Translation Template Learning Based on Hidden Markov Modeling

Nguyen Minh Le
Graduate School of Information Science, JAIST
ISHIKAWA 923-1292,JAPAN
nguyenml@jaist.ac.jp

Akari Shimazu
Graduate School of Information Science, JAIST
ISHIKAWA 923-1292, JAPAN
shimazu@jaist.ac.jp

Susumu Horiguchi
Graduate School of Information Science, JAIST
ISHIKAWA 923-1292, JAPAN
hori@jaist.ac.jp

Abstract

This paper addresses a novel translation method based on Hidden Markov Model using template rules after learning them from the bilingual corpus. The method can enhance the translation accuracy and ensure a low complexity in comparing with the pervious template learning translation method and draws a new perspective for applying statistical machine learning on example based translations. domain.

Keywords: *Machine translation, EBMT, Template learning translation, HMM.*

1 Introduction

Example based machine translation (EBMT), originally proposed by Nagao (Nagao,M.A 1984), and is one of the main approaches to corpus-based machine translation. The main idea behind EBMT is that an input sentence in the source language is compared with the example translations in the given bilingual parallel text to find the closest matching examples so that these examples can be used in the translation of the input sentence. After finding the closest matches for the sentence in the source language, parts of the corresponding target language sentence are constructed using structural equivalences and deviances in the matches. Following Nagao's original proposal, several approaches using the example based method were presented. One of the approaches that applied the idea for translation from English to Turkey is learning translation template (Cicekli,I 1996) (Gűvenir,H.A 1998). This method relies on the technique that uses the similarity and difference from a source sentence and a target sentence in the given bilingual corpus to build template rules for translation. The advantage of this method is that does not need any complex parsing such as syntactic parsing or semantic parsing and overcome the imperfectness of the rule-based machine translation. One of the disadvantages of the method is that a lot of templates can be matched with an input sentence. To overcome this problem, (Öz and Cicekli,I 1998) present a method which allows sorting template rules according to their confident factors. The translation results are sorted using its score through the value of confident factors. However, this method needs to evaluate all matching rules for each input sentence to obtain the output results, while much of them are redundant rules. The exponential calculation problem will arise when an input sentence is long and the number of template rules is large. Following that point, we present a novel method based on an HMM model that uses constraints for set of matching rules with each input sentence. Thus, the translation results of an input sentence are obtained by finding a set of template rules that is most likely with our HMM model.

The remainder of this paper is organized as follows: A template learning algorithm is given in Section 2. Section 3 describes a HMM modeling for translation using template rules. Section 4 show experiments on English Vietnamese translation system and Section 5 give some conclusions and outstanding problems to be solved in future work.

2 Template Learning Translation

The Template learning algorithm (TTL) infers translation templates using similarities and differences between two translation examples E_a and E_b taken from a bilingual parallel corpus. Formally, a translation example E_a: $E_a^1 \leftrightarrow E_a^2$ is composed of a pair of sentences, E_a^1 and E_a^2, that are translations of each other in English and Vietnamese respectively. A similarity between two sentences of a language is a non-empty sequence of common items (root words or morphemes) in both sentences. A difference between two sentences of a language is a pair of two sequences (D_1, D_2) where D_1 is a sub-sequence of the first sentence, D_2 is sub sequence of the second sentence, and D_1 and D_2 do not contain any common item. Given two translation examples (E_a, E_b), we try to find similarities between the constituents of E_a and E_b. A sentence is considered as a sequence of lexical items. If no similarities can be found, then no template is learned from these examples. If there are similar constituents then *a match sequence* $M_{a,b}$ in the following form is generated.

$S_0^1, D_0^1, S_1^1 ..., D_{n-1}^1, S_n^1 \leftrightarrow S_0^2, D_0^2, S_1^2, ..., D_{m-1}^2, S_m^2$

for $1 \leq n, m$

Here, S_k^1 represents *a similarity* (a sequence of common items) between E_a^1 and E_b^1. Similarly, $D_k^1 : (D_{k,a}^1, D_{k,b}^1)$ represents a *difference* between E_a^1 and E_b^1, where $D_{k,a}^1, D_{k,b}^1$ are non-empty differing items between two similar constituents S_k^1, S_{k+1}^1.

For instance, let us assume that the following translation examples are given:

" I bought the book for John" <-> "Tôi đã mua một quyển sách cho John"

" I bought the ring for John" <-> "Tôi đã mua một chiếc nhẫn cho John"

For these translation examples, the matching algorithm obtains the following match sequence.

I bought the (book, ring) for John <-> Tôi đã mua một (quyển sách, chiếc nhẫn) cho John

That is, S_0^1= *I bought the*, D_0^1=(book, ring), S_1^1= for John, S_0^2=*Tôi đã mua một*, D_0^2=(quyển sách, *chiếc* nhẫn), S_1^2=*cho John.*

After a match sequence is found for two translation examples, we used the two different learning heuristics to infer translation templates (Guvenir, H.A 1998) from that match sequence. These two learning heuristic try to locate corresponding differences or similarities in the match sequence respectively. The first heuristic, which is named similarity translation template (STTL), tries to locate all corresponding differences and generate a new translation template by replacing all differences with variables. The second heuristic can infer translation templates by replacing similarities with variables, if it is able to locate corresponding similarities in the match sequence. These translation templates are called difference translation templates (DTTL). The STTL and DTTL are combined as the template learning algorithm (TTL). From the corpus, the TTL algorithm tries to infer translation templates using the two algorithms above. After all translation templates are learned, they are sorted according to their specificities. Given two templates, one that has a higher number of terminals is more specific than the other.

In the following section we address a new method to translate more accuracy and reduce the complexity.

3 Translation Template Learning Base on HMM

To explain translation template learning based on HMM model, some notations are defined, afterward a translation based HMM model is presented in this section.

3.1 Template rules

Let SL and TL be the source language and the target language and $S_1S_2...S_n \leftrightarrow T_1T_2...T_k$ be a template rules, in which S_i is a sequence of word or a variable in SL and T_i is a sequence of words, so called a constant element, or a variable in TL. In addition, each variable in the left side is aligned with each variable in the right side. A variable in the left side and a variable in the right side of a template rule can be received a phrase or a word in SL and TL respectively. Figure 1 depicts an example of a template rules where a sentence containing *"give...up"* in English is translated to a sentence in Vietnamese containing *"tu bo"*.

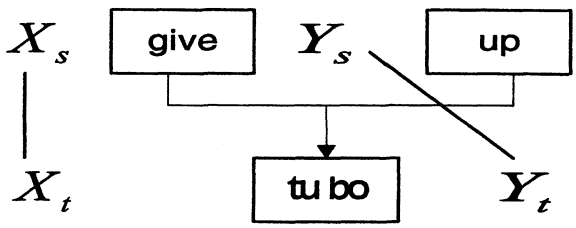

Figure 1. Template rule example

Let a *lexical rule* be a template rule that has no variable inside. A *lexical rule* is a bilingual phrase in SL and TL language.

3.2 Translation base on HMM modeling

3.2.1 The model

The model we propose has two steps. First, we formulate the template learning translation as the equivalent problem that can be solved by using the HMM model based on a set of constraints rules which are observed from the characteristic of SL and TL and on a training corpus. Afterward, a dynamic programming technique, a variant of Viterbi algorithm is used to find the best translation results.

Problem: Given an input sentence $e_1e_2...e_m$ and a set of template rules $r_1, r_2, ..., r_d$, find the set of rules so that their translation results most explain for that sentence. For convenience we will use e [1: m] as shorthand for the input sentence $e_1e_2...e_m$.

The problem is equivalent to find all translation results for each rule r_i (i=1, d). Assuming that the rule r_i is defined as $S_1S_2...S_n \leftrightarrow T_1T_2...T_k$, the original method (Cicekli 1996) (Gűnevir 1998) tries to find all ways to replace variables with phrases in SL so that the input sentence e [1:m] can be produced from this rule. Afterward, find each corresponding phrases in TL within set of lexical rules with a phrase in SL in order to transform the input sentence into the target language. However, when the input sentence is long and the number of rules is large with a much number of variables inside, the original method have to cope with the exponential calculation. To overcome the problem, we propose an approach based on HMM modeling below. Figure 2 shows that an input sentence can be decomposed into many ways using the left side of template rules. Suppose that the variable X and Y have 10 elements respectively whose substrings can be found in the input sentence and which were left sides of lexical rules. For each element in the variable X which have a position k within the input sentence we have to find all elements for the variable Y that has substrings which starts from a position k+1 and was a left side of a lexical rule. Thus, we have to consider 10 x 10 translation ways while most of them need to be cut off. From the example in the figure 2, each constant s_j can be associated with a phrase in the right side of the rule r_i and each variable s_j within the rule r_i can be associated with a set of lexical rules whose left side is a substring that starts from possible positions within the input sentence.

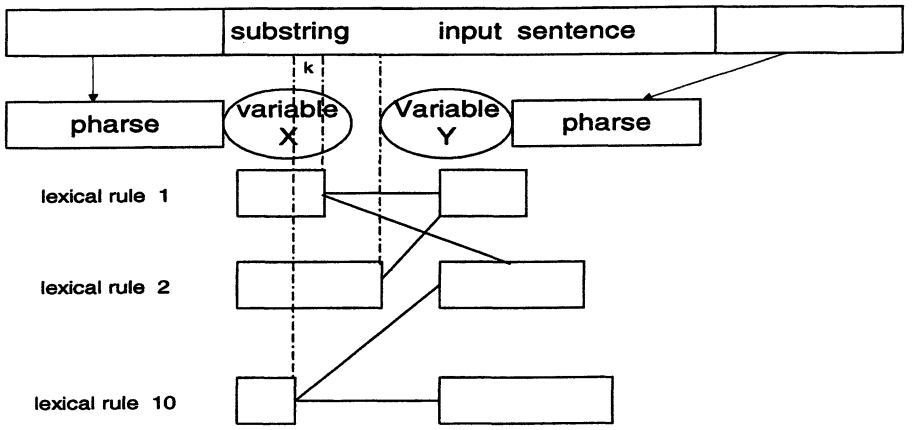

Figure 2. Example of translation based HMM

In such framework, we can asssume that a lexical rule corresponds to a *hidden state* and a substring in the input sentence as an *observed symbol* procduced form the state and the problem of translation is equivalent to find a lexical rule for each variable. Accordingly, the problem can can be solved by using the variant of HMM modeling as mentioned above.

To find the most likely sequence of lexical rules, we must find a sequence of lexical rules that maximized the probability $P(r_i \mid e_1, e_2, ...e_m)$.

Since r_i: $S_1 S_2 ... S_n \leftrightarrow T_1 T_2 ... T_k$

$P(r_i \mid e_1, e_2, ..., e_m) = P(S_1, S_2, ..., S_n \mid e_1, e_2, ..., e_m)$.

$$P(S_1, S_2, ..., S_n \mid e_1, e_2, ..., e_m) = \frac{P(e_1, e_2, ...e_m \mid S_1, S_2, ...S_n)}{P(e_1, e_2, ..., e_m)} \times P(S_1, S_2, ...S_n) \quad (1)$$

Since $e_1 e_2 ... e_m$ is a sequence of input words, and the probability $P(e_1, e_2, ..., e_m)$ is given, we need to maximize the formula below.

$$P(e_1, e_2, .., e_m \mid S_1, S_2, .., S_n) \times P(S_1, S_2, ..., S_n) \quad (2)$$

Using the Bigram model, (2) can be approximated as

$$\prod_{1}^{n-1} P(S_{j+1} \mid S_j) \times \prod_{1}^{n-1} P(e_{j_1} e_{j_2} ... e_{j_k} \mid S_j) \quad (3)$$

where $e_{j_1} ... e_{j_k}$ matches with the left side of a lexical rule matching with S_j.

To find the sequence of lexical rules that maximizes the formula (3), a kind of dynamic programming, the Viterbi algorithm (Viterbi, A.J 1967) can be used. If the rule r_i has n variables and each variable consists of l elements then the complexity is $n \times l^2$, while the recursive way be l^n. In addtion, each rule r_i can be assigned a translation score as the value of the formular (3) and output translations for the input sentence can be sorted according to the score value on the whole of rules in template rules. Therefore, our method using HMM modeling can be avoided the exponential calculation problem by using the dynamic algorithm. In addition, it can sort translation results according to the better accuracy without any complex process on set of template rules. Moreover, it draws a new perspective for applying statistical machine learning theories on the example based translation domain.

3.2.2 Estimate HMM model

The HMM model for translation is estimated by using the Forward-Backward learning (L.E.Baum and J.A.Eagon. 1967) described as follows. The corpus of source sentences and target sentences will be used to generate observed sequences. Each source sentence will be translated by using a sequence of lexical

rules if the right hand side of the rules are the same with the target sentence within the corpus. After obtaining a sequence of lexical rules, the sequence of observed symbols is generated because each observed symbol is a left hand side of a lexical rule. Therefore, using a set of template rules and the corpus we can generate a training data formed as follow:

$O_{t_1}, O_{t_2+1}, ..., O_{m_1} \Leftrightarrow S_{t_1}, S_{t_2+1}, ..., S_{m_1}$

$O_{t_2}, O_{t_2+1}, ..., O_{m_2} \Leftrightarrow S_{t_2}, S_{t_2+1}, ..., S_{m_2}$

....

$O_{t_k}, O_{t_k+1}, ..., O_{m_k} \Leftrightarrow S_{t_k}, S_{t_k+1}, ..., S_{m_k}$

Here, $O_{t_k}, O_{t_k+1}, ..., O_{m_k}$ is a sequence of observed symbols, $S_{t_k}, S_{t_k+1}, ..., S_{m_k}$ is a sequence of lexical rules and $O_{t_k}, O_{t_k+1}, ..., O_{m_k} \Leftrightarrow S_{t_k}, S_{t_k+1}, ..., S_{m_k}$ means that a sequence of observed symbols is associated with the sequence of lexical rules.

Suppose that, $c(l^j), c(l^j, l^k)$ and $c(o^j, l^k)$ be the number of occurrences of lexical rule l^j, the number of occurrence of the lexical rule l^j following the lexical rule l^k and the number of occurrences of a observed symbol o^j corresponding with a lexical rule l^k respectively. With these notations, the initialization algorithm for estimating an HMM model by performing the Forward-Backward algorithm on the training data above is described as follows:

For all lexical l^j do

  For all lexical rule l^k do

$$P(l^k \mid l^j) = \frac{c(l^j, l^k)}{c(l^j)}$$

For all lexical rule l^j do

  For all observed symbols o^l do

$$P(o^l \mid l^j) = \frac{c(o^l, l^j)}{c(l^j)}$$

Figure 3. Algorithm for initializing the parameters of HMM model for template rules

After initialization the probabilistic of observed symbols and lexical rules, the Forward-Backward learning is used to estimate the HMM for translation.

3.2.3 Example

We describe an example of translation using the orignal method and the HMM method for an input sentence with a template rule and a set of lexical rules as shown in the Table 1.

There are three translation outputs when applying the orignal method, which are (1, 2,3), (1,4,5),(1,6,7). Suppose that the probabilistic of two lexical rules in the example are estimated as follows:

P(1|2)=0.2; P(1|4)=0.6; P(1|6)=0.2 ; P(2|3)=0,2; P(4|5)=0.5; P(6|7)=0.2.

Using the formula (3), we have P(1,2,3)=0.04, P(1,4,5)=0.3, P(1,6,7)=0.04. Thus, the translation result is the likely sequence of lexical rules (1,4,5).

Table 1. An example of translation using template translation learning

Input: *I do not think it is necessary to launch a full inquiry at this time*	
Lexical rule	Template rule: X "necessary to launch" Y Z <=> X' "can thiet de bat dau" Y' Z'
1	I do not think it is ↔ toi khong nghi no la
2	a full ↔ su day du
3	Inquriry at this time ↔ doi hoi o thoi diem nay
4	a full inquiry ↔ mot cuoc dieu tra day du
5	at this time ↔ o thoi diem nay
6	a full inquiry at ↔ mot cau hoi day du o
7	this time ↔ thoi gian nay
Human translation: *Toi khong nghi la no thuc su can thiet de bat dau cuoc dieu tra o thoi diem nay.* EBMT(the orignal algorithm have to enumerate all translation results) (1,2,3) : *toi khong nghi la can thiet de bat dau su day du doi hoi o thoi diem nay.* (1,4,5): *toi khong nghi no la can thiet de bat dau mot cuoc dieu tra day du o thoi diem nay.* (1,6,7): *toi khong nghi no la can thiet de bat dau mot cau hoi day du o thoi gian nay.* HMM: (The proposed method obtains a best translation) (1,4,5): *toi khong nghi no la can thiet de bat dau mot cuoc dieu tra day du o thoi diem nay*	

Table 1 shows that the orignal method have to enumerate all translation results and the prposed method can obtain a best translation results by applying a dynamic algorithm.

4 Experiments and Discussion

In order to assert our method can enhance the accuracy in translation while ensuring the complexity is low. We implemented an English Vietnamese translation and tested on a corpus of 1200 bilingual sentences collected manually from some text books and newspapers and experimenting on the HMM model.

4.1 Template Translation Learning

Figure 4 shows the number of template translation learning with the number of sentence within the corpus. This results shows how the size of the template rules for a bilingual corpus of English- Vietnamese language.

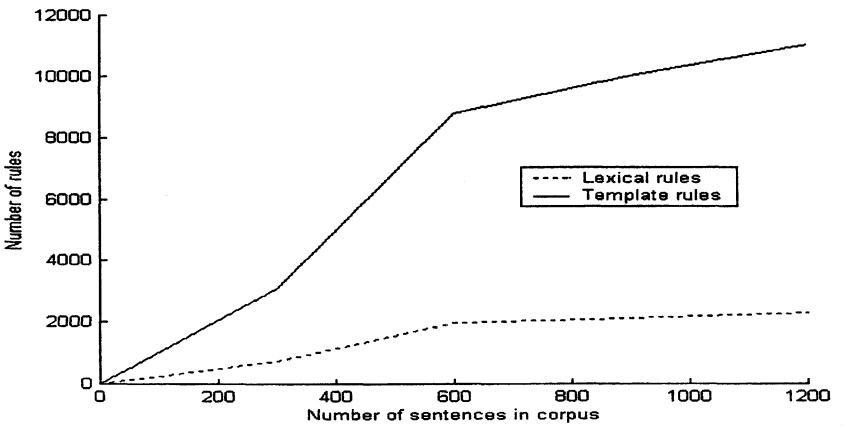

Figure 4. The relation of the number of lexical rules and the number of template rules with the number of sentences within the corpus.

The number of sentences in corpus is from 300 sentences to 1200 sentences. The solid line and the doted line show the relation between the number of template rules and the number of lexical rules with the number of sentences within the corpus respectively.

4.2 HMM model

The number of template rules and the number of lexical rules using the template translation learning is 11,034 rules and 2,287 rules respectively. The number of lexical rules is the number of hidden states in our HMM model. Using the template rules and the data corpus, we obtained the training data for estimating HMM model described in section 3.2.2, then the initialize parameters for the HMM model is estimated by using the algorithm in Figure 3. The training data for estimating the HMM model consists of 1200 observed sequences and each sequence corresponding with a sequence of lexical rules. We used 1100 observed sequence to initialize the parameters for HMM models by performing the algorithm in Figure 3. Afterward, the remained sequences is applied Forward and Backward algorithm to train the model.

4.2 Translation Results

After we generated a set of template rules on the corpus, we estimated the HMM model as mentioned above. We tested the translation accuracy by using the sentences within the corpus.
Using the Viterbi algorithm for each rule, we are able to obtain a list of output translations. We used the lists of translation results of our method and the original method for comparing. We compared by calculating correct translations among the total translation output. We obtained the table as shown in Table 2. The sentences within the corpus were selected randomly and used as inputs for the original method and our method respectively. The first is by the original method (Gűvenir 1998) and the second is by our method. The first column and the second column are the percentage of correct translation results among total translation results by applying the baseline and our method respectively.
Table 2 show that our method achieved the better results in comparing with the orignal TTL algorithm. In addtion, our method achieved a lower complexity $O(n \times l^2)$ in comparing with the original method $O(l^n)$, in which l is the number of lexical rules and n is number of variables in a template rule . This was due to our method based on a dynamic way to avoid the exponential problem.

Table 2. Performance results

Percentage of correct results by the orignal method	Percentage of correct results with HMM
34%	81%

Some examples of our translation method and the orignal method is desribed in the Table 3. Table 3 shows a translation results of our methods in the second column. Table 3 shows a best translation results in testing sentences within the corpus.

Table 3. Some examples of our translation results

Input sentence	Translation output
How long will you stay here ?	Anh sẽ ở lại đây được bao lâu ?
My book is as interesting as yours	Quyển sách của tôi thì lý thú ngang với quyển sách của anh
Several new proposals are being considered by the committee	Nhiều dự án mới đang được ủy ban cứu xét
Before long rice seedlings were big enough to be planted in the field	Chẳng bao lâu sao các cây lúa đó đủ lớn để được cấy vào ruộng đó.
Have you written your report yet ?	Anh viết xong bản báo cáo chưa ?
If she had seen the movie, she would have told you	Nếu cô ta đã nhìn thấy phim, cô ta đã nói với bạn

5 Conclusion

Our method using HMM modeling can avoid the exponential calculation problem by using a dynamic algorithm. In addition, it can sort translation results according to the better accuracy without any complex process and , the preliminary experiment shoved the high translation accuracy compared with the previous method. Moreover, it draws a new perspective for applying statistical machine learning theories on the example based translation domain. Mearging our proposed translation method and rule based translation method is currently under way.

Acknowledgments

This research was supported in part by the international research project grant, JAIST.

References

Nagao,M.A. (1984). *Framework of a mechanical translation between Japanese and English by analogy principle.* in Artificial and Human Intelligence, edited by A. Elithorn and R Banerji, NATO publication: North-Holland, Edinburgh, 1984.pp. 173-180.

Cicekli,I, and Guvenir,H.A. (1996).Learning *Translation Rules From A Bilingual Corpus*.Proceeding of the 2nd International Conference on New Method in Language Proceesing (NeMLaP-2), Ankara, Turky, September 1996:90-97.

Guvenir,H.A, and Cicekli,I. (1998). *Learning translation templates from examples.* In Information System, 23(6):353-363.

Öz,Z. and Cicekli,I. (1998). Ordering *Translation Templates by Assignning Confidence Factors.* Proceeding of the 3rd Conference of Assocaition for Machine Translation in the Mericas, Langhorne, PA, 51-61.

Viterbi, A.J .(1967). *Error bounds for convolution codes and an asymptotically optimal decoding algorithm.* IEEE Trans on Information Theory 13: 260-269.

L.E.Baum and J.A.Eagon. 1967. "An inequality with application to statistical estimation for probabilistic functions of markov processes and to a model of ecology" . Bull. Amer. Math. Soc., 73:360-363.