grammars (TAG). This system is named as TEAM and it is the first system that uses synchronous TAGs for sign language translation.

Hernandez et al. [13] propose a Spanish speech to Spanish Sign Language (LSE) translation system for assisting the deaf people with identity card applying or renewal process. The system converts officer's speech into sign language in real time. It has three components; speech recognizer, natural language translator, and 3D avatar animation. The speech recognizer component translates spoken utterance into word sequences. Then, the natural language translator converts these sequences into LSE glosses by implementing rule-based and statistical methods separately. Finally, the resulting LSE sequences are matched with the predefined HamNoSys notations of the signs and fed into eSIGN editor for avatar animation. The rule-based translator comprises 153 translation rules and achieved 0.578 BLEU score while the statistical translator scores 0.4941. The statistical translator is trained with 266 sentence pairs and tested with 150 sentences. It is important to note that the system has domain constraint and dataset only contains sentence pairs from this domain.

Manzano [14] introduces a Neural Machine Translation (NMT) system to translate English text into ASL. The proposed system is used as a natural language translation component of the Speech2signs project. This project interprets input video and extracts the speech, then converts the speech into text. After, it translates English text into ASL and realizes the ASL signs by virtual avatar. ASLG-PC12 [15] dataset is used as parallel corpus. Train dataset contains 83618 sentence pairs, development dataset 2045 and test has 2046. BLUE score of the system is denoted as 17.73.

Stoll et al. [16] implement a system that converts spoken language into sign language video. Unlike the aforementioned studies, it does not rely on the virtual avatars, instead implements its own sign video generation component with generative adversarial networks. Natural language translation component translates text into glosses. It is trained with German dataset and it is evaluated in terms of the cumulative BLEU scores. PHOENIX14T ⁷ dataset containing 8257 German to German Sign Language (DGS) sentences are used to train the component. This component achieves 50.67 BLUE-1, 32.25 BLUE-2, 21.54 BLUE-3, and 15.26 BLUE-4 scores.

¹⁰

⁷https://www-i6.informatik.rwth-aachen.de/ koller/RWTH-PHOENIX-2014-T/

3. DATASET

Sign languages use visual expressions and they don't have any written form. That makes it challenging to generate a large dataset. Uncertainty of the available sign language data and the lack of the strict grammatical rules also makes it harder. There are several notation systems available to represent the sign languages as described in the chapter 2.

In this study, gloss representation is used to typify the signs in the dataset and official online TİD dictionary is used to acquire the reliable, Turkish to TİD or vice versa, translations.

3.1. Online TID Dictionary Dataset

The Ministry of Family and Social Policies build an online Turkish to TID dictionary [17] containing video and gloss representations of the TİD signs. It also introduces Turkish to TİD sample sentences with relevant glosses. Figure 3.1 shows "Anlamak"⁸ Turkish word in online TİD dictionary.

In this study there is no need for word-to-word translations instead, the sentencealigned, bilingual corpus is required. To do so, sample sentences for each word translation, are used to compose the Turkish to TİD parallel corpus. Online TİD dictionary comprises 2000 words which are grouped alphabetically and it would be challenging to extract the sample sentences by handcraft.

In order to automate the sample sentence extraction task, a website crawler is implemented in javascript. For each letter, it fetches the relevant URL and parses the retrieved page. It extracts the number of available pages. Then, navigates to each page and parses the links of the words. Finally, it fetches these links and extracts Turkish and TİD sentences on the page. It saves these sentences into a file in javascript notation

 $^{^{8}}$ http://tidsozluk.net/tr/Anlamak?d=0012



Figure 3.1. "Anlamak" in online TID dictionary

format (JSON) as shown in figure 3.2. After all, 3561 sentence pairs are retrieved and saved as the bilingual corpus.

Generated corpus is then split into the test, train, and development corpora for different components of the system. Approximately 80% of the corpus is reserved as train corpus while remaining 20% is shared between test and development corpora. In order to ensure the sentence variety in each corpus, sentence pairs for each letter are split individually. In addition to it, each sentence within the split proportion is selected randomly. For instance, 80% of the sentence pairs that are retrieved for the letter "A", are randomly selected and added to the train corpus. Then, half of the remaining 20% is selected randomly and added to the test corpus. Finally, the remaining sentence

```
{
  "word": "Anlamak",
  "TID": "BEN ÖNCE ARABA SÜRMEK BİLMEK^DEĞİL BEN CAHİL BEN SONRA ARKADAS BEN
    ÖĞRETMEK ÖĞRETMEK ŞİMDİ BEN ANLAMAK ARABA SÜRMEK SÜRMEK",
  "Turkish": "Daha önce araba kullanmayı bilmiyordum, ancak arkadaşım bana
    öğretti. Şimdi anladım ve çok iyi araba kullanıyorum."
},
{
  "word": "Anlamak",
  "TID": "BEN OKUL GİTMEK BİR ARKADAŞ GELMEK^DEĞİL BEN ÖĞRETMEN SORMAK O HASTA
    YÜZDEN GELMEK^DEĞİL BEN ANLAMAK",
  "Turkish": "Bir arkadaşım bugün okula gelmedi. Ben de neden gelmediğini
    öğretmene sordum. Hasta olduğu icin gelmemiş, sebebini öğrendim (anlayınca
    rahatladım)."
},
{
  "word": "Anlamak",
  "TID": "BİR ARKADAŞ BEN İNANMAK^DEĞİL İNANMAK^DEĞİL SONRA İNANMAK ŞİMDİ
    ANLAMAK".
  "Turkish": "Önceden arkadaşıma inanmıyordum. Daha sonra inanmam gerektiğini
    anladım."
},
{
  "word": "Anlamak",
  "TID": "BEN SPOR GİTMEK GELMEK ZAYIF OLMAK^DEĞİL ANLAMAK^DEĞİL",
  "Turkish": "Spora qittiğim halde zayıfllayamadım, hiçbir şey anlamadım."
},
{
  "word": "Anlamak",
  "TID": "SEN HASTA AĞIR (COK FAZLA) BEN SEN ANLAMAK AMA MÜDÜR ZOR (BEN) DEMEK
   DEMEK DEMEK BEN BIKMAK BİRAZ İDARE ETMEK",
  "Turkish": "Çok hastasın anlıyorum ama müdür çok baskı yapıyor. Lütfen biraz
    daha idare et."
},
```

[

Figure 3.2. A part of the website crawler's output for letter "A"

pairs are saved as the development corpus. This process is performed for each letter. Eventually, among the 3561 sentence pairs, 2851 randomly selected ones are added to the train corpus, 363 are assigned to the test corpus, and 346 to the development corpus.

4. METHODOLOGY

Turkish to Turkish Sign Language hybrid translation system combines the advantages of the rule-based and statistical machine translation techniques. It consists of the three components; rule-based translation component, preprocessor, and statistical translation component.



Figure 4.1. Hybrid Translation System From Turkish Spoken Language to Turkish Sign Language Architecture

Turkish sentence is first processed by the rule-based translation component and it generates the intermediate sign language translations of the Turkish input sentence. The output of the rule-based translation component is named as system TİD (stid) throughout the study. Then, preprocessor fine-tunes the stid for statistical translation component. Statistical translation component applies the phrase-based statistical translation model by Moses Decoder. Figure 4.1 illustrates an overview of the proposed system.

4.1. Rule-Based Translation Component

One aspect of the main contribution of the thesis is the rule-based translation component. This component first analyzes Turkish input sentence morphologically by natural language analyzer then applies the predefined Turkish to TİD translation rules.

4.1.1. Turkish Natural Language Analyzer

Turkish input sentence must be examined extensively to implement the predefined transformation rules. Turkish language processing tools; ITU NLP Pipeline ⁹ and Zemberek¹⁰ are investigated to analyze the Turkish input sentence morphologically. On the advantage of the accessibility, ease of use and the portability, The Boun Morphological Analyzer is used. It consists of a probabilistic parser and a disambiguator.

The Boun Morphological Parser categorizes each word into word types like noun, adjective, verb and determines the morpheme details like dative, necessity, possessive. It splits input sentence into individual tokens and lists each token in a new line. All possible outputs which are separated with a space character for a token, are listed after the token. For each output, token's stem is itemized first, then stem type is written in square brackets. Each morpheme in the stem is concatenated with "+" character. In order to handle multiple input sentences, "<S> <S>+BSTag" and " $<\backslash S> <<S>+ESTag$ " tags are used to mark the beginning and the end of a sentence respec-

⁹http://tools.nlp.itu.edu.tr/

¹⁰http://zembereknlp.blogspot.com/

```
<S> <S>+BSTag
Sinava sinav[Noun]+[A3sg]+[Pnon]+YA[Dat]
çok çok[Postp]+[PCAbl] çok[Postp]+[PCAbl]+[A3sg]+[Pnon]+[Nom] çok[Adj] çok
[Det] çok[Adv]
az az[Postp]+[PCAbl] az[Postp]+[PCAbl]+[A3sg]+[Pnon]+[Nom] az[Adj] az[Noun]
+[A3sg]+[Pnon]+[Nom] az[Adv] az[Verb]+[Pos]+[Imp]+[A2sg]
kaldı kal(I)[Noun]+[A3sg]+[Pnon]+[Nom]-YDH[Verb+Past]+[A3sg] kal[Verb]+[Pos]
+DH[Past]+[A3sq]
acele acele[Adj] acele[Adv] acele[Noun]+[A3sg]+[Pnon]+[Nom]
ile ile[Postp]+[PCNom] ile[Postp]+[PCNom]+[A3sg]+[Pnon]+[Nom] il[Verb]+[Pos]
+YA[Opt]+[A3sg] il[Noun]+[A3sg]+[Pnon]+YA[Dat] ile[Conj]
arabaya araba[Adj]-[Noun]+[A3sg]+[Pnon]+YA[Dat] araba[Noun]+[A3sg]+[Pnon]+YA
[Dat]
bindim bin[Verb]+[Pos]+DH[Past]+m[A1sg] bin[Noun]+[A3sg]+[Pnon]+[Nom]-YDH
[Verb+Past]+m[A1sg] bin[Adj]-[Noun]+[A3sg]+[Pnon]+[Nom]-YDH[Verb+Past]+m
[A1sg] bin[Adj]-YDH[Verb+Past]+m[A1sg] bindi[Noun]+[A3sg]+Hm[P1sg]+[Nom]
ve ve[Conj]
hemen hemen[Adv]
gittim git[Verb]+[Pos]+DH[Past]+m[A1sg]
 .[Punc]
 +ESTag
```

Figure 4.2. The Boun Morphological Parser Output

tively. Morphological analysis of the Turkish sentence "Sınava çok az kaldı acele ile arabaya bindim ve hemen gittim." is shown in figure 4.2.

Morphological parser output includes all possible morphemes for a word. In order to select the most probable morphological result, The Boun Morphological Disambiguator is used. It gets morphological parser's output as input and replaces the most probable morphological analysis of the token into the first order. Morphological disambiguator output of the Turkish sentence "Sınava çok az kaldı acele ile arabaya bindim ve hemen gittim." is shown in Figure 4.3.

Once the input Turkish sentence is partitioned into words and the relevant morphemes, transformation rules are applied accordingly.

4.1.2. Turkish to Turkish Sign Language Transformation Rules

A deep understanding of Turkish Sign Language is required to define Turkish to TİD transformation rules. To do so, I have attended linguistic readings class in TİD at the linguistic department at Boğaziçi University which was very helpful to compre-

```
<S> <S>+BSTag
Sinava sinav[Noun]+[A3sg]+[Pnon]+YA[Dat]
cok cok[Adv] cok[Postp]+[PCAbl] cok[Postp]+[PCAbl]+[A3sg]+[Pnon]+[Nom] cok
[Adj] cok[Det]
az az[Adj] az[Postp]+[PCAbl] az[Postp]+[PCAbl]+[A3sg]+[Pnon]+[Nom] az[Noun]
+[A3sg]+[Pnon]+[Nom] az[Adv] az[Verb]+[Pos]+[Imp]+[A2sg]
kaldı kal[Verb]+[Pos]+DH[Past]+[A3sg] kal(I)[Noun]+[A3sg]+[Pnon]+[Nom]-YDH
[Verb+Past]+[A3sg]
acele acele[Adj] acele[Adv] acele[Noun]+[A3sg]+[Pnon]+[Nom]
ile ile[Postp]+[PCNom] ile[Postp]+[PCNom]+[A3sg]+[Pnon]+[Nom] il[Verb]+[Pos]
+YA[Opt]+[A3sg] il[Noun]+[A3sg]+[Pnon]+YA[Dat] ile[Conj]
arabaya araba[Noun]+[A3sg]+[Pnon]+YA[Dat] araba[Adj]-[Noun]+[A3sg]+[Pnon]+YA
[Dat]
bindim bin[Verb]+[Pos]+DH[Past]+m[A1sg] bin[Noun]+[A3sg]+[Pnon]+[Nom]-YDH
[Verb+Past]+m[A1sg] bin[Adj]-[Noun]+[A3sg]+[Pnon]+[Nom]-YDH[Verb+Past]+m
[A1sg] bin[Adj]-YDH[Verb+Past]+m[A1sg] bindi[Noun]+[A3sg]+Hm[P1sg]+[Nom]
ve ve[Conj]
hemen hemen[Adv]
gittim git[Verb]+[Pos]+DH[Past]+m[A1sg]
. .[Punc]
 +ESTag
```

Figure 4.3. The Boun Morphological Disambiguator Output

hend various TİD concepts such as tense, aspect, modal, possessives, suffixes, person agreement, word order, and negation. I have also attended Aspects of Visual Grammars course to gain knowledge about more complicated concepts such as epistemic modality, pronouns, role shift, and non-manuals, in different sign languages. In the light of these studies, the following transformation rules are declared. In addition to TİD knowledge, in-depth analysis of the literature is also significant guidance for rule formation phase. In this study 13 Turkish to TİD translation rules are defined and explained in detail below.

4.1.2.1. <u>Infinitive Verb Inflection.</u> Turkish Sign Language does not embody any suffixes. Instead, verbs are represented in infinitive forms while nouns are in nominative forms. TİD fills this gap by employing non-manual markers such as head tilt, eye gaze, and mouthings to convey the additional meanings or implications. This rule omits the suffixes of each word in the Turkish sentence and translates stems of the Turkish words into the correspondent TİD glosses. Stems other than the verbs are translated as they are while verb stems are inflected for their infinitive forms. The infinitive inflection rule is simply performed by inspecting the last vowel in the verb stem. If the last vowel in the verb stem, is a front vowel, it is conjugated with "-mek", otherwise "-mak" suffix is applied.

Turkish sentence:	Piknik	için	plan	yapmıştık
English:	Picnic	for	plan	have done
English translation:	We had	a pla	n for pi	cnic.

Disambiguator result:

Piknik	piknik[Noun]+[A3sg]+[Pnon]+[Nom]
için	<pre>için[Postp]+[PCNom] için[Postp]+[PCNom]+[A3sg]+[Pnon]+[Nom]</pre>
plan	plan[Noun]+[A3sg]+[Pnon]+[Nom]
yapmıştık	<pre>yap[Verb]+[Pos]+mHş[Narr]+YDH[Past]+k[A1p1]</pre>
,	,[Punc]

According to the disambiguator result, given Infinitive Verb Inflection rule converts "yap" verb stem into "YAPMAK" infinitive verb and keeps other stems as they are. So that, it translates "Piknik için plan yapmıştık." Turkish sentence into "PİKNİK İÇİN PLAN YAPMAK" TİD glosses.

On the other hand, passive and causative verbs are exceptions for this rule since they derive new words from the stems. For instance "üzüldüm" passive word is parsed into "üz" verb stem by the disambiguator as shown below. The infinitive verb inflection rule transforms "üz" verb stem into "ÜZMEK" infinitive form instead of "ÜZÜLMEK".

Turkish sentence:	Çok	üzüldüm
English:	Very	I was sorry
English translation:	I was	very sorry.

Disambiguator result:

```
Çok çok[Adv]
üzüldüm üz[Verb]-H1[Verb+Pass]+[Pos]+DH[Past]+m[A1sg]
. . [Punc]
```

.

In order to eliminate this problem, passive and causative verb stems are regenerated by appending the derivative suffixes to the root stem. According to this rule, "Çok üzüldüm" Turkish sentence translated into "ÇOK ÜZÜLMEK" TİD glosses rather than "ÇOK ÜZMEK".

4.1.2.2. <u>Punctuation Marks</u>. Punctuation Marks in Turkish input sentence are eliminated since they are not used in TİD.

4.1.2.3. <u>Conjunctions.</u> In order to cover all conjunctions in Turkish, three different rules are defined.

"-ki" connector (relative pronoun) in Turkish input sentence is omitted since it is nonfunctional in TİD as shown in the sample below.

Turkish sentence: aklı başına geldi Oğlum o kadar yaramaz ki ancak düştükten sonra English: My son naughty but he fell after he came to his sen \mathbf{SO} English translation: My son is so naughty, but after he fell, he came to his sense.

If "-de" connector follows a verb in the Turkish input sentence, the verb is reduplicated in TİD.

Turkish sentence:Ben de sustum.English:I also quieted down.English translation:I also quieted down.

Disambiguator result:

```
ben ben[Pron]+[Pers]+[Alsg]+[Pnon]+[Nom] be[Noun]+[A3sg]+Hn[P2sg]+[Nom]
de de[Conj] de[Verb]+[Pos]+[Imp]+[A2sg] de[Noun]+[A3sg]+[Pnon]+[Nom]
sustum sus[Verb]+[Pos]+DH[Past]+m[Alsg]
```

According to the disambiguator result, the aforementioned rule translates "Ben de sustum." Turkish sentence into "BEN SUSMAK SUSMAK" TİD glosses.

Other conjunctions like "ve", "ama" and "ile" are translated from Turkish into TİD as they are.

4.1.2.4. <u>Person Agreement.</u> This rule is only applied to the verbs in the sentence to extract person information. If a verb has person agreement, the corresponding personal pronoun is added to the beginning of the TİD sentence.

Turkish sentence:	Hemen	hastaneye	gittik	•
English:	Immediately	to hospital	we went	
English translation:	We went to hospital immediately.			

Disambiguator result:

hemen	hemen[Adv]
hastaneye	hastane[Noun]+[A3sg]+[Pnon]+YA[Dat]
gittik	git[Verb]+[Pos]+DH[Past]+k[A1pl]

According to the disambiguator result, Person Agreement rule translates "Hemen hastaneye gittik." Turkish sentence into "BİZ HEMEN HASTANE GİTMEK" TİD glosses.

4.1.2.5. <u>Present Tense Rule.</u> This rule is defined to convey the time information. If any verb in the Turkish input sentence has only progressive feature as the time indicator and also has the first single person agreement, "ŞİMDİ" gloss is added to the head of the TİD sentence as the time adverb.

Turkish sentence:Çoküzülüyorum.English:VeryI'am sorry.English translation:I'am very sorry.

Disambiguator result:

Çokçok[Adv]üzülüyorumüz[Verb]-Hl[Verb+Pass]+[Pos]+Hyor[Prog1]+YHm[A1sg]..[Punc]

According to the disambiguator result, Present Tense rule translates "Çok üzülüyorum." Turkish sentence into "BEN ŞİMDİ ÇOK ÜZÜLMEK" TİD glosses.

4.1.2.6. <u>Past Tense Rule.</u> This rule is defined to convey the time information. If a verb in the Turkish sentence has past tense inflection along with progressive feature, "BİTTİ" gloss is added to the end of the TİD sentence as the time adverb.

Turkish sentence:	Hemen	mum	yaktık	•
English:	Immediately	a candle	we lit	
English translation:	We lit a cand	le immedia	tely.	

Disambiguator result:

Hemen	hemen[Adv]
mum	<pre>mum[Noun]+[A3sg]+[Pnon]+[Nom]</pre>
yaktık	<pre>yak[Verb]+[Pos]+DH[Past]+k[A1pl]</pre>
	.[Punc]

According to the disambiguator result, Past Tense rule translates "Hemen mum yaktık" Turkish sentence "BİZ HEMEN MUM YAKMAK BİTTİ" TİD glosses.

4.1.2.7. <u>Future Tense Rule.</u> Turkish Sign Language does not employ future tense.

4.1.2.8. <u>Necessity Rule.</u> Necessitative which is relayed with "-meli", "-mah" suffixes in Turkish language, is transferred to TİD by "LAZIM" gloss. It is concatenated to the infinite form of the word stem.

Turkish sentence:	Cam	su	şişelerinden	almalısınız	
English:	Glass	water	bottles	should buy	
English translation:	You sh	ould bu	y glass water	bottles.	

Disambiguator result:

cam	<pre>cam[Noun]+[A3sg]+[Pnon]+[Nom]</pre>
su	su[Noun]+[A3sg]+[Pnon]+[Nom]
şişelerinden	<pre>şişe[Noun]+lAr[A3pl]+SH[P3sg]+NDAn[Abl]</pre>
almalısınız	al[Verb]+[Pos]+mAlH[Neces]+sHnHz[A2pl]
•	.[Punc]

According to the disambiguator result, Necessity rule translates "Cam su şişelerinden almalısınız." Turkish sentence into "SİZ PLASTİK ŞİŞE SAĞLIK ZARAR CAM SU ŞİŞE ALMAK LAZIM" TİD glosses.

4.1.2.9. <u>Negation Rule.</u> Privative affixes "-ma", "-me" and "-madan", "-meden" conveys the negation meaning in Turkish while "DEĞİL" gloss is used in TİD. If a verb has privative affix in the Turkish input sentence, "DEĞİL" gloss is attached to the infinite form of the word stem and it is represented as a multi-word expression in TİD.

Turkish sentence:	Müdür	beğenmedi	
English:	Manager	he/she didn't like	
English translation:	Manager o	lidn't like.	

Disambiguator result:

müdür	müdür[Noun]+[A3sg]+[Pnon]+[Nom]
beğenmedi	beğen[Verb]+mA[Neg]+DH[Past]+[A3sg]
•	.[Punc]

According to disambiguator result, Negation rule translates "Müdür beğenmedi." Turkish sentence to "MÜDÜR BEĞENMEK^DEĞİL" TİD glosses. It is also important to note that some words in TID embody separate signs for their negation forms rather than using "DEĞİL" sign. For example, "sevmiyorum" Turkish word is signed with "SEVMEMEK" instead of "SEVMEK^DEĞİL" in TİD. But this exception is not handled in this study since there is no strict rule about it.

4.1.2.10. <u>Possessive Rule.</u> The possessive suffix in Turkish is translated into possessive pronoun in TİD and it is prepended to the relevant word stem.

Turkish sentence:	Arabam	var	•
English:	My car	have	
English translation:	I have a car.		

Disambiguator result:

```
arabam araba[Noun]+[A3sg]+Hm[P1sg]+[Nom]
var var[Adj]
. .[Punc]
```

According to the disambiguator result, Possessive rule translates "Arabam var." Turkish sentence into "BENİM ARABA VAR" TİD glosses.

4.1.2.11. <u>Locative Rule.</u> The locative meaning in Turkish is transferred to TİD by utilizing "İÇİNDE" gloss. If a noun is inflected with locative suffix and followed by a verb in Turkish sentence, it is translated to TİD by appending "İÇİNDE" gloss to its stem.

Turkish sentence:	Doğum günü partimi	evde	yapmayı	düşünüyordum	•
English:	the birthday my party	at home	to make	I was thinking	
English translation:	I was thinking to make	my birthd	ay party at	home.	

Disambiguator result:

Doğum	doğum[Noun]+[A3sg]+[Pnon]+[Nom]
günü	gün[Noun]+[A3sg]+SH[P3sg]+[Nom]
partimi	parti[Noun]+[A3sg]+Hm[P1sg]+NH[Acc]
evde	ev[Noun]+[A3sg]+[Pnon]+DA[Loc]
yapmayı	<pre>yap[Verb]+[Pos]-mA[Noun+Inf2]+[A3sg]+[Pnon]+YH[Acc]</pre>
düşünüyordum	düşün[Verb]+[Pos]+Hyor[Prog1]+YDH[Past]+m[A1sg]
	.[Punc]

According to the disambiguator result, Locative rule translates "Doğum günü partimi evde yapmayı düşünüyorum." Turkish sentence into "BİZ DOĞUM GÜN BENİM PARTİ EV İÇİNDE YAPMAK DÜŞÜNMEK" TİD glosses.

4.1.2.12. <u>Ablative Rule.</u> The ablative suffixes in Turkish sentence, are omitted since they are not used in TID.

4.1.2.13. <u>Proper Nouns.</u> Fingerspelling is the representation of each letter of a word by hand movements in sign languages. If there is a proper noun in Turkish sentence, "fingerspell" mark is appended to its translation in TİD.

Turkish sentence:	İş	bulamayınca	İstanbul'a	taşındım	•
English:	Job	could not find	to İstanbul	moved	•
English translation:	Whe	n I could not fine	d a job, I mov	ved to İstan	bul.

Disambiguator result:

iş	iş[Noun]+[A3sg]+[Pnon]+[Nom]
bulamayınca	bula[Verb]+mA[Neg]-YHncA[Adv+When]
Istanbul'a	<pre>Lstanbul[Noun]+[Prop]+[A3sg]+[Pnon]+'[Apos]+YA[Dat]</pre>
taşındım	<pre>taşın[Verb]+[Pos]+DH[Past]+m[A1sg]</pre>
	.[Punc]

According to the disambiguator result, Proper nouns rule translates "İş bulamayınca İstanbulá taşındım." Turkish sentence into "İŞ BULAMAKDEĞİL İSTANBULÊS TAŞINMAK" TİD glosses.

4.1.3. Rule-Based Translator

The rule-based translator is a python based application that implements the aforementioned rules by utilizing The Boun Morphological Analyzer. It gets input sentences as a file and executes morphological parser, morphological disambiguator and translation rules consecutively. Resulting translations are then saved into the given output file. For debug purposes, outcomes of the disambiguator and the parser are saved as intermediate results into the "out" folder.

python translator.py corpus.tr corpus.stid

When the above command is issued, the translator first reads sentences from the "corpus.tr" file one by one and feeds each sentence into The Boun Morphological Parser individually. After all sentences are processed, parser output is written into "parserResult.txt" file under "out" folder. Then, parser results are given as input to The Boun Disambiguator in order to prioritize the most convenient parser output for each sentence. Disambiguator output is also saved into the "out" folder and named as "disambiguatorResult.txt".

After the Turkish natural language analysis is completed, sign language transformation rules are applied, having the precedence of the rules in mind. First, the infinitive rule is applied to translate the verb stem into TİD. Then, rest of the rules are performed which build upon the infinitive verb inflection rule, by preserving the order they are represented in section 4.1.2.

Finally, the rule-based translator fine-tunes the translation results by extra enhancements. It first trims the sentence then eliminates the rule collisions such as possessive and personal pronoun conflictions.

Turkish sentence:Ailemden ayrıyaşıyorumEnglish word alignments:My family apart from I live.English translation:I live apart from my family.

Applied transformation rules:

Possesive Rule:	ailemden -> BENIM AILE
Person Agreement Rule:	yaşıyorum -> BEN YAŞAMAK
Final Translation:	BEN BENIM AILE AYRI YAŞAMAK

The rule-based translator detects the collision in the above sentence and subtracts the redundant "BENİM" possessive pronoun. So it converts final translation into "BEN AİLE AYRI YAŞAMAK" TİD sequences.

4.2. Preprocessor

The preprocessing stage is required to reduce data sparsity for the evaluation process and statistical machine translation components. In order to calculate consistent BLEU scores for the system evaluation, the translated output and correspondent test sentence should be well aligned in terms of the punctuation, case sensitivity, and sentence length. These divergences mislead training and tuning phases of the machine translation component.

Turkish sentence:	Balık	yemeyi	hiç	sevmiyorum	
English:	Fish	eating	at all	I don't like	
English translation:	I don't	like eatii	ng fish a	it all.	
$T\dot{I}D$ system translation:	BEN E	BALIK YI	EMEK]	HİÇ SEVMEK^DEĞİL	
After TİD preprocessor	ben balık yemek hiç sevmekdeğil				
After Turkish preprocessor	balık yemeyi hiç sevmiyorum				

Moses decoder already implements tokenizers however they are not applicable to this study since sign languages have different syntactic patterns than spoken languages. As shown in the sample above, a generic preprocessor replaces the ^ punctuation with space character, which converts "SEVMEK^DEĞİL" single word into "SEVMEK" and "DEĞİL" words. So, this result confuses the translation model training since it interprets "SEVMEK" and "DEĞİL" as two different words and could not align with "sevmiyorum" Turkish word. In order to overcome the language-specific concerns, custom preprocessors are implemented. One for processing Turkish and one for TİD.

4.2.1. Custom Turkish Preprocessor

Custom Turkish preprocessor first eliminates the expressions in the parentheses, then converts all characters into the lowercase with Turkish encoding. After, it deletes "ki" and "de" conjunctions since they don't have individual representations in TİD. Lastly, it removes all punctuations, empty lines and trims the redundant whitespaces.

A simple python script which gets input and output file names as the parameters is implemented for this purpose.

```
python TurkishPreprocessor.py corpus.tr corpus.processed.tr
```

The above command is issued to process the Turkish input sentences and sample results are listed in table 4.1.

Turkish	Processed Turkish
Oğlum yüzmeyi bilmediği için	oğlum yüzmeyi bilmediği için
sürekli batıyordu, şimdi ona	sürekli batıyordu şimdi ona
öğrettim ve çok güzel yüzüyor.	öğrettim ve çok güzel yüzüyor
Yavru kedi öyle çok ağlıyordu ki	yavru kedi öyle çok ağlıyordu önce
önce ne olduğunu anlamadım. Sonra	ne olduğunu anlamadım sonra gördüm
gördüm ki annesi ölmüş.	annesi ölmüş
Resim konusunda her şeyi bilen	resim konusunda her şeyi bilen bir
(konusuna hakim olan) bir ressamın	ressamın resimlerini çok beğendim
resimlerini çok beğendim ve bir	ve bir tablosunu aldım
tablosunu aldım.	

Table 4.1. Preprocessor results of the Turkish sentences

4.2.2. Custom TID Preprocessor

Unlike Turkish, expressions in the parentheses deliver significant information rather than extra, in TİD. So these expressions are not omitted. Instead, they are treated as standard expressions. Custom TİD preprocessor first extracts the expressions in the parentheses, then removes the punctuations.

In TİD sentence "^" character is used to sign the negations and multi-word expressions such as "GİTMEK^GELMEK". If the circumflex accent is used to convey the negation, preprocessor deletes it and concatenates the negation marker "DEĞİL" to the former word. On the other hand, if it is used to express the multi-words, preprocessor splits these words by replacing the circumflex accent with whitespace.

Finally, preprocessor removes the fingerspell marker "^{FS}" and converts the all characters into lowercase with Turkish encoding.

A simple python script which gets input and output file names as the parameters is implemented for this purpose.

python TIDPreprocessor.py corpus.tid corpus.processed.tid

The above command is issued to process the TID input sentences and sample results are listed in table 4.2.

4.3. Statistical Translation Component

Statistical Translation Component implements statistical machine translation (SMT) techniques to translate the Turkish Spoken Language into the TİD. SMT approach is a state-of-the-art translation methodology which relies on the statistical models that are extracted from the parallel data. Most of the common translators such as Google and Microsoft also utilize SMT methods.

```
TIDPreprocessed TIDBEN DÜĞÜN GITMEK ADAM DÖRT ELBISEben düğün gitmek adam dört elbise(HEP)AYNIhep aynıBEN IŞ GITMEK^GELMEK (BIKMAK) BIRben iş gitmek gelmek bıkmak bir ayAY RAPOR VERMEK RAHAT GEZMEK GEZMEKrapor vermek rahat gezmek gezmekO KIZ KARDEŞ IŞ GITMEKo kız kardeş iş gitmek istemekdeğilISTEMEK^DEĞIL AMA ÇOK MASRAF MASRAFama çok masraf masraf para neredePARA NEREDE?
```

Table 4.2. Preprocessor results of the TID sentences

This component takes the advantage of the Moses Decoder [5] to perform the statistical machine translation. Moses Decoder has two main components; a training pipeline which is a collection of tools for generating language models and a decoder to translate the input sentence. Language modeling and Tuning are also significant parts of the translation system.

4.3.1. Language Model

Language model generates the grammatical pattern of the target language in order to validate the translation output, therefore, it only operates on the target language. RandLM, KenLM, OxLM, NPLM are some of the language model generation tools. In this study, we are using KenLM tool which is included in the Moses Decoder. The following command is executed to generate a trigram language model for TİD. The language model is created in the ARPA format as shown in figure 4.4.

bin/lmplz -o 3 < corpus11/corpus.tid > corpus11/lm/corpus.tid.arpa

In order to reduce memory load time, generated language model in ARPA format is transformed into binary files by the following command.

```
\data\
ngram 1=2981
ngram 2=26551
ngram 3=41085
\1-grams:
-4.378458
                 <unk>
                         0
                 -0.7585538
0
        <S>
                 </s>
-1.677744
                         0
-1.4511296
                 ben
                         -0.4979915
-2.7372835
                 önce
                         -0.15951268
-2.5570967
                 araba
                         -0.28084177
                 sürmek
-2.9989052
                         -0.24098742
-2.7284415
                 bilmekdeğil
                                  -0.26812363
                 cahil
-3.759625
                         -0.14511697
-2.123881
                 sonra
                         -0.28880936
-2.325029
                 arkadas -0.32046622
-3.1314256
                 öğretmek
                                  -0.27599913
                 simdi
-2.496618
                         -0.21225025
-2.9828882
                 anlamak -0.22728384
-2.5397499
                 okul
                         -0.27513638
-2.1068158
                 gitmek
                        -0.3380863
```

Figure 4.4. Part of a sample language model

Produced language model is fed into the training pipeline in order to create a translation model.

4.3.2. Training Pipeline

Training Process consists of several toolkits which are executed as a pipeline. Stages of the pipeline are described below.

4.3.2.1. <u>Corpus Preparation</u>. The first stage of the pipeline is corpus preparation. In order to train the system, a parallel corpus which is also called bitext is required. The parallel corpus contains a collection of sentence pairs in the source and the target languages. It must be aligned at the sentence level and must not have empty lines. Parallel corpus generation is handled in chapter 3.1, once it is ready, the preprocessor structures it syntactically. Moses Decoder already has a tokenizer. It is very practical for spoken languages however not applicable to TİD since it is realized with gloss representation. In order to process the gloss representation, custom preprocessors

which are described in chapter 4.2, are used. A part of the preprocessed parallel train corpus is in table 4.3.

Turkish	TID
daha önce araba kullanmayı	ben önce araba sürmek bilmekdeğil
bilmiyordum ancak arkadaşım bana	ben cahil ben sonra arkadaş ben
öğretti şimdi anladım ve çok iyi	öğretmek öğretmek şimdi ben anlamak
araba kullanıyorum	araba sürmek sürmek
her gün işe gidip geliyorum ve	ben her gün iş gitmek gelmek çocuk
sonrasında eşimle ve çocuğumla	eş ben ilgilenmek ilgilenmek ben
ilgileniyorum o kadar bunaldım bir	boş şişmek bunalmak bir tatil
tatile çıkmak istiyorum	gitmek istemek
bayat balık insanı zehirler balık	balık göre bayat olmak yemek insan
taze yenilmelidir	zehir olmak taze yemek lazım

Table 4.3. A part of the preprocessed train corpus

4.3.2.2. <u>Word Alignment.</u> Word alignment for the training phase is handled by GIZA++ which implements statistical IBM models. Extracted word alignments are utilized for the phrase-based translations. The output of the GIZA++ on a sample Turkish to TİD train data is given below.

Turkish sentence in parallel corpus:	araba	tamirinden	anlıyorum
English:	the car	repair	understand
English translation:	I unders	tand the car	repair
TID sentence in parallel corpus:	ben aral	oa tamir hepsi	i anlamak

GIZA++ word alignments:

#Sentence pair (5) source length 3 target length 5 alignment score : 2.8501e-06 ben araba tamir hepsi anlamak

NULL (1) araba (2) tamirinden (34) anlıyorum (5)

According to GIZA++ word alignments, the word "araba" in Turkish sentence is aligned with the second gloss in TİD sentence which is also "araba" while "tamirinden" Turkish word is aligned with "tamir" and "hepsi" glosses in TİD. Finally "anhyorum" Turkish word is aligned with "anlamak" TİD gloss.

GIZA++ word alignments are extended by applying the grow diagonal final heuristic which is set as the default alignment heuristic. It first aligns the intersections of two alignments then grows by adding other alignment points.

4.3.2.3. <u>Lexical Translation Table.</u> Based on the word alignments, the lexical translation table is generated. Lexical translation table lists the source word, target word and the translation probability between them in the space-separated format as shown below.

numara numara 0.0312500 numara sürekli 0.1250000 sıkışıp sıkmak 0.0476190 sınırında gürcistan 0.3333333 sınırında için 0.0037879 sınırında yer 0.0049020 dünkü dün 0.0322581 voleybol önce 0.0086207 voleybol kıyasıya 0.111111 voleybol NULL 0.0002847 voleybol voleybol 0.4800000

4.3.2.4. <u>Phrase Table.</u> Lexical translation table and the word alignments are used to compose the phrase table. Word alignments are utilized to extract the phrases while lexical translation table is used to score them. Source phrase, target phrase, scores, alignment, counts, sparse feature scores and the key-value properties which are separated by three pipe characters (|||) are listed in the phrase table.

Scores column of the phrase table consists of inverse translation probability, inverse lexical weighting, direct phrase translation probability and the direct lexical weighting in space-separated format. Sample output of the phrase table is shown below.

```
aceleyle evden ||| acele acele ||| 0.5 1.898e-05 0.5 0.0002391 ||| 0-0 |||
2 2 1 ||| |||
aceleyle evden ||| acele ||| 0.0588235 1.898e-05 0.5 0.333333 ||| 0-0 ||| 17
2 1 ||| |||
aceleyle evden çıktım ||| acele acele çıkmak ||| 1 4.77484e-07 1 2.51684e-05
||| 0-0 2-2 ||| 1 1 1 ||| |||
```

4.3.2.5. <u>Reordering Model.</u> Reordering model is used to align the phrases of the source and the target languages with an optimum cost. Moses Decoder utilizes distance-based reordering model by default. This model assigns a linear cost with regard to the skipped words. For example, the cost of skipping over a word is 1 while two words doubles the cost.

Lexicalized reordering models are configured with 5 factors; model type, orientation, directionality, language and collapsing.

Word-based extraction, phrase-based and hierarchical models are the candidates for the model type configuration.

Orientation type parameter defines the ordering types that will be utilized in model training. Monotone, swap, discontinuous, discontinuous-left and discontinuous-right are the different orientation types in SMT. In monotone order, current phrase follows the previous phrase which means that there is no reordering. But if the current phrase is replaced with the previous phrase it is called swap ordering. Besides, if the phrase is placed to any position in the target language, it is called discontinuous ordering. These ordering types are illustrated in figure 4.5.



Figure 4.5. Monotone, swap, and discontinuous orientation classes [5]

The directionality parameter defines the route of the orientation by backward, forward and bidirectional options. Backward directionality employs the orientation based upon the previous phrase while forward directionality relies on the following phrase. Finally, bidirectional directionality incorporates both.

The language parameter determines whether the reordering model leans solely on target language or both target and source languages. "f" and "fe" values are given as the language parameter representing the targe and the source languages respectively.

Collapsing parameter specifies how to handle the scores. "allff" option treats scores individually, while "collaseff" cumulatively.

In this study, reordering model is generated with "msd-bidirectional-fe" parameter which sets "bidirectional" as directionality and "msd" which stands for "monotone", "swap" and "discontinuous", as orientation. In addition, "fe" parameter specifies that both source and target languages are used in the reordering model generation process. Model type and collapsing parameters are used with their default values; "word-based extraction (wbe)" and "allf" respectively. Part of the sample reordering table is shown below.

```
acele ediyordum ama kızım ||| kız ||| 0.2 0.2 0.6 0.2 0.2 0.6
acele etme biraz bekle deyince adam bir ||| adam acele ||| 0.2 0.2 0.6 0.2
0.2 0.6
acele hazırlanıp ||| gelmekdeğil ||| 0.2 0.2 0.6 0.6 0.2 0.2
acele hazırlanıp ||| müdür gelmekdeğil ||| 0.2 0.2 0.6 0.6 0.2 0.2
```

As stated above, training process comprises of several stages and works as a pipeline. In order to train the system with training corpus containing 2852 sentence pairs, the following command which triggers the aforementioned steps consecutively, is executed. The generated language model is also used by the training pipeline.

../scripts/training/train-model.perl -root-dir train -corpus ../corpus11/corpus -f tid -e tr -alignment grow-diag-final-and -reordering msd-bidirectional-fe -lm 0:3:/home/os/Desktop/boun/Thesis/mosesdecoder/corpus11/lm/corpus.tid.blm:8 -external-bin-dir ../tools >& training.out

Upon the completion of the training pipeline, Moses configuration file is generated along with the phrase table, reordering table and other intermediate results such as word alignments and the lexical translation table. Figure 4.6. demonstrates a sample of the Moses configuration file which is then fed into the Tuning phase.

4.3.3. Tuning

The tuning process improves the translation quality of the translation model which is generated by the training pipeline. A parallel corpus other than the training corpus is used to fine-tune the translation model's output by comparing the target sentence in the development corpus with the target sentence that is generated by the translation model, for the same source sentence. In order to find out the best translation, different statistical models are scored. Minimum Error Rate Training (MERT) tuning algorithm is executed with the following command to optimize the translation system with the development corpus. scripts/training/mert-moses.pl corpus11/tuneCorpus.tr corpus11/tuneCorpus.tid bin/moses working11/train/model/moses.ini --mertdir /home/os/Desktop/boun/Thesis/mosesdecoder/bin/ &> mert.out

A sample of the MERT optimized Moses configuration file which is generated after the tuning stage, is shown in Figure 4.7.

4.3.4. Decoder

Decoder implements the beam search algorithm to find out the best translation for the given source language by means of the trained translation model. It is a standalone C++ application that is executed with the command below.

```
bin/moses -f working11/mert-work/moses.ini < corpus11/testCorpus.tr >
working11/testCorpus.translated.tid 2> translation.out
```

The test corpus is fed into the decoder which determines the correspondent target sentences and lists them in the output file.

```
### MOSES CONFIG FILE ###
# input factors
[input-factors]
Θ
# mapping steps
[mapping]
0 T 0
[distortion-limit]
6
# feature functions
[feature]
UnknownWordPenalty
WordPenalty
PhrasePenalty
PhraseDictionaryMemory name=TranslationModel0 num-features=4 path=/home/
os/Desktop/boun/Thesis/mosesdecoder/working11/train/model/phrase-table.gz
input-factor=0 output-factor=0
LexicalReordering name=LexicalReordering0 num-features=6 type=wbe-msd-
bidirectional-fe-allff input-factor=0 output-factor=0 path=/home/os/
Desktop/boun/Thesis/mosesdecoder/working11/train/model/reordering-
table.wbe-msd-bidirectional-fe.gz
Distortion
KENLM name=LM0 factor=0 path=/home/os/Desktop/boun/Thesis/mosesdecoder/
corpus11/lm/corpus.tokenized.tid.blm order=3
# dense weights for feature functions
[weight]
# The default weights are NOT optimized for translation quality. You MUST
tune the weights.
# Documentation for tuning is here: http://www.statmt.org/moses/?
n=FactoredTraining.Tuning
UnknownWordPenalty0= 1
WordPenalty0= -1
PhrasePenalty0= 0.2
TranslationModel0= 0.2 0.2 0.2 0.2
LexicalReordering0= 0.3 0.3 0.3 0.3 0.3 0.3
Distortion0= 0.3
LM0 = 0.5
```

Figure 4.6. Sample of the Moses configuration file

```
# MERT optimized configuration
# decoder /home/os/Desktop/boun/Thesis/mosesdecoder/bin/moses
# BLEU 0.0914003 on dev /home/os/Desktop/boun/Thesis/mosesdecoder/
corpus11/tuneCorpus.tokenized.tr
# We were before running iteration 7
# finished Crs Kas 28 12:43:46 EET 2018
### MOSES CONFIG FILE ###
# input factors
[input-factors]
# mapping steps
[mapping]
0 T 0
[distortion-limit]
# feature functions
[feature]
UnknownWordPenalty
WordPenalty
PhrasePenalty
PhraseDictionaryMemory name=TranslationModel0 num-features=4 path=/home/
os/Desktop/boun/Thesis/mosesdecoder/working11/train/model/phrase-table.gz
input-factor=0 output-factor=0
LexicalReordering name=LexicalReordering0 num-features=6 type=wbe-msd-
bidirectional-fe-allff input-factor=0 output-factor=0 path=/home/os/
Desktop/boun/Thesis/mosesdecoder/working11/train/model/reordering-
table.wbe-msd-bidirectional-fe.gz
Distortion
KENLM name=LM0 factor=0 path=/home/os/Desktop/boun/Thesis/mosesdecoder/
corpus11/lm/corpus.tokenized.tid.blm order=3
# dense weights for feature functions
[weight]
LexicalReordering0= 0.0492443 0.00086267 0.0501674 0.01943 0.238079
0.0313582
Distortion0= 0.0483675
LMO= 0.0943588
WordPenalty0= -0.291252
PhrasePenalty0= 0.0523008
TranslationModel0= 0.000585845 0.0617664 0.0177821 0.0444445
UnknownWordPenalty0= 1
```

Figure 4.7. Sample of the Mert Optimized Moses configuration file

5. EXPERIMENTS AND RESULTS

There are two main approaches to measure the accuracy of the machine translation systems; human evaluation and automated scoring metrics. These two natural language oriented approaches are also applicable to the sign languages.

Human evaluation method already has bottlenecks such as subjectiveness, time consumption, and non-reproducibility, for the evaluation of the spoken language translations. In addition, it has a major drawback for the sign languages; most of the native signers have trouble to express and interpret sign languages in written forms. Since they generally learn the sign languages visually from their family and they don't have a theoretical background about it. In point of TİD, most of the grammatical rules are not well defined yet and it could be misleading to rely on the evaluation of non-signers. Due to the aforementioned obstacles, automated scoring metrics method is used for the system evaluation rather than the human evaluation method.

Bilingual evaluation understudy (BLEU) scoring metric is used to asses the system performance. BLUE calculates the similarity between the original translation and machine translation statistically. It does not take the translation intelligence and grammaticalness into account. In order to compute the similarity score, n-gram models of the original and the machine translations, are compared regardless of their positions. Higher the n-gram precision, higher the BLEU score. BLEU also employs the brevity penalty to eliminate the short sentences which cause high scores.

Performance evaluation of this study is performed by calculating cumulative BLEU scores. Cumulative BLEU score which is called BLEU-n for n-gram precision, weights the individual BLEU scores and calculates the geometric mean of them. BLEU-n formula is given below. λ_i represents the weight of BLEU-i score, in the cumulative score. Brevity penalty is set to 1 as default.

BLEU-n = brevityPenalty exp
$$\sum_{i=1}^{n} \lambda_i \log precision_i$$
 (5.1)

Moses decoder already implements a perl script to compute BLEU-4 cumulative score. This script is modified to calculate each BLEU-n score. Equal weights are assigned to individual precisions during this calculation.

Proposed system's performance is directly proportional to the performance of the translation components. For this reason, performance of the rule-based and statistical translation components are measured individually and compared to the hybrid translation system.

5.1. Rule-Based Translation Component Performance

Rule-based translation component is executed to translate the Turkish test corpus containing 363 sentences into TİD. Then, translation results are processed by the custom TİD preprocessor and BLEU scores are calculated. Rule-based translation component's results are fed into the preprocessor first. Original TİD translations are also processed by the preprocessor. Then, in order to calculate the BLEU scores, these translations are compared. A part of the translation result is listed below. Turkish input sentenceRule-based TID translationEvde montları asmak için bir askıEV LÇ MONT ASKI YOK YENI ASKI ALMAKyoktu, şimdi yeni bir tane aldım.BEN KOYMAK MONT ASMAK ASMAKRahatlıkla montları aslıyorum.EV LÇ MONT ASMAK ASMAK

Doğum günümde annem bana altın küpe hediye etti. Çok şaşırdım. Küpe sevmiyorum ama annem için taktım. BEN DOĞUM GÜN ANNE SÜRPRIZ ALTIN KÜPE HEDIYE ETMEK BEN ŞAŞIRMAK SEVMEK^DEĞIL AMA ANNE IÇIN KÜPE TAKMAK

Bugün günlerden pazar ve koşu yarışması olduğu için yollar saat dörtten sonra açılacak. BUGÜN PAZAR KÖPRÜ KOŞMAK VAR SAAT

Rule-based translation results and the original TID translations are shown in table 5.1.

Rule-based TID translation	Original TID translation
BEN EV MONT ASMAK IÇINDE IÇIN BIR	EV LÇ MONT ASKI YOK YENL ASKI ALMAK
RAHAT MONT	DEN KUIMAR MUNI ASMAR ASMAR

BEN DOĞUM BENIM GÜN BENIM ANNE BEN BEN DOĞUM GÜN ANNE SÜRPRIZ ALTIN ALTIN KÜPE HEDIYE ETMEK IÇINDE ÇOK KÜPE HEDIYE ETMEK BEN ŞAŞIRMAK ŞAŞIRMAK KÜPE SEVMEK^DEĞIL AMA BENIM SEVMEK^DEĞIL AMA ANNE IÇIN KÜPE ANNE IÇIN TAKMAK TAKMAK

BUGÜN GÜN PAZAR VE KOŞU YARIŞMA OLMAK BUGÜN PAZAR KÖPRÜ KOŞMAK VAR SAAT LÇIN YOL SAAT DÖRT SONRA AÇILMAK DÖRT SONRA ARABA YOL AÇIK

Table 5.1. Comparision of the rule-based translation results and the original TID translations

System translations and the original TID translations are then fed into the preprocessor. A part of the results is listed below.



According to the preprocessed translation results above, BLUE-1, BLEU-2, BLEU-3, and BLEU-4 performance scores are measured and illustrated below.



Cumulative BLEU scores of the rule-based translation component

5.2. Statistical Translation Component Performance

Statistical translation component translates the stid into TID as part of the proposed hybrid translation system. In order to evaluate the statistical machine translation technique individually, it is trained to translate Turkish into TID. In order to do that, the system is trained with 2852 Turkish and TID sentence pairs then it is tuned with 346 sentences.

Test corpus containing 363 sentences, is fed into the component and BLEU scores are calculated by comparing the translation results with the preprocessed original TİD translations. A part of the translation results is listed in table 5.2.

Turkish input sentence	Statistical TID translation
Evde montları asmak için bir askı	ben ev montları asmak ben bir askı
yoktu, şimdi yeni bir tane aldım.	yok şimdi yeni bir tane almak almak
Rahatlıkla montları aslıyorum.	montları aslıyorum
Doğum günümde annem bana altın küpe	ben anne doğum günümde altın küpe
hediye etti. Çok şaşırdım. Küpe	hediye etmek ben bakmak şaşırmak küpe
sevmiyorum ama annem için taktım.	sevmekdeğil anne ben takmak
Bugün günlerden pazar ve koşu	bugün günlerden pazar yarışma koşmak
yarışması olduğu için yollar saat	koşmak ben yollar saat dörtten sonra
dörtten sonra açılacak.	açılacak

Table 5.2. Comparision of the statistical translation results and the original TID translations

Statistical component's translation results and preprocessed original TID translations are compared below.

Statistical translation	Preprocessed original TID translation
ben ev montları asmak ben bir askı vok simdi yeni bir tane almak almak	ev iç mont askı yok yeni askı almak ben kovmak mont asmak asmak
montları aslıyorum	ben koymak mont asmak asmak
ben anne doğum günümde altın küpe hediye etmek ben bakmak şaşırmak küpe	ben doğum gün anne sürpriz altın küpe hediye etmek ben şaşırmak sevmekdeğil
sevmekdeğil anne ben takmak	ama anne için küpe takmak
bugün günlerden pazar yarışma koşmak	bugün pazar köprü koşmak var saat
koşmak ben yollar saat dörtten sonra	dört sonra araba yol açık
açılacak	

According to the translation results above, BLUE-1, BLEU-2, BLEU-3, and BLEU-4 performance scores are measured and illustrated below.



Cumulative BLEU scores of the statistical translation component

5.3. Hybrid Translation System Performance

Hybrid Translation System is executed to translate Turkish test corpus containing 363 sentences into TİD. Then BLEU scores are calculated by comparing the translation

results with the original TİD translations of the test corpus. A part of the translation result is listed below.

Hybrid translation
ben ev mont asmak asmak bir askı yok
şimdi yeni almak mont bir tane rahat
ben doğum gün ben küpe anne altın
hediye etmek ben hiç sevmekdeğil ben
mecbur küpe anne takmak
bugün pazar yarışma gün koşmak koşmak
ben araba yol dört açılmak saat
bitmek

Hybrid Translation System's results and processed original TID translations are compared in table 5.3.

Hybrid translation	Processed original TLD translation
ben ev mont asmak asmak bir askı yok	ev iç mont askı yok yeni askı almak
şimdi yeni almak mont bir tane rahat	ben koymak mont asmak asmak
ben doğum gün ben küpe anne altın	ben doğum gün anne sürpriz altın küpe
hediye etmek ben hiç sevmekdeğil ben	hediye etmek ben şaşırmak sevmekdeğil
mecbur küpe anne takmak	ama anne için küpe takmak
bugün pazar yarışma gün koşmak koşmak ben araba yol dört açılmak saat bitmek	bugün pazar köprü koşmak var saat dört sonra araba yol açık

Table 5.3. Comparision of the hybrid translation results and the original preprocessed TID translations

According to translation results above, BLUE-1, BLEU-2, BLEU-3, and BLEU-4 performance scores are measured and illustrated below.



Cumulative BLEU scores of the hybrid translation system

Statistical translation component, rule-based translation component and hybrid translation system performances are compared below in terms of the cumulative BLEU scores.



5.4. Hybrid Translation System and Literature Comparison

Hybrid Translation System is compared to several studies in the literature. These studies are described in section 2 and in order to facilitate the naming, they are called as systems; the study proposed by Hernandez et al. [13] is called as System-1, the study proposed by Manzano [14] is called as System-2 and the study proposed by Stoll et al. [16] is called as System-3. These systems are compared in terms of the BLEU scores as shown below. System-1 and System-2 only calculate the BLEU-4 scores for the evaluation. This is why BLEU-3, BLEU-2, and BLEU-1 scores are marked as 0.



System-1 achieves the best score among the others by 57.8%. This system employs 153 translation rules and limits its translation domain to utterances which are used in identity card office. It is obvious that applying rules to a specific domain will have high performance.

System-2 and System-3 are NMT based systems, therefore, their performance depends on the dataset size. Hybrid Translation System is also affected greatly by the dataset size. So the dataset sizes are compared in table 5.4. Although having the smallest dataset among these systems, Hybrid Translation System scores well.

	System-2	System-3	Hybrid Translation System
Train	83618	Unknown	2851
Develop	2045	Unknown	346
Test	2046	Unknown	363
Overall	87709	8257	3561
	Τa	able 5.4. Da	taset comparison of the systems

5.5. Effects of the Translation Rules on Hybrid Translation System

In order to determine the appropriate translation rules, different rule variations are tried throughout the study. In this section, effect of a rule on the system is analyzed.

Present Tense rule is removed from the rule-based translation component of the system and the new hybrid translation system is named as Model-1 for ease of use. Model-1 is evaluated from scratch with the same dataset.

Negation rule is removed from the rule-based translation component of the system and the new hybrid translation system is named as Model-2 for ease of use. Model-2 is also evaluated from scratch with the same dataset.

Cumulative BLEU scores of Model-1, Model-2 and Hybrid translation system are compared below.



According to the BLEU scores, Negation rule decreases the overall performance by %0.66 while Present Tense rule decreases by %0.38. The difference between the effects of the rules does not give an insight about the importance of the rule. Instead, it indicates that the occurrence frequency of the Negation rule in the test data set is more than the Present Tense rule. In the same manner, bigger test data will increase the performance impact of the rules.

6. CONCLUSION AND FUTURE WORK

This study introduces a hybrid translation system to convert Turkish text into Turkish Sign Language. Rule-based and statistical translation approaches are combined and achieved %12.64 BLEU-4 score.

The Turkish input sentence is first analyzed morphologically by The Boun Morphological Analyzer. According to the parser results, rule-based translator applies the predefined Turkish to TİD transformation rules. Each rule first interprets the Turkish input sentence in various aspects such as tense, person agreement, possessiveness, and conjunctions, then defines the appropriate TİD translation. Rule-based translation component comprises 13 rules. The output of the rule-based translation component is then fed into the statistical translation component in order to enhance the translation quality. Moses Decoder is used to implement statistical machine translation.

In order to train the statistical machine translation component, the bilingual corpus is generated from the online TİD dictionary. A website crawler is implemented to extract the sample sentences from the dictionary. 3561 sentence pairs are obtained as the dataset, then split into train, test, and development corpora.

Translation accuracy is evaluated by the cumulative BLUE scoring metric. The proposed hybrid translation system has achieved %12.64 BLEU-4, %19.28 BLEU-3, %31.48 BLEU-2 and %53.17 BLEU-1 scores. Rule-based and statistical translation components of the system are also evaluated individually. Evaluation results demonstrate that the combination of the rule-based and statistical machine translation techniques increases the overall system performance.

In this study, the input sentence is only interpreted morphologically. In order to increase translation accuracy, it should be analyzed semantically as well, by introducing new rules. In addition to this, dataset should also be extended to increase the system performance. Lastly, translation output should be fed into a virtual avatar tool to realize the gestures of the sign language.

REFERENCES

- Okan Kubus, A. H., "The phonetics and phonology of TID (Turkish Sign Language) bimanual alphabet", Formational units in sign languages, 2017.
- 2. Smith, R., HamNoSys 4.0 User Guide edited by Robert Smith, Ireland, 2013.
- Elliott, R., J. Glauert, J. Kennaway and I. Marshall, "The development of language processing support for the ViSiCAST project", pp. 101–108, 11 2000.
- Kaur, K. and P. Kumar, "HamNoSys to SiGML Conversion System for Sign Language Automation", *Proceedia Computer Science*, Vol. 89, pp. 794–803, 12 2016.
- Koehn, P., MOSES Statistical Machine Translation System: User Manual and Code Guide, 2018.
- Zeshan, U., "Aspects of Türk İşaret Dili (Turkish Sign Language)", Sign Language & Linguistics, Vol. 6, 01 2003.
- Baker, A., B. van den Bogaerde, R. Pfau and T. Schermer, *The Linguistics of Sign Languages: An introduction*, John Benjamins Publishing Company, 2016, https://books.google.com.tr/books?id=IECEDAAAQBAJ.
- Stokoe, W. C., Jr., "Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf", *The Journal of Deaf Studies and Deaf Education*, Vol. 10, No. 1, pp. 3–37, 2005, http://dx.doi.org/10.1093/deafed/eni001.
- Hanke, T., "HamNoSys—Representing sign language data in language resources and language processing contexts", *LREC 2004, Workshop Proceedings: Representation and Processing of Sign Languages. Paris: ELRA*, pp. 1–6, 2004.

- Ehrhardt, U., B. Davies, N. Thomas, M. Sheard, J. Glauert, R. Elliott, J. Tryggvason, T. Hanke, C. Schmaling, M. Wells and I. Zwitserlood, *The eSIGN Approach*, 2004, http://www.visicast.cmp.uea.ac.uk/Papers/eSIGNApproach.pdf.
- Bangham, J., S. Cox, R. Elliott, J. Glauert, I. Marshall, S. Rankov and M. Wells, Virtual Signing: Capture, Animation, Storage and Transmission – an Overview of the (2000), 2000.
- Zhao, L., K. Kipper, W. Schuler, C. Vogler, N. Badler and M. Palmer, "A Machine Translation System from English to American Sign Language", pp. 191–193, 10 2000.
- Hernandez, R., R. Barra-Chicote, R. Cordoba, L. D'Haro, F. Fernández-Martínez, J. Ferreiros, J. Lucas, J. Macias-Guarasa, J. Montero and J. Pardo, "Speech to sign language translation system for Spanish", *Speech Communication*, Vol. 50, pp. 1009–1020, 2008.
- 14. Manzano, D., "English to Asl Translator for Speech2signs", , 2018.
- Othman, A. and M. Jemni, "English-ASL Gloss Parallel Corpus 2012: ASLG-PC12", , 2012.
- Stoll, S., N. C. Camgöz, S. Hadfield and R. Bowden, "Sign Language Production using Neural Machine Translation and Generative Adversarial Networks", *BMVC*, 2018.
- Makaroğlu, B. and H. Dikyuva, Güncel Türk İşaret Dili Sözlüğü [The Contemporary Turkish Sign Language Dictionary], 01 2017.

APPENDIX A: APPLICATION