A Prototype Machine Translation System Between Turkmen and Turkish

A. Cüneyd TANTUĞ [1] , Eşref ADALI [1], Kemal OFLAZER [2]

[1] İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü
34469, Maslak, İstanbul, Türkiye
{cuneyd, adali}@cs.itu.edu.tr

[2] Sabancı Üniversitesi Doğa Bilimleri Fakültesi
Bilgisayar Mühendisliği Bölümü
34956, Orhanlı, Tuzla, Türkiye
oflazer@sabanciuniv.edu

Abstract. In this work, we present a prototype system for translation of Turkmen texts into Turkish. Although machine translation (MT) is a very hard task, it is easier to implement a MT system between very close language pairs which have similar syntactic structure and word order. We implement a direct translation system between Turkmen and Turkish which performs a word-to-word transfer. We also use a Turkish Language Model to find the most probable Turkish sentence among all possible candidate translations generated by our system.

1. Introduction

Robust machine translation (MT) is one of the earliest and most important goals in natural language processing (NLP) field and there exists so much work in the history. Despite the huge amount of funds invested and efforts, even today there are no such systems that can be safely used to translate texts from various domains (newspapers, stories, daily talks and etc.). The problem of automatic translation of arbitrary texts from one language to another is still far to be solved [1]. The fundamental reason for that is the complexity of the task itself. In order to produce a successful translation, one should require techniques from almost all areas of the NLP including morphology, syntax, semantics and discourse analysis.

Since there is no computational model for a full-automatic efficient MT task, most of the current MT systems focus on simple MT problems like rough translation (a rough translation of the input text to understand the general topic and what is said), computer aided translations – CAT (the systems that helps human editors in translation tasks) and MT in limited domains (like translation of weather reports).

1.1. MT Between Very Close Languages

MT between very close language pairs are expected to be easier than MT between different language pairs. The word-to-word translation model can work fine between two languages which have nearly same syntactic structure and similar word order.

The first attempt to translate texts between two close languages was RUSLAN which started in 1985 and was terminated in 1990 because of the insufficient funding [2]. This system aims the automatic translation of the documentation in the domain of mainframe operating systems from Czech to Russian. The system was designed on a rule-based-transfer approach which involves morphological and syntactic analysis of Czech, the transfer rules and a syntactic and morphological generation of Russian. As the evaluation of this system, it was reported that about 40% of the input sentences were translated correctly, about 40% of the input sentences were translated with errors which can be correctable by human editors and about %20 of the input needs re-translation.

Hajič and colleagues presented another MT system between two related Slavic languages: Czech and Slovak [3]. This system, ČESILKO, allows translation only from Czech to Slovak. Like its ancestor work RUSLAN, this MT system is also based on a word-for-word translation approach. The major parts of this MT system are morphological analysis and disambiguation of Czech, bilingual transfer dictionaries (domain related and general purpose) and morphological synthesis of Slovak. In the conclusion part it is claimed that the translation can be easily done between other Slavic languages like Czech-to-Polish in the same manner.

Another MT system between two close languages is *interNOSTRUM* [4]. This system translates texts from various domains between both Spanish-to-Catalan and Catalan-to-Spanish. The system has source language (SL) morphological analyzer, SL tagger, a pattern processor which incorporates with a bilingual dictionary, target language (TL) morphological generator and a post-generator. Most of these modules are implemented as finite state transducers that produces the translation with a high execution speed. *interNOSTRUM* is an indirect MT system using an advanced morphological transfer strategy. It is stated that, the error rates (which are measured as the number of words that have to be inserted, deleted or substituted per 100 words to render the the text acceptable) are around 5% in the Spanish-to-Catalan direction and somewhat worse in the Catalan-to-Spanish direction. In Spain, a daily newspaper, Periódico de Catalunya (www.elperiodico.com), is translated into Catalan or Spanish through this MT system. This may be the first full automated MT translation task for unrestricted test with very successful results.

A MT system between Crimean Tatar and Turkish, which are both Turkic languages, was implemented in 2002 [5]. This system was a word-to-word translation system which uses a Crimean Tatar morphological analyser, transfer rules and a Turkish morphological generator. Despite the fact that the outputs of the system were ambiguous since there was not any disambiguation module in the system, it shows that MT between Turkic languages is a promising area and fully-automated translation systems between Turkic languages can be implemented.

2. Turkic Languages

Turkish has an agglutinative morphology with productive inflectional and derivational suffixes. Because of the suffixes can be added consecutively, one word can convey a lot of information like possessive information, number/person agreement (singular/plural) information, case information, mood and etc.

Turkish and other Turkic languages like Azerbaijani, Turkmen, Uzbek, Kazakh, Kyrgyz, Tatar, Chuvash, Uyghur are all in Ural-Altaic language family. The Turkic languages are also agglutinative just like Turkish and they share common grammatical rules and words. These languages are relative languages and close to each other more than any other language like English or French though the fact that they use different alphabets (some of the alphabets are Arabic or Cyrilic based). The existence of similar grammatical structures and words does not mean that anyone who knows one of the Turkic languages can understand all other Turkic languages. Although one can catch the common or similar words and try to estimate the meaning of the sentence, mostly the real meaning of the sentence is very far away from the estimation. This is generally because of the different usage of common words and morphological structures.

The goal of our project is implementing a machine translation system between Turkish and Turkmen. We have chosen Turkmen language as the first step because it is one the closest language to Turkish and they are both classified in the same sub-tree of Altaic Languages by SIL [6].

3. The Translation System

Our translation system is a direct translation system since the syntactic structure of Turkish and Turkmen languages are nearly same. Additionally these languages have very similar word order that means almost no change is necessary in word places. The main differences between these languages are the morphological differences and of course, different words. So word-by-word translation works fine, there is no need to have a parse tree which generally increases ambiguity.

The translation system has 5 main blocks (Fig-1). The process begins with the tokenization of the Turkmen sentence.

The second block performs the morphological analysis of the tokens in Turkmen language which decomposes the surface form into the word root and other suffixes. The results of this block have morphological ambiguities which mean that there can be more than one result.

The lexical transfer block translates the roots of Turkmen words into their Turkish counterparts by using a bilingual dictionary. Additionally, this block applies transfer rules which transform the morphological structures into the form which the Turkish morphological generator can process.

In the Turkish Morphological Generator block, the input tokens translated to Turkish morphological representations are processed and their surface forms are generated.

The last block of the translation system uses a Turkish language model in order to disambiguate the resulting sentences. Even this non-complex translation procedure has two main sources of ambiguity. The first one is the morphological ambiguity that comes from the Turkmen morphological analysis phase. The second ambiguity source is the lexical transfer phase in which more than one Turkish root can be found as the counterpart of a Turkmen root. These ambiguities are disambiguated by using a Turkish language model.

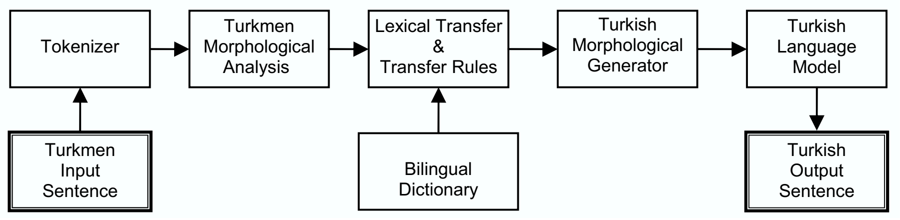

Fig. 1. Translation System Blocks

3.1. Turkmen Morpohological Analyzer

Although both Turkish and Turkmen are morphologically similar, there are some major divergencies. One of them is the existence of some Turkmen tenses which have no similar tenses in Turkish. For example the suffixes "*+makçy/+mekçi*" (literally "*thinking/planning to do sth.*") represent a mood that doesn't exist in Turkish. In Turkmen, some tenses (like future tense *"+jak/+jek"*) and moods (like necessity *"+malı/+meli"*) suppresses the person-agreement information while the same Turkish tenses or moods require this feature.

The first step of the project was planned as the design of a Turkmen morphological analyzer. The lack of NLP related work in Turkmen language makes us implement a finite-state based two-level Turkmen morphological analysis component. We have used Xerox finite-state tools [7] in order to build a high-speed morphological analyzer.

3.2. Lexical Transfer & Translation Rules

For the translation model, we have created a bilingual (Turkish-Turkmen) dictionary with POS tags. For sake of speed, this dictionary lookup is done by a finite-state transducer (FST). The dictionary is used in preparing "lexical transfer" rules. Here is a sample rule (shortened) which translates postpositions:

```
define Postp
"üçin" -> "için",
"garanyñda" -> "göre",
"ýaly" -> "gibi"  ||  [.#. | "\t" | " "] _ "+Postp";
```

Apart from the lexical transfer rules, there are some translation rules which make some processing in word level. These rules re-organize the morphosyntactic features and make some modifications so that the Turkish morphological generator can accept the input. An example of this transfer rule is :

```
  define Rule7 "+Imp+A1sg" -> "+Opt+A1sg";
```

This rule changes the imperative mood (*aglamaýyn*) to optative mood (*ağlamayayım* – *"if only I don't cry"*) for the first-singular person case because there is no usage of imperative mood with first-singular person case in Turkish.

The lexical transfer rules and morphological translation rules are combined together in a big FST which has more than 1000 rules.

3.3. Turkish Morphological Generator

Turkish morphology has been deeply investigated and a well-known, wide-coverage, FST based morphological analyzer and generator is available by Oflazer [8]. We have used this tool for our translation system which accepts the morphologically decomposed input and then generates the surface form of this input. A sample input and output is given below:

```
In:  yetiş+Verb+Pos^DB+Noun+Inf2+A3sg+P3sg+Nom
Out: yetişmesi
```

In Turkish, a root and other morphosyntactic features are enough to determine the word form uniquely (except very rare situations), so this component generates only one or zero output for each input. If the transducer does not accept the input, there will be no output word form. In such a case the transfer model just use Turkish root, so there will be a loss of information conveyed by the dismissed morphological features.

3.4. The Language Model

The lexical transfer block produces lexical ambiguity in addition to the ambiguities generated by the morphological analysis block. There is no known POS Tagger for the Turkmen language so the only option is preserving this morphological ambiguity until the sentence is transferred into Turkish.

Because of these ambiguities, it results in a situation such that one sentence in source language has a lot of target language transfers. In our test set, each sentence has 334 translations on the average.

By using a language model, one can compute the probability of a sentence S ($w_1w_2w_3\ldots w_n$) by the following formula:

$$P(S)=P(w_1)P(w_2|w_1)P(w_3|w_1w_2)\ldots P(w_n|w_1\ldots w_{n-1})$$

This means that the probability of any word w_i can be calculated by using its history ($w_1..w_{i-1}$). There are so many histories and this makes the model not feasible. One way of reducing this overhead is limiting the number of words in the history. If the last n words of the history are taken into account, this model is called as n-gram language model.

We compute our language model by using CMU-Cambridge Statistical Language Modeling Toolkit [9]. Our training corpus contains nearly one million morphologically disambiguated words from a daily Turkish newspaper. In our experiments we have prepared a training corpus which has only root forms by eliminating other morphological features.

In order to find the most probable transfer, an HMM is built for each sentence by using the possible Turkish word forms generated by the MT system. The state observation likelihoods are set as 1, so we only use the language model (state transition) probabilities. Then the most probable sentence is found by using the Viterbi Algorithm [10]. In Figure 2, an HMM built for a simplified input sentence is given.

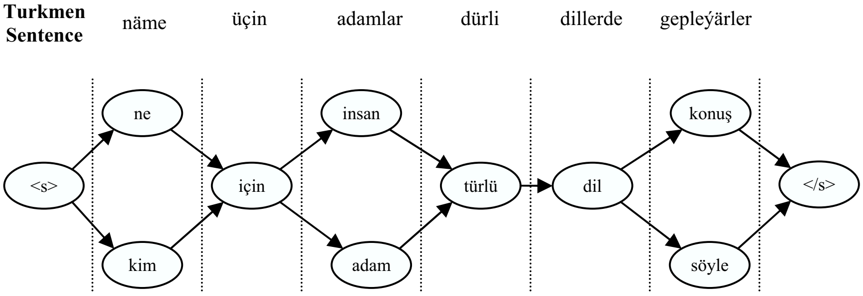

Fig. 2. The process of decoding the most probable target language sentence (the state transition probabilities are determined by the language model)

Table 1. Example decoding of an output sentence by using various LMs

LM Order	Most Probable Sentences	Rank	Log. Prob.
Unigram	ne için insanlar türlü dillerde söylüyorlar	1	-17.2978
	ne için insanlar türlü dillerde konuşuyorlar	2	-17.5196
	ne için adamlar türlü dillerde söylüyorlar	3	-17.7816
Bigram	**ne için insanlar türlü dillerde konuşuyorlar**	1	-18.1625
	ne için adamlar türlü dillerde konuşuyorlar	2	-18.3105
	kim için insanlar türlü dillerde konuşuyorlar	3	-18.6553
Trigram	**ne için insanlar türlü dillerde konuşuyorlar**	1	-18.2265
	kim için insanlar türlü dillerde konuşuyorlar	2	-18.6196
	ne için adamlar türlü dillerde konuşuyorlar	3	-18.6294

We have used unigram, bigram and trigram language models in our recent experiments. The results of the sample sentence above are depicted in Table 1 where the bold sentences indicate the correct word-to-word translation. It can be easily seen that unigram model failed in constructing the right translation because the probability of the verb "söyle" (to tell) P(söyle) is higher than the probability of the verb "konuş " (to talk) P(konuş) which exploits the wrong translation. However, a bigram language model chose the correct translation because the word pair "dil konuş" has more common usage than the word pair “dil söyle”. Note that these probabilities of candidate translations are calculated by using only word roots

4. Successes & Weaknesses

The main problem in our prototype system is the lack of POS tagger for the Turkmen language and lexical ambiguities. Language model block solves most of the lexical ambiguity problems. Luckily, some of the morphological ambiguities don’t need to be handled because the ambiguity disappears when the Turkish morphological generator produces same word forms for the ambigious inputs. This fact is described with two examples in table 2. The Turkmen word “*näme*” (literally “what”) and “*ugry*” (literally “the direction of”) has ambiguous morphological analyses which causes multiple Turkish morphological structures, but this ambiguity disappears after the Turkish morphological generator because two resulting Turkish word forms are same.

Table 2. Some types of ambiguities disappearing after the Turkish morphological generation

Turkmen Surface Form	**Turkmen Morphological Analysis**	**Turkish Morphological Transfer**	**Turkish Surface Form**
näme	nEme+Conj	ne+Conj	ne
	nEme+Adj	ne+Adj	
ugry	ugur+Noun+A3sg+P3sg+Nom	yön+Noun+A3sg+P3sg+Nom	yönü
	ugur+Noun+A3sg+Pnon+Acc	yön+Noun+A3sg+Pnon+Acc	

The language model we have used to select the best candidate translation seems to work fine for disambiguating lexical ambiguities. Even though the language model decoding module handles the short-distance dependencies (actually, the relations of a word with its n previous words for an n-th order LM) , the system fails in finding long-distance relations. For example, for some Turkmen tenses, like future tense, the person agreement feature of an input verb should be determined according to the subject of the sentence. This and a number of similar tasks require a module which operates on the sentence level so that some sentence level work can be done without parsing.

In some cases, the word-to-word translation fails because of multi word expressions which require a phrase-to-phrase translation model.

5. Conclusion

In this work, we have implemented a prototype MT system between Turkmen to Turkish. The current version of our prototype system performs word-to-word translation model that produces promising acceptable translations. This system is still under development to achieve higher quality translations and we believe that with some modifications like sentence level post-processor and multi-word expression transferring module, this system will produce more satisfactory results. As a result, the simplicity and performance of this direct translation transfer approach encourages us to build MT engines between Turkish and other Turkic languages.

6. Future Plans

As the first step, we are planning to build a multi-word transfer block to enable the transfer of multi-word expressions. Additionally, a sentence level post-processing module will improve the translation quality by processing some long-distance relations. Also, it is very important to investigate the effects of different language model types and parameters (order number n, vocabulary size) in the decoding phase.

References

1. Jurafsky, D., Martin J.H.: Speech and Language Processing. Prentice Hall, New York (2000)
2. Oliva, K. : A Parser for Czech Implemented in Systems Q. Explizite Beschreibung der Sprache und automatiche Textbearbeitung XVI, MFF UK Prague (1989)
3. Hajič, J., Hric J., Kubon, V. : Machine Translation of Very Close Languages. Applied NLP Processing, NAACL, Washington (2000)
4. Canals, R., Esteve, A., Garrido, A., et.al. : interNOSTRUM: A Spanish-Catalan Machine Translation System. EAMT Machine Translation Summit VIII, Spain (2001)
5. Altıntas, K., Cicekli, I. : A Machine Translation System Between a Pair of Closely Related Languages, Seventeenth International Symposium On Computer and Information Sciences, Florida, USA (2002)
6. www.sil.org, www.ethnologue.com
7. Karttunen, L., Gaal, T., Kempe, A. : Xerox Finite-State Tools Technical Report, Xerox Research Centre Europe (1997)
8. Oflazer, K. : Two-level Description of Turkish Morphology. Literary and Linguistic Computing, Vol. 9, No:2, (1994)
9. Clarkson, P.R., Rosenfeld R. : Statistical Language Modeling Using the CMU-Cambridge Toolkit. Proceedings ESCA Eurospeech (1997)
10. Fomey, G.D., Jr. : "The Viterbi Algorithm", IEEE Proc. Vol. 61, pp. 268-278 (1973)