

A Pattern-Based Machine Translation System — Yakushite Net MT Engine

Miki Sasaki and Toshiki Murata

Oki Electric Industry Co., Ltd.
2-5-7 Hon-machi, Chuo-ku, Osaka 541-0053, JAPAN
{sasaki234, murata656}@oki.com

Abstract

This paper introduces our pattern-based machine translation system. Our system enables us to register translation patterns including more complicated sentence structures rather than just simple words and limited types of expressions. Having registered in the Japanese to English section of IWSLT05, we aimed to input as many expressions as possible to our system in order to achieve successful translation of travel expression into English within a limited period of time.

1. Introduction

Our machine translation in our system is pattern-based. With this system translation knowledge description is more flexible as the knowledge is written in the form of “translation pattern”. We challenged IWSLT05, the Japanese-English task, manual transcription and C-STAR track with this machine translation system of ours. To tackle this task, we focused on manual registration of translation rules which we extracted from the training corpus. Although the score we got in the evaluation campaign was not what we can be proud of, we proved that the translation patterns in our system are sufficiently effective. The next section presents our system description. Section 3 reports our training methods planned. Section 4 reports our results. Finally, Section 5 presents our conclusions.

2. System Description

Our system (Yakushite Net MT Engine) uses a pattern-based machine translation method [1][2][3][4][5]. In this method, all grammars, word dictionaries, idioms, expressions, and sentences which are needed for translation are treated as translation patterns, and translation is done by preparing massive amount of patterns and using some of those patterns in appropriate combinations. Because grammars and word dictionaries are treated as translation patterns, they both can be registered in the same way to our system.

Likewise, we can register translation patterns including more complicated sentence structures rather than just simple words and limited types of expressions. Idiomatic phrases that are characteristic of travel conversations, which are usually difficult to describe in a general dictionary, can also be registered as translation patterns to the system and used in the machine translation.

2.1. Architecture of Our Pattern-Based Machine Translation System

Figure 1 shows the architecture of our system. Straight arrows show the flow of the translation and dotted arrows show the sequence for dictionary reference.

First, the source sentence is analyzed morphologically, normalizing words and decorating them with morphological features by the morphological analyzer. This decorated sequence of words is then passed to the parser.

The sentence is parsed by using the translation patterns in the appropriate dictionaries. Our system has the user and system dictionaries. When the sentence is parsed successfully, the parse tree is translated by top-down generation of the parse tree of the target language.

Then, the post generator handles the generated tree to produce refined sentences.

Lastly, the morphological synthesizer adjusts inflection and conjugation, and the translated sentence is outputted.

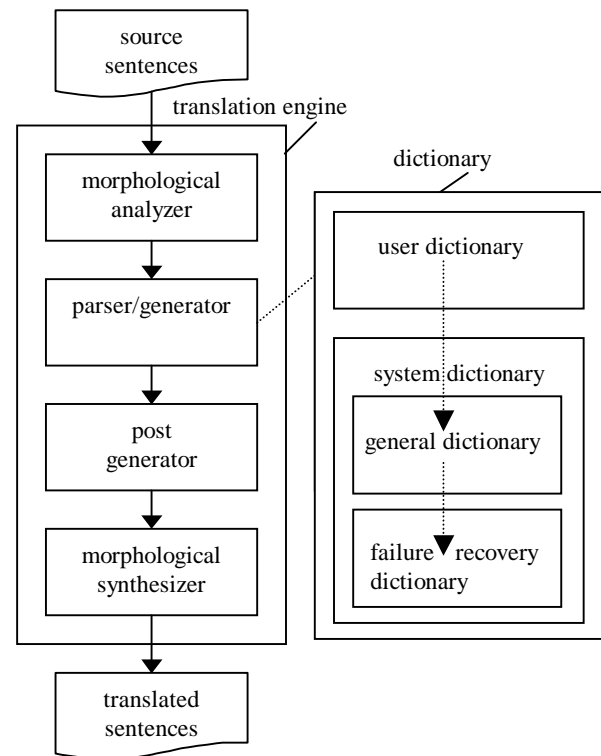


Figure 1: the Architecture of our System

2.2. Translation Patterns

Translation patterns are knowledge necessary to the translation engine. All the word dictionaries, grammars, and syntactical dictionaries are written in this style in our engine. In our parsing and generating method, rules of Context-free Grammar (CFG) are paired, and a sentence is parsed by using the source side of translation patterns, and by combining them. CFG is a formal grammar in which every production rule is of the form

V -> w

where V is a non-terminal symbol and w is a string consisting of terminals and/or non-terminals. In CFG, the left-hand side of a production rule may only be formed by a single non-terminal symbol. The terminal symbols correspond to words in the vocabulary. The non-terminal symbols correspond to legal phrases or sentences that are formed using words from the vocabulary.

Translation patterns are basically written in the format described below:

LHS is Left-hand side of a CFG rule.

RHS is Right-hand side of a CFG rule.

[Name of the language1 : LHS RHS]

[Name of the language2 : LHS RHS]

:

[Name of the language3 : LHS RHS] ;

Analysis uses source language patterns, and generation uses target language patterns. For example, translation from language2 into language1 uses patterns of language2 for analysis, and patterns of language1 for generation. This format allows pattern description of more than one language enabling easier multilingual processing.

A pattern starts with the name of the language, and a category and features on the left-hand side of the CFG rule (the parent node in the parse tree), followed by descriptions of non-bracketed words and bracketed non-terminals on the right-hand side of the CFG rule, in their textual order. “:” is a separator between features of a pattern element, and space is a separator between pattern elements.

Patterns come in pairs: one pattern for each language. The mandatory numerical index in non-terminals allows relating non-terminals elements between source and target patterns.

Analysis uses source language patterns. By applying patterns, word sequences can be reduced to the corresponding left-hand side, and eventually reach the “S” non-terminal (the root of the parse tree).

When the source parse tree has been completed, each node using the corresponding target language pattern is sufficiently converted. Since there is a one-to-one relation between non-terminals in the source and target patterns, generation of the target parse tree is carried out immediately.

Following is an example of translation patterns used in Japanese to English translation. The description of this example is simplified so it is easier to understand. The Japanese sentence is the input sentence, while the English is the output.

Japanese : 「彼(kare)は(wa)どこに(dokoni)行く(iku)か(ka)」

English : “Where does he go?”

(Pattern 1)

[ja:S [1:Sintr:*]]

[en:S [1:Sintr:*] ?;pos=punc];

(Pattern 2)

[ja:Sintr [2:NPJoshi:case=subj:personNum=3sg]
[1:Fsintr] [3:VP:* ;jSentenceType=interrogative]]

[en:Sintr [1:AdvIntr] do;pos=v:personNum=3sg
[2:NP] [3:VP:*]];

(Pattern 3)

[ja:Fsintr どこに;:pos=fs]

[en:AdvIntr where;:pos=adv];

(Pattern 4)

[ja:NPJoshi:case=subj [2:NP:*] 彼;:pos=fj]

[en:NP [2:NP:*:case=subj]];

(Pattern 5)

[ja:NP:personNum=3sg 彼;:pos=ms]

[en:NP he;:pos=prn];

(Pattern 6)

[ja:VP;jSentenceType=interrogative [1:VP:*]

か;:pos=ej]

[en:VP [1:VP:*]];

(Pattern 7)

[ja:VP 行く;:pos=ds]

[en:VP go;:pos=v];

For example, Pattern 1 is a patternized form of the following CFG rules.

Japanese : S -> Sintr

English : S -> Sintr ?

Pattern 1, Pattern 2, Pattern 4 and Pattern 6 are grammatical patterns (relevant to grammars in rule-based machine translation), and Pattern 3, Pattern 5 and Pattern 7 are vocabulary patterns (relevant to word dictionaries in rule-based machine translation). The left-hand side of Pattern 2 is bound to [1:] on the right-hand side of Pattern 1, and the left-hand sides of Pattern 3, Pattern 4 and Pattern 6 are bound respectively to [1:], [2:] and [3:] on the right-hand side of Pattern 2. “pos=v” in Pattern 2 means constraint that the part of speech of the words is verb. “personNum=3sg” in Pattern 2 means constraint that the person of the words is 3rd person and singular.

Constraints are given after the terminal and non-terminal symbols (“:”s) on the right-hand sides. “pos”, “v”, “personNum” and “3sg” are features. The features on the right-hand sides of the source language patterns express conditions. Matching succeeds if all these conditions are satisfied. Features on the left-hand sides of the target patterns act as constraints for the generation processes, and features on the left-hand sides are propagated to child nodes.

Asterisks(“*”) express special constraint. Only one asterisk(“*”) can be given on each right-hand side. A feature marked by an asterisk(“*”) is inherited by the left-hand side non-terminal in the analytical process, and a feature on the

left-hand side is inherited by the node in the generation process.

Next is the example of a grammatical pattern of an idiomatic expression.

To patternize the Japanese sentence which is translated as the English sentence:

Japanese : 「 彼 (kare) は (wa) どこ (doko) でしょう (desyou)か(ka)。 」

English : “Where is he?”

we expressed the translation pair as follows:

Japanese : 「 NP は(wa)どこ(doko)でしょう(desyou)か(ka)。 」

English : “Where is NP ?”

and the actual translation pattern which should be described in our system is:

(Pattern 8)

```
[ja:S [1:SIntr:*] 。 ]
[en:S [1:SIntr:*] ?:pos=punc];
```

(Pattern 9)

```
[ja:SIntr [1:NP:personNum=3sg] は どこ
だ:*:pos=jd う:pos=jd か]
[en:SIntr where
be:*:pos=be:personNum=3sg:conjug=pres [1:NP] ];
```

When there is no necessity to describe constraint in the right-hand side in the translation pattern, the system conducts morphological analysis to automatically convert the parallel texts into a translation pattern and register the pattern.

The following are the examples of the parallel text and the translation pattern:

Japanese : 「 ここ (koko) は (wa) どこ (doko) でしょう (desyou)か(ka)。 」

English : “Where are we?”

(parallel text)

「 ここはどこでしょうか。 」
“Where are we?”

(translation pattern)

(Pattern 10)

```
[ja:S ここ は どこ だ:*:pos=jd う:pos=jd
か 。 ]
[en:S where
be:*:pos=be:personNum=2sg:conjug=pres
we:pos=prn ?:pos=punc];
```

Not only sentences, but most nouns and phrases can also be converted from parallel texts into translation patterns and registered to the system. Non-terminal symbols like NP can also be included in the right-hand sides.

Multiple candidates that appear in parsing process are prioritized generally in this order:

1: One that has less patterns with more than one terminal symbol in the whole tree

2: One with less nodes in the whole tree

Therefore, registered sentences and expressions are prioritized. Adding to this rule, priority control marks described in 2.3. is used to give appropriate translation results to input sentences. With this priority determination system, more ambiguous input sentences are also accurately handled.

The features in the right-hand side of the source language patterns express conditions, either by assigning a specific value for a feature, or expressing a sharing constraint between two features, through unification variables in curly brackets like “{INF}”. Matching succeeds if all these conditions are satisfied.

The following is the example of assigning values of features:

Japanese : 「 彼 (kare) は (wa) 行く (iku) こと (koto) が (ga) できる (dekiru)。 」

English : “He can go.”

Translation pattern with auxiliary verb “can” :

```
[ja:VP:inf={INF} [1:VP:*:inf=rt] こと が でき
る:pos=ds:inf={INF} ]
[en:VP:personNum={PER}:conjug={CON}
can:pos=aux:personNum={PER}:conjug={CON}
[1:VP:*:conjug=bare] ];
```

Inflection of “can” in the right-hand side of the source language is assigned to “VP” in the left-hand side. After the corresponding parse tree is built, the values of the personNum and conjug in the “VP” in the left-hand side of the target language is assigned to “can” in the right-hand side.

The following are the examples of matching conditions. Word selection in the target language is realized by checking features. In the example below, semantics of NPJoshi, namely semantics of NP, matches the semantics of VP(=human). Therefore, “okiru” is translated into “get up”.

Japanese : 「 彼 (kare) は (wa) 起きる (okiru)。 」
English : “He gets up.”

```
! [ja:SSimple
[1:NPJoshi:case=subj:personNum=3sg:sem={SEM}]
[2:VP:*:sSem={SEM}]]
[en:SSimple [1:NP:personNum=3sg]
[2:VP:*:personNum=3sg:conjug=pres] ];
```

```
[ja:NP:personNum=3sg:sem=human 彼:*:pos=ms]
[en:NP he:*:pos=prn];
```

```
[ja:VP:sSem=human 起きる:*:pos=ds]
[en:VP get:*:pos=v up:pos=adv];
```

In the example below, semantics of NPJoshi, namely semantics of NP, doesn’t match the semantics of VP(=human). Therefore, “okiru” is translated into “occur”.

Japanese : 「事故(jiko)は(wa)起きる(okiru)。 」
 English : “An accident occurs.”

```
*[ja:SSimple [1:NPJoshi:personNum=3sg] [2:VP:*] ]
[en:SSimple [1:NP:personNum=3sg]
[2:VP:*:personNum=3sg:conjug=pres] ];
```

```
[ja:NP:sem=concept 事故:*:pos=ms]
[en:NP accident:*:pos=n];
```

```
*[ja:VP 起きる:*:pos=ds]
[en:VP occur:*:pos=v];
```

In the last part of the parsing, two decisions are taken to avoid multiplication of candidates. One of the decisions is that the set of features each non-terminal can have is limited according to a feature definition table.

```
Sintr = { jSentenceType };
VP = { jSentenceType
      personNum
      conjug
      };
```

For instance the CFG rule for “SIntr” does not need any longer conjugation, which is one of the features of head “VP”. With this limitation, every non-terminal symbol has only necessary features, which simplifies parsing trees. This is effective for reducing the number of candidates, in that non-terminals that have the combination of feature values can be merged, and disjunctive tree can be formed from the tree structure during parsing.

The other decision is that generation in the target language is not allowed to fail and backtrack: only one pattern is chosen on the basis of target side constraints if the source side pattern is identical (i.e., the decision is local). Otherwise, failures in feature constraints are ignored, and generation goes on assuming they succeeded.

2.3. Priority Control of Translation Patterns

The system has also priority control mechanism in order to avoid conflicts inherent in the pattern-based approach. The priority control is achieved by attaching an optional priority control mark to each translation pattern.

Multiple candidates given in parsing are ranked according to the priority marks added to the head of the translation patterns. Priority marks are: mark to raise the priority(exclamation(“!”) and plus(“+”)), mark to lower the priority(minus(“-“)), and mark to make the given pattern default(asterisk(“*”)).

Translation patterns starting with exclamation(“!”) have higher priorities. When a parsing process leads more than one result, the translation pattern with exclamation(“!”) is prioritized. exclamation(“!”) prioritizes a pattern, but it does not rule out other candidate patterns. Translation patterns starting with plus(“+”) are outputted prior to other when a source pattern has more than one possible translation. Translation patterns starting with minus(“-“) have lower priorities. When a parsing process leads more than one result, translation patterns with minus(“-“) are least prioritized. Also, patterns starting with asterisks(“*”) or asterisks(“*”) following exclamation(“!”), plus(“+”), or minus(“-“) are

default patterns. When a parsing process leads more than one result, and translation patterns with asterisk(“*”) is not included in the results, the default pattern is used.

By making an unconstrained pattern a default pattern, generation of a default pattern unnecessary to translation is controlled when a specialized pattern with various constraints matches.

2.4. Failure Recovery Dictionary

Our system also has the failure recovery dictionary which becomes active only when the normal parsing process failed. Patterns that are rarely used or grammatically irregular are registered to the failure recovery dictionary. By doing so, patterns unnecessary in normal translation is avoided, and it also prevents speed of simple sentence translation from slowing down. Whenever normal translation fails, the system tries again to translate with more patterns. This process is slower but much more robust.

For example, the Japanese sentence of the following parallel text is grammatically irregular having a verb and a particle omitted.

「魚(sakana)は(wa)どこ(doko)。 」
 ”Where is the fish?”

Therefore we registered the following pattern to the failure recovery dictionary.

```
[ja:SIntr [1:NP] は [2:DmsIntr] ]
[en:SIntr [2:PrnIntr]
be:pos=be:personNum=3sg:conjug=pres [1:NP] ] ;
```

2.5. Post Generator

Generation depends strongly on the structure of source language patterns, therefore pattern-based methods are weak at generating expressions peculiar to the target language.

Some features of the generated tree are handled by the post generator to produce refined sentences. The rule are written in XML notation.

For example, a Japanese sentence 「 VP こと(koto)が(ga)できる(dekiru)だろう(darou) 」 is will be translated to an English sentence “will can VP” by using the following rules:

```
Japanese : VP -> VP こと が できる
English : VP -> can VP
Japanese : VP -> VP だろう
English : VP -> will VP
```

However, correct English should be “will be able to VP”. A verb phrase immediately after an auxiliary verb should usually be generated in a bare form. However, in this example, the verb phrase after the auxiliary verb “will” is “can + verb”. In a case like this, what should be done is not to generate the bare form of “can”, but to change “can” to “be able to”, and generate the bare form of “be able to”.

Followings are the rules to correct the bare form of “can” to “be able to”. These are simple examples. More complicated correction rules can be written.

```
<Rule NAME="can(bare) be(bare) able to">
<StartLeaf>
<FeatureCondAnd>
<Feature NAME="baseForm" VALUE="can"/>
<Feature NAME="pos" VALUE="aux"/>
<Feature NAME="conjug" VALUE="bare"/>
</FeatureCondAnd>
```

```

</StartLeaf>
<Scope TYPE="NEAREST">
  <Feature NAME="category" VALUE="VP"/>
</Scope>
<Exclude>
  <Feature NAME="category" VALUE="NP"/>
</Exclude>
<OriginalLeaves DIR="LtoR">
  <OriginalLeaf ID="CAN" DIR="LtoR">
    <FeatureCondAnd>
      <Feature NAME="baseForm" VALUE="can"/>
      <Feature NAME="pos" VALUE="aux"/>
      <Feature NAME="conjug" VALUE="bare"/>
    </FeatureCondAnd>
  </OriginalLeaf>
</OriginalLeaves>
<EditedLeaves>
  <EditedLeaf ID="CAN" DELTA="0">
    <Feature NAME="baseForm" VALUE="be"/>
    <Feature NAME="pos" VALUE="be"/>
  </EditedLeaf>
  <EditedLeaf ID="CAN" DELTA="1">
    <Feature NAME="baseForm" VALUE="able"/>
    <Feature NAME="pos" VALUE="adj"/>
  </EditedLeaf>
  <EditedLeaf ID="CAN" DELTA="2">
    <Feature NAME="baseForm" VALUE="to"/>
    <Feature NAME="pos" VALUE="prep"/>
  </EditedLeaf>
</EditedLeaves>
</Rule>

```

3. Description of the Planned Training Methods for the Task of IWSLT05

At first, numerous shortcomings have been found in the translation results in which only the existing translation patterns made in the course of our development are used. This is because our system doesn't cover much of colloquial or abbreviated expressions, which are very often seen in BTEC (Basic Travel Expression corpus).

Registering the parallel texts as are given in the corpus would not be very effective when dealing with the actual test data, since those registrations can be used only when exactly the same sentences appear. Moreover, in some cases, two or more Japanese sentences in the corpus correspond to a single English sentence, making it difficult for us to automatically process the sentences.

Therefore, we manually made translation patterns that are highly generalized. Since this kind of manual process takes time and we had limited amount of time, we started with the patterns that appear more frequently instead of trying to cover every pattern that was found in the corpus.

We registered translation patterns from the IWSLT05 training corpus, and used the IWSLT04 test corpus for evaluation before the IWSLT05 test corpus release.

Following is our procedure:

- First, we manually extracted frequently used expressions from both Japanese and English sentences in the IWSLT05 training corpus.

- Secondly, we patternized those expressions replacing some parts with non-terminal symbols, and gave them appropriate translations.
- Thirdly, when some of the extracted patterns already exist in our system, but with some defects, we made corrections to the existing patterns.
- Fourthly, we registered the new patterns to our system.
- Lastly, in order to enhance the effectiveness, we extracted the parallel texts, each of which has only one Japanese sentence, from the IWSLT05 training corpus and the IWSLT04 test corpus, and registered them as-is like shown in Pattern 10 in Section 2.

We thus dealt with the task mainly manually, and although the coverage is limited to a certain part of the corpus due to its volume, the registered patterns and sentences are almost always effective in our experiment.

4. Results of the Training Sets and the Test Sets

Automatic evaluation results of the IWSLT04 test corpus are shown in Table 1. Automatic evaluation results of the IWSLT05 test corpus are shown in Table 2.

- (1) : Before we registered the new patterns.
- (2) : After we registered the new patterns.
- (3) : After we extracted the parallel texts with one Japanese sentence for each IWSLT05 training corpus and IWSLT04 test corpus, and registered them as-is like Pattern 10.

Table 1: IWSLT04 Test Set Results

	BLEU	NIST	WER	PER
(1)	0.1918	6.2283	0.6470	0.5640
(2)	0.2179	6.7882	0.5989	0.5183
(3)	0.7616	12.5216	0.2216	0.1894

Table 2: IWSLT05 Test Set Results

	BLEU	NIST	WER	PER
(1)	0.1918	6.3279	0.6749	0.5624
(2)	0.2222	6.8913	0.6314	0.5258
(3)	0.2639	7.3585	0.6066	0.5065

The following are the numbers of sentences of the IWSLT05 test set.

The number of Japanese original sentences IWSLT05_JE_TESTSET is 506 sentences. The number of input sentences after punctuating is 552 sentences.

Before we registered the new patterns, our system dictionary had approx. 2000 grammatical patterns and approx. 140000 word patterns. Also, our user dictionary had approx. 80000 word patterns, in which 60000 were proper nouns and 20000 were general nouns.

In this explanation, when we use the word "match", we mean that we achieved successful translation results using the translation patterns we registered in each procedure. The following are the numbers of translation patterns that

matched to sentences in the IWSLT05 test set. (a) is the details of the translation pattern registration (3). (b), (c), and (d) are the details of the translation patterns registration (2). Input sentences match the translation patterns in the priority order of (a) to (d).

Pattern 10, Pattern 3, Pattern 9 and Pattern2 in Section 2 is an example of a translation pattern in described in (a), (b), (c) and (d) respectively.

- (a) The number of patterns registered to our system: 17684

(The number of sentences extracted from the corpus as are given in the corpus: 17738 (17238 + 500)

The numbers of Japanese sentences of parallel texts with one Japanese sentence for each IWSLT05 training corpus: 17238

The numbers of Japanese sentences of parallel texts with one Japanese sentence for each IWSLT04 test corpus: 500

The numbers of Japanese original sentences in the IWSLT05 training corpus: 20000

The numbers of Japanese original sentences in the IWSLT04 test corpus: 500)

The number of input sentences matched the patterns in described in (a): 107 / 552

The number of patterns that input sentences matched the patterns in described in (a): 92

- (b) The number of registered patterns made from extracted expressions without non-terminal symbols: 439

The number of input sentences didn't match the patterns in described in (a), but matched the patterns in described in the patterns in described in (b): 6 / 445 (552-107)

The number of patterns that input sentences didn't match the patterns in described in (a), but matched the patterns in described in (b): 6

- (c) The number of registered patterns made from extracted expressions with non-terminal symbols: 1218

The number of input sentences didn't match the patterns in described in (a) and (b), but matched the patterns in described in (c): 41 / 439 (552-(107+6))

The number of patterns that input sentences didn't match the patterns in described in (a) and (b), but matched the patterns in described in (c): 14

- (d) The number of grammatical addition and modifications made to our system: 70

The number of input sentences didn't match the patterns in described in (a), (b), and (c), but matched the patterns in described in (d): 43 / 425 (552-(107+6+14))

The number of entries input sentences that didn't match the patterns in described in (a), (b), and (c), but matched the patterns in described in (d): 14

The following are analyses of (a) to (d) described in the previous paragraph.

(a) The number of times that the registered translation patterns matched the input sentences was 107 for 552 input sentences, which is pretty high. Of the 107 matches, the number of translation pattern types is 92, showing that the

some of the patterns were used repeatedly. These numbers show that many of the expressions given in the training sets and the test sets were same, which was probably because the corpora were of simple travel expressions. The number of the registered translation patterns used in translation of the input sentences was 92 out of 17684 patterns, which is not very high, however the amount of labor in patternizing process was small since the process were automatically registered from the parallel texts.

(b) The reason why the number of times that the registered translation patterns matched the input sentences was as low as 6 for 445 input sentences is probably that many of the input sentences matched the patterns made in (a). The number of the registered translation patterns used in translation of the input sentences was 6 out of 439 patterns, which is higher than the result of 92/17684 in (a).

(c) The number of times that the registered translation patterns matched the input sentences was 41 for 439 sentences, which is higher than the result of 6/445 in (b). The reason why the result here is lower than the result of 107/552 in (a) is that many of the input sentences matched the patterns made in (a). Of the 41 matches, the number of translation pattern types was 14, showing that only several types of patterns are used, and are used repeatedly. This shows that translation patterns match also certain parts of each input sentences since they were patternized using non-terminal symbols.

(d) The number of input sentences with improved translation results due to the grammatical additions and the modifications matched input sentences was 43 sentences out of 425 input sentences, which is higher than the results of 6/445 in (b) and 41/439 in (c). This number for the input sentences other than the ones matched (a), (b), or (c). Thus the grammatical additions and the modifications were highly effective even for the rather large amount of labor. The number of the grammatical additions and the modifications that actually improved translation results was 14 out of 70 items, which also is higher than the results in (a), (b), or (c). Of the 43 improved translations, number of the grammatical additions and the modifications that have effects on the improvement is 14, which is again higher than the results in (a), (b), or (c). This is probably because the training sets and the test sets have many expressions in common.

The following are analyses of the results of the translation using registered translation patterns. The registered translation patterns matched all the relevant input sentences. As to the translation patterns made in (a), (b), or (c) results were improved are prioritized since they were registered in our user dictionary. As to the translation patterns made in (d), results were improved since they are first prioritized in the order as described in 2.2., then with the priority control marks as described in 2.3. Also, the registered translation patterns did not make any unfavorable matches. As to the translation patterns in (b), (c), and (d), which matched the input sentences, the results turned out successful because those patterns were extracted and patternized manually carefully avoiding registration of patterns that could be used for unexpected input sentences. For example, in (b) described below, only the necessary parts of the sentences were extracted by omitting "Yes," and translations that can be used generally were given. In (c),

parts to be replaced with non-terminal symbols are determined appropriately, adding to the procedure described in (b). In (d), grammatical conditions are determined appropriately so the patterns match negative sentences but doesn't match adjective phrase.

The following are examples of registered translation patterns and translation results.

(a)
IWSLT05_JE_training:
Japanese : 素材(sozai)は(wa)何(nan)です(desu)か(ka)。
Translation result (1) : What is a material?
English : What is this made of ?

Registered translation pattern:
[ja:S 素材は何だ.*:pos=jd か。]
[en:S what be.*:pos=be:personNum=3sg:conjug=pres this:pos=det made:pos=adj of:pos=prep ?:pos=punc];

IWSLT05_JE_TESTSET:
Japanese : 素材(sozai)は(wa)何(nan)です(desu)か(ka)。
Translation result (3) : What is this made of ?

(b)
IWSLT05_JE_training:
Japanese : はい(hai)、何(nani)か(ka)ご(go)用(you)です(desu)か(ka)。
Translation result (1) : Yes, does anything be business?
English : Yes, can I be for any help?
English : Yes, may I help you?

Registered translation pattern:
[ja:SIntr 何か御用だ.*:pos=jd か]
[en:SIntr Can:pos=aux I:pos=prn help.*:pos=v:personNum=1sg:conjug=pres you:pos=prn];

IWSLT05_JE_TESTSET:
Japanese : 何(nani)か(ka)ご(go)用(you)です(desu)か(ka)。
Translation result (1) : Does anything be business?
Translation result (3) : Can I help you ?

(c)
IWSLT05_JE_training:
Japanese : ガーゼ(gauze)を(wo)ください(kudasai)。
Translation result (1) : Give gauze.
English : Please give me some gauze.
Japanese : 目覚まし時計(mezamashidokei)を(wo)ください(kudasai)。
Translation result (1) : Give the alarm clock.
English : I'd like an alarm clock, please.

Registered translation pattern:
[ja:SImp [1:NP]を下さる.*:pos=ds:inf=mr]
[en:SImp I:pos=prn would:pos=aux like.*:pos=v [1:NP]];
IWSLT05_JE_TESTSET:
Japanese : ライター(raिता)用(you)の(no)詰めかえ(tsumekae)を(wo)ください(kudasai)。
Translation result (1) : Give the repacking for the lighter.

Translation result (3) : I would like a repacking for a lighter.

(d)
IWSLT05_JE_training:
Japanese : ボール(booru)を(wo)よく(yoku)見(mi)て(te)。
Translation result (1) : You see a ball well and.
English : Watch your ball carefully.
Japanese : つかまえ(tsukamae)て(te)。
Translation result (1) : It catches it and.
English : Catch him.

Extracted expression:
-te form of verbs (conjugated form that leads declinable words) + particle "て(て)" or "で(で)" make imperatives.

Registered translation pattern:
![ja:SImp [1:VP.*:inf=ry:pos=ds] て:pos=sj]
[en:SImp [1:VP.*:conjug=bare]];

IWSLT05_JE_TESTSET:
Japanese : 警察(keisatsu)を(wo)呼ん(yon)で(de)。
Translation result (1) : It calls police and.
Translation result (3) : Call police.
Japanese : 芝生(shibahu)に(ni)入ら(haira)ない(nai)で(de)。
translation result (1) : It does not enter a lawn and.
translation result (3) : Do not enter the lawn.

5. Conclusions

In this paper, we presented our pattern-based translation method, and described how we dealt with the tasks relating to this evaluation campaign.

Our pattern-based translation method enables easier registration of phrasal expressions and grammatical knowledge. And it has been proved that registered patterns are almost always effective in translation in Section 4.

We will also consider adopting an automatic dictionary acquisition technology in our future study.

6. Acknowledgements

This research is supported by a grant from National Institute of Information and Communications Technology (NICT).

7. References

- [1] Takeda, K. "Pattern-Based Context-Free Grammars for Machine Translation", *In proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pp.144-151, 1996.
- [2] <http://www.yakushite.net/> (in Japanese)
- [3] <http://www.yakushite.info/>
- [4] Kitamura, M., and Murata, T. "Practical Machine Translation System allowing Complex Patterns", *In Proceedings of the MT Summit IX*, pp232-239, 2003.
- [5] Murata, T., Kitamura, M., Fukui, T., and Sukehiro, T. "Implementation of Collaborative Translation Environment 'Yakushite Net'", *In Proceedings of the MT Summit IX*, pp479-482, 2003.