# Learning Translation Templates for Closely Related Languages

Kemal Altintas* and Halil Altay Güvenir

Department of Computer Engineering, Bilkent University
Bilkent, 06800 Ankara Turkey
kemal@ics.uci.edu
guvenir@cs.bilkent.edu.tr

**Abstract.** Many researchers have worked on example-based machine translation and different techniques have been investigated in the area. In literature, a method of using translation templates learned from bilingual example pairs was proposed. The paper investigates the possibility of applying the same idea for close languages where word order is preserved. In addition to applying the original algorithm for example pairs, we believe that the similarities between the translated sentences may always be learned as atomic translations. Since the word order is almost always preserved, there is no need to have any previous knowledge to identify the corresponding differences. The paper concludes that applying this method for close languages may improve the performance of the system.

## 1 Introduction

Machine translation has been an interesting area of research since the invention of computers. Many researchers have worked on this subject and developed different methods. Currently, there are many commercial and operational systems and the performances of the machine translation systems are best when the languages are close to each other [2].

There are two main approaches in corpus-based machine translation: statistical methods and example based methods. All corpus-based methods require the presence of a bilingual corpus in hand. The necessary translation rules and lexicons are automatically derived from this corpus.

Example based methods in machine translation use previously translated examples to form a "translation memory" for the translation process [3]. There are three main components of example-based machine translation (EBMT): matching fragments against a database of real examples, identifying the corresponding translation fragments and recombining these to give the target text [7].

* Currently affiliated with Information and Computer Science Department, the University of California, Irvine.

V. Palade, R.J. Howlett, and L.C. Jain (Eds.): KES 2003, LNAI 2773, pp. 756-762, 2003.

A detailed review of example based machine translation systems can be found in [9].

The idea of learning generalized translation templates for machine translation was investigated by Cicekli and Güvenir [5]. They proposed a method for learning translation templates from bilingual translation examples. Their system is based on analyzing similarities and differences between two translation example pairs. There is no linguistic analysis involved in the method and the system totally depends on string matching. The authors claim that the method is language independent and they show that it works for Turkish and English, which are two virtually unrelated languages.

The principal idea of translation template learning framework as presented in [5] is based on a heuristic to infer the correspondences between the patterns in the source and target languages from given two translation pairs. The similarities between the source language sentences are identified and assumed to correspond to the similar parts in the target language. Also, the differences in the source language sentences should correspond to the differences in the target language sentence pair. The system they present identifies the similarities and differences between source and target language pairs and learns generalized translation rules from these examples.

In this paper, we investigate the possibility of applying the same idea to closely related languages by using the corresponding translated sentences themselves instead of using two examples. We take Turkish and Crimean Tatar as the example closely related language pair and we believe that the idea can be developed and applied for other close language pairs.

The rest of the paper is organized as follows: Next section introduces the concept of translation template and Section 3 gives the details of the learning process comparing it against the proposed method in [5]. Section 4 discusses some weak points of the approach that we present here and the last section summarizes the ideas and concludes the paper.

## 2 Translation Templates

A translation template is a generalized translation exemplar pair where some components are generalized by replacing them with variables in both sentences. Consider the following example:

X1 +Verb+Pos+Past+A1sg<=> Y1 +Verb+Pos+Past+A1sg

gel <=> kel

The left-hand side (first) part in this example and in the following examples throughout the paper refers to Turkish and the right-hand side (second) part refers to Crimean Tatar. The first template means that whenever the sequence "+Verb+Pos+Past+A1sg" follows any sequence that can be put in place of the variable X1, it can be translated into "+Verb+Pos+Past+A1sg" provided that it follows another sequence Y1, which is the translation of X1. In other words, after learning this rule, we can translate a sentence ending in "+Verb+Pos+Past+A1sg" provided that the beginning of the sentence can also be translated using the previously

learned rules. The second template is an *atomic template*, which can be read as "gel" (*come*) in Turkish always corresponds to "kel" in Crimean Tatar.

Since Turkish and all other Turkic languages are agglutinative languages, using the surface form (actual spelling) of the words may not be helpful. For example, Turkish word "geliyoruz" (*we are coming*) corresponds to "kelemiz" in Crimean Tatar and they do not show much similarity at first sight. However, if we morphologically analyze the two words we get:

geliyoruz gel+Verb+Pos+Prog1+A1pl

kelemiz kel+Verb+Pos+Prog1+A1pl

The two analyses are similar except for the roots. Thus, using the morphological analyses of the two words may help us to learn much more rules.

For the morphological analysis of Turkish, we used the analyzer developed by Oflazer [8]. For the Crimean Tatar part, we used the analyzer described in [1].

## 3 Learning Translation Templates

Close languages such as Turkish and Crimean Tatar share most parts of their grammars and vocabularies. The word order in close languages can most of the time be the same and even the ambiguities are preserved [6: p.807].

The first phase of translation template learning algorithm is identifying the similarities and differences between the two sentences. A *similarity* is a non-empty sequence of common items in both sentences. Actually, the similarity is an exact matching between sub-strings of the sentences. A *difference* is the opposite of a similarity and it is a non-common sequence of characters between the two sentences. In other words, a difference is what is not a similarity. The following translation pair gives the similarities as underlined:

geliyoruz gel<u>+Verb+Pos+Prog1+A1pl</u>

kelemiz kel<u>+Verb+Pos+Prog1+A1pl</u>

A matching sequence between the sentences is a sequence of similarities and differences with the following properties:

- A similarity is followed by a difference and a difference is followed by a similarity. Two consequent similarities and two consequent differences cannot occur in a match sequence.
- If a terminal occurs in a similarity, it cannot occur in a difference.
- If a terminal occurs in a difference in one language, it cannot occur in a difference in the other language.
- A terminal occurring in both sentences must appear exactly n times where $n >= 1$.
- If a terminal occurs more than once in both sentences, its $i^{th}$ occurrence in both sentences must end up in the same similarity of their minimal match sequence.

If these rules are satisfied, then there is a unique match for the sentences or there is no match. The details of the algorithm that finds the similarities and differences between the two sentences are explained in [4].

Once the similarities and the differences are identified, the system changes the differences with variables to construct a translation template. If there is no difference between the sentences and it is composed of only a single similarity, then it is learned as an atomic template. Many times, Turkish words and their Crimean Tatar correspondings are the same. For example, both the surface and lexical forms of the words "ev = ev+Noun+A3sg+Pnon+Nom" (*house*), "bildim = bil+Verb+Pos+Past+A1sg" (*I knew*) are the same in Turkish and Crimean Tatar. For "ev", the following translation template is learned:

ev+Noun+A3sg+Pnon+Nom <=> ev+Noun+A3sg+Pnon+Nom

Although [5] does not discuss matching pairs with a single similarity, it exists between close languages and can be learned. It is always possible that a variable in the template may have to be replaced with a noun like the one above. Consider the sentence "ev aldım = ev+Noun+A3sg+Pnon+Nom al+Verb+Pos+Past+A1sg" (*I bought a house*). If we have a template like:

X1 al+Verb+Pos+Past+A1sg <=> Y1 al+Verb+Pos+Past+A1sg

we can easily replace X1 with "ev+Noun+A3sg+Pnon+Nom" for the translation.
If the matching sequence is composed of a single similarity and a single difference, then the difference is replaced with a variable and similarity is preserved. Also, the differences and the similarities are learned as separate atomic templates. For the word pair

geldim gel+Verb+Pos+Past+A1sg (*I came*)

keldim kel+Verb+Pos+Past+A1sg

the following templates are learned:

X1 +Verb+Pos+Past+A1sg <=> Y1 +Verb+Pos+Past+A1sg

+Verb+Pos+Past+A1sg <=> +Verb+Pos+Past+A1sg

gel <=> kel

When the similarities are in the beginning then the same rule applies. The differences in the end are replaced with variables and the similarities and differences are learned as separate atomic templates.

When there are two similarities surrounding a single difference in the sentences, the difference is replaced with a variable and the differences and the similarities are learned as separate templates. For the sentence pair "eve geldim = ev+Noun+A3sg+Pnon+Dat gel+Verb+Pos+Past+A1sg" (*I came home*) and "evge keldim = ev+Noun+A3sg+Pnon+Dat kel+Verb+Pos+Past+A1sg" the following rules are learned:

ev+Noun+A3sg+Pnon+Dat X1 +Verb+Pos+Past+A1sg <=>

ev+Noun+A3sg+Pnon+Dat Y1 +Verb+Pos+Past+A1sg

gel <=> kel

ev+Noun+A3sg+Pnon+Dat <=> ev+Noun+A3sg+Pnon+Dat

+Verb+Pos+Past+A1sg <=> +Verb+Pos+Past+A1sg

For the cases where there is more than one difference, the system should learn templates only if at least all but one of the differences have previously learned correspondences. Consider the following sentence pair:

okula geldim (I came to school)

okul+Noun+A3sg+Pnon+Dat gel+Verb+Pos+Past+A1sg

mektepke keldim

mektep+Noun+A3sg+Pnon+Dat kel+Verb+Pos+Past+A1sg

According to [5], the system should not learn anything if it does not know whether "okul" (*school*) is really the translation of "mektep" (*school*) or "kel" (*come*).

Actually it is possible to learn rules without requiring that we know the corresponding differences. The algorithm proposed in [5] requires that at least all but one of the difference correspondences are known. This algorithm is a general method for learning and the system is language independent. The experiments were done for Turkish and English where the word order is clearly different. Thus, for the general system, it might be necessary to verify that all but one of the differences have corresponding translations in hand.

However, for close language pairs, such as Turkish and Crimean Tatar, the word order is almost always preserved in the translation. Thus, if we know that our example translations are fully correct, we can learn the following templates without requiring any preconditions:

X1+Noun+A3sg+Pnon+Dat X2 +Verb+Pos+Past+A1sg <=>

Y1 +Noun+A3sg+Pnon+Dat Y2 +Verb+Pos+Past+A1sg

okul <=> mektep

+Noun+A3sg+Pnon+Dat <=> +Noun+A3sg+Pnon+Dat

gel <=> kel

+Verb+Pos+Past+A1sg <=> +Verb+Pos+Past+A1sg

## 4 Discussions

There are cases where the idea is not applicable. Consider the following phrases:

bildiğim yer (the place where I know)

bil+Verb+Pos^DB+Adj+PastPart+P1sg yer+Noun+A3sg+Pnon+Nom

bilgen yerim

bil+Verb+Pos^DB+Adj+PastPart+Pnon yer+Noun+A3Sg+P1sg+Nom

The difference between the two sentences is that the possessive marker in Turkish follows the past participle morpheme affixed to the verb, whereas the possessive marker in Crimean Tatar follows the noun in this clause. Any translation program in such a case should identify that this is an adjectival clause made with past participle and should move the possessive marker that comes after the verb to its place after the noun.

The current algorithm cannot deal with such a case, regardless of whether we have any prior information or not. Since the differences between the two sentences are only the possessive markers, we cannot have a prior information like:

P1sg <=> Pnon

which is totally wrong. However, the approach which uses example pairs is much safer in this case and can identify a template for this case:

Turkish:

bildiğim yer (the place that I know)

bil+Verb+Pos^DB+Adj+PastPart+P1sg yer+Noun+A3sg+Pnon+Nom

bildiğim ev (the house that I know)

bil+Verb+Pos^DB+Adj+PastPart+P1sg ev+Noun+A3Sg+Pnon+Nom

Crimean Tatar:

bilgen yerim (the place that I know)

bil+Verb+Pos^DB+Adj+PastPart+Pnon yer+Noun+A3sg+P1sg+Nom

bilgen evim (the house that I know)

bil+Verb+Pos^DB+Adj+PastPart+Pnon ev+Noun+A3Sg+P1sg+Nom

From these two examples, we can derive the template:

bil+Verb+Pos^DB+Adj+PastPart+P1sg X1 +Noun+A3sg+Pnon+Nom <=>

bil+Verb+Pos^DB+Adj+PastPart+Pnon Y1 +Noun+A3Sg+P1sg+Nom

However, this is an exceptional case and overwhelming majority of the cases can be covered with the approach that we presented in the paper.

## 5 Conclusion

Corpus based approaches in language processing have attracted more interest. Example based machine translation is also considered as an alternative to traditional rule based methods with its capabilities to learn the necessary linguistic and semantic knowledge from the translation examples.

Cicekli and Güvenir in [5] proposed a method to learn translation templates from bilingual translation examples. They also showed that the method is applicable to Turkish and English, which are two unrelated languages having completely different

characteristics. Their method requires two similar translation example pairs to derive a template. Further they require that the similarities and differences are identified and the corresponding translations for almost all differences are known to derive a template from the given example pair.

In this paper, we extended their approach to closely related languages and taking Turkish and Crimean Tatar as an example, we investigated the possibility of using the translated sentences themselves instead of a pair of sentences to derive some rules.

The first case we saw for close languages is that, it is possible to have cases where the two sentences are exactly the same for both languages. So, this can be learned as an atomic template. Secondly, similarities can always be learned as atomic templates regardless of the number of differences between sentences. Since the word and morpheme order is usually preserved in close languages, it is possible to say that a similarity is always a correspondence between the languages.

Finally, we saw that, in most cases there is no need to know any explicit correspondences between the differences in order to derive templates. Cicekli and Güvenir require that if there are $n > 1$ differences between sentences, we must know at least $n$-1 of the correspondences. However, for close languages, since the word order is preserved, there is usually no need to enforce any preconditions provided that the translations are correct.

## References

[1] Altintas, K., Cicekli, I., "A Morphological Analyser for Crimean Tatar", In *Proceedings of 10th Turkish Artificial Intelligence and Neural Network Conference (TAINN 2001)*, North Cyprus, 2001.

[2] Appleby, S., Prol, M. P., "Multilingual World Wide Web", *BT Technology Journal Millennium Edition*, Vol. 18, No:1, 1999.

[3] Carl, M. "Recent research in the field of example-based machine translation", Computational Linguistics and Intelligent Text Processing, LNCS 2004, pp. 195-196, 2001.

[4] Cicekli, I., "Similarities and Differences", In Proceedings of SCI 2000, pp. 331-337, Orlando, FL, July 2000.

[5] Cicekli, I., and Güvenir, H. A., "Learning Translation Templates from Bilingual Translation Examples", *Applied Intelligence*, Vol. 15, No. 1, pp: 57-76, 2001.

[6] Jurafsky, D., Martin, J. H., Speech and Language Processing, Prentice Hall, 2000.

[7] Nagao, M., A Framework of a Mechanical Translation Between Japanese and English by Analogy Principle, In Artificial and Human Intelligence, Amsterdam, 1984.

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

[9] Somers, H., "Review Article: Example-based Machine Translation", *Machine Translation*, Vol. 14, pp. 113-157, 1999.