Learning Translation Rules From A Bilingual Corpus*

Ilyas Cicekli and H. Altay Güvenir

Dept. of Comp. Eng. and Info. Sc., Bilkent University
06533 Bilkent, Ankara, Turkey
e-mail: {ilyas,guvenir}@cs.bilkent.edu.tr

Abstract. This paper proposes a mechanism for learning pattern correspondences between two languages from a corpus of translated sentence pairs. The proposed mechanism uses analogical reasoning between two translations. Given a pair of translations, the similar parts of the sentences in the source language must correspond the similar parts of the sentences in the target language. Similarly, the different parts should correspond to the respective parts in the translated sentences. The correspondences between the similarities, and also differences are learned in the form of translation rules. The system is tested on a small training dataset and produced promising results for further investigation.

1 Introduction

Traditional approaches to machine translation (MT) suffer from tractability, scalability and performance problems due to the necessary extensive knowledge of both the source and the target languages. Corpus-based machine translation is one of the alternative directions that have been proposed to overcome the difficulties of traditional systems. Two fundamental approaches in corpus-based MT have been followed. These are *statistical* and *example-based* machine translation (EBMT), also called *memory-based* machine translation (MBMT). Both approaches assume the existence of a bilingual text (an already translated corpus) to derive a translation for an input. While statistical MT techniques use statistical metrics to choose the most probable words in the target language, EBMT techniques employ pattern matching techniques to translate subparts of the given input [1].

Exemplar-based representation has been widely used in Machine Learning (ML). According to Medin and Schaffer [7], who originally proposed exemplar-based learning as a model of human learning, examples are stored in memory without any change in the representation. Here, an exemplar is a characteristic example stored in the memory. The basic idea in exemplar-based learning is to use past experiences or cases to understand, plan, or learn from novel situations [4, 6, 10].

EBMT has been proposed by Nagao [8] as *Translation by Analogy* which is in parallel with memory based reasoning [14], case-based reasoning [11] and derivational analogy [2]. Example-based translation relies on the use of past translation examples to derive a translation for a given input [3, 9, 12, 13, 15]. The input sentence to be translated is compared with the example translations analogically to retrieve the *closest* examples to the input. Then, the fragments of the retrieved examples are translated and recombined in the target language. Prior to the translation of an input sentence, the correspondences between the source and target languages

* This research has been supported in part by NATO Science for Stability Program Grant TU-LANGUAGE.

should be available to the system; however this issue has not been given enough consideration by the current EBMT systems. Kitano [5] has adopted the manual encoding of the translation rules, however this is a difficult and an error-prone task for a large corpus. Wu [16] uses a method to extract phrasal translation examples in sentence-aligned parallel corpora using a probabilistic translation lexicon for the language pair. Wu's *inversion transduction grammar* (ITG) formalism is used to model two languages simultaneously. In this paper, we formulate this acquisition problem as a machine learning task in order to automate the process.

In this paper, we propose a technique which stores exemplars in the form of *templates* that are *generalized exemplars*. A template is an example translation pair where some components (e.g., word stems and morphemes) are generalized by replacing them with variables in both sentences, and establishing bindings between the variables. We will refer this technique as GEBMT for *Generalized Example Based Machine Translation*. We assume no grammatical knowledge about languages except morphological structure of some words in the languages.

The algorithm we propose here, for learning such templates, is based on a heuristic to learn the correspondences between the patterns in the source and target languages, from two translation pairs. The heuristic can be summarized as follows: Given two translation pairs, if the sentences in the source language exhibit some similarities, then the corresponding sentences in the target language must have similar parts, and they must be translations of the similar parts of the sentences in the source language. Further, the remaining different parts of the source sentences should also match the corresponding differences of the target sentences. However, if the sentences do not exhibit any similarity, then no correspondences are inferred. Consider the following translation pair given in English and Turkish to illustrate the heuristic:

```
I give+PAST the book to Mary
        ↔ Mary+DAT kitap+ACC ver+PAST+1SG
I give+PAST the pencil to Mary
        ↔ Mary+DAT kurşun kalem+ACC ver+PAST+1SG
```

Similarities between the translation examples are shown as underlined. The remaining parts are the differences between the sentences. We represent the similarities in the source language as "`I give+PAST the` X^S `to Mary`", and the corresponding similarities in the target language as "`Mary+DAT` X^T`+ACC ver+PAST+1SG`". According to our heuristic, these similarities should correspond each other. Here, X^S denotes a component that can be replaced by *any* appropriate structure in the source language and X^T refers to its translation in the target language. This notation represents an *abstraction* of the differences {`book` vs. `pencil`} and {`kitap` vs. `kurşun kalem`} in the source and target languages, respectively. Using the heuristic further, we infer that `book` should correspond to `kitap` and `pencil` should correspond to `kurşun kalem`; hence learning further correspondences between the examples.

Our learning algorithm based on this heuristic is called TRL (*Translation Rule Learner*). Given a corpus of translation cases, TRL infers the correspondences between the source and target languages in the form of translation rules. These rules can be used for translation in both directions. Therefore, in the rest of the paper we will refer these languages as L_1 and L_2. Although the examples and experiments herein are on English and Turkish, we believe that the model is equally applicable to other language pairs.

The rest of the paper is organized as follows. Section 2 describes the underlying mechanisms of TRL, along with sample rule derivations. Section 3 gives more learning examples. Section 4 illustrates the translation process using translation rules. Section 5 concludes the paper.

2 Learning

Our learning algorithm TRL infers translation rules using similarities and differences between a pair of translation examples (E_i, E_j) from a bilingual corpus. A translation example E is also a pair $(E^{L_1} \leftrightarrow E^{L_2})$ where E^{L_1} and E^{L_2} are equivalent sentences in languages L_1 and L_2. Using a matching algorithm, we find a match sequence M^{L_1} representing similarities and differences in $E_i^{L_1}$ and $E_j^{L_1}$, a match sequence M^{L_2} for $E_i^{L_2}$ and $E_j^{L_2}$. From these two match sequences, we learn translation rules.

In our examples, we will use translation examples between English and Turkish. A translation example consists of an English sentence and a Turkish sentence. We will use the lexical level representation[2] for each sentence in our examples. For example, the English sentence *"I broke a pencil"* will be represented by

```
I break+PAST a pencil
```

and its equivalent Turkish sentence *"Bir kurşun kalem kırdım"* will be represented by

```
Bir kurşun kalem kır+PAST+1SG
```

For a pair of translation example $((E_1^{L_1} \leftrightarrow E_1^{L_2}), (E_2^{L_1} \leftrightarrow E_2^{L_2}))$, the matching algorithm produces match sequences M^{L_1} and M^{L_2} to represent similarities and differences in examples in languages L_1 and L_2, respectively. A match sequence M for two different sentences will be in the following form.

$S_1\ D_1\ S_2\ \cdots\ D_n\ S_{n+1}$ where $n \geq 1$

In that sequence, each S_i represents a similarity between sentences. In other words, it is a substring which is common in both of those sentences. Each D_i represents a difference which is a pair of non-empty substrings of sentences, one from the first sentence and the other from the second sentence. For each difference $D_i^1 : D_i^2$, D_i^1 and D_i^2 do not contain any common item. Also, no lexical item in a similarity S_i appear in any previously formed difference D_k for $k < i$. Any of S_1 or S_{n+1} can be empty, however, S_i for $1 < i < n + 1$ must be non-empty. These restrictions guarantee tha there exists either a unique match or no match between two different examples.

For example, in the following translation examples

```
it is a book     ↔ o bir kitap+COP
it is a pencil   ↔ o bir kurşun kalem+COP
```

similarities are underlined and differences are not. The match sequence for English sentences will be

```
it is a book:pencil
```

Note that we have one similarity and one difference between English sentences. The matching sequence for Turkish sentences will be

[2] In our examples, `PAST`, `AOR`, `PRG`, `FUT` denote past, aorist, progressive and future tenses, `COND`, `NEC` denote necessitative and conditional, `ACC`, `DAT`, `LOC`, `ABL` denote accusative, dative, locative and ablative case markers for nouns, `1SG`, `2SG`, `3SG` denote first, second and third singular verbal agreements, `COP` denotes copula in verbs.

o bir kitap:kurşun kalem +COP

where we have two similarities and one difference.

In the example above, the difference in English sentences must correspond to the difference in Turkish sentences, and similarities in them must correspond to each other in that context. TRL can learn the following translation rules from differences and similarities in that example.

book $\leftrightarrow$ kitap
pencil $\leftrightarrow$ kurşun kalem
it is X^E $\leftrightarrow$ o bir X^T +COP where X^E is a translation of X^T

First two rules are learned from differences in English and Turkish sentences, namely **book:pencil** and **kitap:kurşun kalem**. The last rule is learned from similarities in the example. In addition to these three learned rules, we also put two translation rules directly given in the example into our learned rule database. Of course, they are more specific forms of the third learned rule. We order rules from the most specific to the least specific in the database. During translation, the first applicable specific rule will be used for the translation of a sentence as a result of this ordering.

When the number of differences in two match sequences M^{L_1} and M^{L_2} of a pair of translation examples is greater than 1, say n, the learning algorithm only learns new rules if $n-1$ differences can be resolved using already learned rules from previous examples. Otherwise, the current version of the algorithm cannot learn new rules. From the following example,

I give+PAST the book $\leftrightarrow$ Kitap+ACC ver+PAST+1SG
You give+PAST the pencil $\leftrightarrow$ Kurşun kalem+ACC ver+PAST+2SG

we will get the following match sequences.

M^E = I:You give+PAST the book:pencil
M^T = Kitap:Kurşun kalem +ACC ver+PAST +1SG:+2SG

Both M^E and M^T have two differences. If we had not learned anything before this example, there is no way to know whether the difference **I:You** in English sentences corresponds to the difference **Kitap:Kurşun kalem** or **+1SG:+2SG** in Turkish sentences. Now, let us assume that we have already learned the following translation rules from some previous examples.

book $\leftrightarrow$ Kitap
pencil $\leftrightarrow$ Kurşun kalem

Since we now know that the difference **book:pencil** corresponds to the difference **kitap:kurşun kalem**, the difference **I:You** must correspond to the difference **+1SG:+2SG**. Thus, we can learn the following new translation rules from this example.

I $\leftrightarrow$ +1SG where X^E is a translation of X^T
You $\leftrightarrow$ +2SG and Y^E is a translation of Y^T.
X^E give+PAST the Y^E $\leftrightarrow$ Y^T +ACC ver+PAST X^T

For a given pair of translation examples, $((E_1^{L_1} \leftrightarrow E_1^{L_2}), (E_2^{L_1} \leftrightarrow E_2^{L_2}))$, the algorithm of the translation rule learner (TRL) for this pair is given in Figure 1. In that algorithm, first we find match sequences M^{L_1} and M^{L_2} for sentences in languages L_1 and L_2, respectively. Then, we

try to reduce the number of differences in these match sequences to one. At the same time, we construct *Condition* which is a conjunction of translation goals for a translation rule which will be learned later in the algorithm. After this reduction, each of our match sequences will have exactly one difference. So, these unlearned differences must correspond to each other. From this fact, we learn three translation rules given at the end of the algorithm. In the implementation, each learned translation rule is represented in the form of a Prolog fact or rule.

Let $((E_1^{L_1} \leftrightarrow E_1^{L_2}), (E_2^{L_1} \leftrightarrow E_2^{L_2})$ be a pair of translation examples.
$M^{L_1} \leftarrow match(E_1^{L_1}, E_2^{L_1})$;
$M^{L_2} \leftarrow match(E_1^{L_2}, E_2^{L_2})$;
if $\#ofSimilarity(M^{L_1}) = 0$ **or** $\#ofSimilarity(M^{L_2}) = 0$ **then** *exit*;
if $\#ofDifference(M^{L_1}) = 0$ **or**
    $\#ofDifference(M^{L_1}) \neq \#ofDifference(M^{L_2})$ **then** *exit*;
Condition $\leftarrow$ "";
$i \leftarrow 1$;
while $\#ofDifference(M^{L_1}) > 1$ **do**
    begin
      if there exists a D^{L_1} in M^{L_1} and a D^{L_2} in M^{L_2} such that
        the correspondence of D^{L_1} to D^{L_2} has been already learned **then**
        begin
          Replace D^{L_1} in M^{L_1} with a new variable $X_i^{L_1}$;
          Replace D^{L_2} in M^{L_2} with a new variable $X_i^{L_2}$;
          Add "$X_i^{L_1} \leftrightarrow X_i^{L_2} and$" to the end of *Condition*;
          $i \leftarrow i + 1$;
        end
      else *exit*;
    end
Let D^{L_1} in M^{L_1} and D^{L_2} in M^{L_2} be unlearned differences such that
D^{L_1} is $D_1^{L_1} : D_2^{L_1}$ and D^{L_2} is $D_1^{L_2} : D_2^{L_2}$;
Replace D^{L_1} in M^{L_1} with a new variable $X_i^{L_1}$;
Replace D^{L_2} in M^{L_2} with a new variable $X_i^{L_2}$;
Add "$X_i^{L_1} \leftrightarrow X_i^{L_2}$" to the end of *Condition*;
Learn the following translation rules:
    $D_1^{L_1} \leftrightarrow D_1^{L_2}$
    $D_2^{L_1} \leftrightarrow D_2^{L_2}$
    $M^{L_1} \leftrightarrow M^{L_2}$ *if Condition*

Figure 1. Translation Rule Learner Algorithm For Two Translated Sentence Pairs

3 Examples

In order to evaluate the TRL algorithm we have developed a sample bilingual parallel text. In this section, we will illustrate the behavior of TRL on that sample text.

Example 1: Given the example translations "I saw you at the garden" $\leftrightarrow$ "Seni bahçede gördüm" and "I saw you at the party" $\leftrightarrow$ "Seni partide gördüm", their lexical level representations are

i see+PAST you at the garden ↔ sen+ACC bahçe+LOC gör+PAST+1SG
i see+PAST you at the party ↔ sen+ACC parti+LOC gör+PAST+1SG

From these examples, the following translation rules are learned:

i see+PAST you at the X_1^E ↔ sen+ACC X_1^T+LOC gör+PAST+1SG
  if X_1^E ↔ X_1^T
garden ↔ bahçe
party ↔ parti

Example 2: Given the example translations "It is raining" ↔ "Yağmur yağıyor", "He comes" ↔ "Gelir", "If it is raining then you should take an umbrella" ↔ "Eğer yağmur yağıyorsa bir şemsiye almalısın" and "If he comes then we will go to the theater" ↔ "Eğer gelirse tiyatroya gideceğiz", their lexical level representations are

it is rain+PRG ↔ yağmur yağı+PRG
He come+AOR ↔ gel+AOR
if it is rain+PRG then you should take an umbrella
  ↔ eğer yağmur yağı+PRG+COND bir şemsiye al+NEC+2SG
if he come+AOR then we will go to the theater
  ↔ eğer gel+AOR+COND tiyatro+DAT git+FUT+1PL

From the last two examples using first two examples, the following translation rules are learned:

if X_1^E then X_2^E ↔ eğer X_1^T+COND X_2^T
  if X_1^E ↔ X_1^T **and** X_2^E ↔ X_2^T
you should take an umbrella ↔ bir şemsiye al+NEC+2SG
we will go to the theater ↔ tiyatro+DAT git+FUT+1PL

Example 3. Given the example translations "I went" ↔ "gittim", "you went" ↔ "gittin" and "I came" ↔ "geldim", their lexical level representations are

i go+PAST ↔ git+PAST+1SG
you go+PAST ↔ git+PAST+2SG
i come+PAST ↔ gel+PAST+1SG

From the first and second examples where differences are `i:you` and `+1SG:+2SG`, the following translation rules are learned:

X_1^E go+PAST ↔ git+PAST X_1^T
  if X_1^E ↔ X_1^T
i ↔ +1SG
you ↔ +2SG

And from the first and third examples where differences are `go:come` and `git:gel`, the following translation rules are learned:

i X_1^E+PAST $\leftrightarrow$ X_1^T+PAST+1SG
if $X_1^E \leftrightarrow X_1^T$
go $\leftrightarrow$ git
come $\leftrightarrow$ gel

4 Translation

The translation rules learned by the TRL algorithm can be used in the translation directly. The outline of the translation process is given below:

1. First, the lexical level representation of the input sentence is derived.
2. The most specific matching translation rule is found for the input sentence. If the template for the language of the input sentence in a translation rule matches the input sentence, we call it a matching rule. During this matching, certain variables in the template can bind to substrings of the input sentence. Then, translations for these bound variables are sought. Thus, we will get the lexical level representation of the output sentence if these processes are successful. The most specific matching rule contains maximum number matching terminals and minimum number of variables.
3. The surface level representation of the output sentence obtained in the previous step is generated.

Note that, if the input sentence in the source language is ambiguous, then templates corresponding to each sense will be retrieved, and the sentences for each sense will be generated. The translation rules learned by TRL can be used for translation in both directions.

5 Conclusion

In this paper, we have presented a model for learning translation rules between two languages. We integrated this model with an example-based translation model into Generalized Exemplar-Based Machine Translation. We have implemented this model as the TRL (Translation Rule Learner) algorithm. The TRL algorithm is illustrated in learning translation rules between Turkish and English.

The major contribution of this paper is that the proposed TRL algorithm eliminates the need for manually encoding the translations, which is a difficult task for a large corpus. The TRL algorithm can work directly on surface level representation of sentences. However, in order to generate useful translation patterns, it is helpful to use the lexical level representations. It is usually trivial, at least for English and Turkish, to obtain the lexical level representations of words.

Our main motivation was that the underlying inference mechanism is compatible with one of the ways humans learn languages, i.e. learning from examples. We believe that in everyday usage, humans learn general sentence patterns, using the similarities and differences between many different example sentences that they are exposed to. This observation lead us to the idea that a computer can be trained similarly, using analogy within a corpus of example translations.

The accuracy of the translations learned by this approach is quite high with ensured grammaticality. Given that a translation is carried out using the rules learned, the accuracy of the output translation critically depends on the accuracy of the rules learned.

We do not require an extra operation to maintain the grammaticality and the style of the output, as in Kitano's EBMT model [5]. The information necessary to maintain these issues is directly provided by the translation rules.

The model that we have proposed in this paper may be integrated with an intelligent tutoring system (ITS) for second language learning. The rule representation in our model provides a level of information that may help in error diagnosis and student modeling tasks of an ITS. The model may also be used in tuning the teaching strategy according to the needs of the student by analyzing the student answers analogically with the closest cases in the corpus. Specific corpora may be designed to concentrate on certain topics that will help in student's acquisition of the target language. The work presented by this paper provides an opportunity to evaluate this possibility as a future work.

References

1. Arnold D., Balkan L., Humphreys R.Lee, Meijer S. Sadler L.: *Machine Translation*, NCC Blackwell (1994).
2. Carbonell, J.G.: Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. In Jude W. Shavlik and Thomas G. Dietterich (eds), *Readings in Machine Learning*, Morgan Kaufmann (1990) 636–646.
3. Furuse, O. & Iida, H.: Cooperation between Transfer and Analysis in Example-Based Framework, *Proceedings of COLING-92* (1992).
4. Hammond, K.J.: (Ed.) *Proceedings: Second Case-Based Reasoning Workshop.* Pensacola Beach, FL:Morgan Kaufmann (1989).
5. Kitano, H.: A Comprehensive and Practical Model of Memory-Based Machine Translation. In Ruzena Bajcsy (Ed.) *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann V.2 (1993) 1276-1282.
6. Kolodner, J.L.: (Ed.) *Proceedings of a Workshop on Case-Based Reasoning.* Clearwater Beach, FL: Morgan Kaufmann (1988).
7. Medin, D.L. & Schaffer, M.M.: Context theory of classification learning. *Psychological Review, 85* (1978) 207-238.
8. Nagao, M. A.: Framework of a Mechanical Translation between Japanese and English by Analogy Principle (1985).
9. Nirenburg, S., Beale, S. & Domashnev, C.: A Full-Text Experiment in Example-Based Machine Translation. *Proceedings of the International Conference on New Methods in Language Processing, NeMLap* Manchester, UK (1994) 78-87.
10. Ram, A.: Indexing, Elaboration and Refinement: Incremental Learning of Explanatory Cases. In Janet L. Kolodner (ed.), *Case-Based Learning*, Kluwer Academic Publishers (1993).
11. Reisbech, C. & Schank, R.: *Inside the Case-Based Reasoning*, Lawrence Elbaum Associates (1990).
12. Sato, S.: *Example-Based Machine Translation*, Ph.D. Thesis, Kyoto University (1991).
13. Sato, S. & Nagao, M.: The Memory-Based Translation, *Proceedings of COLING-90* (1990).
14. Stanfill, C. & Waltz, D.: Toward Memory-Based Reasoning. *CACM*, Vol.29, No.12 (1991) 185-192.
15. Sumita, E. & Iida, H.: Experiments and Prospects of Example-Based Machine Translation, *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics* (1991).
16. Wu, D.: Grammarless Extraction of Phrasal Translation Examples From Parallel Texts, *Proceedings of the Sixth Int. Conf. on Theoretical and Methodological Issues in Machine Translation* Leuven, Belguim (1995).