

Toward Memory-based Translation

Satoshi SATO and Makoto NAGAO
Dept. of Electrical Engineering, Kyoto University
Yoshida-honmachi, Sakyo, Kyoto, 606, Japan
sato@kuee.kyoto-u.ac.jp

Abstract

An essential problem of example-based translation is how to utilize more than one translation example for translating one source sentence.

This paper proposes a method to solve this problem. We introduce the representation, called *matching expression*, which represents the combination of fragments of translation examples. The translation process consists of three steps: (1) Make the source matching expression from the source sentence. (2) Transfer the source matching expression into the target matching expression. (3) Construct the target sentence from the target matching expression.

This mechanism generates some candidates of translation. To select the best translation out of them, we define the score of a translation.

1 Introduction

Use of extracted information from examples or example-based translation is becoming the new wave of machine translation. The basic idea of example-based translation is very simple: translate a source sentence by imitating the translation example of a similar sentence in the database. The idea first appeared in [Nagao 84], and some research has followed it [Sumita 88][Sato 89][Sadler 89a][Sadler 89b]. But a great deal of effort is still needed to implement the idea.

In our previous work, we show how to select the best target word in case-frame translation based on examples[Sato 89]. In this paper, we concentrate on two problems:

1. How to combine some fragments of translation examples in order to translate one sentence?
2. How to select the best translation out of

many candidates?

We show partial solutions for them in MBT2. MBT2 is the second prototype system in our Memory-based Translation Project. MBT2 can do bi-directional translation between an English word-dependency tree and a Japanese word-dependency tree. It is implemented in Sicstus Prolog.

2 Need to Combine Fragments

The basic idea of example-based translation is very simple: translate a source sentence by imitating the translation example of similar sentence in the database. But in many cases, it is necessary to imitate more than one translation example and combine some fragments of them.

Let's consider the translation of the following sentence.

- (1) He buys a book on international politics.

If we know the following translation example (2) and (3), we can translate sentence (1) into sentence (4) by imitating examples and combining fragments of them.

- (2) *He buys* a notebook.
Kare ha nouto wo kau.
- (3) I read *a book on international politics.*
Watashi ha kokusaiseiji nitsuite kakareta hon wo yomu.
- (4) *Kare ha kokusaiseiji nitsuite kakareta hon wo kau.*

It is easy for a human to do this, but not so for a machine. The ability to combine some fragments of translation examples is essential to example-based translation. A lack of this ability restricts the power of example-based translation. In this paper, we concentrate on the implementation of this ability on machine.

3 Matching Expression

To implement the ability to combine some fragments of translation example in order to translate one sentence, we must determine the following:

- how to represent translation examples
- what is a fragment
- how to represent the combination of fragments

3.1 Translation Database

The translation database is the collection of translation examples. A translation example consists of three parts:

- an English word-dependency tree (EWD)
- a Japanese word-dependency tree (JWD)
- correspondence links

For example, in Prolog,

```
ewd_e([e1,[buy,v],
      [e2,[he,pron]],
      [e3,[notebook,n],
      [e4,[a,det]]]]).
%% He buys a notebook.
jwd_e([j1,[kau,v],
      [j2,[ha,p],
      [j3,[kare,pron]]],
      [j4,[wo,p],
      [j5,[nouto,n]]]]).
%% Kare ha nouto wo kau.
clinks([[e1,j1],[e2,j3],[e3,j5]]).
%% e1 <-> j1, e2 <-> j3, e3 <-> j5
```

Each number with prefix 'e' or 'j' in word-dependency trees represents the ID of the subtree. Each node in a tree contains a word (in root form) and its syntactic category. A correspondence link is represented as a pair of IDs.

3.2 Translation Unit

A word-dependency (sub)tree which has a correspondence link is translatable; e.g. e1, e2, e3, j1, j3, j5. A translatable tree in which some translatable subtrees are removed is also translatable; e.g. e1-e2, e1-e3, e1-e2-e3, j1-j3, j1-j5, j1-j3-j5. Both of them are translatable fragments. Sadler calls them *translation units*[Sadler 89a].

3.3 Matching Expression

Next we will introduce the concept 'matching expression.' Matching expression (ME) is defined as the following:

```
<ME> ::= [<ID>|<ME-Commands>]
<ME-Commands> ::=
    []
    or [<ME-Command>|<ME-Commands>]
<ME-Command> ::=
    [d,<ID>]           %% delete <ID>
    or [r,<ID>,<ME>]   %% replace <ID>
                    %% with <ME>
    or [a,<ID>,<ME>]   %% add <ME> as a
                    %% child of root
                    %% node of <ID>
```

Every ID in an ME should be translatable. We assume the example in Section 3.1 and the following example.

```
ewd_e([e11,[read,v],
      [e12,['I',pron]],
      [e13,[book,n],
      [e14,[a,det]],
      [e15,[on,p],
      [e16,[politics,n],
      [e17,[international,adj]
      ]]]]]).
%% I read a book on international
%% politics.
jwd_e([j11,[yomu,v],
      [j12,[ha,p],
      [j13,[watashi,pron]]],
      [j14,[wo,p],
      [j15,[hon,n],
      [j16,[ta,aux],
      [j17,[reru,aux],
      [j18,[kaku,v],
      [j19,[nitsuite,p],
      [j20,[kokusaiseiji,n]
      ]]]]]]]].
%% Watashi ha kokusaiseiji nitsuite
%% kakareta hon wo yomu.
clinks([e11,j11],[e12,j13],[e13,j15],
      [e16,j20]).
```

Under this assumption, the word-dependency tree (a) can be represented by the matching expression (b).

```
(a) [[buy,v],
     [[he,pron]],
     [[book,n],
     [a,det]],
     [on,p],
     [[politics,n],
     [[international,adj]]]]]]
%% He buys a book on international
%% politics.
(b) [e1,[r,e3,[e13]]]
```

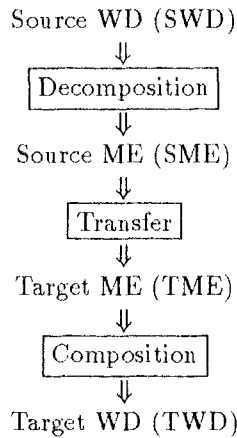


Figure 1: Flow of Translation

The matching expression (b) consists of two translation units: e1-e3, e13. And it has the information to combine them.

4 Translation via Matching Expression

Figure 1 shows the flow of the translation process. The translation process consists of three steps: decomposition, transfer, and composition. This process generates all candidates of translation using Prolog's backtrack mechanism.

4.1 Decomposition

In decomposition, the system decomposes a source word-dependency tree(SWD) into translation units, and makes a source matching expression(SME). For example,

```

SWD = [[buy,v],
        [[he,pron]],
        [[book,n],
         [[a,det]],
         [[on,p],
          [[politics,n],
           [[international,adj]]]]]]]
SME = [e1,[r,e3,[e13]]]

```

The main tasks in this step are to retrieve translation units and compare the source WD with retrieved translation units. To retrieve translation units quickly, we use some hashing techniques. There are two programs to do the comparison task; one for English WDs and one for Japanese WDs. In comparison of Japanese WDs, the order of subtrees is not important.

To reduce the search space and the number of candidates, we define replaceability between syntactic categories. If two nodes are replaceable, system makes only a replace-command. As a result, the system does not make some matching expressions; e.g. [e1,[d,e3],[a,e1,[e13]]]

4.2 Transfer

In the transfer step, the system replaces every ID in the source matching expression with its corresponding ID. For example,

```

SME = [e1,[r,e3,[e13]]]
TME = [j1,[r,j5,[j15]]]

```

4.3 Composition

In the composition step, the system composes the target word-dependency tree according to the target matching expression. For example,

```

TME = [j1,[r,j5,[j15]]]
TWD = [[kau,v],
        [[ha,p],
         [[kare,pron]]],
        [[wo,p],
         [[hon,n],
          [[ta,aux],
           [[reru,aux],
            [[kaku,v],
             [[nitsuite,p],
              [[kokusaiseiji,n]]]]]]]]]]]
%% Kare ha kokusaiseiji nitsuite
%% kakareta hon wo kau.

```

This step divides into two sub-steps; the main composing step and validity checking. In the main composing step, there is no ambiguity with one exception. Because an add-command [a,<ID>,<ME>] specifies only the parent node(<ID>) to add the tree(<ME>), there are some choices in composing English word-dependency trees. In this step, all possibilities are generated.

Validity of the composed word-dependency trees are checked using syntactic categories. Validity is checked in every parent-children unit. For example, in the above target word-dependency tree,

```

[v,[p,p]], [p,[pron]], [p,[n]],
[n,[aux]], ...

```

are checked. A unit is valid if there is a unit which has the same category pattern in the database. A word-dependency tree is valid if all parent-children units are valid.

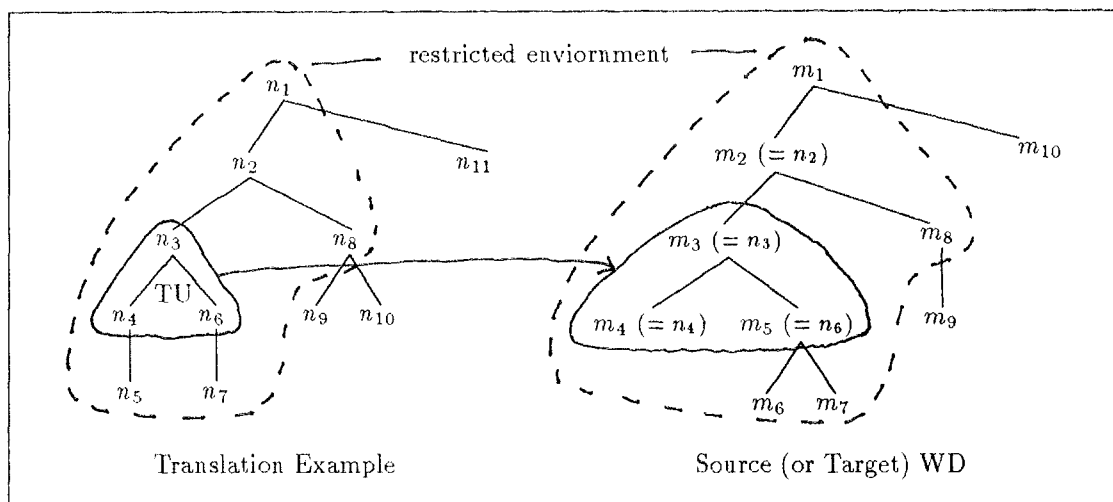


Figure 2: Restricted Environments of TU

5 Score of Translation

To select the best translation out of all candidates generated by system, we introduce the score of a translation. We define it based on the score of the matching expression, because the matching expression determines the translation output. The scores of the source matching expression and the target matching expression are calculated separately.

5.1 Score of Translation Unit

First, we will define the score of a translation unit. The score of a translation unit should reflect the correctness of the translation unit. Which translation unit is better? Two main factors are:

1. A larger translation unit is better.
2. A translation unit in a matching expression is a fragment of a source (or target) word-dependency tree, and also a fragment of a translation example. There are two environments of a translation unit; in a source (or target) tree and in a translation example. The more similar these two environments are, the better.

To calculate 1, we define the size of a translation unit(TU).

$$size(TU) = \text{the number of nodes in TU}$$

To calculate 2, we need a measure of similarity between two environments, i.e. external similarity. To estimate external similarity, we introduce a unit called *restricted environment*. A restricted environment consists of the nodes one link outside of a TU normally. If corresponding nodes are same in two environments, those environments are extended one more link outside. Figure 2 illustrates restricted environments of a TU. We estimate external similarity as the best matching of two restricted environments. To find the best matching, we first determine the correspondences between nodes in two restricted environments. Some nodes have several candidates of correspondence. For example, n_7 corresponds with m_6 or m_7 . In this case, we select the most similar node. To do this, we assume that similarity values between nodes (words) are defined as numeric values between 0 and 1 in a thesaurus. When the best matching is found, we can calculate the matching point between two environments, $mpoint(TU, WD)$.

$$mpoint(TU, WD) =$$

summation of similarity values between corresponding nodes in two restricted environments at the best matching

We use this value as a measure of similarity between two environments.

Finally, we define the score of a translation unit, $score(TU, WD)$.

$$score(TU, WD) =$$

$$size(TU) \times (size(TU) + mpoint(TU, WD))$$

For example, we assume that the following similarity values are defined in a thesaurus.

```

sim([book,n],[notebook,n],0.8).
sim([buy,v],[read,v],0.5).
sim([hon,n],[nouto,n],0.8).
sim([kau,v],[yomu,v],0.5).

```

Then the scores of translation units in the previous section are the followings.

TU	size	mpoint	score
e1-e3	2	0.8	5.6
e13	5	0.5	27.5
j1-j5	4	0.8	19.2
j15	6	1.5	45

5.2 Score of Matching Expression

The score of a matching expression is defined as the following.

$$score(ME, WD) = \frac{\sum_{TU \in ME} score(TU, WD)}{size(WD)^2}$$

For example,

ME	score
[e1, [r, e3, [e13]]]	0.676
[j1, [r, j5, [j15]]]	0.642

5.3 Score of Translation

Finally, we define the score of a translation as the following.

$$score(SWD, SME, TME, TWD) = \min(score(SME, SWD), score(TME, TWD))$$

For example, the score of the translation in the previous section is 0.642.

6 Examples

The English verb *eat* corresponds to two Japanese verbs, *taberu* and *okasu*. For example,

- (4) The man *eats* vegetables.
Hito ha yasai wo *taberu*.
- (5) Acid *eats* metal.
San ha kinzoku wo *okasu*.

Figure 3 shows translation outputs based on example (4) and (5) by MBT2. MBT2 chooses *taberu* for *he eats potatoes* and *okasu* for *sulfuric acid eats iron*.

```

*** Translation Source ***
[[eat,v],
 [[he,pron]],
 [[potato,n]]]
%% He eats potatoes.
*** Translation Results ***
No. 1 (Score = 0.5889)
[[taberu,v],
 [[ha,p],
 [[kare,pron]]],
 [[wo,p],
 [[jagaimo,n]]]]
No. 2 (Score = 0.4556)
[[okasu,v],
 [[ha,p],
 [[kare,pron]]],
 [[wo,p],
 [[jagaimo,n]]]]

*** Translation Source ***
[[eat,v],
 [[acid,n],
 [[sulfuric,adj]]],
 [[iron,n]]]
%% Sulfuric acid eats iron.
*** Translation Results ***
No. 1 (Score = 0.5500)
[[okasu,v],
 [[ha,p],
 [[ryuusan,n]]],
 [[wo,p],
 [[tetsu,n]]]]
No. 2 (Score = 0.4688)
[[taberu,v],
 [[ha,p],
 [[ryuusan,n]]],
 [[wo,p],
 [[tetsu,n]]]]

```

Figure 3: Translation Outputs by MBT2

7 Discussion

Although MBT2 is not a full realization of Nagao's idea, it contains some merits from the original idea.

1. It is easy to modify the system.
The knowledge of the system is in the form of translation examples and thesauri. We can modify the system with addition of translation examples.
2. It can do high quality translation.
The system sees as wide a scope as possible in a sentence and uses the largest translation units. It produces high quality translations.
3. It can translate some metaphorical sentences.
In the system, semantic information is not used as constraints. As a result, the system can translate some metaphorical sentences.

Demerits or problems of the system are:

1. A great deal of computation is needed.
2. Can we make good thesauri?

The first problem is not serious. Parallel computation or some heuristics will overcome it. But the second problem is serious. We have to study how to construct large thesauri.

8 Conclusion

This paper describes how to combine some translation units in order to translate one sentence and how to select the best translation out of some candidates generated by system. To represent the combination of fragments, we introduce the representation called matching expression. To select the best translation, we define the score of translation based on the score of the matching expression.

This framework can be applied to not only the translation between word-dependency trees but also the translation between other data structures. We hope that generation can be implemented in same framework as the translation from a word-dependency tree to a list or string.

acknowledgments

The authors would like to thank Mort Webster for his proof reading.

References

- [Nagao 84] Makoto Nagao, A Framework of a Mechanical Translation between Japanese and English by Analogy Principle, in ARTIFICIAL AND HUMAN INTELLIGENCE (A. Elithorn and R. Banerji, editors), Elsevier Science Publishers, B.V, 1984.
- [Sadler 89a] Victor Sadler, The Bilingual Knowledge Bank(BKB), BSO/Research, 1989.
- [Sadler 89b] Victor Sadler, Translating with a simulated Bilingual Knowledge Bank(BKB), BSO/Research, 1989.
- [Sato 89] Satoshi Sato and Makoto Nagao, Memory-based Translation, IPSJ-WG, NL-70-9, 1989. (in Japanese)
- [Sumita 88] E. Sumita and Y. Tsutsumi, A Translation Aid System Using Flexible Text Retrieval Based on Syntax-Matching, TRL Research Report, TR-87-1019, Tokyo Research Laboratory, IBM, 1988.