A Machine Translation System Between a Pair of Closely Related Languages

Kemal Altintas
Dept. of Computer Engineering
Bilkent University
email:kemal@cs.bilkent.edu.tr

Ilyas Cicekli
Dept. of Computer Science
University of Central Florida
email: ilyas@cs.ucf.edu

ABSTRACT: 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 may suffice most of the time. Usually a semantic analysis may not be needed. This paper 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.

1. Introduction

Using computers for translation has been interesting for people since the invention of computers. During the period after the Second World War, both the United States and the Soviet Union supported projects on machine translation in order to be able to read each other's documents. Later, the importance of machine translation for the replacement of human translators was discovered due to economic reasons. Parties from industry supported and implemented MT systems. Today, many researchers conducting research towards successful machine translation systems among many world languages.

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. 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.

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 into English. Machine translation between close pair of languages was left rather untouched and Turkish [2] and Turkic languages have not attracted too much attention.

This paper explains a finite state method 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 semantics out of the scene, so the research can focus on other issues like the translation of grammar.

Turkish and Crimean Tatar, being two close languages, may be a model for machine translation between closely related languages. Most parts of the grammars and vocabulary of the two languages are common. Usually, the differences in grammars are at the morpheme level, where the morphemes for a specific grammar construct or the order of appearance of morphemes may differ. Most of the root words are used in both languages with smaller differences. Methods developed for this pair of languages can easily be applied to other Turkic languages. Also, similar research on language pairs Czech-Slovak [5] and Spanish-Catalan [2] show that the methods described in this paper are applicable to other closely related language pairs.

The rest of the paper is organised as follows: Section 2 explains the details of the translation process between closely related languages. The translation system is explained in Section 3. Implementation and limitations are discussed in Section 4. Section 5 concludes the paper.

2. MT 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. Some languages may not have certain grammatical structures that are present in the target language. For example, Turkish does not have an explicit perfect tense 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 languages are almost always the languages of people who have similar cultures and common historical roots. 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 usually 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]. 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 Turkish and Uzbek, any machine translation system translating from English to Turkish will also enable us to translate from English to Uzbek. Implementing a system translating from Turkish to Uzbek is easier than developing a system translating from English to Uzbek. So, with lesser effort, we can have a system that is capable of translating from English to several Turkic languages.

In MT of close languages, most of the time a lexical analysis supported by some translation rules may be sufficient, and a semantic analysis may not be required. The number of translation rules is smaller than those of translation between unrelated languages. As a result, hand coding the rules is easier.

A translation system for closely related languages may need morphological analysis and morphological disambiguation tools for the source language, domain specific and general translation lexicons, and morphological generator for the target language. In this case, it will be a word-for-word translation system and it will suggest no translation rules for the grammar. However, even when the languages are very close to each other, there are some differences in the grammars in addition to vocabularies. Otherwise, it would be hard to say the two languages to be different and we could only talk about the same language written with different word domains. We believe that a module to make the necessary translations for the grammars should be included in the system.

Two other systems, [2,5] claim to be using similar methodologies in their translations between closely related languages. The system translating between Czech and Slovak [5] claims to be using a translation memory which stores the previously translated sentence pairs. When a human translator starts translating a new sentence, the system checks the translation memory for the sentence. If it appears in the memory, it is suggested to the user. The user is free to use, modify or reject it. The Spanish-Catalan system [2] uses a similar idea that we used in our system. The basic difference is that they apply the bilingual dictionary before the grammatical translation module. It is a matter of choice and as long as the grammatical translation rules are crafted accordingly, and it does not affect the system performance.

3. Translation System

Translation from Turkish to Crimean Tatar is in general disambiguated word-for-word translation. The grammars of the two languages are very similar, and each morpheme usually 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 usually preserved in Crimean Tatar.

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

- Morphological analysis of Turkish text
- Morphological disambiguation
- 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.

Turkish morphological analyser [7] used in the accepts single word inputs and gives all possible analyses of the input word without considering the context information. A typical output of the morphological analyser can be seen in the analysis of the word "evlerimizden" (from our houses):

evlerimizden
ev+Noun+A3pl+P1pl+Abl

This result indicates that this word is a third person plural noun with possessive 1st person plural and in ablative case.

The translation system was developed using XEROX Finite State Tools (XFST) so it uses the XFST syntax for the translation rules [6]. 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 translation rule is as follows:

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

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

We can categorize the translation rules into the following categories:

1. Most Trivial – No Change

This set of rules includes no change in the roots or in the morphemes. All the roots and the morphemes in Turkish are conserved. No translation rules are applied for these cases

2. Root Change

Only the root is changed, and the rest of the morphemes are not changed. These rules are basically from the bilingual dictionary.

3. Morpheme Change

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

4. Root and Morpheme Change

In addition to the root of the source structure, some of its morphemes are changed too. Actually, these are mostly the word, which are expressed different word category in the target language. The root and the related morphemes are mapped to target morphemes.

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. In this case, the dative morpheme of a noun is changed into ablative if it precedes the certain verbs such as "sormak" (to ask) and "ısmarlamak" (to order) in Turkish.

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" (many) can be singular in Turkish but it cannot be singular in Crimean Tatar.

7. More than One Word Maps to One Word

Sometimes more than one word should be expressed with a single word or one word corresponds to two or more words. For example, "yırlamak" (to sing) in Crimean Tatar is expressed as "şarkı/türkü söylemek" in Turkish.

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, 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".

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 at 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, 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.

Before this output is fed to Crimean Tatar morphological 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 morphological processor. 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 the grammar rules and the root words are translated, the lexical form should be changed to the surface form using a morphological processor [1] for Crimean Tatar. The morphological processor was developed as a part of this project and it 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.

4. Evaluation – Results

4.1 Implementation

The system is implemented using XEROX Finite State Tools for language engineering [6]. 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 was written in Java language. It reads 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.

The input sentence for the translation system 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. As a result, we get all possible combinations of the sentences derived from the morphological analyses of the input words.

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. After the roots are translated, the output is given to the Crimean Tatar morphological processor to generate the surface form.

4.2 Examples

The followings are example translations by the system. The numbers in front of each sentence shows how many times this sentence was generated from separate lexical forms. For example, a 2 in front of a sentence says that two different lexical forms led to the same surface form. The correct translations are marked with a *.

akşam eve geleceğiz
(We will come home in the evening)
2 aqSam evge kelecekmiz *
1 aqSam evge istiqbalmIz

çiçeği suladıkça büyüyor
(The flower/his flower is growing as it is watered)
1 CeCegi suvarGan sayIn Ose *
1 CeCeklni suvarGan sayIn Ose *

4.3 Limatations

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. 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. The coverage of our morphological disambiguator should be improved.

Another limitation of the system is that, although the languages are very similar, there are some problems, which cannot be overcome with a finite state translation tool. 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 easily 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 gel*ir*im” (If you go, I will come) can be translated as “Siz ketseñiz men de kel*ir*im”. However, the same verb in the same tense in the sentence “Ben de bazen gel*ir*im” (I also sometimes come) is translated into “Men de kimerde kel*e*m” (I also am sometimes coming [may not be grammatically correct in English]). There is not a rule for this and it cannot be determined easily even with a parse of the sentence.

One problem with Turkic languages is that verbs do not have regular phonetic 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 "akmak" (to flow) is made causative by –ıt as "akıtmak" (to cause something flow) although phonetically it is similar to "bakmak".

5. Conclusion

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

Experiments with two other systems show that similar methods may be applied to other closely related languages. Provided that the rules are ready in hand, coding them is not very difficult. Finite state transducers for morphological process, translation of grammar rules and bilingual dictionaries may be coded relatively easily with finite compilers.

Close languages are languages of people who usually share a historical background and a common culture. The grammars of such languages do not differ very much. For agglutinative languages, the differences may be expected at the morpheme level. For other languages, it is expected that for any word or morpheme in the source language, a corresponding word and/or morpheme can be found using finite state techniques. Since the cultures and the way of thinking of the people who are speaking close languages are similar, the concepts and terms are usually similar. Both morphological and semantic ambiguities are usually preserved and a morphological disambiguator is usually sufficient. A semantic analyser and a parser may not be needed most of the time.

To sum up, Crimean Tatar language is similar to Turkish, although it has many variations. We tried to cover largest possible rules for a simple pure Crimean Tatar text, without any rule abiding proper names or foreign words.

We believe that Crimean Tatar machine translation system may be a prototype for translation systems between close pair of languages, especially for 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.

References

[1] Altintas, Kemal, Cicekli, Ilyas: 2001, A Morphological Analyser for Crimean Tatar. In *Proceedings of Turkish Artificial Intelligence and Neural Network Conference (TAINN2001)*, North Cyprus

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

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

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

[5] Kubon, Vladislav, Hajic, Jab, Hric, Jan: 2000, Machine Translation of Very Close Languages, in *ANLP-NAACL2000*, Washington.

[6] MLTT Finite State Homepage, (http://www.xrce.xerox.com/research/mltt/fst/home.en.html)

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