

BIDIRECTIONAL MACHINE TRANSLATION BETWEEN TURKISH AND TURKISH SIGN LANGUAGE: A DATA-DRIVEN APPROACH

Merve Selcuk-Simsek¹ and Ilyas Cicekli^{2,3}

¹School of Computer Science, University of Manchester, UK

²Department of Computer Engineering, Hacettepe University, Turkey

³Department of Electrical Engineering and Computer Science, Syracuse University, USA

ABSTRACT

Communication is one of the first necessities for human beings to live and survive. There are many different ways to communicate for centuries, yet there are mainly three ways for today's world: spoken, written and sign languages. According to research on the language usage of deaf people, they commonly prefer sign language over other ways. Most of the times they need helpers and/or interpreters on daily life and they are accompanied by human helpers. We intend to make a bidirectional dynamic machine translation system by using an example-based approach, and apply between Turkish and Turkish Sign Language (TSL) glosses for the first time in literature with the belief of one day this novel work on Turkish would help these people to live independently. Using BLEU and TER metrics for evaluation, we tested our system considering many conditions, and got competitive results especially compared to previous work in this field.

KEYWORDS

Machine Translation, Turkish Sign Language, Example Based Machine Translation

1. INTRODUCTION

Machine translation between spoken natural languages has been done for more than half of a century. The time period overlaps when the world started to accept sign languages as natural languages in 1950s [1], and the machine translation came true. Considering the late formalization of sign languages, it is plausible that combining them with machine translation also became late. Eventually, this is now an area which has a history about 20 years, and it proceeds even faster than expected.

There are differences when machine translation takes place between two spoken natural languages and between a spoken language and a sign language. The differences occur in the layers of computational linguistics. The first difference is the lack of resources for sign languages, which is nonexistent for the translations between spoken languages. The second problem is the decision of how to represent the sign language in translations. Some studies prefer to use a mannequin, or an avatar to visualize the output [2][3][4][5]. A few of the past studies chose to express the results in glosses [6]. This study aims to put forward a dynamic, bidirectional machine translation system between written Turkish and Turkish Sign Language (TSL) glosses. In sign language part, we present our translation output as glosses which are textual representations of a sign language. To provide both dynamic and double-way translation systems while translating sign languages, glosses are perfectly suited. We also have partners working on

producing sign language animations from sign language glosses that are generated by our translation system.

The reason that we call our system as dynamic is the usage of a corpus-based approach. The choice of the corpus-based approach narrowed the methods we considered into two, statistical approach and example-based approach. Both methods need a bilingual translation dataset to start the process. There are resources claiming that they both should be classified as example-based methods [7]. They differ in the ongoing process of learning. We prefer example-based machine translation (EBMT) in this context considering its benefits over statistical approach listed below (Details about the EBMT system are presented in Section 3):

- EBMT is the most suitable approach to process languages whose grammar is barely known, and which has limited resources (in our context, TSL).
- Even with a limited dataset it could produce efficient results compared to other approaches.
- All inputs and outputs in an EBMT system are human-readable, especially comparing with the statistical approaches.

According to a corporate report published by Turkish Statistical Institute and Ministry of Family and Social Policies in 2002, there were 8,431,937 disabled people in Turkey which equals to 12.29% of the total population. The literacy percentage among these people is 63.62%; yet, the highest percentage of graduated school belongs to primary school with 40.97%. Most of the Deaf graduated from high school can't communicate by writing and reading. Consequently, continuing the education with college or a similar degree has a percentage of only 2.42%. Ultimately, participation rate for the Deaf is even less than 15%.

According to Turkey Disability Survey conducted in 2002, most rated items listed by people who have hearing and/or speaking disabilities about the difficulties affecting their lives are:

- Inadequacy of visual warning signs in public places,
- Being unable to solve daily life problems,
- Being lost in communication,
- Inadequacy of visual broadcast in TSL.

These issues can only be fixed in present situation by using human effort which is expensive and causes time delay. The main purpose of this study is to present a dynamic approach as an alternative for all future efforts.

The flow of this paper is as follows. In Section 2, TSL structure is explained. We present our EBMT system between Turkish and TSL in Section 3. The details of the learning method of our EBMT system are explained in Section 3.1. For the translation part, we use a modified Earley parser to match sequences and it is discussed in Section 3.2. System's evaluation is given in Section 4. Section 5 presents some of the related works and the paper is concluded with concluding remarks in Section 6.

2. TURKISH SIGN LANGUAGE (TSL)

TSL (*Türk İşaret Dili (TİD)* in Turkish) is a language commonly used by people who have hearing and/or speaking disabilities in Turkey. Unlike many other sign languages, TSL has always been used freely by people in Turkey since the beginning of their history. The foundation of the first school for the Deaf dates back as early as 1800s by Ottoman Empire [8]. By the time disabled people were excluded from the society and were dispossessed of their rights in other

parts of the world, they had the same rights with others, and were highly respected by the Sultan and the community in Ottomans [1][9].

In modern Turkey, starting with the 1950s the Deaf have been considered to have education in the same class with other students using oralism [10]. On the other hand, researches prove that these people feel more comfortable while communicating in sign language. 70-75% of the Deaf in today's Turkey prefer TSL on their daily life [11].

TSL is a very rich natural language in terms of phonology and morphology [12]. It has a specific grammar, and at least 2607 known signs [13]. Turkish and TSL do not have similarities except the word set used in TSL is a subset of Turkish. Although there is no common sign language between different nationalities, there are common signs in many sign languages enabling people to understand each other in a basic level. However, according to previous research, TSL users do not have common signs to communicate people from other nationalities. This information is a proof of TSL to be a unique sign language been affected from any other languages according to [14].

TSL is considered about 500 years old [12] with the fact that present-day TSL is known to be used since 1950s. Having already a little evidence, if it is proven that sign language used in Ottoman Empire is the same language used today, TSL becomes the oldest and first common used sign language on Earth [14].

2.1. TSL as a Computational Language

To process a language computationally, it may be the first rule to define its grammar rules to the environment. While one of the parts of this study, Turkish, is already ready to be processed based on its long computational history, the other part, TSL doesn't have enough resources to be implemented easily in a computational environment. There are just a few works about TSL among past studies run in a computational environment. Some of them are based on computer vision [15][16], yet one of them is a unidirectional rule-based machine translation system, but it is not eligible for comparison since it hasn't been completed yet [17]. Due to lack of resources, we gathered our own data, and created a notation system for TSL for processing.

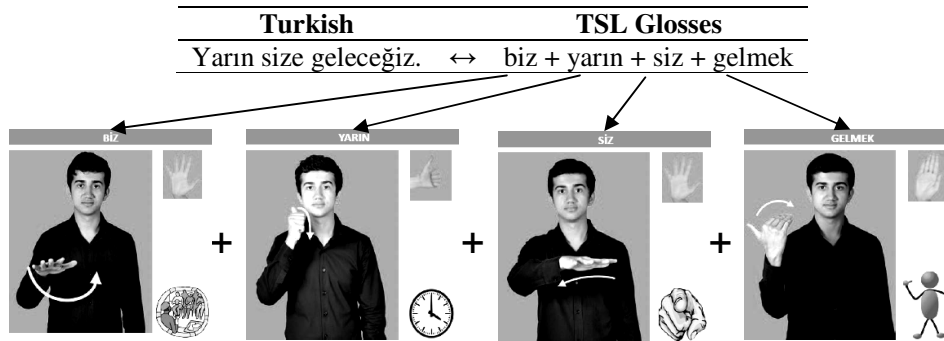


Figure 1. TSL Signs from the TSL Dictionary Published by Turkish Ministry of National Education

There are a few notation systems introduced to represent signs in computational linguistics [18][19][20]. We choose to present our TSL glosses as written versions of signs both for readability and compatibility for later visual representation. An example of our gloss notation is shown in Figure 1.

There is a strong connection between a spoken and a sign language belonging to the same country [12]. However, they might not be similar in terms of their grammars as in the case of Turkish and TSL. Turkish belongs to the Turkic languages group, but word order in TSL glosses does not look like a Turkic language. It rather looks like a Germanic language as in German or English. It can be seen distinctively in Figure 2 and Table 1. Figure 2 illustrates mapping between Turkish words and TSL glosses, and Table 1 describes Figure 2 with English translations of the examples. More differences are presented in Section 2.2.

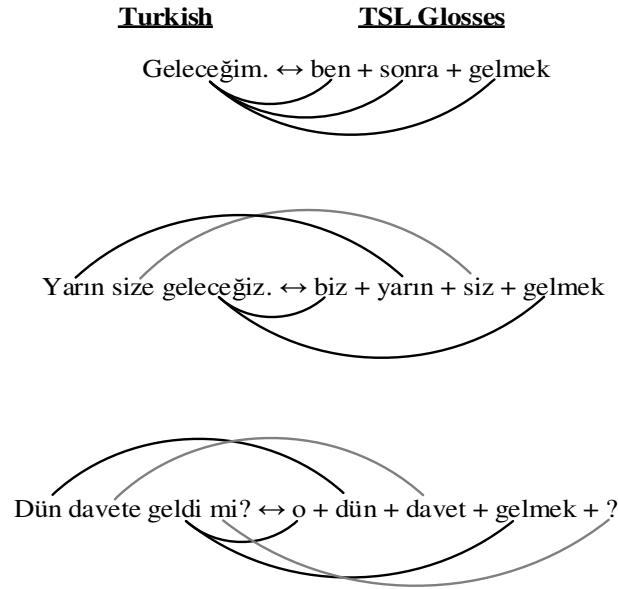


Figure 2. Examples of Mapping between Turkish and TSL Glosses

Table 1. Description of Figure 2

	Turkish		TSL Glosses
	Geleceğim.	↔	Ben + sonra + gelmek
<i>English</i>	<i>I will come.</i>	↔	<i>I + will + come</i>
	Yarın size geleceğiz.	↔	Biz + yarın + siz + gelmek
<i>English</i>	<i>We will come to you tomorrow.</i>	↔	<i>We + tomorrow + you + come</i>
	Dün davete geldi mi?	↔	O + dün + davet + gelmek + ?
<i>English</i>	<i>Did s(he) come to the invitation yesterday?</i>	↔	<i>(s)he + yesterday + invitation + come + ?</i>

Most words in Turkish sentences are represented with suffixes; yet TSL has only two exceptional suffixes which are "-cı" (suffix which indicates a seller of something) and "-lı" (belonging indicator suffix) [12]. As a result of Turkish's agglutinative structure, subjects also can be defined by suffixes. Whenever a subject is defined by a suffix, i.e. an agreement morpheme on predicate, it becomes hidden in the sentence; so, it is called a null subject. However, since every single piece in TSL must be defined by signs, there is no possibility for null subjects to exist in TSL. All the factors mentioned above result TSL to have usually more pieces in a sentence compared to Turkish.

2.2. Differences between Turkish and TSL in Computational Environment

Two languages used in our system differ from each other while processing them in a computational environment. Inputs coming from both languages are firstly processed by lexical supervision component (LSC), and depending on the language (Turkish or TSL), LSC enables or disables some of its functions. Turkish is managed with a compact software tool including morphological processor for Turkish, but for processing TSL we only use an orthography control tool. Both operations are run by LSC.

Turkish is an agglutinative language enriched by suffixes and derivational boundaries. One word in Turkish might be equal to a great deal of words in another language. To make it clear a translation example between Turkish and English [21] is given below:

ölümsüzleştiriveremeyebileceklerimizdenmişsinizcesine
Its Root and the Morphemes:
ölüm + süz + leş + tir + iver + eme + yebil + ecek + ler + imiz + den + miş + sizin + cesine
English Translation:
"(behaving) as if you are among those whom we could not cause hastily to become immortal"

The agglutination of Turkish makes translation almost impossible by only using surface forms. To obtain higher accuracy and to get higher possibility rates while matching sequences, using morphological analysis is almost a must for such languages. We use a PC-KIMMO [22] based morphological analyzer as a Turkish morphological analyzer to get lexical forms of Turkish sentences in our system [23]. An example for Turkish morphological analyzer usage is given in Table 2.

Table 2. An Example of Morphological Analysis for Turkish

Input Sentence: Eve geldin mi? (*English: Did you come home?*)

Tokens in Surface Form	Morphological Analysis Results
Eve ↔	ev+Noun+A3sg+Pnon+Dat
geldin ↔	gel+Verb+Pos+Past+A2sg
mi ↔	mi+Ques+Pres+A3sg
? ↔	?+Punc

Table 3. Disambiguation Illustration of Turkish

Input Sentence: Koşan adamı görmedim. (*English: I didn't see the running man.*)

Tokens	Morphological Analysis Results	Disambiguation Results
Koşan	koşa+Noun+A3sg+P2sg+Nom koş+Verb+Pos^DB+Adj+PresPart	koş+Verb+Pos^DB+Adj+PresPart
adamı	ada+Noun+A3sg+P1sg+Acc adam+Noun+A3sg+P3sg+Nom adam+Noun+A3sg+Pnon+Acc	adam+Noun+A3sg+Pnon+Acc
görmedim	gör+Verb+Neg+Past+A1sg	gör+Verb+Neg+Past+A1sg
.	+.Punc	+.Punc
Output:	koş+Verb+Pos^DB+Adj+PresPart gör+Verb+Neg+Past+A1sg	adam+Noun+A3sg+Pnon+Acc +.Punc

Ambiguity is a part of every natural language, and it might cause wrong translations unless it is handled. We preferred to use a hybrid method introduced in [24] to disambiguate Turkish sentences after eliminating some other disambiguation tools written for Turkish according to their accuracy results. The example in Table 3 illustrates disambiguation of a Turkish sentence. Words *koşan* and *adamı* in the input sentence below have multiple morphological meanings due to suffixes. To be able to translate the input correctly, the right meaning of the tokens has to be identified by applying disambiguation.

Because TSL is a sign language, its glosses are not analyzed morphologically. All glosses are assumed to belong to the type SIGN as shown in Figure 3. The glosses are examined only by using an orthography control tool to check whether there is a misspelling in sign's writing or to eliminate irrelevant symbols in sentences.

Turkish					TSL Glosses				
<u>E</u> ve	<u>g</u> eldin	<u>m</u> i	?	↔	<u>ö</u> n <u>ç</u> e	+ <u>s</u> en	+ <u>e</u> v	+ <u>g</u> el <u>m</u> ek	+ ?
Noun	Verb	Ques	Punc		SIGN	+ SIGN	+ SIGN	+ SIGN	+ SIGN

Figure 3. Turkish vs. TSL Glosses Depending on Part of Speech Tags

3. EBMT SYSTEM BETWEEN TURKISH AND TSL

Our translation system consists of two main parts, learning and translation components; and one sub part, lexical supervision component (LSC) used during both procedures. For Turkish inputs, LSC enables a morphological analyzer, an orthography control tool, and a disambiguator. For TSL inputs, LSC enables only the orthography control tool. The system basically works with taking two inputs: a bilingual dataset, and input sentence/sentences in source language. Afterwards, it produces one output: translation of the input in target language. Figure 4 visualizes a summary of how system works.

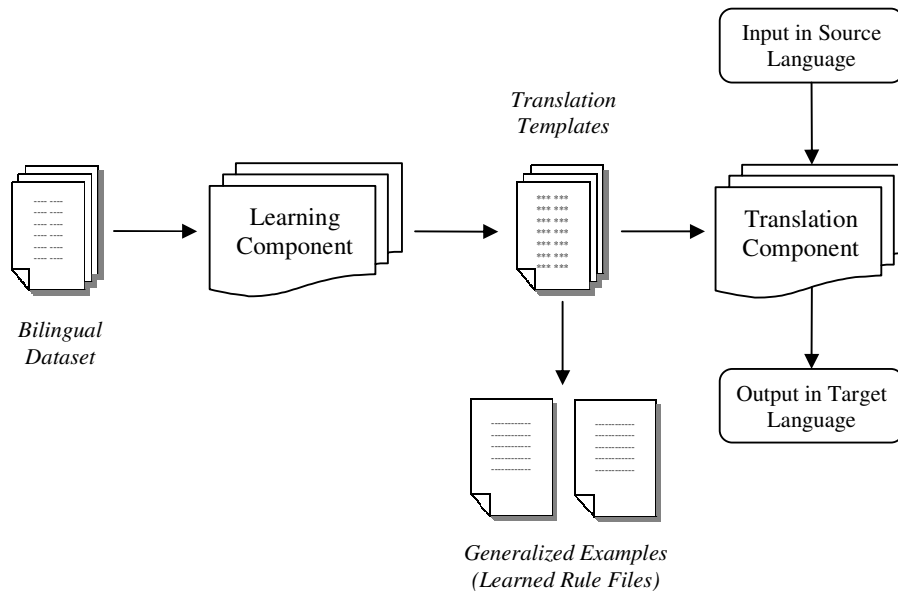


Figure 4. EBMT System's Summary

3.1. Learning Component

Example-based learning aims to use gained knowledge from past experiences [7][24]. In this context, we use bilingual translation examples as past experiences to realize example-based learning. Our learning algorithm is called TTL (Translation Template Learner) [26][27]. Main steps of learning procedure including TTL are shown in Figure 5.

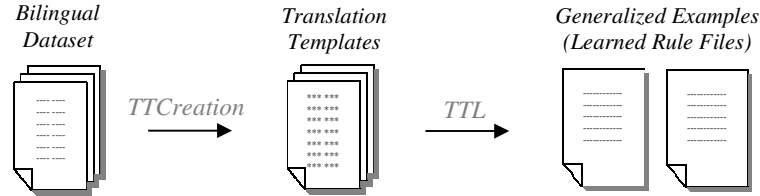


Figure 5. Documentation Steps of Learning

On each intermediate step in Figure 5, learning and saving operations are being executed. *TTCreation* process reads translation pairs from *Bilingual Dataset* in surface level, and morphologically disambiguates Turkish words in order to create *Translation Templates* in lexical level. The most important part, application of *TTL* occurs at the last step after translation templates are ready. Note that templates are stored versions of examples in lexical level. Some examples from translation templates are given below:

```
% kitaplar ↔ çok+kitap(English: Books)
train_pair([kitap+Noun,+A3pl,+Pnon,+Nom], [çok,kitap]).
% Geldim. ↔ ben+önce+gelmek (English: I came.)
train_pair([gel+Verb,+Pos,+Past,+A1sg], [ben,önce,gelmek]).
```

Figure 6. Examples from Translation Templates

In Figure 6 lines starting with **train_pair** indicate translation templates. First (left) part of the template is the Turkish side of the translation in lexical level, and second (right) part of the template stands for TSL. Lines starting with "%" resemble the two-way translation in surface level which belongs to the template downside. Some punctuation marks e.g. "." or "," do not take place in templates because their use might change even for the same sentence in Turkish. Therefore, their consideration may cause a wrong translation.

A little detail here is that generalized examples can also be produced without saving templates. Templates still need to be generated to help producing rules, but saving process in middle stage may be omitted. On the other hand, saving template files makes the system more flexible and human-readable.

We use TTL algorithm to generate learned rules from translation templates. TTL's working principle stands for analyzing relations, i.e. similarity or differences, between templates. It works with at least two translation templates. If there isn't any relationship between pairs, a generalized template from the specified pair can't be produced. More information about TTL could be found in [26][27].

In this study TTL algorithm is used with lexical forms of Turkish sentences. Since there are no types on TSL's part of the translations, there is also no need for type information. However, on

Turkish part, we need type information to place words correctly. To accomplish this, lattice files which include hierarchical type information are used. We have two lattice files, one for Turkish and one for TSL. TSL has only one type (SIGN) in its correspondent file while Turkish lattice file includes type information that gradually increases as the system learns. Although there might be no need to use a file for only one type information, we still use two files due to sustainability of system's dynamic structure for possible usage with different language pairs.

1	<u>ev</u> +Noun, +A3pl, +Pnon, +Nom ↔ çok, <u>ev</u>	(English: Houses)
2	<u>kitap</u> +Noun, +A3pl, +Pnon, +Nom ↔ çok, <u>kitap</u>	(English: Books)

LEARNED TEMPLATES

3	S, +A3pl, +Pnon, +Nom ↔ çok, S [(Noun, SIGN)]	
4	ev+Noun ↔ ev	(English: House)
5	kitap+Noun ↔ kitap	(English: Book)

Among the examples above, numbered items 1 and 2 are sample translations in our dataset. These are both plural formed examples. We expect TTL to result with learning how to make plural forms of Turkish words and TSL glosses. Items 3, 4, and 5 are produced by TTL as learned rules. Item 3 is a generalized example of items 1 and 2. It is basically a learned rule showing how to make a plural form of a type *NOUN* for Turkish and *SIGN* for TSL. *NOUN* and *SIGN* are general types coming from lattice files. From now on, words whose types are ordered under *NOUN* or *SIGN* could be pluralized. Number 4 and 5 are gained word knowledge from the translation samples 1 and 2. These are the results of *differences* between first two sentences. After generalized examples are produced, confidence factors are assigned to them. We use confidence factors to prove reliability of translations [28].

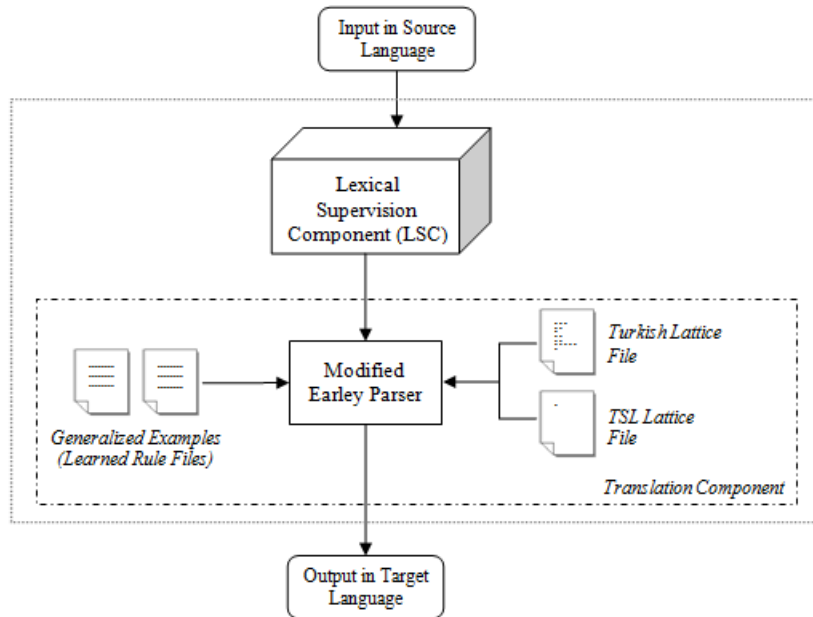


Figure 7. Translation Component of the System

3.2. Translation Component

Translation component is the last part of the system. A summary form of translation component is illustrated in Figure 7. The crucial work here is done by our modified Earley parser. Earley parser was built as a context-free grammar parser [29]. Grammar parsers are used to detect a given sentence's conformity to the specified grammar in general. Though our purpose might seem irrelevant at first sight, in this study we use a grammar parser to find possible translations. We modified the original Earley parser to make it much faster in matching and more efficient in memory usage for our purpose [30].

After an input sentence passes from the LSC, a radix tree carrying all relevant grammar rules is generated. Relevant rules are gathered from the generalized translation examples which carry similar parts to the input sentence [30]. Our parser, or matcher, scans relevant rules coming with a radix tree, makes use of the lattice files when type information is needed, and tries to find a candidate translation for the sentence in source language. At the end, we may still consider the parser here as a grammar parser that decides whether a sentence fits to the grammar or not. If more than one fit (translation) is found, confidence factors determine the final translation. The bigger the confidence factor gets, the more reliable the translation is.

4. EVALUATION

4.1. Evaluation Datasets

Decision of how to compose an evaluation dataset is important for a fair evaluation. In most of the related work, the evaluation dataset is constructed manually according to individual criterion, and under the name of gold standard dataset. Because there is no common criterion composing such sets, we chose to benefit from k -fold cross validation algorithm to determine our training and test sets, and tested our system for k values 3, 5, and 10.

We use three datasets (A , B , and C) to observe different characteristics of the system. A , B , and C are prepared manually with the help of TSL experts due to having no standard dataset in TSL, and they are different from each other in terms of size and features. A is our biggest dataset with size of 2068 translation pairs. With its size it is also one of the biggest datasets in past studies for sign languages regarding [6]. C is a subset of B , and B is a subset of A . B covers examples which contain some specific grammatical rules e.g. plural forms, tenses, pronouns etc. An overview of our datasets is given in Table 4.

Table 4. Statistics of Evaluation Datasets

	A		B		C	
	<i>TR</i>	<i>TSL</i>	<i>TR</i>	<i>TSL</i>	<i>TR</i>	<i>TSL</i>
Sentences	2068		646		544	
Words	10,067	15,070	2542	3779	2282	3551
Distinct Words	2463	1113	844	485	740	453
Singleton Words	526	161	280	108	210	89
Minimum Word Count per Sentence	1	2	1	2	2	3
Maximum Word Count per Sentence	15	18	8	13	8	13
Average Word Count per Sentence	4.87	7.29	3.93	5.85	4.19	6.16
Average TSL Glosses Equals to 1 Turkish Word	1.67		1.62		1.52	

4.2. Tests and Discussion

We use BLEU[31], and TER [32] as evaluation metrics. While testing with BLEU we calculated n -grams regarding $n=4$ with weighted precision in default. We tested our system according to $k=3, 5, \text{ and } 10$ using k -fold cross validation algorithm. Note that better results are equivalent to higher BLEU (BiLingual Evaluation Understudy), and lower TER (Translation Error Rate) values.

Table 5. Details of Training and Test Sets

Corpus	k Value	Training Set Sentence Count	Test Set Sentence Count
A	10	1862	206
B	10	582	64
C	10	490	54

Table 6 summarizes the test results according to datasets in Table 5. Datasets were divided only by k value of 10 (the standard k value) for these tests. The best results of both sides are highlighted in Table 6. Note that due to insufficiency of resources in TSL we didn't use any dictionaries for both sides of translations for the current study.

Table 6. General Test Results

Corpus	TR \rightarrow TSL		TSL \rightarrow TR	
	BLEU	TER	BLEU	TER
A	0.33	0.44	0.56	0.47
B	0.41	0.4	0.56	0.49
C	0.43	0.38	0.59	0.47

Best results are obtained using dataset C according to Table 6. All results in Table 6 are among the highest now in the literature to our knowledge although we didn't use any dictionaries.

Table 7. Test Results of Dataset A According to Different k Values

K Value	TR \rightarrow TSL		TSL \rightarrow TR	
	BLEU	TER	BLEU	TER
3	0.29	0.46	0.53	0.5
5	0.33	0.43	0.56	0.46
10	0.33	0.44	0.59	0.47

Another test we performed on dataset A consists of different k values (Table 7). Obviously seen that the bigger k is, the higher BLEU score gets based on the fact that data-driven approaches get their reliability from the training sets size. An interesting point here is that changing k values in our datasets do not affect the BLEU score much based on Table 7. There are no large amounts of changes between any scores in the table above. All changes related to varied k values occur smoothly. This is the result of well distributed examples resulting effectively learned grammar rules in datasets.

According to tests, translations from TSL to Turkish (TSL \rightarrow TR) resulted better than translations from Turkish to TSL (TR \rightarrow TSL). Success of data-driven approaches depends on data, so it is not easy to pinpoint the source of errors while evaluating these approaches. We suspect a couple

of reasons based on this situation. One of them is that vocabulary of TSL is a subset of Turkish. TSL is likely to have more gained word knowledge compared to Turkish in generalized examples. Lack of vocabulary information affects translation results. It may also cause empty sequences from Turkish to TSL in some cases, and limit the chance to find proper match sequences while translating. The other factor is relevant to the fact that TSL glosses are processed in surface form while Turkish is lexicalized. Matching is more clear and easier with surface forms, yet lexicalization increases the chances to get true translations especially for agglutinative languages. Although Turkish is defined better when lexicalized, its components also become harder to map especially without a dictionary. In other words, this is a challenging design decision.

5. RELATED WORK

Machine translation systems between spoken and sign languages had a start in the late 90s. Being one of the most spoken languages on Earth, English has been among the first and the most common translated languages since then. We examine related work dividing into two subsections: works involving rule-based (traditional) approaches (Section 5.1), and works involving corpus-based approaches (Section 5.2).

The reason that there is no mention of any machine translation system between Turkish and TSL in this section is because there is no previously completed and published work.

5.1. Rule-based Approaches

Starting with the first machine translation experiment in 1954 [33], rule-based systems compose the first and the most used approaches in the background of this area. Unlike newer studies, interlingua method was in demand for the first examples of rule-based translation systems in the late 90s. A multilingual system, Zardozi [34], and TEAM [2] are among these systems. Zardozi was capable of translating into many sign languages like American Sign Language (ASL), Irish Sign Language (ISL), and Japanese Sign Language. TEAM was constructed to translate from English to ASL. To form an interlingua from English to ASL, they used STAG (Synchronous Tree Adjoining Grammar) data structure in TEAM. Neither systems included evaluation information.

Up to date systems comprise more of a transfer-based approaches. We observed one other study in past studies covering a translation system between a spoken language and a sign language's glosses. This is a rule-based, unidirectional translation system from Spanish to Spanish Sign Language glosses [6]. They evaluated their system using various datasets each including 195 sentences. The tests resulted 30% BLEU, and 42% TER.

5.2. Corpus-based Approaches

First study in this field was a statistical machine translation system covering a one-way translation platform from German to German Sign Language (DGS) [3]. Even though at first they had 1399 translation pairs in their system for evaluation, they dropped it into 200 pairs because the dataset had a huge amount of singleton words (34% for German, 16% for DGS). They used 167 pairs for training and 33 for testing. For the test part they preferred to use mWER (minimum Word Error Rate) (resulted 59.9% at best) and mPER (minimum Position-independent Word Error Rate) (resulted 23.6% at best) as evaluation metrics.

Morrissey and Way benefited from an example-based approach in their studies [35] [36]. Their dataset included 561 sentences consisting of 7.89 words on average, 1 word at minimum and 53 words at maximum [35][36]. They conducted a test using randomly chosen 55 sentences from

English and Dutch to Dutch Sign Language, and got results of 119% WER, 78% PER, and 96% SER (Sentence Error Rate)[36].

One other study using example-based approach is [5]. They preferred chunk-based EBMT, evaluated their 203 paired sentence system splitting with LOOCV (Leave-One-Out Cross Validation), and got results of 46.7% WER, and 29.4% PER.

Later studies present work that combines EBMT with statistical approaches[37][4][38]. The example-based method used in these studies is marker-based chunking. The preferred direction in translation is sign language to spoken language in [37]. They used a corpus consists of 595 sentences, and best result is between Irish Sign Language (ISL) to English with 52.18% BLEU. In [38] English to ISL resulted about 39% BLEU, and German to DGS 50% BLEU.

All approaches summarized above cover unidirectional translation systems with using smaller datasets compared to ours. The best results apparently belong to combined corpus-based approaches; yet we reached better results by only using an EBMT system. Furthermore, a hybrid approach between Turkish and TSL might give better results, and that's why, this could be examined in future work.

6. CONCLUSION

In this study, we introduce a bidirectional, dynamic machine translation system between Turkish and TSL glosses for the first time in relevant studies. TSL glosses are represented as written signs in our system to make them easily understandable and maintainable for possible further visual studies. We prefer one of the most prevalent methods in the literature, a corpus-based approach, example-based machine translation (EBMT) to make our system both dynamic and scalable, and to obtain considerable results even with a language lack of resources as TSL. For evaluation we used three differently scaled and prepared datasets. Furthermore, to determine training and test sets we benefited from k-fold cross validation method with k values of 3, 5, and 10. We tested our system using BLEU and TER metrics. Because EBMT strictly relies on corpus, we got different results for different sized and/or featured datasets. Finally, we had competitive and encouraging results for such a proof-of-concept study especially comparing with past work in terms of BLEU (43% for $TR \rightarrow TSL$, 59% for $TSL \rightarrow TR$), and TER (38%, 47%).

REFERENCES

- [1] Y. K. Kemaloğlu. "Konuşamayan İşitme Engellilerin (Sağırların) Tarihi - History Of The Speechless Hearing Disabled People (The Deaf)." *KBB ve BBC Dergisi*, vol. 22(1), pp. 14–28, 2014.
- [2] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler, and M. Palmer. "A Machine Translation System from English to American Sign Language." *Envisioning Machine Translation in the Information Future*, vol. 1934, pp. 54–67, 2000.
- [3] J. Bungeroth and H. Ney. "Statistical Sign Language Translation," in *Proceedings of LREC Workshop on Representation and Processing of Sign Languages (LREC 2004)*, 2004, pp. 105–108.
- [4] S. Morrissey and A. Way. "Joining Hands: Developing A Sign Language Machine Translation System with and for the Deaf Community," in *Proceedings of the Conference and Workshop on Assistive Technologies for People with Vision & Hearing Impairments (CVHI)*, 2007.
- [5] A. Almohimeed, M. Wald, and R. I. Damper. "Arabic Text to Arabic Sign Language Translation System for the Deaf and Hearing-Impaired Community," in *Proceedings of the 2nd Workshop on Speech and Language Processing for Assistive Technologies*, 2011, pp. 101–109.
- [6] J. Porta, F. Lopez-Colino, J. Tejedor, and J. Colas. "A rule-based translation from written Spanish to Spanish Sign Language glosses." *Computer Speech and Language*, vol. 28, no. 3, pp. 788–811, 2014.
- [7] H. Somers. "Review article: Example-based Machine Translation." *Machine Translation*, vol. 14, pp. 113–157, 1999.
- [8] Y. K. Kemaloğlu and P. Y. Kemaloğlu. "The history of sign language and deaf education in Turkey." *Kulak Burun Bogaz Ihtis. Derg.*, vol. 22, no. 2, pp. 65–76, 2012.

- [9] M. Miles. "Signing in the Seraglio: Mutes, Dwarfs and Gestures at the Ottoman Court 1500-1700." *Disability & Society*, vol. 15, no. 1, pp. 115–134, 2000.
- [10] M. A. Melekoglu, O. Cakiroglu, and K. W. Malmgren. "Special education in Turkey." *International Journal of Inclusive Education*, vol. 13, no. 3, pp. 287–298, 2009.
- [11] Ç. Gürboğa and T. Kargin. "İşitme Engelli Yetişkinlerin Farklı Ortamlarda Kullandıkları İletişim Yöntemlerinin/Becerilerinin İncelenmesi." *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi*, vol. 36, no. 1–2, 2003.
- [12] O. Kubuş. "An Analysis of Turkish Sign Language (TİD) Phonology and Morphology." MSc. Thesis, Middle East Technical University, Ankara, Turkey, 2008.
- [13] Turkish Ministry of National Education. "Turkish Sign Language Dictionary." Internet: http://orgm.meb.gov.tr/alt_sayfalar/duyurular/1.pdf, 2015 [June 10, 2017].
- [14] U. Zeshan. "Aspects of Türk İşaret Dili (Turkish Sign Language)." *Sign Language & Linguistics*, vol. 6, no. 1, pp. 43–75, 2003.
- [15] M. A. Guvensan and H. Haberdar. "Detection and Tracking of Face and Facial Features for Recognition of Turkish Sign Language," in *Proceedings of IEEE 15th Signal Processing and Communications Applications (SIU 2007)*, 2007, pp. 1–4.
- [16] A. Memis and S. Albayrak. "Turkish Sign Language recognition using spatio-temporal features on Kinect RGB video sequences and depth maps," in *Proceedings of 21st Signal Processing and Communications Applications Conference (SIU)*, 2013, pp. 1–4.
- [17] C. Eryiğit, H. Köse, M. Keleş, and G. Eryiğit. "Building machine-readable knowledge representations for Turkish sign language generation." *Knowledge-Based Systems*, vol. 108, pp. 179–194, 2016.
- [18] W. C. Stokoe. "Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf," in *Studies in Linguistics. Occasional Papers*, vol. 8. New York: Buffalo University Press, 1960.
- [19] T. Hanke. "HamNoSys - Representing sign language data in language resources and language processing contexts," *Lr. 2004, Work. Proc. Represent. Process. sign Lang.*, 2004, pp. 1–6.
- [20] V. Sutton. *Lessons in Sign Writing, Textbook and Workbook (Second Edition)*. The Center for Sutton Movement Writing, Inc., 1995.
- [21] H. Sak, T. Güngör, and M. Saraçlar. "Resources for Turkish morphological processing." *Language Resources & Evaluation*, vol. 45, no. 2, pp. 249–261, 2011.
- [22] E. L. Antworth. *PC-KIMMO: A Two-level Processor for Morphological Analysis*. Dallas, TX: Summer Institute of Linguistics, International Academic Bookstore, 1990.
- [23] K. Oflazer. "Two-level description of Turkish morphology." *Literary and Linguistic Computing*, vol. 9, no. 2, pp. 137–148, 1994.
- [24] M. Kutlu and I. Cicekli. "A Hybrid Morphological Disambiguation System for Turkish," in *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP)*, 2013, pp. 1230–1236.
- [25] D. L. Medin and M. M. Schaffer. "Context theory of classification learning." *Psychological Review*, vol. 85, no. 3, pp. 207–238, 1978.
- [26] I. Cicekli and H. A. Güvenir. "Learning Translation Templates from Bilingual Translation Examples," in *Recent Advances in Example-Based Machine Translation*. M. Carl and A. Way, Ed. Boston: The Kluwer Academic Publishers, 2003, pp. 247–278.
- [27] I. Cicekli. "Inducing Translation Templates with Type Constraints." *Machine Translation*, vol. 19, no. 3–4, pp. 283–299, 2005.
- [28] Z. Öz and I. Cicekli. "Ordering Translation Templates by Assigning Confidence Factors," in *Proceedings of the Association for Machine Translation in the Americas (AMTA)*, 1998, pp. 51–61.
- [29] J. Earley. "An Efficient Context-Free Parsing Algorithm." *Communications of the ACM*, vol. 13, no. 2, pp. 94–102, 1970.
- [30] S. Polat, M. Selçuk-Simsek, and I. Cicekli. "A Modified Earley Parser for Huge Natural Language Grammars." *Research in Computing Science*, vol. 117, pp. 23–35, 2016.
- [31] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. "BLEU: A Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02)*, 2002, pp. 311–318.
- [32] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of Association for Machine Translation in the Americas*, 2006, pp. 223–231.
- [33] J. Hutchins. "The Georgetown-IBM Demonstration." *MT News International*, no. 8, pp. 15–18, 1994.

- [34] T. Veale, A. Conway, and B. Collins. “The Challenges of Cross-Modal Translation: English-to-Sign-Language Translation in the Zardoz System.” *Machine Translation*, vol. 13, no. 1, pp. 81–106, 1998.
- [35] S. Morrissey and A. Way. “An Example-Based Approach to Translating Sign Language,” in *Proceedings of the Workshop in Example-Based Machine Translation (MT Summit X)*, 2005, pp. 109–116.
- [36] S. Morrissey and A. Way. “Lost in Translation: The Problems of Using Mainstream MT Evaluation Metrics For Sign Language Translation,” in *Proceedings of the 5th SALT MIL Workshop on Minority Languages at LREC-06*, 2006, pp. 91–98.
- [37] S. Morrissey, A. Way, D. Stein, J. Bungeroth, and H. Ney. “Combining Data-Driven MT Systems for Improved Sign Language Translation,” in *Proceedings of Machine Translation Summit XI*, 2007, pp. 329–336.
- [38] S. Morrissey. “Data-Driven Machine Translation for Sign Languages,” PhD Thesis, Dublin City University, Ireland, 2008.

AUTHORS

Merve Selcuk-Simsek received her BSc degree in computer engineering from TOBB University of Economics and Technology, and her MSc degree in computer engineering from Hacettepe University, both in Ankara, Turkey. This work is a part of her MSc thesis, and was mostly done during her studies in Hacettepe University. She is currently a research assistant in University of Manchester, UK.



Prof. Dr. Ilyas Cicekli received his MS and Ph.D. degrees in computer science from Syracuse University, USA. He is a faculty member of the Department of Computer Engineering at Hacettepe University, Turkey. He is also currently a visiting professor of the Department of Electrical Engineering and Computer Science at Syracuse University. His current research interests include natural language processing, example-based machine translation, text summarization, question-answering, social-media text analysis and information extraction.

