TURKISH to CRIMEAN TATAR MACHINE TRANSLATION SYSTEM

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE OF

BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

By

Kemal Altıntaş

July, 2001

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İlyas Çiçekli (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Özgür Ulusoy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Attila Gürsoy

Approved for the Institute of Engineering and Science

Prof. Dr. Mehmet Baray
Director of the Institute of Engineering and Science

ABSTRACT

TURKISH TO CRIMEAN TATAR MACHINE TRANSLATION SYSTEM

Kemal Altıntaş
MS in Computer Engineering
Supervisor: Asst.Prof.Ilyas Cicekli
July, 2001

Machine translation has always been interesting to people since the invention of computers. Most of the research has been conducted on western languages such as English and French, and Turkish and Turkic languages have been left out of the scene. Machine translation between closely related languages is easier than between language pairs that are not related with each other. Having many parts of their grammars and vocabularies in common reduces the amount of effort needed to develop a translation system between related languages. A translation system that makes a morphological analysis supported by simpler translation rules and context dependent bilingual dictionaries would suffice most of the time. Usually a semantic analysis may not be needed.

This thesis presents a machine translation system from Turkish to Crimean Tatar that uses finite state techniques for the translation process. By developing a machine translation system between Turkish and Crimean Tatar, we propose a sample model for translation between close pairs of languages. The system we developed takes a Turkish sentence, analyses all the words morphologically, translates the grammatical and context dependent structures, translates the root words and finally morphologically generates the Crimean Tatar text. Most of the time, at least one of the outputs is a true translation of the input sentence.

Keywords: Natural Language Processing, Machine Translation, Turkish, Turkic Languages, Crimean Tatar.

ÖZET

TÜRKÇE'DEN KIRIMTATARCA'YA OTOMATİK ÇEVİRİ SİSTEMİ

Kemal Altıntaş
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. İlyas Çiçekli
Temmuz, 2001

Bilgisayarın keşfinden beri otomatik çeviri işlemi insanların ilgisini çekmiştir. Bu konuda bugüne kadar yapılan araştırmaların çoğu İngilizce ve Fransızca gibi Batı Dilleri üzerinde yapılmış, Türkçe ve Türk dilleri sahnenin dışında kalmıştır. Yakın diller arasındaki otomatik çeviri işlemi birbiriyle ilişkisi olmayan diller arasındaki çeviri işleminden daha kolaydır. Gramer ve kelime hazinelerinin önemli bir kısmı ortak olduğundan, yakın diller arasında çeviri yapacak bir sistem geliştirmek daha az çabayla mümkün olabilir. Yakın diller arasında tercüme yapmak için sınırlı tercüme kuralları ve karşılıklı sözlükler tarafından desteklenecek bir biçimbirimsel çözümleme çoğu zaman yeterli olacaktır. Genelde bir anlam çözümlemesine gerek olmayabilir.

Bu tezde Türkçe ve Kırımtatarca arasında tercüme için sonlu durumlu teknikler kullanan bir otomatik çeviri sistemi anlatılmaktadır. Türkçe ve Kırımtatarca arasında geliştirilen otomatik çeviri sistemimizin yakın diller arasında geliştirilecek sistemlere bir model teşkil edeceğini ummaktayız. Geliştirdiğimiz sistem Türkçe bir cümleyi alıp biçimbirmsel olarak çözümlemekte, gramer yapılarını ve çevirisi duruma bağlı olan sözcükleri çevirmekte, kökleri çevirmekte ve son olarak da Kırımtatarca cümleyi biçimbirimsel olarak üretmektedir. Üretilen cümlelerden en az biri çoğu zaman girdi olarak alınan cümlenin doğru bir tercümesi olmaktadır.

Anahtar Kelimeler: Doğal Dil İşleme, Otomatik Çeviri, Türkçe, Kırımtatarca, Tatarca, Türk Dilleri

ACKNOWLEDGEMENT

I would like to express my deep gratitude to my supervisor Dr. İlyas Çiçekli for his guidance and suggestions throughout the development of this thesis. I feel lucky for having worked with him.

I am also indebted to Dr. Özgür Ulusoy and Dr. Attila Gürsoy for showing keen interest to the subject matter and accepting to read and review this thesis.

I would like to thank Dr. Zuhal Yüksel, Dr. Hakan Kırımlı and İsmet Yüksel for their moral support and encouragement and their supplying the necessary material to be able to work on this thesis. Without their support, this thesis would not be possible.

My biggest gratitude is to my family. I am grateful to my parents and to my brothers for their infinite help throughout my life. The help and friendship my brother Erdal provided during my studies is invaluable. I thank my wonderful fiancee Hümeyra who always supported me during all the difficult times of my study.

Qırımtatar Halqınıñ Adsız Qaramanlarına

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Overview

People use language as a communication tool. Every people need a language to interact with others. This may be in the form of speech or a written document that is necessary to be read. Sometimes it is not possible to communicate since people do not know each other's language. In those cases, a person or a tool is needed to translate the source language material into the target language so that it is intelligible.

Traditionally, human translators helped people to understand written documents and speech in a foreign language. However, it is not always possible to find a human translator, who can do the job for us. Also, the amount of written material that one person can translate in unit time is very limited. The translation process is time consuming especially when we need an accurate and diplomatic copy of the document in the target language. Moreover, having a human translator is costly. For this reason, people and companies are in the search of finding alternative methods for the translation process.

Using computers for machine translation proposes a solution for this costly process. Machine translation aims to reduce the cost of the translation process. The quality of the translation depends on the system, the languages and the domain of the texts, however any machine translation system helps human translators. Even a system that can give a rough translation of the source text may be helpful, in the sense that it helps to eliminate unrelated material. Most of the time, human translators make a draft translation and it is checked a second time for grammar and vocabulary details. A machine translation system can be put in the place of the first translator.

Most of the time, at MT research, people have worked on western languages such as English and French. When other languages are included, again most of the research has been trying to translate from or to English. Machine translation between close pair of languages was left rather untouched and Turkish and Turkic languages have not attracted any attention.

This thesis tries to develop some methods for translation between closely related languages, which we believe, is needed to construct language domains that will make the translation process from other languages possible. Developing such a system is easier than developing independent systems between language pairs. Also, the process by nature will take some of the issues like word order and most of the time the meaning out of the scene, so the research can focus on other issues like the translation of grammar.

Turkish and Crimean Tatar may be a model for machine translation between closely related languages. Methods developed for this pair of languages can easily be applied to other Turkic languages. Also, similar research on language pairs Czech-Slovak [6] and Spanish-Catalan [7] shows that the methods described in this thesis are applicable to other closely related language pairs.

1.2. Machine Translation

As soon as the emergence of the computers, the idea of using them in the automatic translation process gained attention. At the beginning, people thought a message that was written in a foreign language as having originally been written in their own language, in an encrypted form. The translation process was a process of decrypting the encrypted message. However, they soon realised that it is much more complicated than just deciphering [1].

The serious research on machine translation began in 1950's. The first aim of the research was being able to translate Russian sentences into English, namely aiming political and military purposes. Throughout 1950's and 1960's, many research groups were initiated in all parts of the world, especially in the US and the USSR.

In 1964, the government sponsors of MT in the United States formed the Automatic Language Processing Advisory Committee (ALPAC) to examine the prospects. In the famous 1966 report, ALPAC concluded that MT was slower, less accurate and twice as expensive as human translation and that "there is no immediate or predictable prospect of useful machine translation" [2]. The effects of this report were very deep and it brought a virtual end to the MT research in the US for over a decade.

While the focus of research in the United States was on Russian to English, and in the USSR on English to Russian, the need and the problems in Europe and in Canada were different. Canada, being a bilingual country, needed the copies of official documents both in English and in French. Similarly, in European Community countries, the need was translating scientific, technical, administrative and legal documents from and into all the Community languages. Thus, the research activities switched from US to Europe and Canada.

In Canada, the first successful machine translation system, METEO, was developed and became operational in 1976. This system was specifically developed for translating weather reports from English to French every day. The language used in these reports, both in terms of vocabulary and grammar, was very limited and the METEO system was successfully used.

Throughout 1970's, the research focused on interlingua approaches and several systems developed using this idea. However, the results were not very promising and the direct transfer method from one language to another gained more popularity.

With 1980's, some successful products started to appear both in the US and in Europe. At the same time, Japanese researchers introduced many products, using a variety of methods and capable of translating into and from Japanese, Korean, Chinese and some other languages.

During the first half of 1980's, the main focus was on transfer-based systems generally with a restricted domain and language. In the second half, the idea of using an interlingua again gained importance.

From the beginning of 1990's, the use of corpus for statistical learning came to the scene. First a group at IBM published results for a pure statistical system and others followed them. Many systems, using statistical methods and a combination of statistical methods with others, were developed.

1.2.1. Methods Used in Machine Translation

The methods used in machine translation can be grouped into four:

1. Direct Translation
2. Transfer Based Approach
3. Interlingua Approach
4. Statistical Methods

1.2.1.1. Direct Translation

The first method used in machine translation was directly giving the meanings of words in the target language. At first sight, this may seem to be working; however, an ordinary word in a dictionary has more than one meaning. This is more dramatic for very common words. In general, we can say that the more common a word is, the more it has entries in a dictionary. Meanings of most of the words can be understood from the context in which they appear. Let us consider the word 'book' in the following two sentences:

I bought a *book* yesterday.
I asked him to *book* a room for us.

Although the word 'book' has the same format in the two sentences, the type of information it carries is different, so is the meaning. In the first one, it is a noun used instead of "a set of written, printed, or blank pages fastened along one side and encased between protective covers". In the second one, it is the name of an act, namely "to arrange for in advance; reserve". Without considering the context information, it is not possible to correctly translate this word into another language.

Moreover, the order of the words in one language may not be, and usually is not, the same as the order in another. Some of the languages have Subject-Verb-Object form such as English and German. Some have Subject-Object-Verb order such as Turkish and Finnish and some other have Verb-Subject-Object form such as Arabic and Hebrew. Directly translating between languages in different groups may cause serious misunderstandings. Even there may be variances among languages in the same group. The relative position of adjective compared to noun or the prepositions may differ from language to language.

Another problem is that, some languages are agglutinative and others are not. Languages like Turkish and Finnish are called agglutinative languages and meaning is added to a sentence by adding different morphemes to one or more words. Namely, more than one words, sometimes a whole sentence in a language like English may correspond to a single word in languages like Turkish. Also, a word must be in accordance with the other words in the sense of sex, number, case etc.

In conclusion, a direct translation of words may have some meaning only in certain restricted situations, but it is not useful most of the time. Thus it is not a preferred as a method of translation.

The most famous machine translation system using direct translation technique is SYSTRAN [23, 24].

1.2.1.2. Transfer Based Approach

In order to overcome the problems of the direct translation method, the source text can be analysed to some extent depending on the language pair, the analysed text can be transferred to a representation of the target language and the target text can be generated from this transferred representation. This method is called transfer method and can be seen in Figure 1.

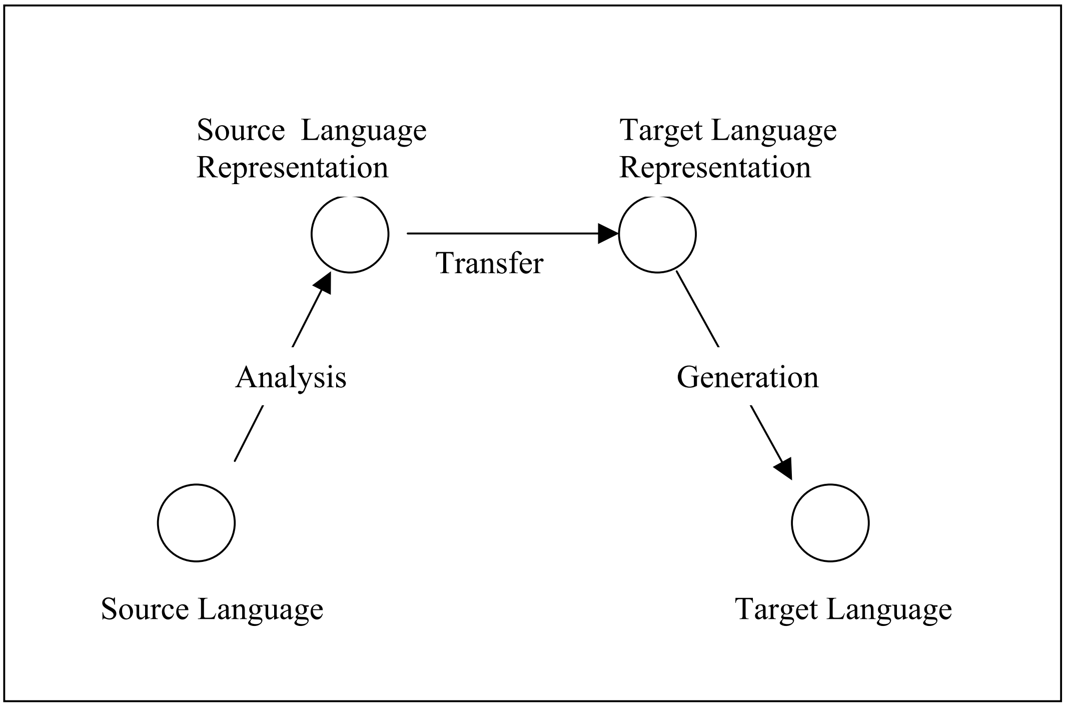
Source Language Representation
Target Language Representation
Transfer
Analysis
Generation
Source Language
Target Language

Figure 1. Transfer Based Translation

In the analysis process, different tools are used to get the most possible meaning from the source. Morphological analysis, syntactic analysis and even some semantic analysis may be necessary. Then the hand coded transfer rules are applied to this analysed text and it is transformed into some representation of the target text. The last procedure of this process is the generation of the target text.

The amount of analysis depends on the language pairs worked on. For languages belonging to very far language groups, like Turkish and English, a deeper analysis at the morphological, syntactic and semantic levels may be necessary. While translating from Turkish to English for example, a morphological analysis will extract the root, subject, tense and case from a single word which are all represented by separate words in English. Part of speech information must be determined and transfer rules that will map Turkish roots to English roots must be applied. Also rules for reordering the words of the sentence must be used since word order in Turkish and English are not the same. In the last step, English text must be generated.

The transfer-based systems have been successful, and many of the commercial systems used this approach. The main drawback of this approach is the difficulty of determining and coding the transfer rules. Since all the rules are hand coded, it requires a time consuming work. The person or the members of the team must have extensive knowledge on the system, as well as the source and target languages.

The performance of this approach is best when the languages are close to each other [1]. Since the languages are similar in structure, the word order and part of speech information, the number of rules required to transfer from one language to the other is limited. Most of the time, even the ambiguities are preserved from one language to the other. Phrases and word order usually are not changed. However, most of the transfer-based systems translate from and to English.

The MT systems GETA [26], SUSY [25] and many other systems use transfer-based machine translation.

1.2.1.3. Interlingua Approach

Sometimes, there is need for a system which has to translate among many languages. In European Union for example, many documents are to be translated into many languages at a time. Developing independent translation systems between language pairs is an expensive task. Having this idea in mind, people came up with the idea of using an interlingua for the system.

In the interlingua approach, the source text is translated into a language that is capable of representing the meaning of all languages. From this interlingua representation, the target text can be generated in any of the languages the system can generate. The interlingua systems are usually supported by a knowledge base in order to analyse the source more accurately. Every detail of the source text must be captured because not only syntactic information but also the meaning is represented in the interlingua representation.

The major advantage of an interlingua system is its decreasing the effort needed for a multilingual system. The amount of the effort needed for an *n* language system in the transfer based approach is *n(n-1)* since an independent system is to be developed for each language pair and translations must work in both directions for this language pair. However, when an intermediate language is used, in order to add a new language to the system, only a two-way translation program to and from the interlingua will suffice. Thus, for an *n* language system, *2n* translation will be enough.

However, there is not an interlingua in hand, which covers all the world languages. Capturing meaning is dependent on the world knowledge and most of the time needs ontology. The words and phrases may mean different in different cultures and some words and concepts may be totally inexistent in some languages. The interlingua must consider all these and an ideal interlingua must be compatible with all the world languages, at least with those covered by the system. Usually, developing an interlingua which covers all these aspects is almost impossible and adding a new language is most of the time not just a matter of adding a two-way translation program translating from and to the interlingua. The system designer must ensure that all the other programs are working fine with the added system.

Among interlingua systems, the followings are noteworthy: Rosetta [27], KBMT [28]

1.2.1.4. Statistical Approach

Statistical approach to machine translation is to translate a text using the information automatically learned from previously translated texts. For this purpose, large corpora of the source and target languages are needed. The translation rules, the dictionary and context information for each word can be derived from a sufficiently large corpus.

In the statistical process, usually the training corpora are given to the system to train and prepare it for the actual translation. Two types of alignment are necessary. First is the sentence alignment, that is the alignment of the bilingual texts at the sentence level. The second is word alignment, which is the alignment of each source word in the target language. The system learns from this aligned corpus how to translate words, phrases and grammar rules. The frequency of each word pair appearing together can be derived from this corpus and the rules can be applied to the actual text. The results of the translation are usually added to the training corpus and the system performance is tried to be improved.

Statistical systems usually do not use linguistics information and mainly focus on only the information gathered from the sequence of words. During the translation process, depending on the model used, each translation is assigned a score and the translation, which has the highest score, is assigned a higher priority. However, different methods may return with different scores for the same word or phrase. Since usually no linguistic information is employed within the process, wrong results may be returned just because they get a higher score due to defects in the training corpus or in the method used.

Statistical methods usually return with acceptable results provided that a sufficient training corpus is present. However, it is usually not present, especially for lesser-studied languages like Turkish [3, 31]. Even when a relatively large bilingual corpus is present, it is rarely aligned at the sentence and word levels and as a raw text, it cannot be used. Aligning the corpus is a time consuming and tiring job, which must be done by hand by those who know the both languages well.

The most popular work for statistical machine translation belongs to the researchers at IBM [29, 30].

1.3. Machine Translation Between Closely Related Languages

Translation is a hard job due to various reasons. First of all, different societies have different cultures. The concepts that each society has in mind and the names that they give to objects and abstract concepts may be different. For example, the Hebrew "adonai roi (The Lord is my shepherd)" cannot be translated to a language of a culture that has no sheep [4, p.819].

All languages in the world are claimed to be equally complex. Some may have simpler syntax, but they have more complex phonology and morphology to compensate this [5, p.9]. Some may not have certain grammatical structures that are present in the target language. For example, Turkish does not have an explicit perfect tense construct and translation of perfect tense from English to Turkish may cause some problems. Another problem with translation is the ambiguity. Since one word may have many meanings, the process of choosing the correct sense among the alternatives is not an easy task.

However, for languages that are very close to each other, some of these problems are not present. These kinds of languages are almost always the languages of people who have a similar culture and somewhere in the history they have the same roots. Russian and Ukrainian are very close languages and the historical roots of these two people are same. Turkish and Crimean Tatar are two Turkic languages, which throughout the history had great interaction.

Cultural differences between people speaking closer languages are not very significant most of the time. Even when they have different cultures and concepts, the concepts of the other culture is present in the language since they have great interaction. Also when the two languages are closer to each other, the grammatical differences and inexistence of some words are limited. Ambiguities are usually preserved in the two languages. For example, in the sentence “John saw the girl with binoculars”, the part ‘with the binoculars’ is ambiguous since it may belong to John or the girl. This may be a problem while translating this sentence into Turkish. However, the ambiguity is preserved in French and it is not a problem for a translation into French [4, p.807]. As a result, the closer the languages of people, the easier to make translation between them.

People usually have worked on translation systems for languages that are not directly related. However, translation of closely related languages is also very important. First of all, the research for translation between similar languages will contribute a lot to the overall machine translation techniques. Since the structures of the languages are similar, many features of the two languages may be ignored. For example, Turkish is a free word order language whereas English is more strict in the word order. In the translation process from Turkish to English, we have to consider the word order. On the other hand, the translation from Turkish to Kazakh, which is also a free word order language, would usually not require consideration of word order. Thus research may focus on other features of translation process.

Another advantage of translation between closely related languages is its creating a domain of interchangeable languages. In other words, having a system that is capable of successfully translating between Russian and Ukrainian, any machine translation system from English to Russian will also enable us to translate from English to Ukrainian. Implementing a system translating from Russian to Ukrainian is easier than developing a system translating from English to Ukrainian. So, with lesser effort, we can have a system that is capable of translating from English to several Slavic languages.

These are also applicable to Turkish and Turkic languages which are close relatives of each other. The grammars for Turkish, Crimean Tatar, Kazan Tatar, Azeri, Kazakh, Kirgiz, Uzbek and other Turkic languages have many intersections and the vocabularies have many words in common. The sentence structure and part of speech information is often preserved in a translation. Most of the time, the translation is word-for-word translation. Many times, the ambiguities in one language are preserved in others.

Turkish and Crimean Tatar, being one of the closest pairs of Turkic languages, may be a model for translation between Turkic languages and between any pair of close languages. They have most parts of their grammar in common although morphemes and expressions may differ. For example, the narrative morpheme is –miş for Turkish and –gen for Crimean Tatar. The use of narration in both languages is almost the same and a narration can directly be translated. But it is not straightforward to translate some phrases, idioms and even some grammatical structures.

1.4. Machine Translation Process for Closely Related Languages

As stated above, translation between closely related languages is easier than translation between languages belonging to different language families. Most of the time, a semantic analysis is not required and a lexical analysis supported by some translation rules may be sufficient. The number of translation rules or at least the groups of translation rules are much lesser than those of translation between unrelated languages are. Thus, hand coding the rules is easier.

We can summarise the translation process as follows:

- **Morphological analysis of the source text**
  The words in a language are composed of morphemes, the smallest meaningful units that cannot be divided further. In some languages like English, the words themselves are the morphemes and suffix and prefix morphemes are rare compared to many other languages. For example, the word “man” expresses a noun that is in third person singular and a suffix –ed comes after a verb is a morpheme that gives a past or past participle meaning. In agglutinative languages like Turkish, words are enriched through morphemes and each morpheme usually has a single meaning. For example,

the word “gelmiştik” (we had come) is composed of four morphemes each has a single meaning: gel(Verb:come)+miş(Narrative)+ti(Past)+k(1stPersonPlural). In languages like Russian, a single morpheme may have several meanings at the same time. The suffix –om in “Borisom” (with Boris) expresses a masculine, singular noun in instrumental case. The morphological analysis process seeks for all meanings of a word based on its morphemes.

- **Disambiguation**

  Sometimes a word may have more than one meaning when analysed independent of its context. Usually, only one of the possible analyses is true in a given context. For example, for the Turkish word “bilen”, there are two possible analyses: “the one who knows” and “get yourself sharpened/support your hatred”. Using a syntactic analysis, the first one can be selected as a more probable alternative if the following word is a verb or a noun. In a semantic analysis, if the context is a war, a fight or a struggle, the second meaning is most probable. The disambiguation process tries to select the most probable analysis in a given context.

- **Translation of grammatical rules and context dependent structures**

  Even when the languages are close to each other, there may be some differences in the grammars. For example the order of appearance, the relative positions of adjectives, construction of some phrases between Czech and Polish are different [6]. Some words may be translated into different words depending on the context. Turkish “durmak” may be translated to Crimean Tatar and other Kipchak languages as “turmaq” when it means, “stay in a situation/position” as in “gelip duruyor” (he continuously comes). It should be translated as “toqtamaq” when it means to stop as in “araba durdu” (the car stopped). The previous and following words may determine how to translate a word when the context information is important.

- **Translation of domain specific structures using a bilingual dictionary**

  Certain words may have a different usage when they are used in a certain domain. Many of the everyday words may have different meanings when they are used in a technical context. The phrase “hand shaking” expresses two persons holding each other’s hands in everyday context whereas it means the communication of two computers when the context is computer networks.

- **Translation of the roots using a bilingual dictionary**
  Roots have to be translated from the source language to the target language. The other morphemes are translated in the other steps including generation and the roots should be translated before the target morphological representation is tried to be generated into everyday spelling.

- **Generation of the target text**
  The representation after the translation process is in an internal format depending on the system. This internal format (lexical form) must be replaced with a corresponding everyday representation (surface form).

Any differences between languages can be dealt within the disambiguation and translation stages. For example, in [6], it is said that in some Slavic languages part of speech ambiguities are more common whereas in others ambiguity of gender, number and case is more frequent. Then, they claim, without the analysis of noun phrases, it is hard to make the translation right. Alternatively, a morphological disambiguator can be used to overcome the problem.

Another system translating between Catalan and Spanish, developed at Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, Spain, basically uses the same methodology [7]. They use a morphological analyser, a part of speech tagger, a pattern matching module, a morphological generator and a post processor respectively, and they claim that they get successful results.

1.5. Finite State Techniques in Machine Translation

Successful applications of finite state techniques in various areas of natural language processing have already been done [8]. Among the machine translation methods mentioned above, morphological analysis and the translation mechanisms are interesting to us.

Finite state transducers read their input symbol by symbol and each time they read a symbol, they give a corresponding output and move to a new state. This improves the processing speed fundamentally. Practically, the processing speed is independent of the size of the rules [7].

Morphological analysis can be considered as a finite state process. Each word in a language is composed of a root and possible morphemes affixed to that root. The finite state transducer takes a word in and checks all possible roots and morphemes affixed to that root. Many previously determined rules work in parallel and they check the possibilities at any state. If all of the rules accept the input, then the input is accepted. Finite state morphological analysers for many languages including Finnish, Swedish, Russian, English, Swahili, Turkish and Arabic have been developed [9]. For more compact information on finite state morphological analysis process, see [9, 10, 11].

Apart from the morphological analysis process, large dictionaries can successfully be stored in finite state transducer [8]. Maohri gives the experimental results for a large finite state dictionaries and claims that it is efficient both in the sense of time and space. Since many words have their first few characters in common, they share the same path in automata. As a result, the storage required for a dictionary structure may be less than storing each word separately. Figure 2 presents a basic English-Turkish dictionary structure using FST.

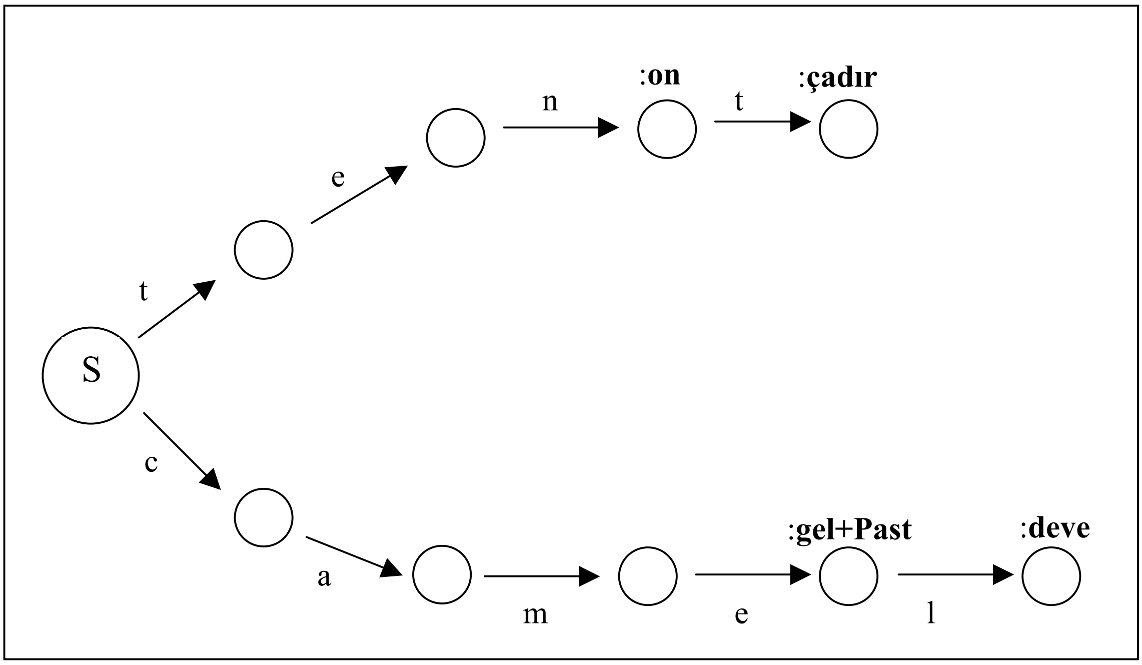

Figure 2. An English-Turkish Dictionary Structure using FST

Similarly, translation rules may be operated in parallel or in a certain sequence, and input may efficiently be transformed to a new form. Parallel operation of transformation rules, especially when the number of rules are high, will be more efficient than a procedural approach, where probably each rule will have to check the input several times, although most of them are not employed at each run. In case of rule interferences, the order of rules may be adjusted so that they do not make unnecessary changes.

1.6. Layout of the Thesis

The organisation of the thesis is as follows. In Chapter 2, we make a comparison of the Turkish and Crimean Tatar morphologies and grammars. In Chapter 3, we give the details of the translation system. In Chapter 4, the details of Crimean Tatar morphological processor are given. Experimental results and evaluation are given in Chapter 5. In Chapter 6, the weak points of the system with the reasons and possible improvements are discussed and the conclusion is given.

Chapter 2

Comparison of Turkish and Crimean Tatar

2.1. Introduction

Being two Turkic languages, Crimean Tatar and Turkish have many parts of their grammars and vocabularies in common. Close relations between Crimea and the Ottoman Empire helped the interaction between the languages of two peoples. Crimean Tatar, originally being a Kipchak oriented language, in time, gained many Oghuz properties. Vocabulary of Crimean Tatar derived many words from Turkish. As a result, it became a transition language between Turkish and Kipchak languages like Kazakh and Kirgiz.

There are three main dialects of Crimean Tatar. Northern dialect, which is called “çöl şivesi” (steppe dialect) in Crimean Tatar, shows much more Kipchak properties and is close to Kazakh and Kirgiz. The central dialect is called “Bahçesaray Şivesi” referencing Bahçesaray, the capital city of Crimean Khanate and is the basic literary dialect. The southern dialect is “Yalıboyu şivesi” (coastal dialect) and is very close to Anatolian Turkish [21].

In this thesis, we implemented the system compatible with Bahçesaray dialect since it is the literary language. Throughout the thesis, the term Crimean Tatar means "Bahçesaray dialect of Crimean Tatar language" and the term Turkish means "literary Turkish language spoken in Turkey".

Most of the root words in Crimean Tatar are common with Turkish [12, 22]. However, today, the differences both in roots and in grammatical rules are not negligible. Many words, especially in northern dialect, are completely different from Anatolian Turkish.

Azbar : avlu (yard)
Kökrek : göğüs (chest)
Yengil : hafif (light)

Many words are present in both languages, however they mean different:

"Taşlamaq" in Crimean Tatar means "to leave something at somewhere", however in Turkish "taşlamak" means "to stone".

"Salmaq" in Crimean Tatar means "to put or add", and "salmak" in Turkish means "to let something go".

There are many variances between the grammars of Turkish and Crimean Tatar. For example, the second tense of a verb is written as a separate word in Crimean Tatar while it is joined to the root in Turkish. Also, the narrative suffix in Crimean Tatar is –gen or its equivalents according to harmony rules, while narration is expressed with –miş or with its equivalence class in Turkish. For example "kelgen edi" is written as "gelmişti" (he had come) in Turkish. The present progressive tense in Crimean Tatar many times can correspond to simple present tense of Turkish. For example, the sentence in Turkish "Buraya sık sık gelir" (He often comes here), which is in simple present tense, can be translated to "Mında sıqlıqnen kele", which is in present progressive tense.

Living under Russian rule for more than two centuries, the effects of Russian is heavily felt over Crimean Tatars. Not only there are many words derived from Russian, sometimes even Russian grammatical rules are applied to Crimean Tatar words. However, since these are not valid structures for Crimean Tatar, they have not been considered for the system we developed. Words, especially related to technology and usually the counterparts of Turkish words that come from western languages, are mostly derived from Russian. Some examples are:

Televizor	: televizyon (television)
Avtobus	: otobüs (bus)
Peçqa (peçka)	: soba (stove)

The following sections describe the Crimean Tatar morphology and the grammar compared to Turkish in a more detailed and structured form.

2.2. Crimean Tatar Morphology

Being two closely related Turkic languages, Crimean Tatar and Turkish have most parts in common. The word order and the duties of words in the sentence are most of the time similar. The roots are usually similar, but sometimes they may have different meanings in the two languages. For example the word “kaldırmak” means “to lift” in Turkish, whereas it means “to leave something at somewhere” in Crimean Tatar context.

The actual spelling of a word is called *surface form* or *surface level*, and the representation that is a concatenation of morphemes, the smallest units of meaning in a language, making up the word is called *lexical form* or *lexical level*. The lexical form of a word in the following tables given as a set of morphemes separated by plus signs (+). For the Turkish word “bakıyor”, the surface form is “bakıyor” and the lexical form is “bak+Hyor”. The capital letters in the lexical forms are special characters representing a set of consonants or vowels that can be realised according to Turkish vowel and consonant harmony rules. For a detailed explanation of lexical forms see Section 4.2.

The differences between Turkish and Crimean Tatar are usually in morphemes rather than in deeper levels of grammar. In other words, different morphemes are used to get the same meaning.

The following sections present a tabular comparison of the two grammars. This is not a complete analysis, but rather a comparison to give some idea about Crimean Tatar grammar. It covers main aspects of Turkic languages. Details of Crimean Tatar grammar are explained in [12, 13, 14, 15].

2.3. Alphabet

Crimean Tatar used to be written in the Arabic based alphabet up to the first quarter of twentieth century. After the establishment of the Soviet rule, first a Latin based alphabet was used and then Crimean Tatars were forced to use the Cyrillic alphabet. During the Soviet period, everything was printed in the Cyrillic alphabet. After the collapse of the Soviet Union, a Latin based alphabet was accepted by Crimean Tatar National Assembly. Now both Cyrillic and Latin alphabets are used. Newspapers and journals today are printed in both alphabets.

The current alphabet is the same as Turkish alphabet. There are three letters that differ: â, ñ, q. The letter â is a sound that is between a and e as in lâle (tulip), kâğıt (paper). The letter ñ is for nasal n and is the counterpart of Ottoman character, “nûn-ı türkî”. It is mostly used in the second person morpheme such as kelesiñ (geliyorsun – you are coming), köyüñiz (köyünüz – your village) and in words such as deñiz (sea), sıñır (boundary). The last of these letters is used for Turkish k, however it is always paired with back vowels : qalmaq (kalmak – to stay), qurultay (kurultay – meeting), qaysı (hangi – which).

2.4. Tenses

All the tenses present in Turkish are also present in Crimean Tatar. The usages of the tenses are almost the same. In the tables below, the left column gives a brief structure for Crimean Tatar and the right column for Turkish. In those cases, the Turkish and Crimean Tatar examples correspond to each other. The first line of each explanation gives the lexical morpheme and the second line is the corresponding surface morphemes. The third line, if present, is for necessary explanations.

Present progressive tense in Turkish is constructed with –(ı/i/u/ü)yor. When the last letter of the previous syllable is a vowel, the vowel before –yor is omitted. When the previous syllable ends in a consonant, a vowel before –yor is inserted according to vowel harmony rules of Turkish.

Telefon *çalıyor*. (The telephone is ringing)
Birazdan *geliyorum*. (I am coming in a second)

In Crimean Tatar, present progressive tense is constructed with –a or –e when the root ends in a consonant and with –y when it ends in a vowel.

Endi mında *kele*. (He is coming here now)
Apaylar çamaşır *yuvalar*. (Women are washing the clothes)
Yabancı adamlar sizniñ eviñizni *soraylar*. (Strangers are asking for your home)

Table 1 gives a comparison of Turkish and Crimean Tatar present progressive tenses.

Narration in Turkish is done with –miş/mış/muş/müş according to vowel harmony rules.

Adam *ölmüş*. (The man died)
Misafirler *gelmiş*. (The guests have arrived)

In Crimean Tatar, narration is represented by –KAn and done with –gen/ğan/ken/qan according to vowel and consonant harmony rules.

Bağçasına gül *saçqan*. (He planted roses in his garden)

Kelgen ketkenlerden bizni *soraǧan*. (He asked for us from those who came back and forth)

A comparative explanation of narration in these two languages is given in Table 2.

Crimean Tatar		**Turkish**	
-A -a/e/y -vowel harmony rules apply -can correspond to English simple present tense and present progressive tense	baq + A = baqa kel + A = kele sora + A = soray	-Hyor -(ı/i/u/ü)yor -the suffix -yor does not coincide with vowel harmony rules	bak + Hyor = bakıyor (s/he is looking) gel + Hyor = geliyor (s/he is coming) sor + Hyor = soruyor (s/he is asking)

Table 1. Present Progressive Tense in Turkish and Crimean Tatar

Crimean Tatar		Turkish	
-Kan -gen/ken/Gan/qan -vowel and consonant harmony rules apply	kel + KAn = kelgen tOk + KAn = tOkken sora + KAn = soraGan saC + KAn = saCqan	-mHş -mış/miş/muş/müş	gel + mHş = gelmiş (s/he came) dök + mHş = dökmüş (s/he poured) sor + mHş = sormuş (s/he asked) ek + mHş = ekmiş (s/he planted)

Table 2. Narrative in Turkish and Crimean Tatar

While forming future tense in Turkish, the consonant 'y' is inserted before –acak/ecek if the previous syllable ends in a vowel. Otherwise –acak or –ecek is used according to vowel harmony rules.

Birazdan güneş *doğacak*. (The sun will rise soon)

Susuzluktan ağaçlar *kuruyacak*. (The trees will die because of lack of water)

In Crimean Tatar, –acaq or –ecek is used if the previous character is a consonant, and –ycaq or –ycek is used according to vowel harmony when the previous letter is a vowel.

İstanbuldan Anqarağace *yürecekler*. (They will walk from Istanbul to Ankara)

Toplaşuvda bir çıqış *yapacaq*. (He will make a speech at the meeting)

Baladan adını *soraycaq* ola. (He wants to ask the child his name)

Table 3 compares the formation of future tense in Turkish and in Crimean Tatar.

Crimean Tatar		**Turkish**	
-AcAK	al + AcAK = alacaq	-yAcAk	al + yAcAk = alacak
-acaq/ecek/ycaq/ycek		-(y)acak/(y)ecek	(s/he will take)
-vowel harmony rules apply	kOr + AcAK = kOrecek		gör + yAcAk = görecek (s/he will see)
	sora + AcAK = soraycaq		sor + yAcAk = soracak (s/he will ask)
	tile + AcAK = tileycek		dile + yAcAk = dileyecek (s/he will wish)

Table 3. Future Tense in Turkish and Crimean Tatar

2.5. Compound Tenses

In Crimean Tatar, the second tense that comes to the root is not joined with the root, but written separately with the verb "emek". In Turkic languages, the second tense normally is past or narrative. So, the second tense in Crimean Tatar comes after the root as "edi" or "eken". There is an exceptional case here for vowel-consonant harmony rules, which say that narrative suffix that comes after a vowel is written as –gen. However, here it is written as –ken. The person suffixes are added to the second tense, rather than the root whereas passive and causative suffixes are added to the root as explained in Table 4.

Crimean Tatar		Turkish	
-edi -edi -for past tense as second tense	yazGan ediñ	-DH -dı/di/du/dü/tı/ti/tu/tü -for past tense as second tense -it is joined to the root	yazmıştın (you had written)
-eken -eken -for narrative as the second tense	yapacaq eken	-mHş -mış/miş/muş/müş -narrative as second tense	yapacakmış (s/he would have done it)

Table 4. Compound Tenses in Turkish and Crimean Tatar

2.6. Cases

Although the meaning given by the case suffixes is the same as Turkish, the suffixes themselves and formation rules are different from Turkish.

Accusative case marker in Crimean Tatar is –nı/ni and there is no –nu/nü form. The sound "n" is a part of the morpheme and it is always written and said even if it follows a syllable ending in a consonant.

Kitapnı oqudı. (He read the book)

Aqçanı körmeden iş yapmaz edi. (He did not work without seeing the money first)

Dative case is represented by –KA and realised as –ge/ğa/ke/qa according to vowel and consonant harmony rules.

Qolundaki cevizlerni *balağa* berdi. (He gave the nuts in his hand to the child)

Köyge yaqınlaşqanda ağlap başladı. (When they approached the village, she started to cry)

Genitive case is constructed with –nıñ/niñ and as in the accusative case, the sound "n" is never dropped. Also there is no corresponding –nuñ/nüñ morpheme.

Ametniñ qalemi pek balaban. (Ahmet's pencil is very big)

Köyümizniñ ocası yoq. (Our village does not have a teacher)

Instrumental case in Crimean Tatar is done with –nen and there is no –nan morpheme. The sound "n" again is not dropped.

Samalyötnen kelgenler. (They came by plane)

Mambetnen kettik. (We went with Mambet)

Tables 5, 6, 7 and 8 explain the formation of accusative, dative, genitive and instrumental cases respectively.

Crimean Tatar		**Turkish**	
-nM	ev + nM = evni	-yH or nH	ev + yH = evi
-nı/ni		-(y)ı/(y)i/(y)u/(y)ü	(the house [Acc])
-no corresponding –nu/nü	qol + nM = qolnI	-(n)ı/(n)i/(n)u/(n)ü	kol + yH = kolu
-the sound n is the part of the morpheme and is never dropped	baca + nM = bacanI	-the sounds y and n joining sounds and can be dropped if morpheme follows a root ending in consonant	(the arm [Acc]) baca + yH = bacayı (the chimney [Acc])
-vowel harmony rules apply		-the vowel harmony rules for Turkish apply	

Table 5. Accusative Case in Turkish and Crimean Tatar

Crimean Tatar		Turkish	
-KA -ge/ke/Ga/qa -vowel and consonant harmony rules apply	deñiz + KA = deñizge qoranta + KA = qorantağa kökrek + KA = kökrekke at + KA = atqa	-yA -(y)a/(y)e -if root ends in a consonant, the joining sound y is dropped	deniz + yA = denize (to the sea) aile + yA = aileye (to the family) göğüs + yA = göğüse (to the chest) at + yA = ata (to the horse)

Table 6. Dative Case in Turkish and Crimean Tatar

Crimean Tatar		Turkish	
-nMN -nıñ/niñ -no corresponding nuñ/nüñ -the sound n is the part of the morpheme and is never dropped	ev + nMN = evniN horaz + nMN = horaznIN quyu + nMN = quyunIN	-nHn -(n)ın/(n)in/(n)un/ (n)ün -the sound n is a joining sound and can be dropped if morpheme follows a root ending in consonant -the vowel harmony rules for Turkish apply	ev + nHn = evin (of the house) horoz + nHn = horozun (of the hen) kuyu + nHn = kuyunun (of the well)

Table 7. Genitive Case in Turkish and Crimean Tatar

Crimean Tatar		Turkish	
-nen -no -nan form is present -le/la is rarely used under Turkish influence	Amet + nen = Ametnen avtobus+nen= avtobusnen soqur + nen = soqurnen	-(y)le/(y)la -y is the joining sound and drops when the root ends in a consonant -vowel harmony rules apply	Ahmet + ylA = Ahmet'le (with Ahmet) otobüs + ylA = otobüsle (by bus) kör + ylA = körle (with the blind)

Table 8. Instrumental Case in Turkish and Crimean Tatar

2.7. Adjective Derivation

Adjective derivation with the narrative suffix is different from Turkish in structure. Two different structures in Turkish correspond to Crimean Tatar adjectives constructed with –KAn. The corresponding structures are explained in Table 9.

2.8. Comparison of Grammar Rules and Semantics

Although these two languages are in the same language group and very close to each other, there are some differences in their grammars. While having the same functionality, some morphemes may have different structure. For example, instrumental case marker in Turkish is –(y)la/(y)le whereas it is –nen in Crimean Tatar. A brief explanation of morphological differences is given in Section 2.2.

The use of tenses is almost the same in these two languages. Past, narrative and future tenses are used in the same way. Although there is a simple present tense in Crimean Tatar, the meaning expressed in simple present tense of Turkish, when the speaker is talking about a continuous action, is usually expressed in present progressive in Crimean

Crimean Tatar		**Turkish**	
-Kan -gen/ğan/ken/qan -one meaning is “something that has already happened”	Ol + KAn = Olgen sat + Hl + KAn = satılğan bit + KAn = bitken	-mHş -mış/miş/muş/müş -has the same meaning	öl + mHş = ölmüş (dead) sat + Hl + mHş = satılmış (sold) bit + mHş = bitmiş (finished)
-the second meaning is “something that is currently continuing”	çap + KAn = çapqan yür + KAn = yürgen	-yAn -(y)an/(y)en	koş + yAn = koşan (running) yürü + yAn = yürüyen (walking)

Table 9. Adjective Derivation in Turkish and Crimean Tatar

Tatar. The simple present tense is usually used for continuous actions of first person and for others, usage of present progressive is more common. For example, Turkish sentence “Bazen buraya *gelir*” (He sometimes comes here) in simple present tense can be translated as “Kimerde mında *kele*” in present progressive tense. Actually this property is also present in Turkish and the same meaning can be given in present progressive.

However, when the meaning expressed in simple present tense is a promise, intention, desire or guess for future actions, the simple present tense is used in both languages. Turkish sentence “Siz giderseniz, onlar da *gelirler*” (If you go, they will also come) is translated as “Siz ketseñiz, olar da *kelirler*”.

The compound tenses in Turkish are written jointly in the verb where it is separated with verb “emek” in Crimean Tatar. For example, “vermişti” (he had given) is translated as “bergen edi”.

Nouns and verbs in the two languages have the same structure and usage. However, the use of specific words may differ. The cases of objects of specific verbs are different in the two languages. For example, the object of verb “bakmak” (to look) is in dative case in Turkish: “*belgelere* bakmak*”* (to look at the documents. But the same verb is used with an accusative object in Crimean Tatar: “*dökümentlerni* baqmaq*”.* There is no rule for the cases of objects of verbs in Turkish and Turkic languages. Each verb is learned with the case of its objects. For example, the verb “bakmak” (to look) is used with dative case in Turkish as in the previous example. But the verb “görmek” (to see), which has almost the same meaning and refers to a similar action, is used with accusative case as in “*belgeleri* gördüm” (I saw the documents).

Some of the verbs that have their objects with different cases are shown in Table 10:

Turkish		**Crimean Tatar**	
Verb	**Case**	**Verb**	**Case**
bakmak (to look)	dative	baqmaq	accusative
vurmak (to hit)	dative	urmaq	accusative
sormak (to ask)	dative	soramaq	ablative
ısmarlamak (to order)	dative	sımarlamaq/ smarlamaq	ablative
acımak (to feel sorry for)	dative	acımaq	accusative
evlenmek (to get married)	instrumental	evlenmek	dative

Table 10. Case Changing Verbs in Turkish and Crimean Tatar

The use of past participle as adjective is different in two languages. In Turkish, the possessive information comes after the verb with past participle where in Crimean Tatar it comes after the noun. For example *“geldiğim köy”* (the village from which I came) is translated as *“kelgen köyüm”*. Notice that the past participle morpheme in Turkish is -dik and it is -gen in Crimean Tatar. The case marker in the noun is not lost since the possessive marker precedes the case marker. The phrase *“geldiğim köy**de**”* (in the village from which I came) is translated as *“kelgen köyüm**de**”*. However, when both the adjective and the noun have possessive markers, some information is lost. In the phrase *“geldiğim köyünüz”* (your village from which I came), the adjective has first person singular possessive marker and the noun has second person plural possessive marker. When it is translated into Crimean Tatar, the possessive information of the noun is lost and the phrase becomes *“kelgen köyüm”* (the village from which I came).

Question meaning in both languages is expressed with “mi” or its equivalent morpheme according to vowel harmony rules. In Turkish, this is written as a separate word as in “Uyudun *mu*?” (Did you sleep?), whereas in Crimean Tatar it is joined to the previous word: *“Yuqladıñmı?”* The relative position of the question morpheme in Turkic languages depends on what is questioned and what is emphasized. In the sentence “Ben *mi* geldim?” (Did *I* come?), the emphasis is on the person, I. However in “Ben geldim *mi*?” (Did I *come*?), the emphasis is on the action, to come.

In Turkish the suffix –(y)arak/(y)erek has the meaning “by doing so, with the way of, using as a means of”. The same suffix is not present in Crimean Tatar and the same meaning can be expressed with the suffix –(ı)p/(i)p/(u)p/(ü)p. In the following examples, the first of the sentence pairs is in Turkish and the second one is in Crimean Tatar.

> Antlaşmayı *imzalayarak* durumu kabullendi. (*By signing* the treaty, he admitted the situation)
> Añlaşmanı *imza etip* vaziyetni qabul etti.
>
> *Yürüyerek* geldi. (He came *by walking*).
> *Yürüp* keldi.

The suffix –(y)ıp/(y)ip/(y)up/(y)üp is present also in Turkish and is used to give the meaning “after doing so”. The same suffix is used in Crimean Tatar for “after doing so”

Kapıyı *örtüp* gel. (Come here *after closing* the door)

Qapını *yapıp* kel.

Positive ability in Turkish is expressed with the auxiliary verb –(y)abil/(y)ebil. Positive ability in Crimean Tatar is expressed with –(y)abil/(y)ebil and "–(i)p ol". Both forms are valid.

Okuyabiliyor. (He *can* read)

Oquyabile.

Bitirebilirse tatile cıkacağız. (If he *can* finish, we will go to a vacation)

Bitirip olsa raatlanmağa ketecekmiz.

Negative ability in Turkish is constructed with –(y)ama/(y)eme. The same meaning in Crimean Tatar, however, expressed with –"(i)p ol(a)ma" or –(a)lma.

Okuyamadı. (He could not read)

Oqup olamadı.

Ben burada *yaşayamadım*. (I could not live in this place)

Men bu yerde *yaşalmadım*.

The morpheme –ken in Turkish when comes after the aorist morpheme –ar/er/ır/ir gives the meaning "while". The aorist morpheme actually does not function to give aorist meaning and the time of event is understood from the main verb of the sentence. The same meaning in Crimean Tatar is usually expressed by past participle morpheme –gen with locative case. The aorist morpheme is lost. Sometimes, "-ır/ir/ar/er ekende" is also used.

Geçerken görmüştük. (While we were passing by, we had seen it)

Keçkende körgen edik.

Gelirken alacak. (He will buy it while he was coming).

Kelir ekende alacaq.

The singular-plural agreement in Turkish is not very strict. It is possible to have a plural adjective with a singular noun or a plural subject in a sentence with a singular verb.

However, the singular-plural agreement is more strict in Crimean Tatar. Adjectives and nouns must agree in number with each other. The agreement between subject and the verb is more common in Crimean Tatar.

Birkaç gün sonra geldi. (He came few days later)
Bir qaç künlerden soñ çıqıp kelir.

Toplantıya sınıfımızdaki *öğrenciler katıldı*. (The students in our class attended the meeting)
Toplaşuvğa sınıfımızdaki *talebeler qoşuldılar*.

The conditional situations in Turkish are constructed with –sa/se in all tenses.

Çalıştıysa başarır. (If he studied, he will pass)
Konuşacaksa hazırlanmalı. (If he will make a speech, he must prepare)
Duymamışsa gelmesine gerek yok. (If he had not heard, he does not need to come)

On the other hand, in Crimean Tatar different morphemes are used for conditionals in different tenses. Conditionals in past are constructed with –sa/se followed by "edi".

Bilse edim aytar edim. (If I knew, I would tell)
Körmese edi bile qıdırır edi. (Even if he had not seen, he would have looked for)

Conditionals in narrative in Crimean Tatar are constructed with –gen/ken/ğan/qan followed by "olsa".

Tapqan olsa qaytarır edi. (If he had found, he would have returned)
Oqumağan olsa laqırdı etmez edi. (If he had not read, he would not have talked about it)

Simple present conditionals are done with only –sa/se. The aorist morpheme –ar/er/ır/ir that is present in Turkish is lost.

Sorasa yoq dep aytıñız. (If he asks me, tell him that I am not here)

Future conditionals are constructed with "–acaq/ecek/ycaq/ycek olsa" or sometimes with only –sa/se.

Aşaycaq olsa aşatmañız. (If he wants to eat, do not let him eat)

Kelmeycek olsa haber etiñiz. (If he will not come, inform me about it)

Açlıqtan *ölsem* ballarıma saip çıqıñız. (If I die because of hunger, take care of my children).

Desire in Turkish is also constructed with –sa/se. In Crimean Tatar, desire is expressed with the –ğay/gey/qay/key morpheme.

O kitabı *alsaydım*. (If only I had bought that book)

O kitapnı *alğay* edim.

Yemeseydiniz. (You should not have eaten)

Aşamağaydıñız.

The morpheme –dikçe in Turkish has the meaning "as it happens" and expresses the continuous happening of an event. The corresponding Crimean Tatar structure is "–gen sayın".

Ağladıkça anlayacaksın. (As you cry, you will understand)

Ağlağan sayın añlaycaqsıñ.

Hatırladıkça aniden ağlamaya başlıyor. (As she remembers, she suddenly cries)

Esine *tüşken sayın* qıçırıp ala.

The structure –e/a kadar in Turkish has the meaning "until" or "up to that point". The same meaning in Crimean Tatar is expressed with –gece/ğace.

Çocuklar *eve kadar* koştu. (The children ran until the house)

Ballar *evgece* çapqaladılar.

Balıkçılar *adaya kadar* yüzdüler. (The fishermen swam until they reached the island)

Balıqçılar *adağace* yaldadılar.

The nouns of phrases of "başlamak" are written in dative case in Turkish. The same meaning is given with "–ip başlamaq" in Crimean Tatar.

Saat sekizde misafirler *gelmeye başladı* (The guests started to come at eight o'clock).
Saat sekizde qonaqlar *kelip başladılar*.

Yazla birlikte meyveler *olgunlaşmaya başladı*. (The fruits started to ripen with the summer)
Yaznen birge meyvalar *pişip başladı*.

Turkish idiomatic phrase "–esi gelmek" expresses some sense of desire and corresponds to English "feel like". This same phrase is not present in Crimean Tatar, but the same meaning is expressed with future participle as "–ecegi kelmek".

Meyveleri görünce *yiyesim geldi*. (When I saw the fruits, I felt like eating them)
Meyvalarnı körgende *aşaycağım keldi.*

Öğrencilerle sohbet edince *okuyasım geldi*. (When I chatted with the students, I felt like studying)
Talebelernen subetleşkende *oquycağım keldi*.

The time expression –ince gives the meaning "when, at the time of" in Turkish. Crimean Tatar counterpart of the same construction is noun in locative case done with past participle.

Sabah *kalkınca* yüzünü yıkadı. (When he got up in the morning, he washed his face)
Erten *turğanda* betini yuvdı.

Saat altıya kadar *gelmeyince* merak ettik. (We wondered when he did not come until six o'clock)
Saat altığace *kelmegende* meraqlandıq.

Instrumental case is sometimes written separately with the conjunction “ile” in Turkish. However, instrumental case has to be joined to the previous word as –nen in Crimean Tatar.

Ayşe *ile* birlikte yemek pişirdik. (We cooked the dish with Ayşe)
Ayşenen birge aşnı pişirdik.

Kalem *ile* yazmayı öğretti. (He taught how to write with a pen)
Qalemnen yazuvnı ögretti.

The meaning of “try to do so” in Turkish is given by “–maya çalışmak” and includes some intention with or without action. The same meaning is given in Crimean Tatar with “–acaq olmaq”.

Haberi duyunca *gelmeye calışmış*. (When he heard the news, he tried to come)
Haberni eşitkende *kelecek olğan*.

Anlatmaya çalıştım ama başaramadım. (I tried to tell but could not succeed)
Añlatacaq oldım amma beceralmadım.

For more information on Crimean Tatar grammar, see [12, 13, 14, 15].

Chapter 3

Translation System

3.1. Introduction

Translation from Turkish to Crimean Tatar is most of the time word-for-word translation. The grammars of the two languages are similar, and each morpheme has a corresponding morpheme with or without change. Finite state transducers, which can transfer the grammar differences, context dependent structures and roots, are most of the time sufficient. Ambiguities in Turkish are most of the time preserved in Crimean Tatar. For example, the word “gelecek” in Turkish has four morphological analyses, all of which are preserved in Crimean Tatar with the same representation.

Sometimes the translation of one word is dependent on the context in which it appears. For example, the word “durmak” (to stop) in Turkish is translated as “toqtamaq”. However, if it comes after a verb with the meaning “staying in a position of action” as in “bakıp durmak” (to stay in a staring position), it is translated as “turmaq”.

The steps of the translation process can be listed as follows:

- Morphological analysis of Turkish text
- Application of context dependent and grammatical translation rules
-

- General one-to-one translation of words.
- Morphological generation of Crimean Tatar text

After the input text is morphologically analysed, it needs to be disambiguated. Then the phrases and the context dependent structures of the disambiguated text are translated. Phrases that consist of more than one word and words that depend on the previous and following words must be translated before the roots in order not to lose the context information. In the following step, one-to-one translation of words is done using a bilingual dictionary between Turkish and Crimean Tatar. The morphological generation of the processed text is the last step. Figure 3 gives a description of the structure of the system.

3.2. Turkish Morphological Analyser

The first step of the translation process is the analysis of Turkish text by a morphological analyser, developed by Oflazer. The details of this system can be seen in [18]. Consider two typical Turkish words, “evlerimizden” (from our houses) and “gitmiştim”, analysed by the system:

evlerimizden
1. ev+Noun+A3pl+P1pl+Abl

gitmiştim
1. git+Verb+Pos+Narr+Past+A1sg

The analysis states that the root is ‘ev’ (house) and it is a noun (ev+Noun). The agreement of this noun is third person plural (A3pl), since the morpheme –ler appears in the word. Then the first person plural possessive morpheme (P1pl) –imiz comes and the case of the word is ablative (Abl) due to the morpheme –den.

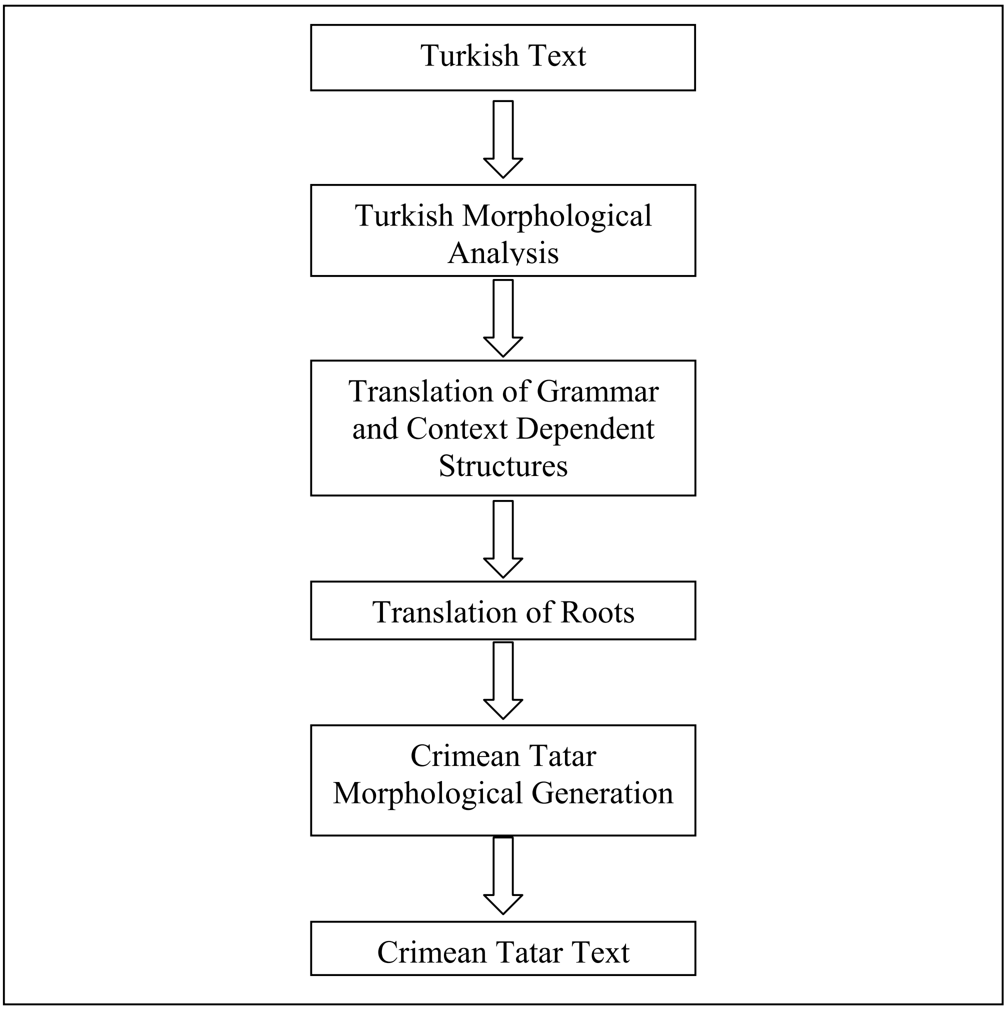
Turkish Text
Turkish Morphological Analysis
Translation of Grammar and Context Dependent Structures
Translation of Roots
Crimean Tatar Morphological Generation
Crimean Tatar Text

Figure 3. Structure of the Translation System

The root of the second word is ‘git’ and it is a verb in positive sense (git+Verb+Pos). The morpheme –miş shows us that the act is narrated (Narr) and the following –ti states that it was in the past (Past). The last –m shows it was done by the first person singular (A1sg).

Any input, entered to the system is first divided into its words and these words are analysed by the morphological analyser. For example, the input sentence “Annem gelecek .” is first divided into three words, “annem”, “gelecek” and “.”. Then these three words are analysed by the morphological analyser and the following results are returned:

annem

1. anne+Noun+A3sg+P1sg+Nom

gelecek
1. gel+Verb+Pos+Fut+A3sg
2. gel+Verb+Pos^DB+Adj+FutPart+Pnon
3. gelecek+Noun+A3sg+Pnon+Nom
4. gelecek+Adj

.
1. .+Punc

The words “annem” and “.” have only one analysis according to this morphological analyser. However, the word “gelecek” has four analyses, only one of which is true in this context.

After the morphological analysis is complete, the sentences are formed again with the analyses in order to save the context information. In many cases, the order in which the words appear is important and the meaning of the sentence is directly dependent on that order. When the words are ambiguous, all possible combinations with the words are generated. For example, if one of the words in a sentence has two morphological analyses and another word has three analyses, then 2x3=6 sentences are to be generated at the end of the analysis process. For the sentence above, the following four sentences are generated:

anne+Noun+A3sg+P1sg+Nom gel+Verb+Pos+Fut+A3sg .+Punc
anne+Noun+A3sg+P1sg+Nom gel+Verb+Pos^DB+Adj+FutPart+Pnon .+Punc
anne+Noun+A3sg+P1sg+Nom gelecek+Noun+A3sg+Pnon+Nom .+Punc
anne+Noun+A3sg+P1sg+Nom gelecek+Adj .+Punc

The main problem with the morphological analysis step is the ambiguity present in the language. In a typical Turkish text, almost one third of the words are ambiguous [3, p.61]. In order to be able to find the correct meaning of the word, it has to be disambiguated in the context it appears. Hakkani-Tur, in her PhD thesis [3] states that the system they developed has reached an accuracy of over 90% for average Turkish text. Our translation system however currently does not use a morphological disambiguator.

3.3. Translation of Grammar and Context Dependent Structures

As explained in the previous chapter, most parts of the Turkish and Crimean Tatar grammars are similar. However, there are also some differences. When the differences are at the morpheme level, the translation part leaves them to the morphological generator. For example, the narration morpheme in Turkish is –miş and it is transformed to '+Narr' after the morphological analysis. When this morpheme is fed to the Crimean Tatar morphological generator, it then generates +gen/ken/qan/ğan according to the vowel and consonant harmony rules. Thus, simple morpheme changes are left unchanged during the transformation process.

However, there are some grammatical rules that have to be changed in order to have a correct expression. Consider the following phrase:

"geldiğimiz yer" (the place from/to where we came)

gel+Verb+Pos^DB+Adj+PastPart+P1pl   yer+Noun+A3sg+Pnon+Nom

It is translated into Crimean Tatar as

"kelgen yerimiz"

kel+Verb+Pos^DB+Adj+PastPart+Pnon   yer+Noun+A3Sg+P1Pl+Nom

The morphological generator directly maps the past participle morpheme –dik of Turkish to –gen (or its equivalent according to vowel/consonant harmony). The main difference between the two analyses is the possessive information. In Turkish, the possessive marker of past or future participle is placed directly after the verb. In Crimean Tatar, it comes after the noun, so it must be transported from the verb to the noun.

However, when the participle appears alone without a noun, the possessive marker should stay untouched and only a straightforward mapping of the morphemes by the morphological generator will be sufficient for the translation.

geldiğimizde (when we came)

gel+Verb+Pos^DB+Noun+PastPart+A3sg+P1pl+Loc

kelgenimizde

kel+Verb+Pos^DB+Noun+PastPart+A3Sg+P1Pl+Loc

Similarly, there are some situations where the translation of a root or a phrase depends on the context. For example, Turkish word "söz" (something said; saying) is normally translated as 'laf' or 'laqırdı'. However, when it appears in the idiom "söz vermek" (to promise), then it is translated as "söz bermek". Thus, the context information is important for the translation of this word. Another example may be the word "zaman" (time). Normally it is translated as "vaqıt", but when it appears after a past participle adjective, it is omitted and the information is saved in the previous verb: "geldiğin zaman" (when you came) and "kelgende".

Among the grammar rules given in the previous section, the followings are to be changed by the grammar and context dependent rules FST:

- Question word: mi
- Case changing verbs
- "By doing so" to "after doing so"
- The possessive information in past participles
- Ability expressed with "olmaq"
- Compound tenses
- Past and narrative morphemes that come after nouns
- Singular plural harmony
- "While" to "past participle"
- Desire and condition
- Until

- "As" to "past participle"
- "Feel like" to "future participle"
- "when" to "past participle"
- Joining of instrumental
- Some words like "zaman" (time), "başlamak" (to start), "durmak" (to stop) according to context

3.4. Translation of Roots

Many of the roots in Turkish and Crimean Tatar are the same such as "ev" (house), "yol" (road), "at" (horse), "devlet" (state). Some of the roots are actually pronounced similarly, with a slight difference, such as "k" in Turkish is written as "q" in Crimean Tatar. In the following examples, The words on the left are in Turkish and the words on the right are in Crimean Tatar.

bakmak (to look)	– baqmaq
kaya (rock)	– qaya

Some other words have the similar structure, but few of the sounds in the word are pronounced differently. These differences are especially in d-t, g-k, g-ğ, v-b, but there are also other examples, which have the differences in other sounds.

dil (tongue/language)	– til
düşünmek (to think)	– tüşünmek
gelmek (to come)	– kelmek
geniş (large)	– keniş
gayret (effort)	– ğayret,
garip (poor/strange)	– ğarip
vermek (to give)	– bermek
duvar (wall)	– divar.

However, this is not a strict rule and these pair of sounds are not always mapped to each other. The following are words used in both of the languages:

demek (to say)
ders (lesson)
gemi (ship)
göl (lake)
vatan (home country)
vilayet (county)

The other words are different in the two languages both in written and spoken form.

çocuk (child)	– bala
genç (young)	– yaş
yaşlı (old)	– qart

Some of the words in Turkish may have different translations according to the part of speech information. For example the word “çok” is translated as “ziyade” if it is an adverb and is changed to “çoq” if it is an adjective.

The translation system we developed does not translate the words that are the same in the two languages. They are left untouched during the translation process and the morphological generator generates the corresponding words in surface form. A simple pattern matching and replacement FST translates those words that are written differently. The type of a word also plays a role in the replacement process.

3.5. Translation Rules

The translation system is developed using finite state tools of XEROX, so the translation rules given in this section for the system are in the XFST syntax. The general structure of the rules is context dependent replacement. The corresponding phrase or word in a given context replaces one phrase or a word. The structure of a rule is as follows:

```
[ source -> target || LeftContext _ RightContext ];
```

Operator	Meaning	Explanation
`->`	Replace	Maps the source text to target text

<table>
<tr><td>@-></td><td>Longest Match Replace</td><td>Chooses the longest candidate of the source string and replaces it with the target text</td></tr>
<tr><td>||</td><td>Condition Marker</td><td>Marks the boundaries of condition and separates the target text from the conditions. Used as:
[source -> target || LC _ RC]</td></tr>
<tr><td>,</td><td>Parallel Operation</td><td>Runs two rules in parallel. Used when there is no context information for the rules.</td></tr>
<tr><td>,,</td><td>Parallel Operation</td><td>Runs two rules in parallel when their contexts are different from each other.</td></tr>
<tr><td>|</td><td>Logical Or</td><td>Logical or operator that denotes any of the operands may appear at this location.</td></tr>
<tr><td>*</td><td>Zero or More Times Occurs</td><td>It denotes that the token that comes just before this sign may appear zero or more times in the text.</td></tr>
<tr><td>+</td><td>One or More Times Occurs</td><td>It denotes that the token that comes just before this sign may appear one or more times in the text.</td></tr>
<tr><td>%</td><td>Escape Character</td><td>Eliminates any special meaning of the following character.</td></tr>
<tr><td>0</td><td>Epsilon</td><td>A symbol that represents the empty string "".</td></tr>
<tr><td>?</td><td>Any Character</td><td>A symbol that represents any symbol that occurs in the same regular expression and any unknown symbol.</td></tr>
<tr><td>^</td><td>N-ary Concatenation</td><td>Used as A^n. The set of strings or pairs of strings obtained by concatenating A with itself <i>n</i> times. For example A^3 is equivalent to [A A A].</td></tr>
</table>

Table 11. Operators Used in XFST

The source is mapped to target if it appears in the given context. The underscore character determines the position of the replacement. Context information is not obligatory and if it is not given, the source text is always mapped to target in any context.

The operators used in the rules are shown in Table 11. For more information on xfst syntax see [17].

In the following sections, we categorise the rules and give examples for each category. The first word of each pair is in Turkish and the second word is in Crimean Tatar. English meanings are given in parentheses next to each Turkish word.

3.5.1. Most Trivial

This set of rules includes no change in the roots or in the morphemes. All the roots and the morphemes are conserved.

evimiz (our house)	: ev+Noun+A3sg+P1pl+Nom
evimiz	: ev+Noun+A3Sg+P1Pl+Nom

evlerindekiler (those who/which are in their houses):
    ev+Noun+A3pl+P3sg+Loc^DB+Det^DB+Noun+Zero+A3pl+Pnon+Nom
evlerindekiler :
    ev+Noun+A3Pl+P3Sg+Loc^DB+Det^DB+Noun+Zero+A3Pl+Pnon+Nom

No translation rules are applied for these cases.

3.5.2. Root Change

Only the root of the source and the type information are changed, and the rest of the morphemes are not changed.

çocukların (of the children)	: çocuk+Noun+A3pl+Pnon+Gen
ballarnıñ	: bala+Noun+A3Pl+Pnon+Gen
güvenimizi (our trust [acc])	: güven+Noun+A3sg+P1pl+Acc
işançımıznı	: iSanC+Noun+A3Sg+P1Pl+Acc
dönüşümüze (to our return)	: dön+Verb+Pos^DB+Noun+Inf+A3sg+P1pl+Dat
qaytuvımızğa	: qayt+Verb+Pos^DB+Noun+Inf+A3Sg+P1Pl+Dat

These three root change rules are represented as follows in XFST syntax:

```
[ çocuk %+ Noun -> bala %+ Noun ,
  güven %+ Noun -> işanç %+ Noun,
  dön %+ Verb -> qayt %+ Verb];
```

These rules can be applied in parallel and since no context information is given, they are joined to each other by a single comma.

3.5.3. Morpheme Change

Some of the morphemes are to be changed without touching the root of the word. For example, as explained in the previous section, Turkish “FeelLike” morpheme is to be changed into “FutPart” without effecting the other parts of the word in lexical form.

yiyesim (I felt like eating)

ye+Verb+Pos^DB+Noun+FeelLike+A3sg+P1sg+Nom

aşaycağım

aSa+Verb+Pos^DB+Noun+FutPart+A3Sg+P1Sg+Nom

```
[FeelLike -> FutPart];
```

These rules directly map one morpheme or a sequence of morphemes into another.

3.5.4. Root and Morpheme Change

In addition to the root of the source structure, some of the morphemes are changed. Actually, these are mostly the roots, which are expressed differently. In the following example, the root of Turkish structure is a verb, whereas the root of the Crimean Tatar counterpart is an adjective, which is later changed into a verb.

sakinleşti (calmed down)	: sakin+Adj^DB+Verb+Become+Pos+Past+A3sg
tındı	: tIn+Verb+Pos+Past+A3Sg

```
[sakin %+ Adj %^ DB %+ Verb %+ Become -> tIn %+ Verb ];
```

The root and the related morphemes are mapped to target morphemes.

3.5.5. Verbs That Effect Its Object

Some verbs change the case of their objects. In other words, the same verb is used with different cases of its object in the two languages. For example, in Turkish something is asked to a person whereas something is asked from someone in Crimean Tatar.

çocuğa sordu (noun+dat sor+verb) = baladan soradı (noun+abl sora+verb)
çocuğa (to the child)
çocuk+Noun+A3sg+Pnon+Dat sor+Verb+Pos+Past+A3sg
baladan (from the child)
bala+Noun+A3Sg+Pnon+Abl sora+Verb+Pos+Past+A3Sg

```
[Dat -> Abl || _ [% ]+ [sor | ısmarla ] %+ Verb];
```

The dative morpheme of a noun is changed into ablative if it precedes the verbs “sormak” (to ask) and “ısmarlamak” (to order).

3.5.6. Grammar Structures That Effect the Previous and Following Words

These are the rules that effect the previous and following words. For example, the past participle morpheme –dik in Turkish corresponds to –gen in Crimean Tatar and the possessive morpheme is added to the verb in Turkish but it is added to the noun in Crimean Tatar. Another such rule is that the noun coming after the word “çok” can be singular in Turkish but it cannot be singular in Crimean Tatar.

oturduğum yer (verb+pastpart+poss noun) = oturğan yerim (verb+pastpart noun+poss)
oturduğum yer (the place where I sit/stay)
otur+Verb+Pos^DB+Adj+PastPart+P1sg yer+Noun+A3sg+Pnon+Nom
oturGan yerim

otur+Verb+Pos^DB+Adj+PastPart+Pnon yer+Noun+A3Sg+P1Sg+Nom

```
[P1sg -> Pnon || DB %+ Adj %+ PastPart %+ _  [% ]+ [?+ %+
     Noun %+ ] ,,
 Pnon -> P1sg || DB %+ Adj %+ PastPart %+ P1sg [% ]+ [?^<10]
     %+ Noun %+ [Inf %+]* A ?^3 %+ _ ];
```

This is composed of two rules that are running in parallel. The first part changes the possessive information in the verb part to “Pnon”. The second part changes the possessive information of the noun to the possessive information of the verb. Since the contexts of the two rules are different, they are run in parallel with double comma operator. Notice that the given rule runs only for the first person singular possessive marker and similar rules for other possessive markers are to be written.

3.5.7. More Than One Word Maps to One Word

Sometimes more than one word should be expressed with only a single word or one word corresponds to two or more words. For example “yırlamak” in Crimean Tatar is expressed as “şarkı/türkü söylemek” in Turkish.

geldiğim zaman (verb+pastpart+poss zaman+noun) = kelgende (verb+pastpart+loc)

geldiğim zaman (when I came)

gel+Verb+Pos^DB+Adj+PastPart+P1sg zaman+Noun+A3sg+Pnon+Nom

kelgende

kel+Verb+Pos^DB+Noun+PastPart+A3Sg+Pnon+Loc

```
[Adj %+ PastPart %+ [P1sg | P2sg | P3sg] [% ]+ zaman %+ Noun
%+A3sg %+ Pnon %+ Nom -> Noun %+ PastPart %+ A3sg %+ Pnon %+
Pnon %+ Loc];
```

Two words that are separated by one or more space characters are mapped to another pattern. The `[%]+` part denotes that there is at least one space character between the words.

3.5.8. One Word Maps to More Than One Words

In Crimean Tatar, the compound tenses are written separately. Whenever a second tense follows the first one, then it is separated from the first one. Also, sometimes one Turkish word should be translated as a group of words such as “sunmak” (to present) translated as “taqdim etmek”.

gelmişti = kelgen edi

gelmişti (he had come)	: gel+Verb+Pos+Narr+Past+A3sg
kelgen edi	: kel+Verb+Pos+Narr+A3Sg e+Verb+Pos+Past+A3Sg

```
[Narr %+ Past -> Narr %+ A3sg [%] e %+ Verb %+ Pos %+ Past];
```

This rule is similar to the previous one. `[%]` part ensures that there is one space character between the two inserted words. The absence of + character is to avoid infinite insertion of spaces.

3.5.9. Rule Order

The order of rules normally is not important. Mostly, they can be applied in any order. However, the rules that change the roots must be applied as the last step. The system is dependent on the Turkish roots and it checks the Turkish roots and morphemes when it checks the previous and next tokens. Thus, to have a reliable system, the rules that change the roots must be applied at the end. If for any reason a root is changed or a word is inserted in an intermediate step, an exclamation mark is inserted so that the root changing rules do not change it.

If, at anywhere, a rule order is important, it can be placed in the correct position in the rules. The architecture of the system is such that it applies the first rule to the input, then applies the second rule to the output of the first and so on. Parallel rules are applied in parallel at the same time in the order that they appear in the rules. If for any reason, it is possible to give more than one output for the given input, all possible generations are given. This is helpful, especially in parallel runs, since more than one rule may effect the input.

Chapter 4

Crimean Tatar Morphological Processor

4.1. Morphological Process

A morpheme is the smallest unit of meaning in a language. It can be a single word like *world, moon, cat* or a meaningful unit that follows a root or other morphemes like –s in eat<u>s</u>. Morphemes are usually grouped into two as stems and affixes. Stem is the main structure that gives the main meaning to the word and affixes are usually the morphemes that add some meaning to it. Affixes can come before, after, to both sides of and inside words.

Morphemes in different languages may have different properties. Some languages like English usually have single words as morphemes in the sentence whereas languages like Turkish are mostly composed of roots followed by other morphemes. Most of the time, there are special spelling rules to concatenate morphemes to each other. For example, when the plural morpheme –s is added to English word *city*, the –y is changed into –ie before it acquires –s.

Morphology is the study of the way words are built from morphemes. Storing all possible words in lexicons is practically very difficult for many languages, especially when the language is rich in morphemes. Languages like Turkish virtually have infinite number of words in the lexicon. Any verb can be turned into a noun with several morphemes or any noun can be turned into different nouns with the addition of the determiner morpheme –ki infinite number of times. For this reason, it is practically necessary to analyse words to determine its morphemes.

Thinking in the other way, it is sometimes necessary to build a morphological generator, that is a program that generates words in the actual spelling provided the correct morphemes. For example, after translation processes, the output is usually in a format that represents the morphemes of the word and it needs to be rewritten in the actual spelling.

In order to build a morphological processor, a program that can analyse words to decompose them into their morphemes and generate words given the correct morphemes in the right order, we need to have two main information sources in addition to a lexicon that lists the root words with type information.

Orthographic Rules

These are the spelling rules that determine how the morphemes are fixed to each other. Morphemes are usually represented by an abstract symbol that is to be changed into the actual spelling when it is processed. For example in our system, we use symbol D for sounds ‘d’ and ‘t’ and D is changed into one of them according to vowel and consonant harmony rules.

Morphotactic Rules

Morphotactic rules determine the order in which the morphemes appear in a word. For example, in English, -ing morpheme comes after verbs rather than adjectives. In Turkish, possessive marker comes after root nouns when it is singular and always follows the plural morpheme –lAr when the noun is plural.

A morphological processor is a finite state transducer that combines these two parts with the root words.

4.2. Overview of Two-Level Morphology

Two-level morphology is a way of handling morphological structures by executing pseudo-parallel rules [10]. There are two levels of the system, *surface level* and *lexical level*. Surface level representation is the direct representation of input, as it is represented in the original language. Lexical level is the decomposed form of the input and is the output of the system when the surface representation is given as input. In a finite state transducer, normally the surface and lexical levels are represented as two expressions separated by a colon. For example, an expression like *a:b* is usually expected to mean "lexical form a is derived from the surface form b".

Rules that denote the morphological modifications and variations are all executed in parallel and all the rules work on the same input. If all of the rules accept the input, then the machine accepts the input. However, if the input is rejected by any of the rules, then the machine rejects the input directly.

There are four different rule types in such a system:

> a:b => LC _ RC : Lexical *a* is mapped to surface *b* if it appears in these left and right contexts. However, its appearing in this context does not require such a mapping. In other words, if *a* is mapped to *b*, then it must be in this context and cannot happen in another context.
>
> a:b <= LC _ RC : lexical form *a* is mapped to surface form *b* if it appears in LC and RC. However, it is also possible to map *a* to *b* in another context.
>
> a:b <=> LC _ RC : a lexical *a* is always mapped to a surface *b* in this context and this is possible only in this context.
>
> a:b /<= LC _ RC : a lexical *a* is never mapped to a surface *b* in the given context.

The morphotactic rules are compiled to a finite state transducer and are joined with these rules. The system as a whole tries to locate the roots and possible following suffixes for a given surface form input. If the system at any stage cannot locate a valid suffix or it discovers a situation violating the morphological modification rules, it returns with no answer. For a detailed explanation of two-level morphology, see [16].

4.3. The Alphabet

As it is the case for all Turkic languages, Crimean Tatar was also written using the Arabic Script in the beginning of twentieth century. After the formation of Soviet rule, the Cyrillic alphabet of Russian language was started to be used. Now, a Latin based alphabet which is the same as Turkish alphabet with few additions was accepted by the Crimean Tatar National Assembly and is being used. The Latin based Crimean Tatar alphabet is:

Aa Ââ Bb Cc Çç Dd Ee Ff Gg Ğğ Hh Iı İi Jj Kk Ll Mm Nn Ññ Oo Öö Pp Qq Rr Ss Şş Tt Uu Üü Vv Yy Zz

In the program, the letters that are present in the ASCII characters are used as is in lowercase. Both in surface form and lexical form, we represented the letters which are absent in ASCII, with the capital form of the closest symbol. The correspondences are as follows:

ç – C ğ – G ı – I ö – O ş – S ü – U ñ – N

The system directly maps the characters in the alphabet to themselves and any straightforward mapping is shown in the alphabet section of the rules. For example, a single ‘m’ in the alphabet shows that the character m is always mapped to itself in both surface level and morphological structure. For some of the symbols, there is a single occurrence of it in addition to pairing of it with another character. Symbol n appears alone and in the ‘V:n’ pair in the alphabet. This shows that symbol n is mapped to both n and V in the morphological structure.

At the lexical level, however, we need to use some extra characters to represent one to many mappings and exceptions. For this purpose, we use the following capital letters, which are used only in the program and are invisible to the user:

J – ç that does not change to c	: kUJ + U = kUCU
P – p that does not change to b	: saP + I = sapI

Q – q that does not change to ğ : baQ+a = baqa

T – t that does not change to d : beT + i = beti

W – k that does not change to g : teW + i = teki

H – corresponds to symbols I, i, u, U according to vowel harmony

M – corresponds to symbols I, i according to vowel harmony

A – corresponds to a or e according to vowel harmony rules

Y – corresponds to u or U according to vowel harmony rules.

K – corresponds to g, k, G, q according to consonant harmony rules

D – corresponds to d or t according to consonant harmony rules.

Z – s that does not drop as a joining sound : alim + Ziñ = alimsin

We also use the following groupings in the two level morphology rules.

The vowels are (VOWEL) = a e I i o O u U A H M Y â

The consonants are (CONS) = b c C d f g G h j k l m n N p q r s S t v y z K Z B P Q J W

The other groupings are as follows :

Back Vowel	(BACKV)	= a I u o â;
Front Vowel	(FRONTV)	= e i O U;
Front Unrounded Vowel	(FRUNROV)	= i e;
Front Rounded Vowel	(FRROV)	= O U;
Back Rounded Vowel	(BKROV)	= u o;
Back Unrounded Vowel	(BKUNROV)	= a I â;
Soft Consonants	(SEDALI)	= b c d g G j v z l m n N r y h B;
Hard Consonants	(SEDASIZ)	= p C t k q S f s Z P Q J W;
Joining Consonants	(X)	= s y;

There is also a special surface character 0 (epsilon) which can be mapped to any morphological level character. A pair like +:0 states that instead of an invisible character 0 in the surface level, a plus sign (+) is inserted where necessary in the morphological structure.

4.4. Vowel and Consonant Harmony Rules

"A realized as a"

A:a => [:BACKV] [CONS]* (%+:0) [CONS: | :CONS | :0]* _ ;

After a back vowel, the following A must be represented as an a.

bala + lAr -> balalar (çocuklar – kids)

qoy + lAr -> qoylar (koyunlar - sheep)

"A realized as e"

A:e => [:FRONTV] [CONS]* (%+:0) [CONS: | :CONS | :0]* _ ;

Similar to the following rule, this follows the grammar rule stating that front vowels are to follow front vowels :

kOy + lAr -> kOyler (köyler – villages)

gUl + DAn -> gUlden (gülden – from the rose)

"A realized as y"

A:y <=> [:VOWEL] %+:0 _;

In Crimean Tatar, present progressive tense suffix is –y and future suffix is -ycAK if the root ends in a vowel:

sora + A -> soray (soruyor – s/he is asking)

qorCala + AcAK -> qorCalaycaq (koruyacak – s/he will protect)

"H realized as u"

H:u => .#. [CONS]* [:BKROV] [CONS]* (%+:0) [CONS: | :CONS | :0]* _ ;

If there is only one syllable in the root, which means there is only one vowel before H and if it comes after a back rounded vowel (o, u), it is resolved to u:

soN+HncH -> soNuncI (sonuncu – the last)

"H realized as U"

H:U => .#. [CONS]* [:FRROV] [CONS]* (%+:0) [CONS: | :CONS | :0]* _ ;

In other cases, namely H coming in the second syllable and following a front rounded vowel (U, O), it is resolved to U:

kOy + ZHz -> kOysUz (köysüz – without a village)
UC + HncH -> UCUnci (üçüncü – the third)

"H realized as i"

H:i => [:VOWEL] [CONS]* (%+:0) [CONS: | :CONS | :0]* [:FRONTV]
[CONS]* (%+:0) [CONS: | :CONS | :0]* _ ;
.#. [CONS]* [:FRONTV] [CONS]* (%+:0) [:CONS | CONS: | :0]* _;

In Crimean Tatar language, the u and ü in suffixes can appear only in the second syllable, and for the same suffix, it is written as ı or i in the third and the later syllables. Few exceptional morphemes, such as past morpheme –di and accusative morpheme –ni, are most of the time written with ı/i even if they appear in the second syllable. Here are two rules operationg in parallel. The first rule checks whether there are at least two syllables. If there are, then H is resolved to i after all front vowels, namely e, i, O, U. If there is one syllable, then the second rule runs and maps H to i after only front unrounded vowels.

kOr + DH -> kOrdi (gördü – s/he saw)
sUt + sHz + lHK -> sUtsUzlik (sütsüzlük – milklessness)

"H realized as I"

H:I => [:VOWEL] [CONS]* (%+:0) [CONS: | :CONS | :0]* [:BACKV]
[CONS]* (%+:0) [CONS: | :CONS | :0]* _ ;
.#. [CONS]* [:BACKV] [CONS]* (%+:0) [:CONS | CONS: | :0]* _;

This is the corresponding rule for the previous one. It checks whether there are at least two syllables. If there are, it maps H to I if the previous vowel is a back vowel, namely a, â, ı, o, u. If there is one syllable, it maps H to I only if it comes after a back unrounded vowel:

azbar + HmHz -> azbarImIz (bahçemiz – our garden)
qal + DH -> qaldI (kaldı – he stayed)

"H is dropped after a vowel, before a morpheme"

H:0 <=> [:VOWEL] %+:0 _ ;

If H comes in the beginning of a morpheme and the last symbol of the previous morpheme is a vowel, then H drops :

eki + HncH -> ekinci (ikinci – second)

tile + Hr -> tiler (diler – s/he wishes)

"M realized as i"

M:i => [:FRONTV] [CONS]* (%+:0) [CONS: | :CONS | :0]* _ ;

Sometimes, morphemes are written with ı/i and never written with u/ü. The accusative morpheme "nı/ni" is an example for this. M is paired with i when it comes after a front vowel.

ev + nM -> evni (evi – the house)

gUl + nM -> gUlni (gülü – the rose)

"M realized as I"

M:I => [:BACKV] [CONS]* (%+:0) [CONS: | :CONS | :0]* _ ;

When M comes after a back vowel, it is resolved as ı.

qol + nM -> qolnI (kolu – the arm)

bala + nM -> balanI (çocuğu – the child)

"M is dropped after a vowel, before a morpheme"

M:0 <=> [:VOWEL] %+:0 _ ;

If M comes ın the begınnıng of a morpheme after a vowelö then ıt ıs dropped.

sora + Mp -> sorap (sorarak – by asking)

"Y realized as u"

Y:u => [:BACKV] [CONS]* (%+:0) [CONS: | :CONS | :0]* _ ;

In some cases, morphemes are written with u/ü and never with ı/i. An example for such morphemes is -uv/-üv which makes nouns from verbs. Y is used and resolved into u if it follows something resolved into a back vowel:

toplaS + Yv -> toplaSuv (toplanış / toplanma – gathering / meeting) oq + Yv -> okuv (okuma – reading / education)

"Y realized as U"

Y:U => [:FRONTV] [CONS]* (%+:0) [CONS: | :CONS | :0]* _ ;

If Y comes after some symbol that is resolved into a front vowel, then it is resolved into U:

kel + Yv -> kelUv (geliş / gelme – coming)

"Y is dropped after a vowel, before a morpheme"

Y:0 <=> [:VOWEL] %+:0 _ ;

If Y comes in the beginning of a morpheme and after a vowel, it is dropped:

sayla + Yv -> saylav (seçim - election)

"K realized as k"

K:k => [:FRONTV] [:CONS]* [:SEDASIZ] %+:0 _ [:FRONTV] [CONS: | :CONS | :0]* ;
[:FRONTV] (%+:0) _ [.#.|[(%+:0) [CONS]]];

In Crimean Tatar language, there are two different k sounds which are also represented seperately in writing : one is represented by k and the other is by q. "k" is paired with front vowels and q is with back vowels. Also there are "sedalı" (soft) and "sedasız" (hard) consonants which affect their changes in the words. In morphemes, k softens to g, and q softens to ğ. Please note that the sound ğ is not the same as the soft g in Turkish and it is much harder a sound. All these forms are represented by the capital K symbol.

K corresponds to k when it comes in the beginning of a morpheme where the last sound in the root is a hard consonant and the last vowel is a front vowel or the root ends in a front vowel.

ket + Kan -> ketken (gitmiş – s/he went)
kel +mAK -> kelmek (gelmek)

"K realized as q"

K:q => [:BACKV] [:CONS]* [:SEDASIZ] %+:0 _ [:BACKV] [CONS: | :CONS | :0]*;
[:BACKV] (%+:0) _ [.#.|[(%+:0) [:CONS]]];

If K is in the beginning of a morpheme and preceded by a back vowel and a hard consonant, or it follows a back vowel, it is reallized as q.

saC + Kan -> saCqan (ekmiş - planted)
baq +AcAK + mMz -> baqacaqmIz (bakacağız – we will look)

"K realized as g"

K:g => [[:FRONTV] [:CONS]* [:SEDALI] | [:FRONTV]] %+:0 _ [:FRONTV] [CONS: | :CONS | :0]*;
[:FRONTV] _ %+:0 (:0) [:FRONTV];

This rule and the following one are the pair stating the rules for softening the k and q. K corresponds to g if it is preceded by a front vowel and a soft consonant (sedalı) or if it is preceded and followed by front vowels.

kel + AcAK + Hm -> kelecegim (geleceğim – I will come)
piSir + KAn -> piSirgen (pişirmiş – s/he cooked)

"K realized as G"

K:G => [[:BACKV] [:CONS]* [:SEDALI] | [:BACKV]]%+:0 _ [:BACKV] [CONS: | :CONS | :0]*;
[:BACKV] _ %+:0 (:0) [:BACKV];

K is paired with G when it comes between two back vowels or it follows a back vowel and a soft consonant.

qal + AcAK + Hm -> qalacaGIm (kalacağım – I will stay)
al + KAn -> alGan (almış – s/he took)

"X is deleted after a consonant"

X:0 <=> [:CONS | CONS:] %+:0 _ ;

The symbols n, s and y are not written if they follow a consonant but written if they follow a vowel. For example s in the following morpheme is deleted :

ev + sH -> evi

"D realized as t"

D:t <=> [:SEDASIZ] %+:0 _ ;

D is realized as t if and only if it follows a symbol that corresponds to a hard consonant (sedasız). Otherwise it is realized as d.

ket + DH -> ketti (gitti – s/he went)
kitap + DAn -> kitaptan (kitaptan – from the book)

"k realized as g"

k:g <=> [VOWEL] _ %+:0 (X:0) [:VOWEL];

The symbol k in the end of a word is realized as g if it is followed by a vowel in the following morpheme, possibly with a sound dropping in between:

yürek + Hm -> yüregim (yüreğim – my heart)
eSek + sH -> eSegi (eşeği – his/her donkey)

Note that this is not the same as K realized as g in the previous rules.

"q realized as G"

q:G <=> [VOWEL] _ %+:0 (X:0) [:VOWEL];

Symbol q is changed into a G if it succeeds a vowel and the beginning of the following morpheme is a vowel.There may possibly be a dropping joining sound such as s.

ayaq + sH -> ayaGI (ayağı – his/her foot)
qaSIq + HmHz -> qaSIGImIz (kaşığımız – our spoon)

"C realized as c"

C:c => [VOWEL] _ %+:0 (X:0) [:VOWEL];

The character C corresponds to c if it comes between two vowels with a possible dropping sound.

aGaC + HmHz -> aGacImIz (tahtamız – our wood)

"p realized as b"

p:b <=> [VOWEL] _ %+:0 (X:0) [:VOWEL];

The symbol p is changed into b if it is followed by a vowel.

kitap + sH -> kitabI (kitabı – his/her book)

Garip + Hm -> Garibim (garibim – my poor)

"c realized as C"

c:C => [:SEDASIZ] %+:0 _ [:VOWEL];

The symbol c corresponds to a C after a root or a morpheme ending in a hard sound (sedasız). This is especially for the morpheme -cı / -ci which makes nouns from nouns.

qurt + cH -> qurtCu (kurtçu – wolf trainer)
aS + cH -> aSCI (aşçı – cook)

For the last few rules, it is necessary to state that there are exceptional cases. For example, for the word "sap + sH", it becomes "sapI", namely the symbol p does not change into b. Or for the word "tek", again there is no change. However, for "tUp", there is a change when the "sH" morpheme is added: "tUbU". There is no strict rule for these kinds of words. The way we handle them is changing all the C's or p's whenever it is possible, and writing those words, which do not change with a different symbol. For example, the word "tek" will internally be written as "teW" and "sap" as "saP". Note that all the symbols are normally lower case symbols except for special Turkish characters. The rest of the uppercase characters are special cases handled in different situations.

4.5. Morphotactics

4.5.1. Roots

The root words for this application are compiled from pieces of literary works. They include words from different dialects of the language. There are 5300 root words included in our lexicon. There are different reasons behind the fact that the total number of roots is not very high. First of all, our lexicon does not include many words derived from Russian and other languages. Only a very small part of proper names are included in the system and technical words are not considered. Moreover, Crimean Tatar language could not find a fertile area to develop during Soviet period, leaving us with a relatively small lexicon. We hope to improve the total area covered by the roots in time. The list of words are grouped as follows:

a) Nouns

b) Verbs

c) Adjectives

d) Adverbs

e) Proper Names

f) Simple Numbers

g) Pronouns

h) Connectives

4.5.2. Morphotactic Rules For Crimean Tatar

Morphotactics of a language determine the order of morphemes that appear in a word. Although Crimean Tatar is located basically in Kipchak group of Turkic languages, the morphotactic rules of Crimean Tatar mostly comply with those of Turkish. In other words, the morphemes themselves are sometimes different from those of Turkish, however the meaning they imply and the order they appear in the word are usually the same as Turkish.

In the system, the finite state machine starts from a start state and checks the possible constructs beginning with the root and possible following morphemes. Each list of roots and possible following morphemes are expressed in a lexicon file. If a root is matched, then the machine gives the appropriate output and goes to the next state.

Below a sample part of the lexicon can be seen:

```
LEXICON NOUNS
abide+Noun:abide        POST-NOUN;
abla+Noun:abla          POST-NOUN;
aC+Noun:aJ              POST-NOUN;
```

```
acderha+Noun:acderha          POST-NOUN;
acet+Noun:acet                POST-NOUN;
...

LEXICON POST-NOUN
+A3Sg:                PLURAL;
+A3Pl:+lAr            PLURAL;

LEXICON PLURAL
+Pnon:                POSSESSIVE;
+P3Sg:+sH             POSS-3;
+P1Sg:+Hm             POSSESSIVE;
+P2Sg:+HN             POSSESSIVE;
+P1Pl:+HmHz           POSSESSIVE;
+P2Pl:+HNHz           POSSESSIVE;

LEXICON VERBS
abdIra+Verb:abdIra    POST-VERB;
aC+Verb:aC            POST-VERB;
acI+Verb:acI          POST-VERB;
adal+Verb:adal        POST-VERB;
adaS+Verb:adaS        POST-VERB;
afIr+Verb:afIr        POST-VERB;
aGIr+Verb:aGIr        POST-VERB;
...

LEXICON POST-VERB
```

```
              NEGATION;
              ABILITY;
              CAUS-PASS;

LEXICON ABILITY
+Pos^DB+Verb+Able:+yAbil  VERBAL-STEM;
^DB+Verb+Able+Neg:+yAmA   VERBAL-STEM;

LEXICON NEGATION
+Pos:               VERBAL-STEM;
+Pos:               AOR;
+Neg:+mA            NEG-MA;

LEXICON NEG-MA
                    VERBAL-STEM;
+Aor+A1Sg:+m        FINAL;
+Aor+A2Sg:+zsHN     FINAL;
+Aor:+z             NEG-AORIST;
+Aor+A1Pl:+mHz      FINAL;
+Aor+A2Pl:+zsHNHz   FINAL;
+Aor+A3Pl:+zlAr     FINAL;
+Aor^DB+Adj+Zero:+z FINAL;
```

For example, for a surface noun form like “abidesi” (the statue of something/someone), we can think of the internal representation as “abide + sH” which is created with the help of vowel and consonant harmony rules. The system would first check the roots for “abide” and as it finds the word there, it outputs the lexical form “abide + Noun” and goes to the next state indicated by POST-NOUN. At this state, the possible morpheme accepted is +lAr. Otherwise, the system goes to the next state, PLURAL, with zero input (epsilon transition) giving the output +A3Sg. Now the output is “abide + Noun + A3Sg”. At the PLURAL state, the system recognises the input morpheme +sH and goes to the next state

POSS-3 after giving the input +P3Sg. This continues until the system reaches the final state or a state that does not accept the input. If the input is not accepted, the output is not returned to the user.

Similarly, the verbal form “aGIrmazlar” (they do not ache) can be represented as “aGIr+mA+zlAr” with the help of vowel and consonant harmony rules. The system first checks all possible nouns, verbs, pronouns etc. to find “aGIr”. As it is found in the verbal list, the system gives the output “aGIr+Verb” and directly goes to the next state, POST-VERB, and then to all three states. The following morpheme “+mA” is listed in NEGATION and although not listed here, the other states do not reach to a final state for this input. The output “+Neg” is given for the morpheme “+mA” and the following state NEG-MA is reached. The system follows direstly to VERBAL-STEM and checks in parallel the possible paths through that state. Also since the morpheme “+zlAr” is listed in this state, the output “Aor+A3Pl” is given and the FINAL state is reached. The final output for this input is “aGIr+Verb+Neg+Aor+A3Pl” and the input is accepted. If, at any state, the system fails to find a path to the FINAL state, then it rejects the input and gives no output.

The finite state diagrams for Crimean Tatar morphotactics is given in Figure 4 and Figure 5.

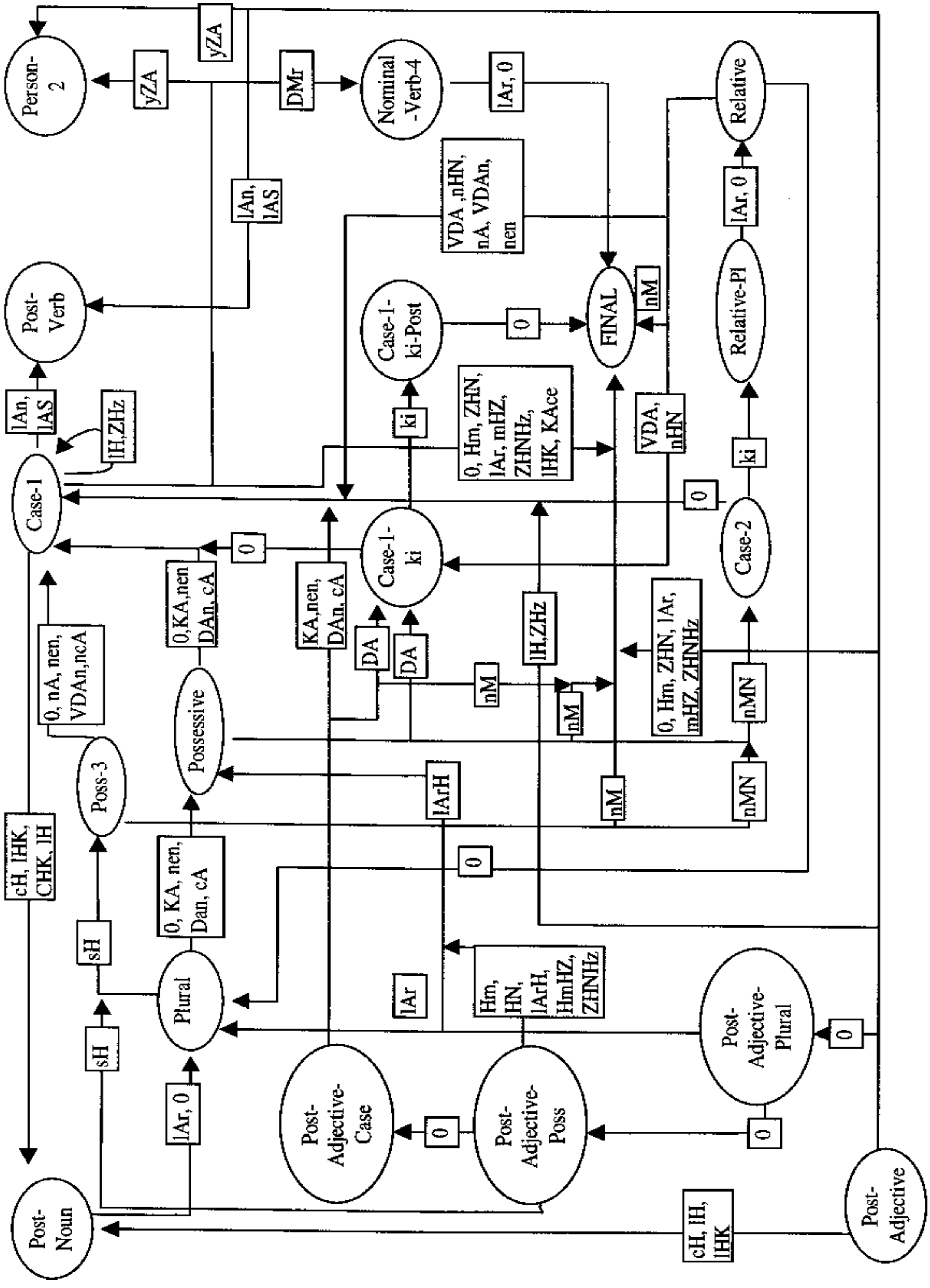

Figure 4. FSA for Crimean Tatar Nouns and Adjectives

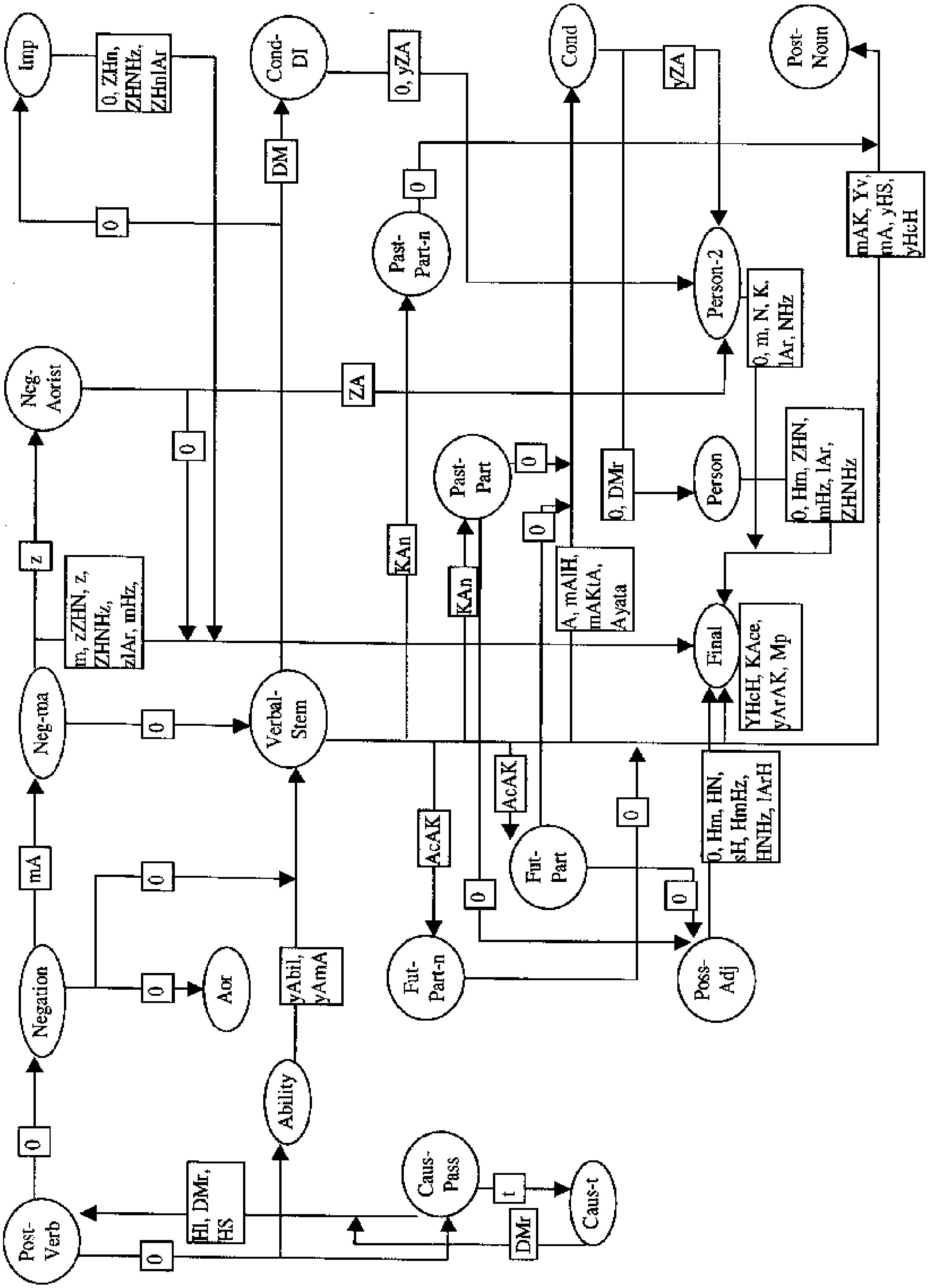

Figure 5. FSA for Crimean Tatar Verbs

Chapter 5

Evaluation – Results

5.1. Implementation

The system is implemented using XEROX Finite State Tools for language engineering which are available to researchers [17].

Xerox finite-state tool (XFST) is a general-purpose utility for computing with finite-state networks. It enables the user to create simple automata and transducers from text and binary files, regular expressions and other networks by a variety of operations. The user can display, examine and modify the structure and the content of the networks. The result can be saved as text or binary files. TWOLC is a compiler that converts two-level rules into deterministic, minimized finite-state transducers. The Finite-State Lexicon Compiler (LEXC) is an authoring tool for creating lexicons and lexical transducers. It is designed to be used in conjunction with transducers produced with the Xerox Two-level Rule Compiler (TWOLC).

The interface of the translation system is written in Java language. It needs the input from a text file and extracts the tokens. The tokens are organised and fed to XEROX tools, which are launched as external applications. The output of each transducer is fed to the next one and the final result is shown on the screen.

5.2. Morphological Processors

Turkish morphological processor was implemented by Kemal Oflazer [18]. The FST is ready in binary format readable by XFST and LEXC.

Two level vowel and consonant harmony rules of the Crimean Tatar morphological processor are compiled with TWOLC. The lexical rules are compiled with LEXC. The basic dialect of Crimean Tatar used for the lexical rules is Bahçesaray dialect. However the lexicon includes words from other dialects. There are a total of more than 5300 root words compiled from around 80.000 word text that includes pieces from different literary works and public literature.

The program runs in both ways. Given the surface form, the lexical form is produced by the program. Similarly, when the lexical form is given, the corresponding surface form is produced. The mappings are not one to one due to ambiguities in the language, so it is always possible to get more than one result.

The output of the morphological analyser starts with the root of the surface form entered by the user. Then the type of the word is given. The following morpheme for nouns is agreement morpheme and possessive and case markers follow it. Adjectives and pronouns are similar. For verbs, the third morpheme is sense. Tense and agreement markers follow it. Changes in the type of the word are marked with a derivational boundary (^DB) and it is followed by the new type of the word.

Below are a few example outputs of the system:

kelem	kel+Verb+Pos+Prog1+A1Sg
	(geliyorum – I am coming)
eviNiz	ev+Noun+A3Sg+P2Pl+Nom
	(eviniz – your house)
qalacaGIm	qal+Verb+Pos+Fut+A1Sg
	(kalacağım – I will stay)
	qal+Verb+Pos^DB+Adj+FutPart+P1Sg

	(kalacağım [adjective] – [of the place] related to my stay)
	qal+Verb+Pos^DB+Noun+FutPart+A3Sg+P1Sg+Nom
	(kalacağım [bana ait olan kalma eylemi] – my prospective stay)
	qal+Verb+Pos^DB+Noun+FutPart+A3Sg+Pnon+Nom^DB+Verb+Zero+Pres+A1Sg
	(kalacak olanım – I am the one who will stay)
	qal+Verb+Pos^DB+Noun+FutPart+A3Sg+P3Sg+Nom^DB+Verb+Zero+Pres+A1Sg
	(kalacağıyım – I am his prosepective stay)
suvaruv	suvar+Verb+Pos^DB+Noun+Inf+A3Sg+Pnon+Nom
	(sulama – watering)
yazGanlarGa	yaz+Verb+Pos^DB+Noun+PastPart+A3Pl+Pnon+Dat
	(yazanlara – to those who have written)

5.3. Transformation System

Several transducers compiled by XFST handle translation of grammar rules, context dependent structures and roots. The system is implemented according to the grammar rules in [12, 13, 14, 15] and mostly on the comparative translations between Turkish and Crimean Tatar by Zuhal Yuksel of Gazi University in [19].

The input sentence is first read from the input device and divided into its words, then each word is passed through Turkish morphological analyser. All possible analyses are generated by the fst and then they are again joined so that the context information, the original order in which the words appeared is not lost.

Consider the sentence “akşam eve geleceğiz”. It is first divided into its tokens, “akşam”, “eve”, “geleceğiz”. Then each of these words are passed through Turkish analyser and the following results are taken:

akşam:

1. akşam+Noun+A3sg+Pnon+Nom

eve:

1. ev+Noun+A3sg+Pnon+Dat

geleceğiz:

1. gel+Verb+Pos+Fut+A1pl
2. gelecek+Noun+A3sg+Pnon+Nom^DB+Verb+Zero+Pres+A1pl
3. gelecek+Adj^DB+Verb+Zero+Pres+A1pl

Since the last token returned with three analyses, there are three possible combinations of this sentence. They are all constructed and we get these three sentences:

akşam+Noun+A3sg+Pnon+Nom ev+Noun+A3sg+Pnon+Dat
gel+Verb+Pos+Fut+A1pl

akşam+Noun+A3sg+Pnon+Nom ev+Noun+A3sg+Pnon+Dat
gelecek+Noun+A3sg+Pnon +Nom^DB+Verb+Zero+Pres+A1pl

akşam+Noun+A3sg+Pnon+Nom ev+Noun+A3sg+Pnon+Dat
gelecek+Adj^DB+Verb +Zero+Pres+A1pl

These sentences are then given to the translation FST that checks the sentences for compatibility with Crimean Tatar grammar. All necessary grammar changes and context dependent transformations are made by this FST.

The output of the translation FST is again broken down to its words and this time each word is given to the FST that translates the roots. The output of the of the root FST for the previous sentences is as follows:

akşam+Noun+A3Sg+Pnon+Nom ev+Noun+A3Sg+Pnon+Dat kel+Verb+Pos+Fut+A1Pl

akşam+Noun+A3Sg+Pnon+Nom ev+Noun+A3Sg+Pnon+Dat
    istiqbal+Noun+A3Sg+Pnon+Nom^DB+Verb+Zero+Pres+A1Pl

akşam+Noun+A3Sg+Pnon+Nom ev+Noun+A3Sg+Pnon+Dat
    kelecek+Adj^DB+Verb+Zero+Pres+A1Pl

As seen from the sentences, the first "gel+Verb" is translated by the FST to "kel+Verb". For demonstrative purposes, "akşam" which should be transfromed to "aqSam" is left unchanged.

Before this output is fed to Crimean Tatar generator, one final transformation is possible. Many of the words in Turkish and in Crimean Tatar are the same except that they are written with 'k' in Turkish and with 'q' in Crimean Tatar. The rule is strict and any 'k' that precedes or follows any of "a, ı, o, u" are to be changed into a 'q'. In addition, since the system we developed does not operate on Turkish characters and special upper case characters are used instead of them, we need to change the Turkish characters into the form recognised by the system. As a result, we can apply this rule to the input so that many words that are not covered by the translation lexicon can be recognised by the generator. After applying this FST, the output becomes:

aqSam+Noun+A3Sg+Pnon+Nom ev+Noun+A3Sg+Pnon+Dat kel+Verb+Pos+Fut+A1Pl

aqSam+Noun+A3Sg+Pnon+Nom ev+Noun+A3Sg+Pnon+Dat
    istiqbal+Noun+A3Sg+Pnon+Nom^DB+Verb+Zero+Pres+A1Pl

aqSam+Noun+A3Sg+Pnon+Nom ev+Noun+A3Sg+Pnon+Dat
    kelecek+Adj^DB+Verb+Zero+Pres+A1Pl

If the tokens of these sentences are given to Crimean Tatar morphological generator and the output is formatted, we get

2 aqSam evge kelecekmiz
1 aqSam evge istiqbalmIz

The number in front of each sentence states how many times this surface form of the sentence appeared in the output. The first sentence appeared twice and the second one once. Thus, two of our input sentences are translated into the same sentence. The adjective form and the verb form of “gelecek” are translated to the same surface form in Crimean Tatar although they have different morphological representations.

Chapter 6

Conclusion

6.1. Problems

In Turkish, many of the words are ambiguous, that is there are more than one meaning for many of the words. Usually only one of them is true and acceptable in a given sentence. For example, the word “kalem” has basically two meanings: one is “a tool used for writing” and the other is “my castle”. In a sentence like “Kalemle yazdı” (He wrote with a pencil), the word is used most probably in the first meaning. “Kalem şehri kuşatmıştı” (My castle had surrounded the city) implies however the second meaning. Which one of these should be accepted is totally dependent on the context. Morphological disambiguators use context information and statistical processing to guess the correct analysis for a word.

Research done on Turkish is very limited. Resources available are not sufficient most of the time to use directly in a program. There are not many algorithms developed for agglutinative languages, which are also applicable to Turkish. Thus it is hard to find the appropriate tools and methods to use in such a system.

The main problem with the system we developed is the absence of a morphological disambiguator. We did not have an easy to use morphological disambiguator. One such tool was available in hand [20]. However, the format of its input and output was not suitable for our system and implementing an interface for that would require a lot of time and effort which would take us from the main focus of our research.

Since we do not use a disambiguator, we generate all possible combinations of the sentences from the input. After processing all these sentences, we try to guarantee that at least one of the outputs is true. Actually, most of the time ambiguities in Turkish are partially or fully preserved in Crimean Tatar and the surface forms of the output sentences are the same. The Czech-Slovak system [6] and the Spanish-Catalan system [7] mentioned earlier announce to be using such a disambiguator and they claim that their results are more successful.

Another problem with the system is that, although the languages are very similar, there are some problems, which cannot be overcome with a lexical analysis. Turkish and Crimean Tatar are free word order languages and theoretically words of a sentence may be organised in many different ways to give the same meaning. It is better for the object to be close to the verb, but it is not a must. As we explained in a previous section, the cases for objects of some verbs are different in two languages. When the object does not come just before the verb, it cannot be covered by our system. Consider the sentence “Rus kızıyla evlendi” (He got married to a Russian girl). The system will successfully translate it to “Rus qızına evlendi”. However, the sentence “Rus kızıyla Moskova’da evlendi” (He got married to a Russian girl in Moscow) cannot be covered without a parse. Similarly, the sentence “Rus qızıyla memnuniyetle evlendi” (He got married to a Russian girl with pleasure) will probably generate a wrong result since the noun in instrumental case that precedes the verb “evlenmek” (get married) is “memnuniyet” (pleasure).

The use of present progressive for simple present meaning is more common in Crimean Tatar. The sentence “Siz giderseniz ben de gelirim” (If you go, I will come) can be translated as “Siz ketseñiz men de kelirim”. However, the same verb in the same tense in the sentence “Ben de bazen gelirim” (I also sometimes come) is translated into “Men de kimerde kelem” (I also am sometimes coming). There is not a rule for this and it cannot be determined easily even with a parse of the sentence. Idioms, proverbs and culture specific issues are also hard, if not impossible, to translate.

Crimean Tatars lived under Russian rule for about 230 years and the last 80 years were in the Soviet and post-Soviet period. They had to use Russian at all official and public places and took their education in Russian. They were forced to leave the language and the authorities tried to impose Russian as a communication medium even among themselves. As a result Crimean Tatar could not find a fertile area to develop and rejuvenate. Printed materials in Crimean Tatar were very limited during the Soviet times and they are increasing in number recently. Important part of the young people cannot speak Crimean Tatar perfectly. All of these in the overall cause the language to be and remain weak. New words for the newly invented things and concepts could not be generated. They mostly derived the words from Russian. For example words like television, aeroplane, computer, concrete, tax are words that do not have Crimean Tatar counterparts. So, it is sometimes very hard to find the corresponding Crimean Tatar word for a Turkish one. The same problems also arise for Turkish, however it is in a better situation than Crimean Tatar.

One problem with Turkic languages is that verbs do not have regular rules to get the aorist, causative and passive morphemes. The verb “bakmak” (to look) in Turkish and in other Turkic languages is made causative by –tır as in “baktırmak” (to have/cause somebody look). However, the verb the verb “akmak” (to flow) is made causative by –ıt as “akıtmak” (to cause something flow) although phonetically it is similar to “bakmak”.

The aorist morpheme has a similar problem. The verb “kurmak” (to set) takes the aorist morpheme –ar to become “kurar” (he sets). However the verb “durmak” (to stop) takes –ur to become “durur” (he stops). The two verbs are phonetically similar, since the vowels are both ‘u’, but the suffixes they take to have aorist meaning are different. There is no explicit rule for the suffixes that verbs take. There has to be a list of verbs for each suffix and such lists are not available for Turkic languages.

6.2. Future Work

We are planning to add a morphological disambiguator to the system. We believe it will improve the system performance considerably.

The root lexicons in the system are not sufficient to cover every text. Without forgetting the problems mentioned in the previous section, we are planning to enrich the lexicons. Also it is possible that we did not cover some of the grammar rules and context dependent structures in the translation process. We would like to determine whether there are such rules and fix any problems.

The system is working under Unix and is using XFST as an external application. However, most of the ordinary computer users are using PC's. We are planning to write the interface and the implementation of the FST's again so that the program runs under MS Windows and available to ordinary users.

Numbers, dates, proper names and Russian words are not included. They require some more research and programming, but they have to be included in the system.

6.3. Conclusion

Although there is much influence of Anatolian Turkish over Crimean Tatar, it is a prototype for Kipchak oriented Turkic languages. Rules and systems developed for Crimean Tatar may be applied to other Kipchak languages such as Kazan Tatar, Kazakh or Kirgiz. Having morphological analysers ready in hand, we expect machine translation among these languages to be relatively easy.

Provided that the rules are ready in hand, coding them is not very difficult. However, in some cases, the grammatical rules themselves are not very clear. Having to write their language with Cyrillic letters, and after a long period of obligatory Russian education, many of the Turkic people lost the nuances in their languages. Most of the time, their use of their mother tongue was limited to daily life issues. Sometimes, the writers of the grammar books do not agree on some rules. For example, in some Crimean Tatar grammar books, it is said that, the known past morpheme -di is not written with u/ü. According to this, we can not use DH as past morpheme, but should use some similar morpheme such as DM instead. However, some other books say that it is valid to write -du / -dü. Actually, in Crimean Tatar language, there are very few cases where the symbols u, ü, o, ö appear in the syllables after second. Thus, even if we use DH, most of the time it will be represented as dI/di due to this nature of the language itself.

Also, in this thesis, we do not consider numbers, proper names and the words that are derived from some foreign languages especially from Russian. Since Crimean Tatars have lived under Russian rule for more than two hundred and twenty years, there are many Russian words appearing in the language. However, it is a very deep subject, which requires a considerable knowledge of Russian grammar and vocabulary.

To sum up, Crimean Tatar language is similar to Turkish in many aspects, although it has some variations. We tried to cover largest possible rules for a simple pure Crimean Tatar text, without any rule abiding proper names or foreign words. There is a lot of work to do in the area and during the ongoing research, we will find out the missing parts and recover them.

We believe that Crimean Tatar machine translation system may be a prototype for translation systems between Turkish and other Turkic languages. They have similar properties with Crimean Tatar and we believe rules and methods developed for Crimean Tatar may be applicable to other languages with relatively little changes.

BIBLIOGRAPHY

[1] Stephen Appleby, Marta Pombo Prol, Multilingual World Wide, Web, BT Technology Journal, Millenium Edition, Vol 18, No. 1.

[2] W.John Hutchins, Machine Translation: A Brief History, Concise history of the language sciences: from the Sumerians to the cognitivists. Edited by E.F.K.Koerner and R.E.Asher. Oxford: Pergamon Press, 1995. Pages 431-445]

[3] Hakkani-Tur, Dilek. Statistical Modelling of Agglutinative Languages, PhD Thesis, Department of Computer Engineering, Bilkent University, 2000.

[4] Daniel Jurafsky, James H. Martin, Speech and Language Processing, Prentice Hall, 2000

[5] D. Arnold, L. Balkan et.al, Machine Translation, Blackwell Publishers, Cambridge, MA, 1994.

[6] Vladislav Kubon, Jab Hajic, Jan Hric, Machine Translation of Very Close Languages, Unpublished Paper, 2001.

[7] Raül Canals, Anna Esteve, Alicia Garrido et.al. , interNOSTRUM: A Spanish-Catalan Machine Translation System , Machine Translation Review, Issue No. 11, December 2000 - pages 21-25.

[8] Mehryar Mohri. On Some Applications of Finite-State Automata Theory to Natural Language Processing . Natural Language Engineering, 2:1-20, 1996

[9] Lauri Karttunen, Ronald M. Kaplan, and Annie Zaenen, Two-Level Morphology with Composition, In the proceedings of Coling 92. International Conference on Computational Linguistics, Vol. I 141-148. July 25-28, 1992. Nantes, France.

[10] Antworth, E. L. PC-KIMMO: a two-level processor for morphological analysis. Occasional Publications in Academic Computing No. 16, Summer Institute of Linguistics, Dallas, Texas. 1990

[11] Ritchie, Graeme D. Languages Generated by Two-level Morphological Rules. Research Paper 496. Department of Artificial intelligence, University of Edinburgh, 1990

[12] Asanov, Ş. A., Garkavets, A. N., Useinov, S. M. Krimskotatarsko-Russkiy Slovar.

Radyanska Shkola, Kiev, 1988

[13] Çobanzade, Bekir. Qırımtatar İlm-i Sarfı. Qırım Hükümet Neşriyatı, Aqmescid, 1925

[14] Dermenci, E., Şemsedinoma, A. Qırımtatar Tili Dersligi. Qırım ASSR Devlet Neşriyatı, 1940

[15] Memetov, Ayder. Tatar Tili Grammatikasınıñ Praktikumı. Okituvçı, Taşkent, 1984

[16] Oflazer, Kemal. Morphological Analysis, chapter in Syntactic Wordclass Tagging Hans van Halteren, Editor, Kluwer Academic Publishers, 1998

[17] http://www.xrce.xerox.com/research/mltt/fst/home.en.html

[18] Oflazer, Kemal. Two-level Description of Turkish Morphology. Literary and Linguistic Computing, Vol. 9, No:2, 1994

[19] Kösoğlu, Nevzat et al. Başlangıcından Günümüze Kadar Türkiye Dışındaki Türk Edebiyatları Antolojisi 13: Kırım Türk-Tatar Edebiyatı, T.C. Kültür Bakanlığı, Ankara, 1999

[20] Kemal Oflazer and Gökhan Tür, Combining Hand-crafted Rules and Unsupervised Learning in Constraint-based Morphological Disambiguation in Proceedings of the ACL-SIGDAT Conference on Empirical Methods in Natural Language Processing, May 1996, Philadelphia, PA, USA

[21] Useinov, S. M., Mireev, V. A. Izuchayte Krimskotatarskiy Yazik. Tavriya, Simferopol, 1991

[22] Useinov, S. M. Qırımtatarca Rusça Luğat. Dialog, Akmescid, 1994

[23] Hutchins W. J., Somers H. L. An Introduction to Machine Translation Academic Press, London, 1992

[24] Y. Wilks, Systran: It Obviously Works, But How Much Can It Be Improved? Computers in Translation: A Practical Appraisal, Routledge, London, pp.166-188, 1992

[25] Maas H.D. The MT System SUSY Machine Translation Today: the State of the Art, Proceedings of the Third Lugano Tutorial, Edinburgh University Press, Edinburgh, pp. 209-246, 1987.

[26] Vauquois B., Boitet C. Automated Translation at GETA Grenoble: GETA, 1984.

[27] Landsbergen J. Isomorphic Grammars and Their Use in the Rosetta Translation System Machine Translation Today: the State of the Art, Proceedings of the Third Lugano Tutorial, Edinburgh University Press, Edinburgh, pp.351-377, 1987.

[28] Goodman K., Nirenburg S. The KBMT Project: A Case Study in Knowledge-Based Machine Translation Morgan Kaufmann, San Mateo, California, 1991.

[29] Brown P.F., Cocke J., Della Pietra S.A., Della Pietra V.J., Jellinek Lafferty J., Mercer R.L., Rossin P.S. a Statistical Approach to French/English Translation Proceedings, Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1988

[30] Brown P.F., Cocke J., Della Pietra S.A., Della Pietra V.J., Jellinek Lafferty J., Mercer R.L., Rossin P.S. a Statistical Approach to Language Translation Proceedings of the 12th International Conference on Computational Linguistics, Vargha D. (ed.) COLING Budapest, John von Neumann Society for Computing Sciences, pp.71-76, 1988

[31] Cicekli Ilyas, Guvenir H. Altay Learning Translation Templates from Bilingual Translation Examples, Applied Intelligence, Vol. 15, No. 1, (2001), pp: 57-76.

Appendices

Translation Rules

```
define P1sgtoPnon [P1sg -> Pnon || DB %+ Adj %+ PastPart %+
_  [% ]+ [?+ %+ Noun %+ ] ,,
Pnon -> P1sg || DB %+ Adj %+ PastPart %+ P1sg [% ]+ ?+ %+
Noun %+ [Inf %+]* [A3sg | A3pl] %+ _ ];

define P2sgtoPnon [P2sg -> Pnon || DB %+ Adj %+ PastPart %+
_  [% ]+ [?+ %+ Noun %+ ] ,,
Pnon -> P2sg || DB %+ Adj %+ PastPart %+ P2sg [% ]+ ?+ %+
Noun %+ [Inf %+]* [A3sg | A3pl] %+ _ ];

define P3sgtoPnon [P3sg -> Pnon || DB %+ Adj %+ PastPart %+
_  [% ]+ [?+ %+ Noun %+ ] ,,
Pnon -> P3sg || DB %+ Adj %+ PastPart %+ P3sg [% ]+ ?+ %+
Noun %+ [Inf %+]* [A3sg | A3pl] %+ _ ];

define P1pltoPnon [P1pl -> Pnon || DB %+ Adj %+ PastPart %+
_  [% ]+ [?+ %+ Noun %+ ] ,,
Pnon -> P1pl || DB %+ Adj %+ PastPart %+ P1pl [% ]+ ?+ %+
Noun %+ [Inf %+]* [A3sg | A3pl] %+ _ ];

define P2pltoPnon [P2pl -> Pnon || DB %+ Adj %+ PastPart %+
_  [% ]+ [?+ %+ Noun %+ ] ,,
Pnon -> P2pl || DB %+ Adj %+ PastPart %+ P2pl [% ]+ ?+ %+
Noun %+ [Inf %+]* [A3sg | A3pl] %+ _ ];

define P3pltoPnon [P3pl -> Pnon || DB %+ Adj %+ PastPart %+
_  [% ]+ [?+ %+ Noun %+ ],,
Pnon -> P3pl || DB %+ Adj %+ PastPart %+ P3pl [% ]+ ?+ %+
Noun %+ [Inf %+]* [A3sg | A3pl] %+ _ ];
```

```
define pastpart [P1sgtoPnon .o. P2sgtoPnon .o. P3sgtoPnon
.o. P1pltoPnon .o. P2pltoPnon .o. P3pltoPnon];

define zamanTOgen [%^ DB %+ Adj %+ PastPart %+ [P1sg | P2sg
| P3sg | P1pl | P2pl | P3pl] [% ]+ zaman %+ Noun %+ A3sg %+
Pnon %+ Nom -> %^ DB %+ Noun %+ PastPart %+ A3Sg %+ Pnon %+
Loc];

define durmak [dur -> tur || %+ Adv %+ AfterDoingSo [% ]+ _
];

define ipbasla [%^ DB %+ Noun %+ Inf %+ A3sg %+ Pnon %+ Dat
[% ]+ başla %+ Verb -> %^ DB %+ Adv %+ AfterDoingSo %  başla
%+ Verb];

define yanTOyaq [yan -> yaq || [bir | öbür] %+ Adj [% ]+ _];

define arakTOip [ByDoingSo -> AfterDoingSo];

define ileTOnen [%+ Nom [% ]+ ile %+ [Conj | Postp %+ PCNom]
-> %+ Ins];

define sart [Past %+ Cond  -> Narr %+ A3Sg %  ol %+ Verb %+
Pos %+ Desr ,
Narr %+ Cond  -> Narr %+ A3Sg %  ol %+ Verb %+ Pos %+ Desr ,
Desr %+ Past  -> Narr %+ A3Sg %  ol %+ Verb %+ Pos %+ Desr ,
Desr %+ Past  -> Narr %+ A3Sg %  ol %+ Verb %+ Pos %+ Desr ,
%+ [Aor | Fut] %+ Cond -> %+ Desr];
define kenTOgen [ %+ [Narr | Aor] %^ DB %+ Adv %+ While ->
%^ DB %+ Noun %+ PastPart %+ A3Sg %+ Pnon %+ Loc];

define esiTOacak [FeelLike -> FutPart];

define incaTOgen [%+ Adv %+ When -> %+ Noun %+ PastPart %+
A3Sg %+ Pnon %+ Loc];

define CaseChangingVerbs [Dat -> Acc || _ [% ]+ [bak | dürt
| vur | acı] %+ Verb ,, Dat -> Abl || _ [% ]+ [sor |
ısmarla] %+ Verb];

define Dativekadar [%+ Dat [% ]+ kadar %+ Postp %+ PCDat ->
%+ Nom %^ DB %+ Adv %+ Until];

define dikce [%^ DB %+ Adv %+ As -> %^ DB %+ Adj %+ PastPart
%+ Pnon %   %!
SayIn %+ Noun %+ A3sg %+ Pnon %+ Nom];

define quest [[% ]+ [mi | mu | mı | mü] %+ Ques [% ] @-> %+
Ques [% ]];

define removeSigns [A3sg -> A3Sg , A2sg -> A2Sg , A1sg ->
A1Sg , A1pl -> A1Pl, A2pl -> A2Pl , A3pl -> A3Pl , %! -> 0 ,
P3sg -> P3Sg , P2sg -> P2Sg , P1sg -> P1Sg , P1pl -> P1Pl,
P2pl -> P2Pl , P3pl -> P3Pl];

define olmakYapabilir [%^ DB %+ Verb %+ Able %+ Neg ->
%+ Pos %^ DB %+ Adv %+ AfterDoingSo %  ol %+ Verb %^ DB %+
Verb %+ Able %+ Neg];

define birlesikzaman [ Narr %+ Past -> Narr %+ A3sg [% ] e
%+ Verb %+ Pos %+ Past ,
Narr %+ A3pl %+ Past -> Narr %+ A3sg [% ] e %+ Verb %+ Pos
%+ Past %+ A3pl ,
Prog1 %+ Past -> Prog1 %+ A3sg [% ] e %+ Verb %+ Pos %+
Past ,
Prog1 %+ A3pl %+ Past -> Prog1 %+ A3sg [% ] e %+ Verb %+ Pos
%+ Past %+ A3pl , Aor %+ A3pl %+ Past -> Aor %+ A3sg [% ] e
%+ Verb %+ Pos %+ Past %+ A3pl ,
Aor %+ Past -> Aor %+ A3sg [% ] e %+ Verb %+ Pos %+ Past ,
Aor %+ Narr -> Aor %+ A3sg [% ] e %+ Verb %+ Pos %+ Narr ,
Aor %+ A3pl %+ Narr -> Aor %+ A3sg [% ] e %+ Verb %+ Pos %+
Narr %+ A3pl,
Prog1 %+ Narr -> Prog1 %+ A3sg [% ] e %+ Verb %+ Pos %+ Narr
,
Prog1 %+ A3lp %+ Narr -> Prog1 %+ A3sg [% ] e %+ Verb %+ Pos
%+ Narr %+ A3pl ,
Fut %+ A3pl %+ Past -> Fut %+ A3sg [% ] e %+ Verb %+ Pos %+
Past %+ A3pl ,
Fut %+ Past -> Fut %+ A3sg [% ] e %+ Verb %+ Pos %+ Past ,
Fut %+ A3pl %+ Narr -> Fut %+ A3sg [% ] e %+ Verb %+ Pos %+
Narr %+ A3pl ,
Fut %+ Narr -> Fut %+ A3sg [% ] e %+ Verb %+ Pos %+ Narr ];

define Coqplural [A3sg -> A3pl || [çok %+ Adj | birçok %+
Det ][% ]+ [?^<10] %+ Noun [%+ Inf]* %+ _];

define total1 [zamanTOgen .o. pastpart .o. quest .o. durmak
.o. yanTOyaq .o. kenTOgen .o. incaTOgen];

define total2 [CaseChangingVerbs .o. dikce .o. arakTOip .o.
sart .o. ipbasla];

define total3 [Dativekadar .o. birlesikzaman .o.
olmakYapabilir .o. esiTOacak .o. ileTOnen];

define total [total1 .o. total2 .o. total3 .o. removeSigns];
```

Morphotactic Rules

```
LEXICON Root
                NOUNS;
                VERBS;
```

```
               ADJECTIVES;
               PRONOUN;
               PROPER;
               CONJ;
               NUM;
               ADV;
               POSTP;
               INTERJ;
               DETERMINER;
               PUNCT;

LEXICON PRONOUN

men+Pron+PersP+A1Sg:men  PRONOUN-POS;
sen+Pron+PersP+A2Sg:sen  PRONOUN-POS;
o+Pron+PersP+A3Sg:o      PRONOUN-POS;
biz+Pron+PersP+A1Pl:biz  PRONOUN-POS;
biz+Pron+PersP+A1Pl:bizler PRONOUN-POS;
siz+Pron+PersP+A2Pl:siz  PRONOUN-POS;
o+Pron+PersP+A3Pl:olar   PRONOUN-POS;
men+Pron+PersP+A1Sg:maNa PRN-I-DAT;
sen+Pron+PersP+A2Sg:saNa PRN-I-DAT;
o+Pron+PersP+A3Sg:oNa    PRN-I-DAT;

          PRN-POS;
          PRN-DEM;
          PRN-Q;
          PRN-INDEF;
          PRN-REF;

LEXICON PRN-REF
Oz+Pron+ReflexP:Oz       PRN-REF-POST;

LEXICON PRN-REF-POST
```

```
+A3Sg+P3Sg+Nom:                FINAL;
+A1Sg+P1Sg:+Hm           PRN-REF-CASE;
+A2Sg+P2Sg:+HN           PRN-REF-CASE;
+A3Sg+P3Sg:+sH           PRN-REF-CASE;
+A1Pl+P1Pl:+HmHz         PRN-REF-POS;
+A2Pl+P2Pl:+HNHz         PRN-REF-POS;
+A3Pl+P3Pl:+lArH         PRN-REF-CASE;

LEXICON PRN-REF-CASE
+Dat:+VA                 CASE-1;
+Nom:                    CASE-1;
+Loc:+VDA                CASE-1-KI;
+Acc:+nM                 CASE-1;
+Abl:+VDAn               CASE-1;
+Ins:+nen                CASE-1;
+Equ:+cA                 CASE-1;

LEXICON PRN-REF-POS
+Dat:+KA                 CASE-1;
+Nom:                    CASE-1;
+Loc:+VDA                CASE-1-KI;
+Acc:+nM                 CASE-1;
+Abl:+VDAn               CASE-1;
+Ins:+nen                CASE-1;
+Equ:+cA                 CASE-1;
+Gen:+nMN                CASE-2;

LEXICON PRONOUN-POS
+Pnon+Nom:               FINAL;
+Pnon+Loc:+VDA           CASE-1-KI;
+Pnon+Acc:+nM            FINAL;
+Pnon+Dat:+KA            FINAL;
+Pnon+Abl:+VDAn          FINAL;
+Pnon+Ins:+nen           FINAL;
```

```
+Pnon+Equ:+cA             FINAL;

LEXICON PRN-Q
qaC+Pron+PrQ:qaC          PRN-Q-FOL;
nere+Pron+PrQ:nere        PRN-Q-FOL;
qaysI+Pron+PrQ:qaysI      PRN-Q-FOL;
ne+Pron+PrQ:ne            PRN-Q-FOL;

LEXICON PRN-Q-FOL
+A3Sg+P3Sg:+sH            PRN-Q-FOL-I;
+A3Sg+P1Pl:+HmHz          PRN-Q-FOL-I;
+A3Sg+P2Pl:+HnHz          PRN-Q-FOL-I;
+A3Sg+P2Sg:+Hn            PRN-Q-FOL-I;
+A3Sg+P3Sg:               PRN-Q-FOL-I;

LEXICON PRN-Q-FOL-I
+Nom:                     FINAL;
+Dat:+KA             FINAL;
+Loc:+VDA            FINAL;
+Abl:+VDAn           FINAL;
+Gen:+nMN            FINAL;

LEXICON PRN-INDEF
birev+Pron+PrInDef:birev       PRN-INDEF-POS;
biri+Pron+PrInDef:biri         PRN-INDEF-POS;
ep+Pron+PrInDef:ep             PRN-INDEF-POS-II;
ep+Pron+PrInDef:epi            PRN-INDEF-POS;
iCbir+Pron+PrInDef:iCbiri      PRN-INDEF-POS;
bazI+Pron+PrInDef:bazI         PRN-INDEF-POS;

LEXICON PRN-INDEF-POS
+A1Pl+P1Pl+Nom:+HmHz           FINAL;
+A2Pl+P2Pl+Nom:+HNHz           FINAL;
+A3Pl+P3Pl+Nom:+lArH           FINAL;
```

```
+A3Sg+P3Sg+Nom:+sH            FINAL;
+A3Sg+P3Sg+Gen:+nMN           FINAL;
+A3Sg+P3Sg+Acc:+nM            FINAL;
+A3Sg+P3Sg+Dat:+KA            FINAL;
+A3Sg+P3Sg+Loc:+VDA           FINAL;
+A3Sg+P3Sg+Abl:+VDAn          FINAL;


LEXICON PRN-INDEF-POS-II
+A3Sg+P3Sg+Nom:+ZH            FINAL;


LEXICON PRN-DEM
bu+Pron+DemonsP+A3Sg:bu            PRN-DEM-POS;
Su+Pron+DemonsP+A3Sg:Su            PRN-DEM-POS;
o+Pron+DemonsPn+A3Sg:o             PRN-DEM-POS;
bu+Pron+DemonsP+A3Pl:bular         PRN-DEM-POS;
Su+Pron+DemonsP+A3Pl:Sular         PRN-DEM-POS;
o+Pron+DemonsP+A3Pl:olar           PRN-DEM-POS;
mInavI+Pron+DemonsP+A3Sg:mInavI    PRN-DEM-POS;
mInavI+Pron+DemonsP+A3Pl:mInavlar  PRN-DEM-POS;
anavI+Pron+DemonsP+A3Sg:anavI      PRN-DEM-POS;
anavI+Pron+DemonsP+A3Pl:anavlar    PRN-DEM-POS;

bu+Pron+DemonsP+A3Sg:buNa     PRN-DEM-DAT;
Su+Pron+DemonsP+A3Sg:SuNa     PRN-DEM-DAT;
o+Pron+DemonsP+A3Sg:oNa       PRN-DEM-DAT;


LEXICON PRN-DEM-POS
+Pnon+Nom:          FINAL;
+Pnon+Gen:+nMN      FINAL;
+Pnon+Acc:+nM       FINAL;
+Pnon+Loc:+VDA      FINAL;
```

```
+Pnon+Abl:+VDAn     FINAL;
+Pnon+Ins:+nen      FINAL;
+Pnon+Dat:+KA       FINAL;

LEXICON PRN-DEM-DAT
+Pnon+Dat:          FINAL;

LEXICON PRN-POS
men+Pron+PersP+A1Sg:menim     PRN-POS-FOL;
sen+Pron+PersP+A2Sg:seniN     PRN-POS-FOL;
o+Pron+PersP+A3Sg:onIN        PRN-POS-FOL;
biz+Pron+PersP+A1Pl:bizim     PRN-POS-FOL;
siz+Pron+PersP+A2Pl:siziN     PRN-POS-FOL;
olar+Pron+PersP+A3Pl+Pnon+Gen:olarnIN FINAL;

LEXICON PRN-POS-FOL
+Pnon+Gen:               FINAL;
+Pnon+Equ:+cA            FINAL;



LEXICON PRN-I-DAT
+Pnon+Dat:               FINAL;



LEXICON POST-NOUN

+A3Sg:                   PLURAL;
+A3Pl:+lAr               PLURAL;

LEXICON PLURAL

+Pnon:                   POSSESSIVE;
+P3Sg:+sH                POSS-3;
+P1Sg:+Hm                POSSESSIVE;
```

```
+P2Sg:+HN                POSSESSIVE;
+P1Pl:+HmHz              POSSESSIVE;
+P2Pl:+HNHz              POSSESSIVE;

LEXICON POSSESSIVE

+Nom:                    CASE-1;
+Dat:+KA                 CASE-1;
+Ins:+nen                CASE-1;
+Loc:+DA                 CASE-1-KI;
+Abl:+DAn                CASE-1;
+Acc:+nM                 FINAL;
+Equ:+cA                 CASE-1;
+Gen:+nMN                CASE-2;

LEXICON CASE-1-KI

^DB+Det:+ki              CASE-1-KI-POST;
                         CASE-1;

LEXICON CASE-1-KI-POST
^DB+Noun+Zero:           POST-NOUN;
                         FINAL;

LEXICON CASE-2

^DB+Pron:+ki                  RELATIVE-PL;
                              CASE-1;

LEXICON RELATIVE-PL
+A3Pl:+lAr                    RELATIVE;
+A3Sg:                        RELATIVE;

LEXICON RELATIVE
```

```
+Loc:+VDA                     CASE-1-KI;
+Loc:+VDA                     CASE-1;
+Gen:+nMN                     CASE-1-KI;
+Gen:+nMN                     CASE-1;
+Acc:+nM                      FINAL;
+Dat:+nA                      CASE-1;
+Abl:+VDAn                    CASE-1;
+Ins:+nen                     CASE-1;
                              PLURAL;

LEXICON CASE-1
                              FINAL;
^DB+Verb+Zero+Pres+A1Sg:+Hm   FINAL;
^DB+Verb+Zero+Pres+A2Sg:+ZHN  FINAL;
^DB+Verb+Zero+Pres+A1Pl:+mHz  FINAL;
^DB+Verb+Zero+Pres+A2Sg:+ZHNHz     FINAL;
^DB+Verb+Zero+Pres+A2Pl:+ZHNHz     FINAL;
^DB+Verb+Zero+Pres+A3Pl:+lAr  FINAL;
^DB+Verb+Zero+Pres+Cop:+DMr   NOMINAL-VERB-4;
^DB+Verb+Zero+Cond:+yZA       PERSON-2;
^DB+Noun+Agt:+cH              POST-NOUN;
^DB+Noun+Ness:+lHK            POST-NOUN;
^DB+Adj+FitFor:+lHK           FINAL;
^DB+Noun+Small:+CHK           POST-NOUN;
^DB+Adj+With^DB+Noun+Zero:+lH POST-NOUN;
^DB+Adj+With:+lH              CASE-1;
^DB+Adj+Without:+ZHz          CASE-1;
^DB+Verb+Acquire:+lAn         POST-VERB;
^DB+Verb+Become:+lAS          POST-VERB;
^DB+Adv+Until:+KAce           FINAL;

LEXICON NOMINAL-VERB-4
```

```
+A3Sg:                         FINAL;
+A3Pl:+lAr                     FINAL;


LEXICON POSS-3

+Nom:                          CASE-1;
+Dat:+nA                       CASE-1;
+Ins:+nen                      CASE-1;
+Loc:+VDA                      CASE-1-KI;
+Abl:+VDAn                     CASE-1;
+Acc:+nM                       FINAL;
+Equ:+ncA                      CASE-1;
+Gen:+nMN                      CASE-2;



LEXICON FINAL
+Ques:+mH                      #;
                               #;



LEXICON POST-VERB
                               NEGATION;
                               ABILITY;
                               CAUS-PASS;

LEXICON CAUS-PASS
^DB+Verb+Pass:+Hl              POST-VERB;
^DB+Verb+Caus:+DMr             POST-VERB;
^DB+Verb+Caus:+t               CAUS-T;
^DB+Verb+Recip:+HS             POST-VERB;
```

```
LEXICON CAUS-T
^DB+Verb+Caus:+DMr              CAUS-PASS;
^DB+Verb+Caus:+DMr              POST-VERB;



LEXICON ABILITY
+Pos^DB+Verb+Able:+yAbil        VERBAL-STEM;
^DB+Verb+Able+Neg:+yAmA         VERBAL-STEM;

LEXICON NEGATION
+Pos:                           VERBAL-STEM;
+Pos:                           AOR;
+Neg:+mA                        NEG-MA;

LEXICON AOR
+Aor:+Hr                        COND;
+Aor:+Ar                        COND;



LEXICON NEG-MA
                                VERBAL-STEM;
+Aor+A1Sg:+m                    FINAL;
+Aor+A2Sg:+zsHN                 FINAL;
+Aor:+z                         NEG-AORIST;
+Aor+A1Pl:+mHz                  FINAL;
+Aor+A2Pl:+zsHNHz               FINAL;
+Aor+A3Pl:+zlAr                 FINAL;
+Aor^DB+Adj+Zero:+z             FINAL;

LEXICON NEG-AORIST
+A3Sg:                          FINAL;
+Cond:+ZA                       PERSON-2;

LEXICON  VERBAL-STEM
```

```
^DB+Noun+Inf:+mAK             POST-NOUN;
^DB+Noun+Inf:+mA              POST-NOUN;
^DB+Noun+Inf:+Yv              POST-NOUN;
^DB+Noun+Inf:+yHS             POST-NOUN;
^DB+Adj+Agt:+yHcH             FINAL;
^DB+Noun+Agt:+yHcH            POST-NOUN;
:+KAn                         PAST-PART;
:+KAn                         PAST-PART-N;
:+AcAK                        FUT-PART;
:+AcAK                        FUT-PART-N;
+Prog1:+A                     COND;
+Past:+DM                     COND-DI;
+Neces:+mAlH                  COND;
+Prog2:+mAKtA                 COND;
+Imp:                         IMP;
+Desr:+ZA                     PERSON-2;
^DB+Adv+ByDoingSo:+yArAK      FINAL;
^DB+Adv+AfterDoingSo:+Mp      FINAL;
^DB+Adv+Until:+KAnce          FINAL;
+Duration:+Ayata              COND;

LEXICON COND-DI
                              PERSON-2;
+Cond:+yZA                    PERSON-2;

LEXICON COND
                              PERSON;
+Cond:+yZA                    PERSON-2;
+Cop:+DMr                     PERSON;

LEXICON IMP
+A2Sg:                        FINAL;
+A2Pl:+HNHz                   FINAL;
+A3Sg:+ZHn                    FINAL;
```

```
+A3Pl:ZHnlAr                   FINAL;

LEXICON FUT-PART
+Fut:                          COND;
^DB+Adj+FutPart:               POSS-ADJ;

LEXICON FUT-PART-N
^DB+Noun+FutPart:              POST-NOUN;



LEXICON PAST-PART
+Narr:                         COND;
^DB+Adj+PastPart:              POSS-ADJ;

LEXICON PAST-PART-N
^DB+Noun+PastPart:             POST-NOUN;

LEXICON POSS-ADJ
+Pnon:                         FINAL;
+P1Sg:+Hm                      FINAL;
+P2Sg:+HN                      FINAL;
+P3Sg:+sH                      FINAL;
+P1Pl:+HmHz                    FINAL;
+P2Pl:+HNHz                    FINAL;
+P3Pl:+lArH                    FINAL;


LEXICON PERSON-2
+A3Sg:                         FINAL;
+A1Sg:+m                       FINAL;
+A2Sg:+N                       FINAL;
+A1Pl:+K                       FINAL;
+A2Pl:+NHz                     FINAL;
+A3Pl:+lAr                     FINAL;
```

```
LEXICON PERSON
+A3Sg:                        FINAL;
+A1Sg:+Hm                     FINAL;
+A2Sg:+ZHN                    FINAL;
+A1Pl:+mHz                    FINAL;
+A2Pl:+ZHNHz                  FINAL;
+A3Pl:+lAr                    FINAL;

LEXICON POST-ADJ
                              FINAL;
                              POST-ADJ-PLURAL;
^DB+Verb+Zero+Pres+A1Sg:+Hm   FINAL;
^DB+Verb+Zero+Pres+A2Sg:+ZHN  FINAL;
^DB+Verb+Zero+Pres+A1Pl:+mHz  FINAL;
^DB+Verb+Zero+Pres+A2Pl:+ZHNHz     FINAL;
^DB+Verb+Zero+Pres+A2Sg:+ZHNHz     FINAL;
^DB+Verb+Zero+Pres+A3Pl:+lAr  FINAL;
^DB+Verb+Zero+Pres+Cop:+DMr   NOMINAL-VERB-4;
^DB+Verb+Zero+Cond:+yZA       PERSON-2;
^DB+Noun+Agt:+cH              POST-NOUN;
^DB+Noun+Ness:+lHK            POST-NOUN;
^DB+Noun+Zero^DB+Adj+With^DB+Noun+Zero:+lH   POST-NOUN;
^DB+Noun+Zero^DB+Adj+With:+lH      CASE-1;
^DB+Adj+Without:+ZHz               CASE-1;
^DB+Verb+Acquire:+lAn              POST-VERB;
^DB+Verb+Become:+lAS               POST-VERB;

LEXICON POST-ADJ-PLURAL

^DB+Noun+Zero+A3sg:                POST-ADJ-POSS;
^DB+Noun+Zero+A3pl:+lAr            PLURAL;
^DB+Noun+Zero+A3pl+P3pl:+lArH      POSSESSIVE;
```

```
LEXICON POST-ADJ-POSS

+Pnon:                        POST-ADJ-CASE;
+P3sg:+sH                     POSS-3;
+P1sg:+Hm                     POSSESSIVE;
+P2sg:+HN                     POSSESSIVE;
+P1pl:+HmHz                   POSSESSIVE;
+P2pl:+HNHz                   POSSESSIVE;
+P3pl:+lArH                   POSSESSIVE;


LEXICON POST-ADJ-CASE
+Dat:+KA                      CASE-1;
+Ins:+nen                     CASE-1;
+Loc:+DA                      CASE-1-KI;
+Abl:+DAn                     CASE-1;
+Acc:+nM                      FINAL;
+Equ:+cA                      CASE-1;
+Gen:+nMN                     CASE-2;
```

Translation Examples

The following are example translations by the system. The correct translations are marked with a *.

akşam eve geleceğiz

2 aqSam evge kelecekmiz *

1 aqSam evge istiqbalmIz

uykusunun arasında konuşuyor

1 yuqusInIN arasInda qonuSa *

1 yuqusInIN arasInda laf ete *

çiçekleri suladıkça büyüyorlar

1 CeCekleri suvarGan sayIn Oseler *

1 CeCeklerni suvarGan sayIn Oseler *

2 (..) suvarGan sayIn Oseler

eve geldiğim zaman seni görmek istiyorum

1 evge kelgende seni kOrUv/kOrmek/kOrme/kOrUS isteyim *

1 evge kelgenim vaqIt seni kOrUv/kOrmek/kOrme/kOrUS isteyim

resmine baktıkça onu hatırlıyorum

3 resimini baqqan sayIn (..) hatIrlayIm

3 resimiNni baqqan sayIn (..) hatIrlayIm

3 (..) baqqan sayIn (..) hatIrlayIm

1 resimini baqqan sayIn onI hatIrlayIm *

1 resimiNni baqqan sayIn onI hatIrlayIm *

1 (..) baqqan sayIn onI hatIrlayIm

kalem ile yazmayı öğretti

1 (..) ilge yazmaqnI/yazmanI/yazuvnI/yazISnI Ogretti

1 qalem ilge yazmaqnI/yazmanI/yazuvnI/yazISnI Ogretti

2 (..) yazmaqnI/yazmanI/yazuvnI/yazISnI Ogretti

2 qalemnen yazmaqnI/yazmanI/yazuvnI/yazISnI Ogretti *

1 (..) ilge (..) Ogretti

1 qalem ilge (..) Ogretti

2 (..) (..) Ogretti

2 qalemnen (..) Ogretti

kedi yavrusunu bahçede dolaştırıyordu

2 mISIq balasInI azbarda/baGCada (..) edi

1 mISIq balasInI azbarda/baGCada aylandIra edi

gözlerinin böyle sevinçle parladığını görmemiştim

2 kOzleriniN bOyle quvanCnen yIltIraGanInI kOrmegen edim *

2 kOzleriNniN bOyle quvanCnen yIltIraGanInI kOrmegen edim *

4 (..) bOyle quvanCnen yIltIraGanInI kOrmegen edim

2 kOzleriniN bOyle quvanCnen yIltIraGanINnI kOrmegen edim

2 kOzleriNniN bOyle quvanCnen yIltIraGanINnI kOrmegen edim

4 (..) bOyle quvanCnen yIltIraGanINnI kOrmegen edim

2 kOzleriniN bOyle quvanCnen yIltIraGanInI (..)

2 kOzleriNniN bOyle quvanCnen yIltIraGanInI (..)

4 (..) bOyle quvanCnen yIltIraGanInI (..)

2 kOzleriniN bOyle quvanCnen yIltIraGanINnI (..)

2 kOzleriNniN bOyle quvanCnen yIltIraGanINnI (..)

4 (..) bOyle quvanCnen yIltIraGanINnI (..)

bugün hava serin olacağa benziyor

3 bu kUn (..) (..) olacaqqa oSay

3 bu kUn ava (..) olacaqqa oSay

1 bu kUn (..) seriNiz olacaqqa oSay

1 bu kUn ava seriNiz olacaqqa oSay

1 bu kUn (..) salqIn olacaqqa oSay

1 bu kUn ava salqIn olacaqqa oSay *

kırlangıçların mahallesinde serçeler kenarda kalır

1 qarIlGaClarnIN mallesinde torGaylar Cette qalIr/qalar *

1 qarIlGaClarIN mallesinde torGaylar Cette qalIr/qalar

1 qarIlGaClarnIN mallesinde torGaylar Cette (..)

1 qarIlGaClarIN mallesinde torGaylar Cette (..)

doktor kırık kemiği çekince bağırdı

4 dohtur sInIq sUyegi (..) (..)

4 dohtur (..) sUyegi (..) (..)

4 dohtur sInIq sUyekni (..) (..)

4 dohtur (..) sUyekni (..) (..)

1 dohtur sInIq sUyegi tartqanda (..)

1 dohtur (..) sUyegi tartqanda (..)

1 dohtur sInIq sUyekni tartqanda (..)

1 dohtur (..) sUyekni tartqanda (..)

4 dohtur sInIq sUyegi (..) cekirdi/qICIrdI/baqIrdI

4 dohtur (..) sUyegi (..) cekirdi/qICIrdI/baqIrdI

4 dohtur sInIq sUyekni (..) cekirdi/qICIrdI/baqIrdI

4 dohtur (..) sUyekni (..) cekirdi/qICIrdI/baqIrdI

1 dohtur sInIq sUyegi tartqanda cekirdi/qICIrdI/baqIrdI

1 dohtur (..) sUyegi tartqanda cekirdi/qICIrdI/baqIrdI

1 dohtur sInIq sUyekni tartqanda cekirdi/qICIrdI/baqIrdI *

1 dohtur (..) sUyekni tartqanda cekirdi/qICIrdI/baqIrdI

birçok insan erkenden kalkıp işe gidiyorlar

2 bir Coq adamlar (..) turIp (..) baralar/keteler/kedeler

1 bir Coq adamlar erteden turIp (..) baralar/keteler/kedeler

2 bir Coq adamlar (..) turIp iSke baralar/keteler/kedeler

1 bir Coq adamlar erteden turIp iSke baralar/keteler/kedeler *

ırmağın kenarında kuşlara bakarak şarkı söylerdi

1 OzeniN Cetinde/Cedinde quSlarnI baqIp yIr aytIr/aytar/aydIr/aydar edi

1 OzenniN Cetinde/Cedinde quSlarnI baqIp yIr aytIr/aytar/aydIr/aydar edi *

1 OzeniN CetiNde/CediNde quSlarnI baqIp yIr aytIr/aytar/aydIr/aydar edi

1 OzenniN CetiNde/CediNde quSlarnI baqIp yIr aytIr/aytar/aydIr/aydar edi

2 OzeniN Cetinde/Cedinde quSlarnI baqIp (..) aytIr/aytar/aydIr/aydar edi

2 OzenniN Cetinde/Cedinde quSlarnI baqIp (..) aytIr/aytar/aydIr/aydar edi

2 OzeniN CetiNde/CediNde quSlarnI baqIp (..) aytIr/aytar/aydIr/aydar edi

2 OzenniN CetiNde/CediNde quSlarnI baqIp (..) aytIr/aytar/aydIr/aydar edi

1 OzeniN Cetinde/Cedinde quSlarnI baqIp yIr (..)

1 OzenniN Cetinde/Cedinde quSlarnI baqIp yIr (..)

1 OzeniN CetiNde/CediNde quSlarnI baqIp yIr (..)

1 OzenniN CetiNde/CediNde quSlarnI baqIp yIr (..)

2 OzeniN Cetinde/Cedinde quSlarnI baqIp (..) (..)

2 OzenniN Cetinde/Cedinde quSlarnI baqIp (..) (..)

2 OzeniN CetiNde/CediNde quSlarnI baqIp (..) (..)

2 OzenniN CetiNde/CediNde quSlarnI baqIp (..) (..)

ilginç düşünceleriyle ninesini meraklandırdı

1 meraqlI tUSUncelerinen qartanayInI meraqlandIrdI *

2 meraqlI (..) qartanayInI meraqlandIrdI

ince elbiseler içinde neredeyse üşütüyorduk

1 (..) urbalar iCinde (..) suvuqlana/suvuqlandIra/toNdIra edik

1 ince urbalar iCinde (..) suvuqlana/suvuqlandIra/toNdIra edik

1 (..) urbalar iCiNde (..) suvuqlana/suvuqlandIra/toNdIra edik

1 ince urbalar iCiNde (..) suvuqlana/suvuqlandIra/toNdIra edik

2 (..) urbalar iCinde temiz suvuqlana/suvuqlandIra/toNdIra edik

2 ince urbalar iCinde temiz suvuqlana/suvuqlandIra/toNdIra edik *

2 (..) urbalar iCiNde temiz suvuqlana/suvuqlandIra/toNdIra edik

2 ince urbalar iCiNde temiz suvuqlana/suvuqlandIra/toNdIra edik

sabaha kadar horuldayarak uyudular

1 sabahGa qadar huruldap yuqladIlar

1 sabahGace huruldap yuqladIlar *

yol açılınca arabalar gelmeye başladılar .

1 yol aCIlGanda/acIlGanda maSinalar (..) baSladIlar .

1 (..) aCIlGanda/acIlGanda maSinalar (..) baSladIlar .

1 yol (..) maSinalar (..) baSladIlar .

1 (..) (..) maSinalar (..) baSladIlar .

1 yol aCIlGanda/acIlGanda maSinalar kelip baSladIlar . *

1 (..) aCIlGanda/acIlGanda maSinalar kelip baSladIlar .

1 yol (..) maSinalar kelip baSladIlar .

1 (..) (..) maSinalar kelip baSladIlar .

Kırım'da kaldığımız otel çok güzeldir

2 qIrImda qalGan gastinitsamIz ziyade dUlberdir *

2 qIrImda qalGanImIz gastinitsa ziyade dUlberdir

2 qIrImda qalGanImmIz gastinitsa ziyade dUlberdir

2 qIrImda qalGan gastinitsamIz Coq dUlberdir *

2 qIrImda qalGanImIz gastinitsa Coq dUlberdir

2 qIrImda qalGanImmIz gastinitsa Coq dUlberdir

2 qIrImda qalGan gastinitsamIz (..) dUlberdir

2 qIrImda qalGanImIz gastinitsa (..) dUlberdir

2 qIrImda qalGanImmIz gastinitsa (..) dUlberdir