DEVELOPING A STATISTICAL TURKISH SIGN LANGUAGE TRANSLATION SYSTEM FOR PRIMARY SCHOOL STUDENTS

by

Buse Buz

B.S., Computer Engineering, Boğaziçi University, 2016

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

DEVELOPING A STATISTICAL TURKISH SIGN LANGUAGE TRANSLATION SYSTEM FOR PRIMARY SCHOOL STUDENTS

APPROVED BY:

Prof. Tunga Güngör (Thesis Supervisor)	. . . .
Assoc. Prof. Arzucan Özgür	. . . .
Tuğba Yıldız, Ph.D.	. . . .

DATE OF APPROVAL: 07.01.2019

ACKNOWLEDGEMENTS

I would first like to thank my thesis advisor Prof. Tunga Güngör. His office door was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this study to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank the experts who were involved in this study by helping Turkish Sign Language structure and translations: Prof. Sumru Özsoy and PhD student Aslı Özkul. Without their passionate participation and input, this study could not have been successfully conducted.

I would also like to acknowledge Assoc. Prof. Arzucan Özgür and Dr. Suzan Üsküdarlı, I am gratefully indebted to them for their very valuable support on everything on my life and comments on this thesis.

I am also thankful for the help and support I have received from my friends who I spent my undergraduate and ongoing life together, Basri Yılmaztürk, Oğuzhan Karaçakır and Utku Sarıdede. Without them, I couldn't love this department and even I couldn't be a computer engineer.

Finally, I must express my very profound gratitude to my beloved family and especially to my fiance, Birkan, who has sleepless nights with me for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Most importantly, I know that my beloved father knows what I have accomplished and he is proud of me. Words fail to express my gratitude to you.

ABSTRACT

DEVELOPING A STATISTICAL TURKISH SIGN LANGUAGE TRANSLATION SYSTEM FOR PRIMARY SCHOOL STUDENTS

Nowadays, as the access to information in the field of education increases, new technologies are developing for primary school children. However, deaf and dumb children still have limited access to the information especially in their school lives. One of the most important reasons for this problem is the lack of studies in the Turkish Sign Language domain. In this study, for the first time, translation from Turkish to Turkish Sign Language has been performed with statistical machine translation approach. The data required for translation were taken from the textbooks of primary school children and data processing was performed by using various algorithms. The system has been used with Moses Decoder and the results have been tested with different evaluation metrics. Because there is no other SMT based study for Turkish Sign Language in the literature, the results obtained from this study can not be compared. Nevertheless, it is seen that the scores obtained from results are motivating for new studies.

ÖZET

İLKÖĞRETİM ÖĞRENCİLERİ İÇİN İSTATİSTİKSEL TÜRK İŞARET DİLİ ÇEVİRİ SİSTEMİ GELİŞTİRME

Günümüzde, eğitim alanındaki bilgiye erişim arttıkça, ilkokul çocukları için yeni teknolojiler gelişmektedir. Ancak sağır ve dilsiz çocukların özellikle okul hayatlarındaki bilgiye erişimi sınırlıdır. Bunun en önemli sebeplerinden biri Türk İşaret Dili(TİD) alanındaki çalışmaların eksikliğidir. Bu çalışmada ilk kez istatistiksel makina çevirisi yaklaşımıyla Türkçe'den Türkçe İşaret Diline çeviri yapılmıştır. Tercüme için gerekli veriler ilkokul çocuklarının ders kitaplarından alınmış ve veri işleme çeşitli algoritmalar kullanılarak yapılmıştır. Sistem, Moses Decoder ile kullanılmış ve sonuçlar farklı değerlendirme ölçütleriyle test edilmiştir. Literatürde başka bir SMT tabanlı çalışma bulunmadığından, bu çalışmadan elde edilen sonuçlar karşılaştırılamamaktadır. Buna rağmen, sonuçlardan elde edilen skorların yeni çalışmalar için motive edici olduğu görülmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

BLEU-n	BLEU score of n-grams
lex	Lexical weighting
w	Word translation table
ϕ	Phrase Translation Probability

LIST OF ACRONYMS/ABBREVIATIONS

BLEU	Bilingual Evaluation Understudy
DEPREL	Dependency Relation
DSG	German Sign Language
EN	English Language
FEATS	Features
GER	German Language
ITU	Istanbul Technical University
LM	Language Model
MERT	Minimum Error Rate Training
NLP	Natural Language Processing
PER	Position Independent Word Error Rate
RO	Romanian Language
SMT	Statistical Machine Translation
TER	Translation Error Rate
TİD	Turkish Sign Language
UPOS	Universal Part of Speech Tags
XPOS	Language Specific Part of Speech Tags
WER	Word Error Rate

1. INTRODUCTION

With the development of technology, accessibility became a more important issue than before. This issue has a vital role for impaired people, especially when we think of children. For cognitive development of deaf and dumb children, primary school education has a crucial impact.

Studies in recent years show that, deaf children have encounter lots of problems due to their disabilities [1]. Most of them learn how to read and write in a few years while their peer groups learn it within a few months. While their peer groups can evolve their language and communication skills, deaf children can not do that because of lack of language skills and problems in their social lives. However, if they can express their thoughts and feelings with a language, with that way they can learn a way for communication. They can also learn the written and spoken language just like their peer groups with help of the communication technique they have learned. Sign languages are actually the communication instrument of the deaf children. A sign language is a visual language which is build by the positioning and movements of upper body as well as the facial expressions. So, if children know sign language, mostly they can learn written and spoken languages with the help of the sign language.

Our aim is to create an automatic translation system for Turkish Sign Language (TİD) using Natural Language Processing (NLP) methods. These methods allow us to create a system which converts the text in human language to any other language. For primary-school children, the materials are mostly children stories and introduction to reading and writing books. There are already lots of studies in language processing, but not many for Turkish Sign Language because of the complex structure of Turkish language. Thus, in this study we want to help deaf and dumb primary school children to learn a written language with using Sign Language.

One of the biggest problems in creating such a translation system is that the number of previous studies is low for TİD. Furthermore, the number of competent individuals who know TİD is quite a little. Because of these reasons, it was very challenging for us to create a dataset which is one of the most important things required for a statistical translation. Here, we would like to mention that the translation of the sign language was done by us who don't know the sign language, but accompanied by supervisors who are either people who know TİD or researchers who study on TİD. So, the data here is not necessarily one to one translations for TİD users. Another problem is that Turkish is an agglutinative language that has a lot of derivational suffixes and inflectional suffixes, while such attachments are not suitable for TİD. Thus, one can say that the two languages are too far away from each other in the morphological form. Therefore, for statistical translation, we also examined TİD and made some preprocessing according to the findings. Examples about the complex structure of TİD - Turkish duo can be seen in Table 1.1.

In this thesis, we propose a translation method which uses both morphological properties of TİD and statistical translation techniques. Also we have created a parallel corpus for other researchers to use. This corpus can be extended and corrected by TİD signers and researchers. If the language can be studied in more detail, better systems can be created to translate the sign language with better outputs. In spite of all these deficiencies, it was a starting point since there is no such study with TİD and, considering other languages, our system is successful enough according to the state-of-art studies.

Table 1.1. Turkish - TİD sentence pairs

Turkish Sentence	**TİD Sentence**
Anne ve babası, heyecanlı olmasının doğal olduğunu söylediler. (His/her parents said it was natural to be excited.)	ANNE VE BABA (parents) SÖYLEMEK (to say) O HEYECANLI OLMAK (s/he is excited) BU NORMAL (this is normal)
Annem izin almak için okulun hangi bölümüne gitmelidir? (What part of the school should my mother go to get permission?)	BEN (I) ANNE (mother) İZİN ALMAK İÇİN (to get permission) OKUL (school) HANGİ BÖLÜM (which part) GİTMEK (to go)?
Tanımadığımız kişilerle ilişkilerimizde dikkatli olmalıyız. (We must be careful in our relations with people we don't know.)	BİZ (we) KİŞİ (someone) TANIMAKˆ DEĞİL (not to know) BİZ (we) İLİŞKİ (relation) DİKKATLİ OLMAK (to be careful) LAZIM (necessary).
Sonbaharda ağaçlar yapraklarını döker. (In the autumn the trees drop their leaves.)	SONBAHAR (autmn) AĞAÇ (tree) YAPRAK (leaf) DÖKMEK (to drop)

2. RELATED WORKS

A child's cognitive development depends on the communication and language skills. In the [1] *Yorganci et. al* already mentioned the communication problems for deaf and dumb children. While their peer groups can learn their natural language in the first year of primary school, for deaf children it is not possible to learn it even until third year. To overcome this problem, the researchers created an avatar named Merry which helps deaf children to translate text to Avatar-based Interface. They set up an experiment with a test from social studies book that was designed for primary school children. Children can read these questions by themselves, or understand the questions while watching Merry. The results show that, for deaf children, Sign Language interface has an important role. The results are shown in Table 2.1.

Table 2.1. Results with Text and Merry [1]

Accuracy	**Correct Answers**	**Wrong Answers**
Text only	45.33%	32.50%
Text and Merry	66.11%	27.08%

Sign languages and spoken languages are different from each other in terms of lexical, morphological and syntactic levels. In [2], researchers have developed a system which creates a machine-readable sign language notation and use an avatar to represent it. They use rule-based model approach for the translation problem because to develop a statistical model one needs to have a large amount of dataset. However, like some other languages, TİD lacks electronic resources which creates a difficulty for the researchers. TİD also lacks the definition of signs and there is a variation in the use of lexical items which causes confusion as the same word can be represented by different ways for different TİD resources. Because of these problems, they have created their own TİD dictionary. They developed a test corpus which consists of both NLP-tagging Turkish sentences and written TİD tagging sentences. This corpus is crucial for the researchers who study on TİD.

In [3], researchers proposed a translation system from sign language to spoken language. If we focus on translation part, the researchers used a statistical approach instead of conventional rule-based approach. In their study, it's clear that statistical approach is comparable to traditional approach. In general, two problems have been mentioned. (i) lack of large corpora and (ii) lack of notion standard. About the first problem, it can be seen that most corpus for translation contain about 1 million sentences, while there are no more than 2000 sentences in the corpus for sign languages. As for the second problem, each sign language has its own rules. Thus, every signer can show a sentence with a different way. When we take into account that the number of people who know the TİD is a quite little, we have also encountered these problems while doing this research.

In the same study, to perform experiments, training and testing data and an objective error measurement is needed. In total, 1399 sentences have been used. The corpus divided into training samples (83% of the sentences) and testing samples (17% of the sentences) [3]. The training is performed by using both IBM Model 1-4 (*Brown et al. 1993*) and Hidden Markov Model (*Ney and Och, 2000*). For evaluation metrics, mWER (word error rate) and mPER (position-independent word error rate) have been used [4]. If we consider the results, we can say that the results are promising on behalf of the statistical translation.

Table 2.2. Results for German to German Sign Language (DGS) [3]

	mWER(%)	**mPER(%)**
Single words	85.4	43.9
Alignment Templates	59.9	23.6

Moses is a statistical translation tool which uses phrase-based translation approach.In phrase-based translation, adjacent segments of words in the input sentence are mapped to adjacent segments of words in the output sentence [5]. For source language sentence s and target language sentence t, Moses tries to find:

$$\hat{t} = argmax_t P(t|s)$$

Where t̂ is the translation of s with highest probability and $P(t|s)$ is the probability model. In order to create the model Moses uses SRILM. SRILM implements an efficient representation of the phrase translation table [6]. It uses binary format so it works faster compared to other similar tools. Moses also uses GIZA++ for word-based alignment [7].

In [8], researchers work on a translation system from English to Indian Sign Language which uses Moses as decoder. Railways' announcements and reservation information are used as a domain. One of the problems they have faced is again lack of large corpus. 326 sentences were studied. Apart from this problem, the challenge of ambiguity is also mentioned in the study. For example book can be used for 2 words: book a ticket(verb), read a book(noun). This means, probabilities for source sentences may be spared. In other words, the problem is the change in the probabilities of word alignments. Despite all of these problems, the success of the system is quite good. The results from this study can also be compared with rule-based systems.

In [9], Moses has been used as Statistical Machine Translation decoder again. For Word-alignment, GIZA is used. In addition to GIZA, Jaro-Winkler distance is also used for word alignment because the same words are used in the both natural language and its sign language.

The most common opinion about corpus size on SMT is "the more the better". However, [10] shows that rule-based and statistical approaches can be compared in the sign language domain. Previously, for the statistical approach, we mentioned that small corpus is the main problem. However, in this study, different size of corpus were used. JRC-Acquis-L is a large corpus and JRC-Acquis-S is a small corpus drawn from the the same data. "3-grams work generally the best" (*Rousu, 2008*) motto has been used for evaluation. 4 languages were used for translation which are from English (EN) to Romanian (RO), Romanian to English, German (GER) to Romanian and Romanian to English. If we compare BLEU [11] and TER [12] scores for different language pairs, we can see that a large data set does not make one of the scores superior to the other. Thanks to this study, we provided the necessary motivation for our work.

Table 2.3. BLEU vs TER Scores

Score	**JRC-Acquis-S**	**JRC-Acquis-L**
BLEU (EN to RO)	0.4801	0.4015
TER (EN to RO)	0.5032	0.5023
BLEU (RO to EN)	0.4904	0.4255
TER (RO to EN)	0.4509	0.4457
BLEU (GER to RO)	0.2811	0.3644
TER (GER to RO)	0.6658	0.6113
BLEU (RO to GER)	0.2926	0.3726
TER (RO to GER)	0.6816	0.6112

3. METHODOLOGY

3.1. Architecture

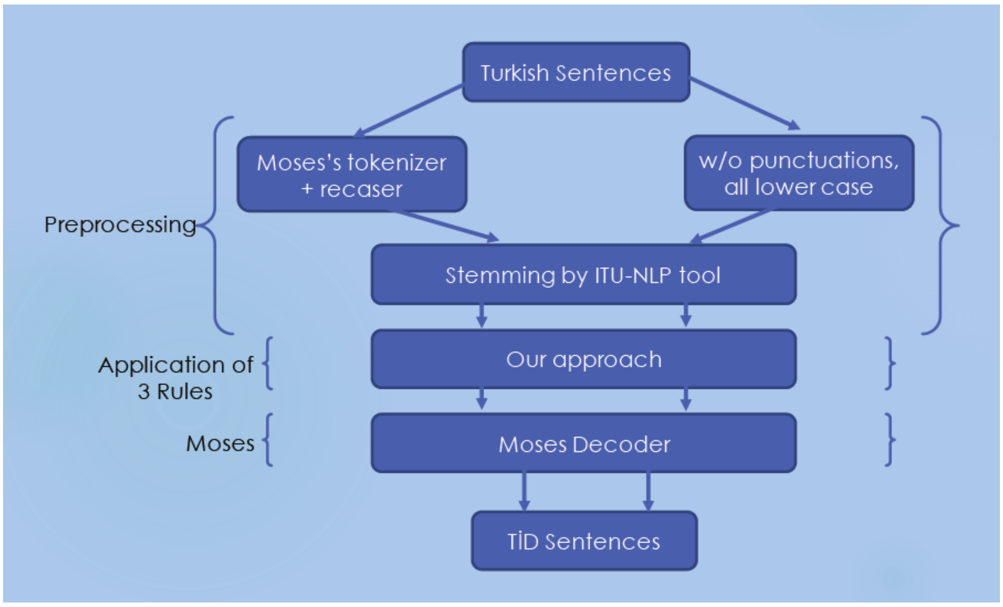


Figure 3.1. Methodology Architecture

The system consists of 3 steps. It starts with preprocessing part which includes tokenization, recasing and stemming. Then the proposed rules apply. As a final step, the parallel corpora is given to Moses tool.

3.2. Preprocessing

Before training and testing our system, some processes have been done to our corpus. Tokenization means splitting up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. As a first step in our study, tokenization has been applied by Moses' tokenizer. After tokenization, Moses's recaser has been used. The recaser checks the first tokens of sentences to be sure whether they are starting with capital letter or not. Then the initial words in each sentence are converted to their most probable casing. In this way, data sparsity has been reduced.

After preparing the data for training the translation system, stemming is applied. Stemming is the act of reducing inflected or derived tokens to their roots. The aim of stemming in our study is to reduce inflectional forms of a word to a common root. Different forms of a word can be seen in Table 3.1.

Table 3.1. Different forms of a word "Okul" (*School*)

okulun (of school)	okul (school)
okula (to school)	okul (school)
okuldan (from school)	okul (school)

The most important reason for using preprocessing for this study is the fact that TİD does not use inflectional suffixes. ITU-NLP tool [13] is used for stemming in this study to perform such preprocessing operations which are shown in Figure 3.2.

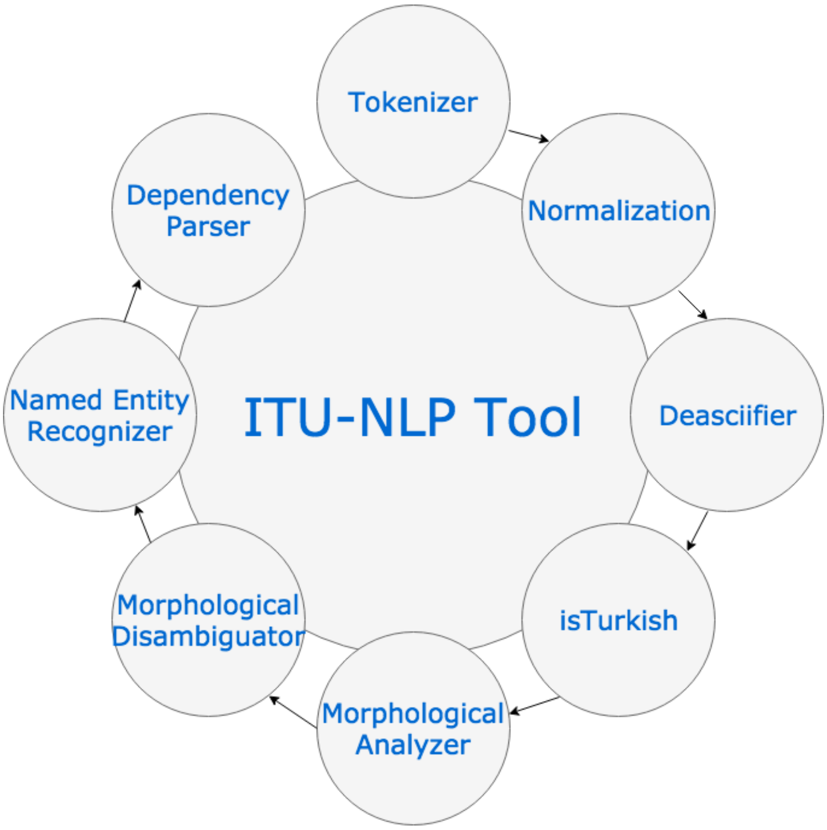
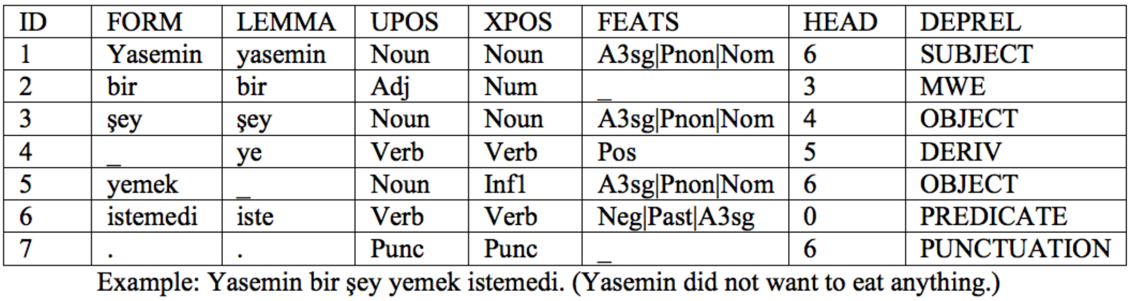


Figure 3.2. Operations by ITU-NLP tool.



ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL
1	Yasemin	yasemin	Noun	Noun	A3sg|Pnon|Nom	6	SUBJECT
2	bir	bir	Adj	Num	_	3	MWE
3	şey	şey	Noun	Noun	A3sg|Pnon|Nom	4	OBJECT
4	_	ye	Verb	Verb	Pos	5	DERIV
5	yemek	_	Noun	Inf1	A3sg|Pnon|Nom	6	OBJECT
6	istemedi	iste	Verb	Verb	Neg|Past|A3sg	0	PREDICATE
7	.	.	Punc	Punc	_	6	PUNCTUATION

Example: Yasemin bir şey yemek istemedi. (Yasemin did not want to eat anything.)

Figure 3.3. Example output from ITU-NLP tool.

Example output from NLP-tool can be seen in Figure 3.3. All tokens have Universal Part-of-speech tags which is important for further approaches.

Table 3.2. Universal Part-of-speech tags

+Noun	Noun
+Adj	Adjective
+Adv	Adverb
+Cond	Condition
+Verb	Verb
+Postp	Postpositive
+Pron	Pronoun
+Punc	Punctuation

The list of morphological features from the universal feature inventory or from a defined language-specific extension can be seen below:

- Nominal forms get the following inflectional markers:
  Number/Person Agreement (Table 3.3) + *Possessive Agreement* (Table 3.4) + *Case* (Table 3.5)
- Verbs also get number/person agreement and the following markers:
  Polarity (Table 3.6) + *Tense/Aspect* (Table 3.7) + Number/Person Agreement

Table 3.3. Number/Person Agreement

+A1sg	1. singular
+A2sg	2. singular
+A3sg	3. singular
+A1pl	1. plural
+A2pl	2. plural
+A3pl	3. plural

Table 3.4. Possessive Agreement

+P1sg	1. singular
+P2sg	2. singular
+P3sg	3. singular
+P1pl	1. plural
+P2pl	2. plural
+P3pl	3. plural
+Pnon	Pronoun (no overt agreement)

Table 3.5. Case

+Nom	Nominative
+Acc	Accusative/Objective
+Dat	Dative (to ...)
+Abl	Ablative (from ...)
+Loc	Locative (on/at/in ...)
+Gen	Genitive (of ...)
+Ins	Instrumental (with ...)
+Equ	Equative

Table 3.6. Polarity

+Pos	Positive
+Neg	Negative

Table 3.7. Tense - Aspect

+Past	Past Tense
+Narr	Narrative Past Tense
+Fut	Future Tense
+Aor	Aorist
+Pres	Present Tense
+Desr	Desire/Wish
+Cond	Conditional
+Neces	Necessitative
+Opt	Optative
+Imp	Imperative
+Prog1	Present cont., process
+Prog2	Present cont., state

Table 3.8. Before-After Stemming

Turkish Sentence	Yasemin okula başlıyor. *(Yasemin is starting to school)*
After Stemming	Yasemin *(Yasemin)* okul *(school)* başla *(to start)* .
TİD Sentence	YASEMİN *(Yasemin)* OKUL *(school)* BAŞLAMAK *(to start)*

After stemming is applied on Turkish sentences, such pair can be seen in Table 3.8. As can be seen in the example, when stemming is applied to these sentences, the structure becomes more appropriate for translation. In Chapter 5, it can be seen how important the stemming is when evaluating the translation.

3.3. Our Approach

In addition to preprocessing, the structures of Turkish and TİD are examined and according to the information gained, a few more operations has been added. The reason for using additional operations is that the inflectional suffixes are not used in TİD as it was mentioned before and this was a problem while making the translation. After using these operations, parallel data is given to Moses and scores are compared.

3.3.1. Adding negation

In Turkish, if the verb is negative, that suffix is added to the verb.

gelmedi ⇒ gel + Verb + Neg | Past | A3sg
S/he didn't come ⇒ to come + Verb + . . .

In TİD, there is no such suffix, instead DEĞİL tag is used after the verb.

gelmedi ⇒ O GELMEK ˆ DEĞİL
S/he didn't come ⇒ S/he + to come + not

To increase accuracy in the translation system, it is necessary to take these suffixes into account. For this case, the verbs are checked for each word in the dataset and if the verb is negative +Neg tag is added to the verb. Thus, when making the alignment in the statistical translation section, the words geldi*(s/he came)* and gelmedi*(s/he didn't come)* are divided into stems and first one is labeled as gel*(to come)*, while the second one is labeled as gelNeg*(to come+Neg)*. The system can learn the difference between these verbs.

Table 3.9. Example of First Approach

Turkish Sentence	**After First Approach**
Yasemin bir şey yemek istemedi. *(Yasemin did not want to eat anything.)*	Yasemin *(Yasemin)* bir şey *(anything)* ye *(to eat)* isteNeg *(to want+Neg)* .
Ayşegül, sürücüyü tanımıyordu. *(Ayşegül did not know the driver.)*	Ayşegül *(Ayşegül)* sürücü *(driver)* tanıNeg *(to know+Neg)* .

3.3.2. Adding pronoun to Noun

In Turkish, the possesive suffix is added to the noun.

kalemim ⇒ kalem + Noun + P1sg
my pencil ⇒ pencil + Noun + . . .

In TİD, again because there is no such suffix, pronoun is added to the noun.

kalemim ⇒ BEN KALEM
my pencil ⇒ I + pencil

To solve this problem before giving the parallel corpus to Moses, and to increase the alignment scoring, the suitable pronoun is added to the noun which has the possessive suffix. This step also reduces data sparsity.

The pronoun is used as a prefix to given noun and now the tokens are referring to the same noun. Thus, after second approach, the translation score is expected to increase.

Table 3.10. Example of Second Approach

Turkish Sentence	**After Second Approach**
Arkadaşlarımla tanışıyorum. *(I meet my friends.)*	Ben *(I)* arkadaş *(friend)* tanış *(to meet)* .
Arkadaşlarımıza ve arkadaşlarımızın eşyalarına zarar vermeyelim. *(Let's not hurt our friends and the belongings of our friends.)*	Biz *(we)* arkadaş *(friend)* ve *(and)* biz *(we)* arkadaş *(friend)* o eşya *(the belonging)* zarar verNeg *(to hurt+Neg)* .

3.3.3. Adding pronoun to Verb

In Turkish, personal suffixes added to the verb.

$$\text{okudum} \Rightarrow \text{oku} + \text{Verb} + \text{Past}|\text{A1sg}$$
$$\textit{I read} \Rightarrow \textit{to read} + \textit{Verb} + \ldots$$

In TİD, according to the verb of the sentence, the pronoun which indicates who made the action, is added to the sentence.

$$\text{okudum} \Rightarrow \text{BEN OKUMAK}$$
$$\textit{I read} \Rightarrow \textit{I} + \textit{to read}$$

In this step, the suffixes for each verb are examined and the pronoun is added to the verb to inform who was performing the action. So okudum*(I read)* and okudu*(s/he read)* are indicated the same as the root, but the translation is indicated as different words. Sample sentences can be seen in Table 3.11.

Table 3.11. Example of Third Approach

Turkish Sentence	**After Third Approach**
Arkadaşlarımla oynarken işitme cihazıma zarar vermemek için dikkatli oluyorum. *(I'm careful not to damage my hearing aid while playing with my friends.)*	ben *(I)* arkadaş *(friend)* oyna *(to play)* işit *(hearing)* ben *(I)* cihaz *(device)* zarar verNeg için *(to damage+Neg)* dikkatli *(careful)* ben *(I)* ol *(to be)* .
Şimdi sizleri tanımak istiyorum. *(Now, I want to know you.)*	Şimdi *(now)* siz *(you)* tanı *(to know)* ben *(I)* iste *(to want)* .

3.4. Moses

Moses is an implementation of the statistical approach to machine translation. In statistical machine translation (SMT), translation systems are trained on large sizes of parallel data. Parallel data is a collection of sentences in two different languages, which is sentence-aligned, in that each sentence in one language is matched with its translated sentence in the other language [5].

The training process in Moses uses the parallel data and co-ocurrences of words and phrases to understand translation correspondences between the two languages. In phrase-based machine translation, these correspondences are simply between continuous sequences of words.

3.4.1. Training

3.4.1.1. Prepare Data and Run GIZA++. The parallel corpus has to be converted into a format that is suitable for the GIZA++ toolkit [7]. In order to prepare data, tokenization and truecasing steps are done by Moses scripts. Here *train_tr.txt* and *train_tid.txt* are the given parallel corpus. In the commands we use tid and tr options which mean the files are going to be tokenized for the given Turkish (tr) language

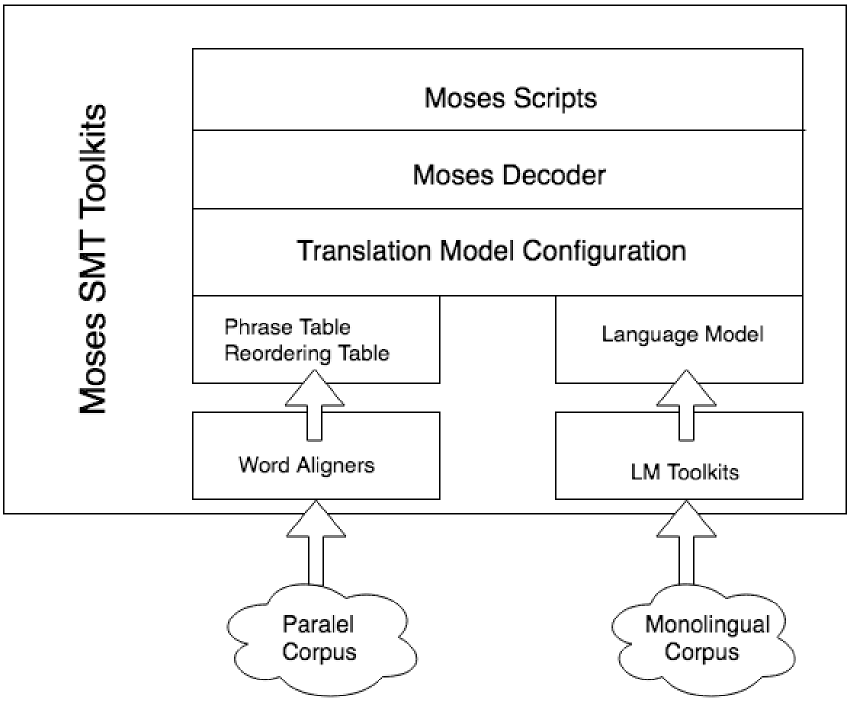


Figure 3.4. Moses Architecture from [14]

or TİD (tid). However, because Moses does not support tr and tid, it uses English tokenizer instead. After tokenization, truecaser is trained. Then with the truecase models for both tid and tr, the dataset truecased. We also applied the same steps to development and test sets. After these operations, we have .true files to use in the training system.

Tokenization:

```
~/moses/scripts/tokenizer/tokenizer.perl -l tid < ~/corpus/train_tid.txt >
~/corpus/train.tok.tid
~/moses/scripts/tokenizer/tokenizer.perl -l tr < ~/corpus /train_tr.txt >
~/corpus /train.tok.tr
```

Training truecaser with the given data:

```
∼/moses/scripts/recaser/train-truecaser.perl −−model ∼/corpus /truecase-model.tid −−corpus ∼/corpus /train.tok.tid
∼/moses/scripts/recaser/train-truecaser.perl −−model ∼/corpus /truecase-model.tr −−corpus ∼/corpus/train.tok.tr
```

Truecasing the data:

```
∼/moses/scripts/recaser/truecase.perl −−model ∼/corpus /truecase-model.tid < ∼/corpus/train.tok.tid > ∼/corpus/train.true.tid
∼/moses/scripts/recaser/truecase.perl −−model ∼/corpus /truecase-model.tr < ∼/corpus /train.tok.tr > ∼/corpus /train.true.tr
```

To train the system, the given command is run for word alignment (using GIZA), phrase extraction and scoring, and creating lexicalized reordering tables. Also, here /bin folder contains the necessary GIZA++ files.

```
cd ∼/working
nohup nice ∼/moses/scripts/training/train-model.perl -root-dir train -corpus ∼/corpus/train.true -f tr -e tid -alignment grow-diag-final-and -reordering msd-bidirectional-fe -lm 0:3∼/lm/train.blm.tid:8 -external-bin-dir ∼/bin/ > & training.out &
```

Two vocabulary files are generated with the given commands and the parallel corpus is converted into a numberized format by GIZA++. The vocabulary files contain words, integer word IDs and number of occurrences of the word which can be seen in Table 3.12.

Table 3.12. Vocabulary Files

TR-ID	**tr.vcb**	#	**TID-ID**	**tid.vcb**	#
1	UNK	0	1	UNK	0
2	ve *(and)*	235	2	BİZ *(we)*	584
3	ol *(to be)*	114	3	VE *(and)*	222
4	ne *(what)*	112	4	BEN *(I)*	181
5	yap *(to do)*	104	5	O *(s/he)*	140
6	’	87	6	YAPMAK *(to do)*	96
7	söyle *(to say)*	87	7	SEN *(you)*	89
8	bir *(a)*	79	8	NE *(what)*	88
9	et *(make)*	75	9	SÖYLEMEK *(to say)*	87
10	bu *(this)*	72	10	OLMAK *(to be)*	85
.	.	.	.	.	.
324	Yasemin *(Yasemin)*	5	320	YASEMİN *(Yasemin)*	5

The sentence-aligned corpus contains only integers and looks like this:

1
324 756 1169 (Yasemin erkenden kalktı) *(Yasemin got up early)*
320 739 1187 (YASEMİN ERKEN KALKMAK) *(Yasemin + early + to get up)*

A sentence pair now consists of three lines: First, the frequency of the sentence. In our training process frequency of a sentence is mostly 1 because almost all sentences appears only once. The other two lines below contain Word IDs of the Turkish and the TİD sentences.

GIZA++ also requires words to be placed into word classes (Table 3.13). This is done automatically by calling the mkcls program. Word classes are only used for the IBM reordering model in GIZA++.

Table 3.13. Examples of Word Classes

Token	**Class ID**
ABLA *(sister)*	22
ACELE *(rush)*	6
ACI *(pain)*	28
ACİL *(urgent)*	46
AD *(name)*	22

Our word alignments are taken from the intersection of bidirectional runs of GIZA++ plus some additional alignment points from the union of the two runs. Running GIZA++ is the most time consuming step in the training process. GIZA++ learns the translation tables of IBM Model 4, but we are only interested in the word alignment file for our study:

Sentence pair (1) source length 3 target length 3 alignment score : 0.242363
Yasemin *(Yasemin)* erken *(early)* kalk *(to get up)*
NULL () YASEMİN *(Yasemin)* (1) ERKEN *(early)* (2) KALKMAK *(to get up)* (3)

Sentence pair (2) source length 2 target length 2 alignment score : 0.390061
kahvaltı *(breakfast)* hazır *(ready)*
NULL () KAHVALTI *(breakfast)* (1) HAZIR *(ready)* (2)

In this file, after some statistical information and the Turkish sentence, the TİD sentence is listed word by word, with references to aligned foreign words: The first word YASEMİN (1) is aligned to the first Turkish word “Yasemin”. The second word ERKEN (2) is aligned to “erken” *(early)*. And so on.

3.4.1.2. Align Words. The alignment file contains alignment information, one alignment point at a time, in the form of the position of the Turkish and TİD words.

<u>3.4.1.3. Get Lexical Translation Table.</u> Given this alignment, it is quite straight forward to estimate a maximum likelihood lexical translation table. We estimate the w(TİD|Turkish) as well as the inverse w(Turkish|TİD) word translation table. Top translations for "arkadaş" *(friend)* into TİD with the probabilities can be seen in Table 3.14.

Table 3.14. Probabilities for translation

ARKADAŞLAR *(friends)*	Arkadaş *(friend)*	0.1911765
BEN *(I)*	Arkadaş *(friend)*	0.1176471
O *(s/he)*	Arkadaş *(friend)*	0.0882353
ARKADAŞ *(friend)*	Arkadaş *(friend)*	0.5147059
BİZ *(we)*	Arkadaş *(friend)*	0.0882353

<u>3.4.1.4. Extract Phrases.</u> In the phrase extraction step, all phrases are dumped into one big file. The content of this file is for each line: Turkish phrase, TİD phrase, and alignment points. Alignment points are pairs (Turkish, TİD). Here is the top of that file:

arkadaş *(friend)* ad *(name)* ||| ARKADAŞ *(friend)* AD *(name)* ||| mono mono

arkadaş *(friend)* ad *(name)* öğren *(to learn)* ||| ARKADAŞ *(friend)* AD *(name)* ÖĞRENMEK *(to learn)* ||| mono mono

<u>3.4.1.5. Score Phrases.</u> To estimate the phrase translation probability ϕ(TİD|Turkish) we proceed as follows: First, the extract file is sorted. This ensures that all TİD phrase translations for a Turkish phrase are next to each other in the file. Thus, we can process the file, one Turkish phrase at a time, collect counts and compute ϕ(TİD|Turkish) for that Turkish phrase. To estimate ϕ(Turkish|TİD), the inverted file is sorted, and then ϕ(Turkish|TİD) is estimated for a TİD phrase at a time.

arkadaş *(friend)* ad *(name)* ||| ARKADAŞ *(friend)* AD *(name)* ||| 1 0.65 1 0.46146

arkadaş *(friend)* ad *(name)* öğren *(to learn)* ||| ARKADAŞ *(friend)* AD *(name)* ÖĞRENMEK *(to learn)* ||| 1 0.65 1 0.184584

Currently, four different phrase translation scores are computed:

- inverse phrase translation probability ϕ(Turkish|TİD)
- inverse lexical weighting lex(Turkish|TİD)
- direct phrase translation probability ϕ(TİD|Turkish)
- direct lexical weighting lex(TİD|Turkish)

3.4.1.6. Build Lexicalized Reordering Model. By default, only a distance-based reordering model is included in final configuration. This model gives a cost linear to the reordering distance.

3.4.1.7. Build Generation Models. The generation model is build from the target side of the parallel corpus. By default, forward and backward probabilities are computed.

3.4.1.8. Create Configuration File. As a final step, a configuration file for the decoder is generated with all the correct paths for the generated model and a number of default parameter settings.

3.4.2. Building a Language Model

The language model (LM) is used to ensure fluent output, so it is built with the target language (i.e TİD in this case). The following command builds an appropriate 3-gram language model. Then it converts the language model file into binary form. In order to store these files, language modeling folder has been created.

```
~/moses/bin/lmplz -o 3 < ~/corpus /train.true.tid > ~/lm/train.arpa.tid
~/moses/bin/build_binary ~/lm10/train.arpa.tid ~/lm/train.blm.tid
```

KenLM is a language model which is distributed with Moses and compiled by default. With KenLM, ARPA file is created. ARPA file contains probabilities of the texts. However these information can be stored in binary format which is more efficient in terms of storage. Here you can see the 3-grams that the ARPA model includes:

```
    1-grams:
-3.7576616 <unk> 0
0 <s> -0.53296804
-1.0332772 < /s> 0
-3.6057224 1 -0.10019073
-3.095953 ÜNİTE -0.17581995
-3.6057224 OKULUMUZDA -0.10019073
-3.0082843 HAYAT -0.3387933
.
.

    2-grams:
-0.98600936 SEVİNMEK YASEMİN -0.058950756
-1.3822894 YASEMİN ERKEN -0.058950756
-1.2861999 SABAH ERKEN -0.058950756
-0.98629045 ERKEN KALKMAK -0.058950756
.
.

    3-grams:
-0.51113063 DAL KIRMAK BİTKİ
-0.61218756 HAYVAN İLE ELDE
-0.8561048 BİTKİ İLE ELDE
-0.98953664 İLE ELDE DİLMEK
```

The actual probabilities are replaced by their logs. So the negative numbers are seen, not the numbers between 0 and 1.

3.4.3. Tuning

In the decoding layer, Moses scores translation hypotheses using a linear model. In the traditional approach, the features of the model are the probabilities from the language models, phrase/rule tables, and reordering models, plus word, phrase and rule counts. Tuning process tries to find the optimal weights for the linear model, where optimal weights are those which maximize translation on a small set of parallel sentences (the development/tuning set). By default, tuning is optimizing the BLEU score of translating the specified tuning set using Minimum Error Rate (MERT). MERT was introduced by Och in [4]. This line-search based method is a tuning algorithm which is still used widely, and the default option in Moses and it measures the translation performance with BLEU [11].

Tuning requires a small amount of parallel data, separate from the training data. Here development set is used. Development set is already ready to use from the corpus preparation.

```
    cd ∼/working
    nohup nice ∼/moses/scripts/training/mert-moses.pl ∼/corpus/dev.true.tr
∼/corpus/dev.true.tid ∼/moses/bin/moses train/model/moses.ini −−mertdir
∼/moses/bin/ & > mert.out &
```

3.4.4. Testing

As a final step, the system takes test set as an input and tries to create a translated sentences given the language model, phrase table and the reordering table. After the translation, one can measure the BLEU score with again Moses's script. The experiments are detailed in Chapter 4.

Testing takes more time if the dataset is getting larger. In order to make it faster, the phrase table and lexicalized reordering models can be binarised. To do this, a directory can be created (here the directory named as binarised-model) and the models can be binarised as follows:

```
    ∼/moses/bin/processPhraseTableMin -in train/model/phrase-table.gz -nscores
4 -out binarised-model/phrase-table
    ∼/moses/bin/processLexicalTableMin -in train/model/reordering-table.wbe
-msd-bidirectional-fe.gz -out binarised-model /reordering-table
```

Then it is needed to make a copy of the ∼/working/mert-work/moses.ini in the binarised-model directory and change the phrase and reordering tables to point to the binarised versions, as follows:

- Change PhraseDictionaryMemory to PhraseDictionaryCompact
- Set the path of the PhraseDictionary feature to point to ∼/working/binarised-model/phrase-table.minphr
- Set the path of the LexicalReordering feature to point to ∼/working/binarised-model/reordering-table

After these steps, translation is going to be faster. Then the trained model can be filtered for this test set, means that only the entries are needed to translate the test set are retained. This will make the translation a lot faster.

```
    cd ∼/working
    ∼/moses/scripts/training/filter-model-given-input.pl filtered mert-work/
moses.ini ∼/corpus/test.true.tr -Binarizer ∼/moses/bin/processPhraseTableMin
```

Finally the translation can be done and the decoder can be tested while running the BLEU script.

```
    nohup nice ∼/moses/bin/moses -f filtered/moses.ini <
```

```
~/corpus/test.true.tr > test.translated.tid 2> test.out ~/moses/scripts/generic/multi-bleu.perl –lc ~/corpus/test.true.tid < test.translated.tid
```

4. EXPERIMENTS

4.1. Dataset

The dataset consists of Turkish-TID sentence pairs where Turkish sentences are collected from first grade students' book of Life Science of the Ministry of National Education of Turkey. The book consists of 6 units. Respectively;

(i) Okulumuzda Hayat *(Life in our School)*
(ii) Evimizde Hayat *(Life at Home)*
(iii) Sağlıklı Hayat *(Healthy Life)*
(iv) Güvenli Hayat *(Safe Life)*
(v) Ülkemizde Hayat *(Life in our Country)*
(vi) Doğada hayat *(Life in Nature)*

Each unit mentions related issues. In general, sentence structures are quite simple and sentences are quite short. Translation was done by uourselves who did not know sign language but worked on sign language structure. During the translation, information about how to do the translation is provided by Prof. Sumru Özsoy. Apart from her, PhD. candidate Aslı Özkul has consulted for some problems were encountered in translations. It should be noted again that the translation was not done by a native user and incorrectly translated sentences may be encountered. Some examples without any preprocessing of pairs can be seen in Table 4.1.

Considering all data, there are total of 1950 sentences and about 13 thousand tokens. The corpus has about 1450 unigram 5500 2-grams and 6650 3-grams. Also number of words in sentences are given in Figure 4.1 and Figure 4.2. 1500 of these sentences have been used for train and 250 for development and 200 for test. In the Results section, train and development sets of different sizes were also studied.

Table 4.1. Dataset

Turkish Sentence	**TİD Sentence**
Kaç arkadaşınızın adını öğrendiniz? *(How many names of your friends have you learned?)*	SEN *(you)* KAÇ ARKADAŞ *(how many friends)* AD *(name)* ÖĞRENMEK *(to learn)* ?
Boya kalemlerini evde unuttuğunu fark etti *(S/He realized s/he left her/his crayons at home.)*	BOYA KALEM *(crayon)* EVDE *(at home)* UNUTMAK *(to forget)* O *(s/he)* FARK ETMEK *(to realize)*
Yemek yerken nelere dikkat ediyorsunuz? *(What do you pay attention to when eating?)*	YEMEK YEMEK *(to eat)* SEN *(you)* NE *(what)* DİKKAT ETMEK *(to be careful)* ?

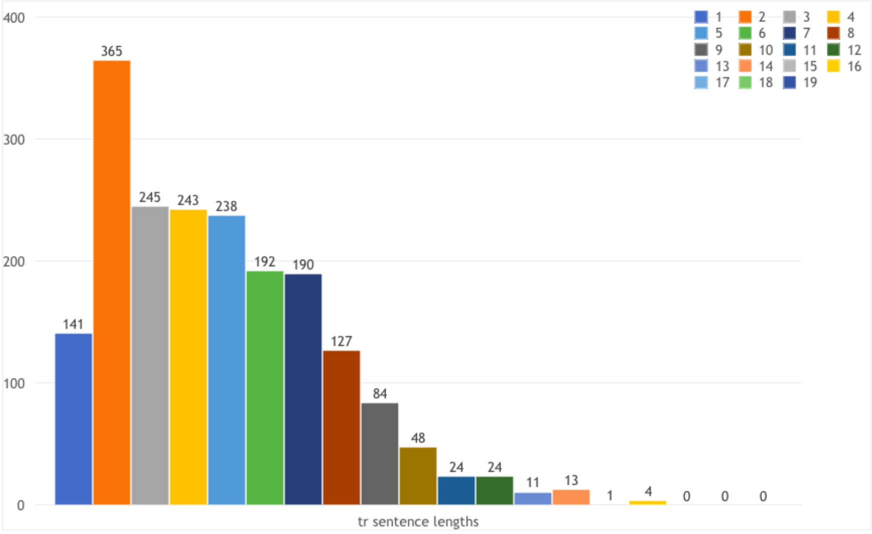


Figure 4.1. Lengths of Turkish Sentences

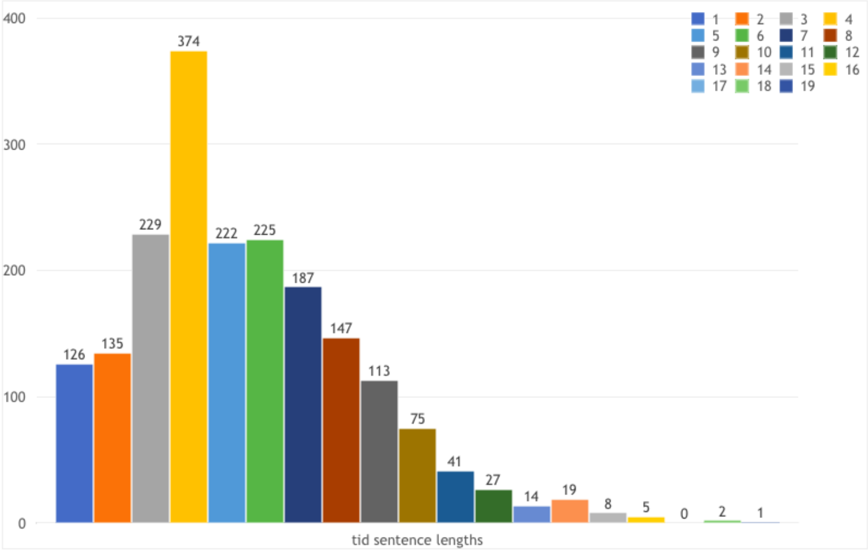


Figure 4.2. Lengths of TİD Sentences

4.2. Evaluation Metrics

The main evaluation metrics we have used in this study are BLEU [11] and WER (word error rate) [4]. After finding the translated sentences, each metric has been calculated with reference sentences. Also the metrics are used for different proportions of training and development sets.

BLEU calculates n-gram overlap between machine translation output and reference translation (Equation 4.1). In other words, it is basically the averaged percentage of n-gram matches. For each i-gram where i = 1,2, . . . , N , it computes the percentage of the i-gram tuples in the hypothesis that also occurs in the references (this is also called the precision).

$$\text{BLEU} = min(1, \frac{\text{output-length}}{\text{reference-length}})(\prod_{i=1}^{4} precision_i)^{1/4} \tag{4.1}$$

WER is the minimum number of editing steps to transform output sentence to reference sentence (Equation 4.2). There are 4 possible editing steps:

match: words match, no cost

insertion: add word

deletion: drop word

substitution: replace one word with another

$$\text{Word Error Rate} = \frac{\text{insertion} + \text{deletion} + \text{substitution}}{\text{number of words in reference}} \tag{4.2}$$

5. RESULTS AND ANALYSIS

As it was mentioned before, the sentences were short. Therefore, unigram alignments were thought to be appropriate in the first place, but for the following approaches BLEU-2 and BLEU-3 became more important for the study. For each approach, the data set was randomly divided into train, development and test sets. Also, it can be seen that the scores from Moses Preprocessing are relatively low compared to other scores. Because recaser of Moses has been trained with a small size of train set, it can not decide the first token of a sentence whether it should start with a capital letter or not. Also, because punctuation was not removed while using Moses scripts, there are more tokens in corpus which means probabilities of tokens are changed.

No operations were performed on the data as the first approach. Pairs in the parallel corpus first were given to the Moses' tokenizer and recaser. With the results from it, the system is trained (Process-1). As another step, tokenization has been done manually and also punctuation is removed. Then the data has been given to the Moses again (Process-2). The results are reported in the Figure 5.1.

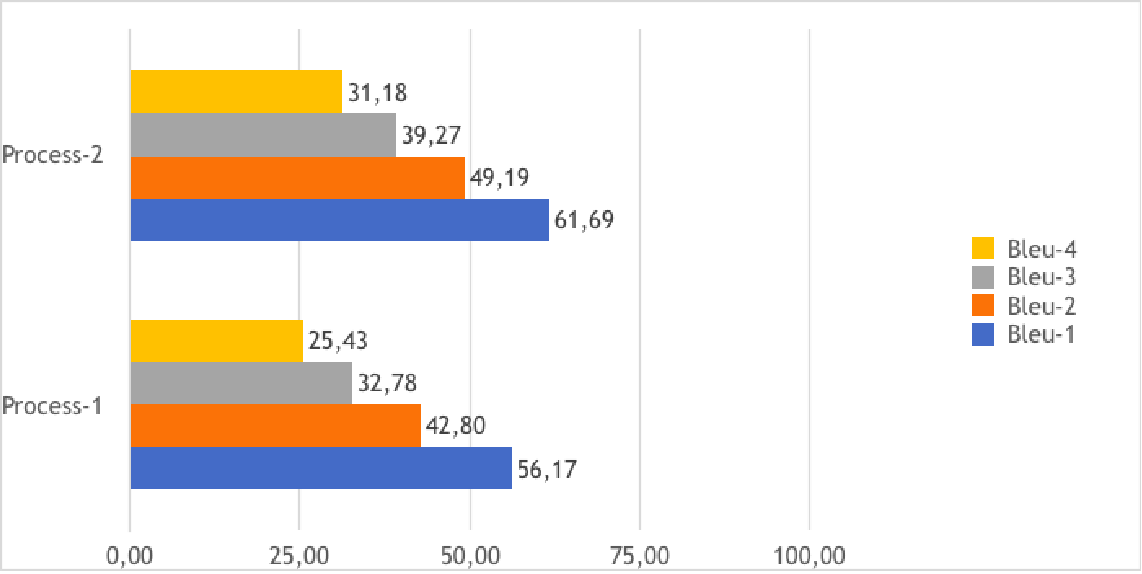


Figure 5.1. Without any preprocessing

In this step, ITU-NLP tool has been used and stemming have been applied. The output was first run by the Moses' tokenizer and recaser, and then the system has been trained (Process-3). As another approach, Moses' preprocessing tools have not been used again. Punctulation has been removed and the output from stemming have been directly used for training (Process-4). The rise of post-stemming results was something was expected. This approach eliminates data sparsity.

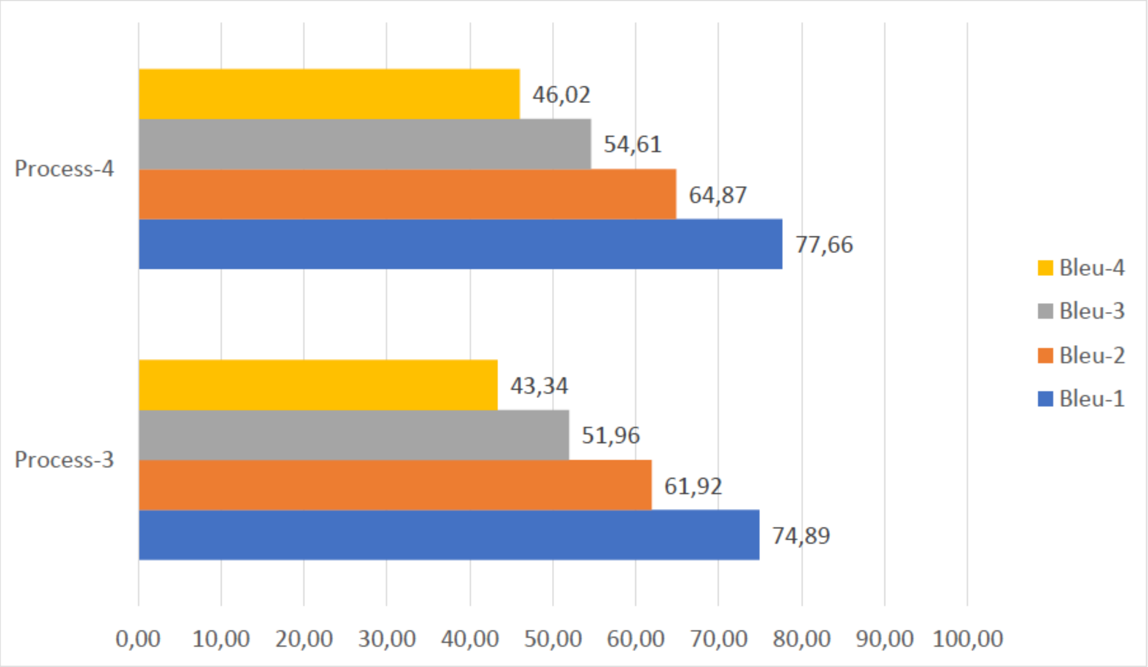


Figure 5.2. After Stemming

The first step of our approach was used for the following results. First, each token was examined to find out whether the verb is negative or positive meaning. If there is a negative tag for a verb, +neg tag is added. By this way, negative and positive verbs did not lose their meanings after stemming. Again, in the Process-5 Moses' tokenizer and truecaser have been used. In Process-6, punctuation is removed.

The next step is the second step of our approach. Again, in Process-7, Moses scripts have been used and in Process-8 Moses scripts have not been used, instead punctuation is removed. Also, in this approach each token has been examined. If the token is noun then it is checked whether it has possessive suffix or not. For these

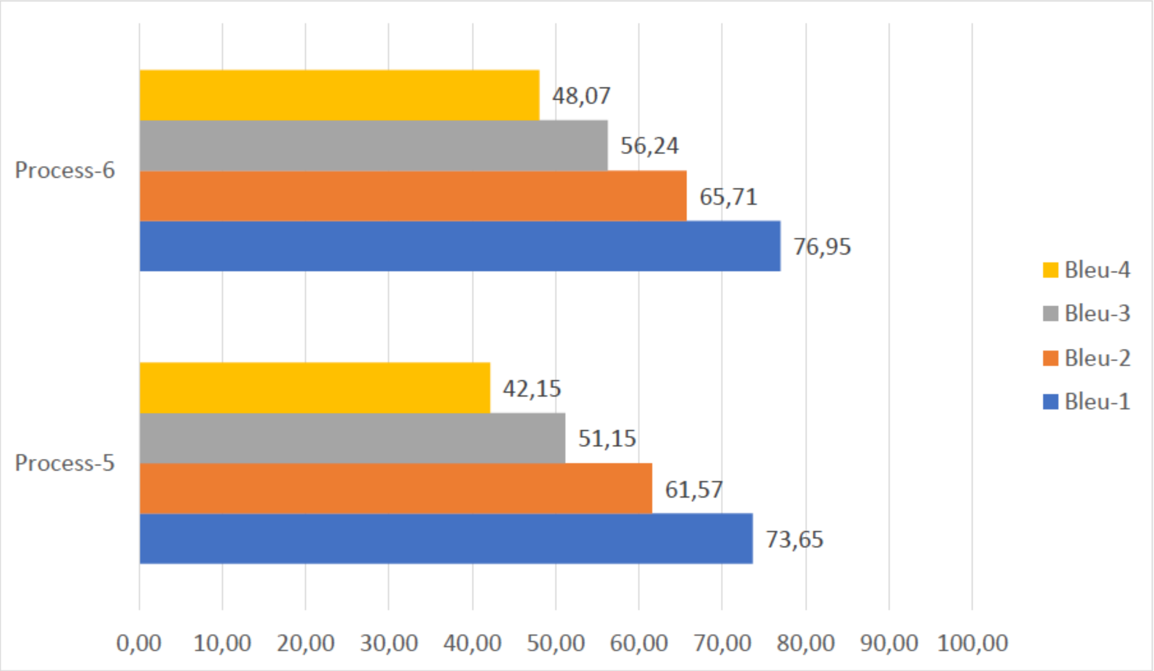


Figure 5.3. After 1st Approach

nouns, pronounce is added to them. By this way, the Bleu-1 score has been increased. However, adding pronouns to any possessive suffix, caused having large number of pronouns in the sentence, and in this case Bleu-2 and Bleu-3 scores dropped. Results can be seen in Figure 5.4.

The third step is the last step. In this step, verbs are examined. If the verb has person agreement then pronoun is added to the sentence. With this method, the fewness in BLEU-2 and BLEU-3 scores can be explained while the increment in BLEU-1 score has already been known previously. The pronouns from the second approach and the pronouns from the third approach led to extra tokens in the sentence. Due to the lack of fully established rules within the TİD, it is not easy to choose the pronoun for given verbs and nouns.

The reason why BLEU-3, BLEU-4 and WER scores fall after certain stage is that the system continuously adds pronouns without examining the structure and elements of the sentence because of 2nd and 3rd rules. This means, word alignments get better

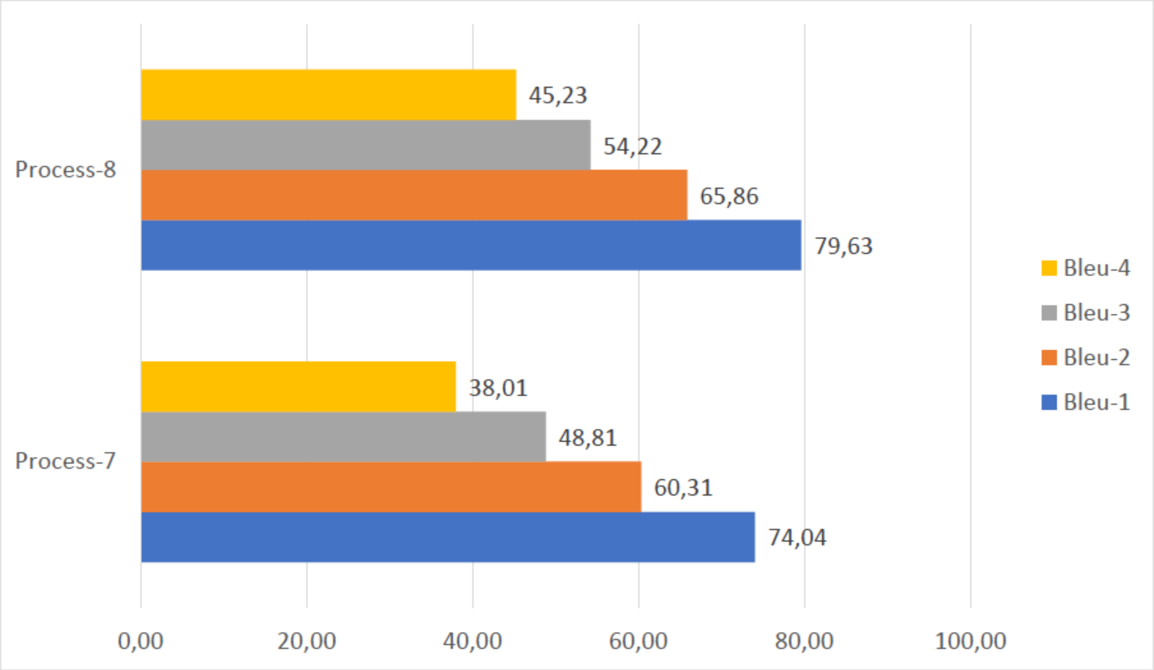


Figure 5.4. After 2nd Approach

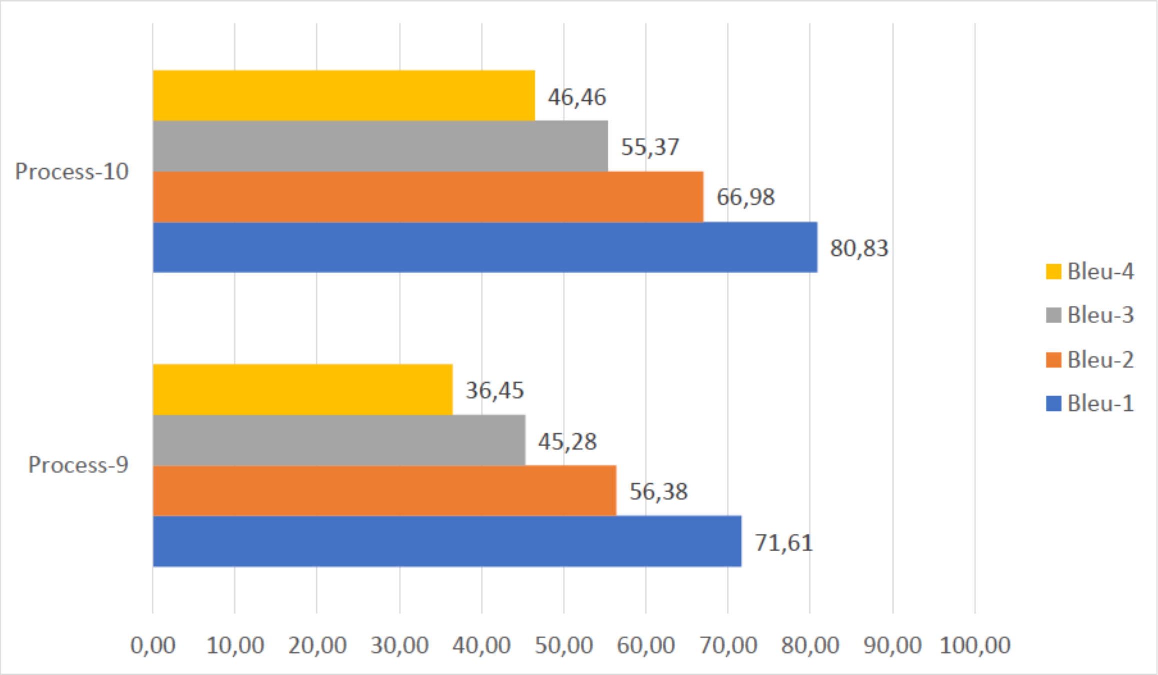


Figure 5.5. After 3rd Approach

because every token can be translated, however, because there are lots of pronouns coming from nouns and verbs, number of overlapping n-grams increases. For example, for each noun which has possessive suffix, pronoun "ben" (*I*) is added to the sentence given below.

Turkish sentence: Ben, ablam, annem, babam, babaannem ve büyükbabam birlikte yaşıyoruz.

(I, my sister, mother, father, grandmother and grandfather live together.)

After all processes: ben ben abla ben anne ben baba ben babaanne ve ben büyükbaba birlikte biz yaşa

(I I sister I mother I father I grandmother I grandfather together we to live)

As a final step, Word Error Rates and BLEU scores can be compared in Table 5.2. Baseline is choosed as word to word comparison without SMT approach can be seen in Table 5.1. Also for Process-10, 10-fold cross validation has been applied and final results can be seen in Table 5.3.

Table 5.1. Baseline

Without SMT	**Bleu-1**	**Bleu-2**	**Bleu-3**	**Bleu-4**	**WER**
Process-2	32.73	20.79	14.40	10.26	68%
Process-4	55.73	42.98	32.86	24.85	46%
Process-6	54.45	43.24	34.57	27.90	48%
Process-8	60.19	42.80	29.75	20.09	55%
Process-10	57.65	39.79	28.82	21.31	63%

Table 5.2. Results

	Bleu-1	**Bleu-2**	**Bleu-3**	**Bleu-4**	**WER**
Process-1	56.17	42.80	32.78	25.43	50%
Process-2	61.69	49.19	39.27	31.18	42%
Process-3	74.89	61.92	51.96	43.34	38%
Process-4	77.66	64.87	54.61	46.02	30%
Process-5	73.65	61.57	51.15	42.15	33%
Process-6	76.95	65.71	56.24	48.07	28%
Process-7	74.04	60.31	48.81	38.01	35%
Process-8	79.63	65.86	54.22	45.23	32%
Process-9	71.61	56.38	45.28	36.45	41%
Process-10	80.83	66.98	55.37	46.46	29%

Table 5.3. 10-fold Cross Validation

k	**Bleu-1**	**Bleu-2**	**Bleu-3**	**Bleu-4**	**WER**
1	76.29	65.15	55.70	47.03	28%
2	77.57	65.11	54.98	46.56	29%
3	77.34	65.46	55.56	46.72	31%
4	77.47	67.23	58.00	50.01	28%
5	80.46	68.68	59.28	50.15	29%
6	74.52	61.79	51.42	42.59	34%
7	74.69	63.14	53.11	44.59	34%
8	77.81	64.00	51.96	42.19	32%
9	75.75	63.36	52.86	44.15	32%
10	78.65	65.40	55.27	47.07	29%
Average	77.05	64.93	54.81	46.10	30%

6. CONCLUSION AND FUTURE WORK

With this study, for the first time, translation from Turkish into TİD was performed by using SMT. This system also adds a new approach to the TİD studies which are quite few in the Machine Translation field. Also about 2000 new TİD translations are added to the literature. It is shown that the size of the corpus, which is thought to be the most important issue in statistical translation, is not crucial for us to represent a closed domain in itself. If this study is appreciated as a take of point, SMT will be approved on more complex algorithms and more data. Currently, even the current translations are quite meaningful.

The system was tested with different approaches using the Moses decoder. In this way, after each approach different information was obtained about the TİD and the system was developed a little more at each step. In particular, this study was conducted with a book that children of primary school age encountered at school every day. Thus, it was aimed to overcome the limit of access to information for deaf and dumb children of primary school age. This system can be used as an intermediate step, and each word/phrase that we translate can be shown with animation for further studies.

Obviously, since there were no previous and similar works in this area and we created and used this dataset for the first time, we are not able to make a comparison with other systems. Instead we used baseline (word to word translation) scores for comparison. However, the BLEU score was 30.23 in the French-English dataset used by Moses as an example [15], while the highest score after 10-fold cross validation is 46.010 in this study.

Another point to note is that the data is translated by ourselves. If the data is translated by native TİD users in future works, there may be differences in translations. Thus, native TİD users and researchers are still needed to validate this dataset. This work was used to show that the sign language is also suitable for SMT. But the point is that; for a language with little studies compared to other languages, a previously untried approach has been presented. The system has been evaluated with BLEU and WER. It was mentioned in previous studies that these metrics were important in translation [4].

The system has been also tested in different situations. The approach which has relatively highest score was attempted with the 10-fold cross validation and the train/development sets of different sizes. This shows us that with more studies on TİD, different approaches can be created and the translation system can be improved.

With this study followings can be deduced,

- SMT can also be meaningful with little data.
- System performance can be improved with different approaches and data to be added in the domain.
- Such a system may be included in the translation system from Turkish written sources to the TID visual sources.

As future work, more data and algorithms can be added to the system. Another task to do next can be adding visualization of translation for primary school children. Also, Neural Machine Translation can be tried for sign language translation. This way we can update our work to new era of deep learning.

REFERENCES

1. Yorganci, R., A. Alp Kindiroglu and H. Kose, “Avatar-based Sign Language Training Interface for Primary School Education”, *Workshop: Graphical and Robotic Embodied Agents for Therapeutic Systems*, 2016.

2. Eryigit, C., H. Kose, M. Kelepir and G. Eryigit, “Building machine-readable knowledge representations for Turkish sign language generation”, *Knowledge-Based Systems*, Vol. 108, pp. 179–194, 2016.

3. Bungeroth, J. and H. Ney, “Statistical sign language translation”, *Workshop on representation and processing of sign languages, LREC*, Vol. 4, pp. 105–108, 2004.

4. Och, F. J., “Minimum error rate training in statistical machine translation”, *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Vol. 1, pp. 160–167, 2003.

5. Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, “Moses: Open source toolkit for statistical machine translation”, *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pp. 177–180, 2007.

6. Stolcke, A., “SRILM – An Extensible Language Modeling Toolkit”, *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, Vol. 2, 2004.

7. Och, F. J. and H. Ney, “A Systematic Comparison of Various Statistical Alignment Models”, *Computational Linguistics*, Vol. 29, No. 1, pp. 19–51, 2003.

8. Mishra, G. S., A. K. Sahoo and K. K. Ravulakollu, “Word based statistical machine translation from english text to indian sign language”, *ARPN Journal of Engineering and Applied Sciences*, Vol. 12, No. 2, 2017.

9. Othman, A. and M. Jemni, "Statistical Sign Language Machine Translation: from English written text to American Sign Language Gloss", *International Journal of Computer Science Issues*, Vol. 8, pp. 65–73, 2011.

10. Gavrila, M. and C. Vertan, "Training Data in Statistical Machine Translation-the More, the Better?", *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pp. 551–556, 2011.

11. Papineni, K., S. Roukos, T. Ward and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation", *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318, 2002.

12. Snover, M., B. Dorr, R. Schwartz, L. Micciulla and J. Makhoul, "A study of translation edit rate with targeted human annotation", *Proceedings of association for machine translation in the Americas*, Vol. 200, No. 6, 2006.

13. Eryiğit, G., "ITU Turkish NLP web service", *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1–4, 2014.

14. Jabin, S., S. Samak and S. Kim, "How to Translate from English to Khmer using Moses", *International Journal of Engineering Inventions*, Vol. 3, pp. 71–81, 2013.

15. Koehn, P., *Statistical Machine Translation System User Manual and Code Guide*, 2010, `http://www.statmt.org/moses/manual/manual.pdf`, accessed at January 2019.