A Method of Automatically Adapting an MT System to Different Domains

Setsuo YAMADA, Hiromi NAKAIWA, Kentaro OGURA, Satoru IKEHARA
NTT Communication Science Laboratories
1-2356 Take, Yokosuka-shi, Kanagawa-ken, JAPAN 238-03
{syamada,nakaiwa,ogura,ikehara}@nttkb.ntt.jp

Abstract

In order to achieve high translation quality for existing documents in a special domain using conventional MT systems, a domain adaptive translation method based on bilingual corpora has been proposed.

In this method, source sentences in a bilingual corpus are translated by the MT system and the results are compared with the target expressions in the corpus. The identifying parse trees of the machine translations with parse trees of the manual translations are investigated for three levels of mismatches: words, predicates, and sentences. The method proceeds as follows. First, it extracts poorly translated expressions by comparing parse trees and classifies the expressions into three levels of the differences. Second, it modifies the MT system at different levels corresponding to the three kinds of mismatches.

The experiments for the word level mismatches showed 84% of incorrectly translated words and their correct translations can be found automatically for subsequent registration in a user dictionary.

1 Introduction

Although various MT systems have been developed, no existing system can translate all linguistic phenomena. To achieve high quality translation for text using rule-based MT systems, two methods are considered. The first method rewrites sentences of the source language to enable easier translation [1]. This method has the advantage of being able to use existing translation functions for the translation of difficult-to-translate expressions. The second method adapts the MT system to the target domain. This improves the system performance for expressions which appear frequently in the target domain but the system can not easily translate, hereafter referred to as poorly translated expressions. This paper discusses the latter method.

For the second method, users must make a user dictionary for each target domain at present. In addition, other dictionaries and rules in the MT system must be modified in order to be able to translate the poorly translated expressions. When users try to adapt the MT system to the target domain, they must be familiar with the MT system itself. In order to adapt the MT system to the target domain more easily, various methods have proposed. One method is to use records of pre and post-editing [2]. This method is effective because translation quality for poorly translated expressions improves through incremental editing. But this kind of adaptation is both labor intensive and expensive. The other method is to improve translation quality automatically by using corpora. Several methods of this type have been proposed. Some methods add appropriate dictionaries or rules for MT target domains to a rule based MT system from a corpus [3, 4, 5, 6, 7]. These methods have the advantage of being able to acquire some useful knowledge for MT without manual load. Other methods get appropriate translations for the target domain from a corpus to combine a rule based MT system with an example based MT system [8, 9]. When these methods which use corpora are applied effectively and efficiently to a practical MT system, the following two conditions must be met:

1. To make the best use of existing dictionaries and rules.

2. To resolve the problems according to how poor translations can be solved by the MT system.

Existing methods which use corpora ignore existing dictionaries and rules. Therefore they do not satisfied condition 1 above. If a human selects the appropriate dictionaries or rules, the manual load remains excessive. Moreover, they resolve only some parts of the problems or all problems with one method. Each problem, however, requires a different solution. For example, a particular poor translation may be resolved if only dictionaries are modified, but another poor translation may be resolved if only rules are modified. Therefore existing methods do not satisfy condition 2 above.

Recently, a method that achieves condition 1 was proposed [10]. The method automatically extracts effective translation patterns from the differences between a machine translation and a correct manual translation. However, since this method deals only with an example based MT system, it is difficult to apply the method to a rule based MT system. Past methods do not consider comprehensive improvement so they are unsuitable for practical rule based MT systems.

We propose a new method where we compare the parse tree of the machine translation with the parse tree of the manual translation from a bilingual corpus. The system can automatically adapt to the target domain as follows. First, it extracts poorly translated expressions and classifies the expressions into three types, according to how they are mistranslated. The types are such expressions in which words, predicates (which includes verbs and predicative use of adjectives), or the whole sentence itself are mistranslated. Second, it modifies the MT system at different levels corresponding to the three kinds of mismatches. Moreover, this paper describes experimental results from applying a partial implementation of the proposed method to the MT system Japanese to English ALT-J/E [11] in the domain of technical manuals.

2 An Method for Adapting MT System to Target Domains

This section describes a method for automatically adapting a MT system to a target domain that both identifies poor translations and classifies them according to how they can be translated. The system has two functions as follows.

1. Identifying poor translations and finding the source of the translations:
   This function identifies poor translations by comparing parse trees of machine translations to parse trees of manual translations from bilingual corpora in the target domain. Then, it determines how each poor translation should be corrected according to the mismatching between the parse tree of the machine translation and the parse tree of the equivalent manual translation.

2. Adding to or modifying dictionaries or rules:
   This function takes the output of function 1 and perfect Matches, and resolves the poor translations by adding to or modifying domain adapted dictionaries or rules.

Function 1 effectively determines why correct translations can not be generated. The translation quality is improved by executing these functions repeatedly and adapting the dictionaries or rules to the MT target domain. Note that it creates domain adapted dictionaries and rules so as not to have a bad influence another domains.

Fig. 1 shows how the proposed method can be combined with the Japanese to English machine translation system **ALT-J/E** [11]. **ALT-J/E** has a word dictionary, a pattern transfer dictionary, and transfer rules. The word dictionary is used when translating words in a sentence. The pattern transfer dictionary is used when translating the structure of a simple sentence around a predicate. The transfer rules are used when translating more complex language

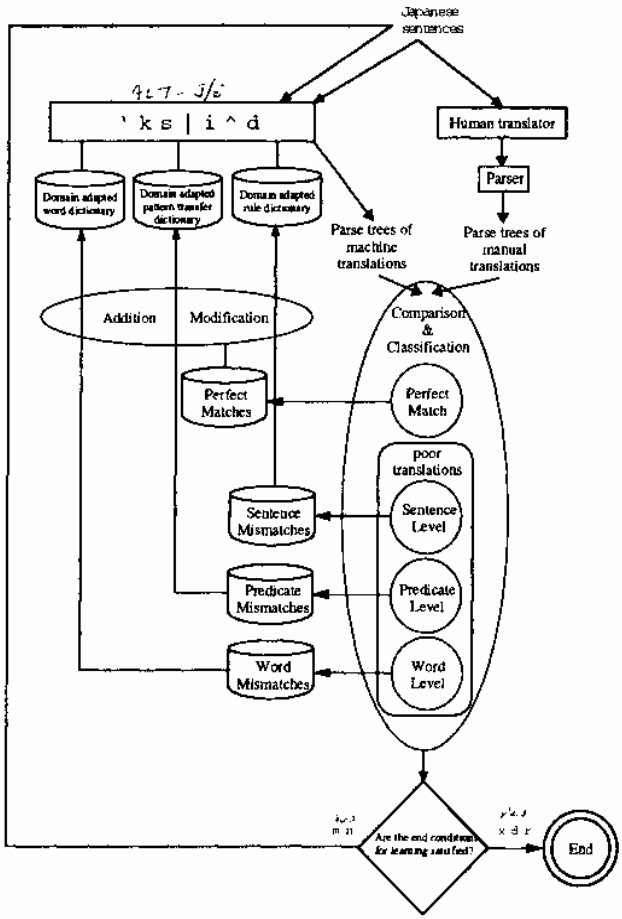

Figure 1: A Method for domain adaptation

structures such as complex, compound, and embedded sentences. Executing each of the above functions once for all sentences in the corpus, it creates a word dictionary, a pattern transfer dictionary, and rules. It is assumed that the same sentence in the source sentences has only one target sentence. Then, it translates the same Japanese sentences again. These processes are repeated until translation quality meets or exceeds the required quality.

3 Comparing Parse Trees

This section describes how to compare the parse tree of a machine translation with the parse tree of the equivalent manual translation. Several methods have defined a quantitative distance between the trees [12, 13]. The basic idea is that large distances imply that more parts must be modified or replaced to make the two trees identical. This definition fails, however, to treat differences between parse trees, i.e. changes to the node such as a part of speech. We expand 1 he definition to consider at what level the trees differ. If changing one part means that many other parts must be change as well, the distance is larger than indicated by existing methods. For example, if changing one verb, prepositions for the verb may be changed at the same time. While, if changing one noun, no part need to be changed. Therefore the former distance is larger than the latter distance. In this paper, we compare between the parse trees based on this extended definition.

We classified differences between the manual translations with the machine translations into four types. Those are perfect match, word mismatch, predicate mismatch, and sentence mismatch. We show examples of three mismatch case as follows:

(A) Word level mismatch:

Jap:	*watashi-wa*	*tokidoki*	*sofuto-o*	*kau*
Gloss:	I-TOP(SUBJ)	sometimes	software-OBJ	buy
Manual:	I sometimes buy **software**			
Machine:	I sometimes buy a **soft hat**			

These translations differ at the word level. In this case, the system can produce the manually derived sentence by modifying the word dictionary.

(B) Predicate level mismatch:

Jap:	*watashi-wa*	*fune-ni*	*noru*
Gloss:	I-TOP(SUBJ)	ship-LOC	board
Manual:	I **board** the ship		
Machine:	I **take** the ship		

These translations differ at the predicate level. In this case, the system can produce the manually derived sentence by modifying the pattern transfer dictionary.

(C) Sentence level mismatch:

Jap:	*kono-syutsuryoku-wa*	*rokuon-kanou-dearu*
Gloss:	This-output-TOP(SUBJ)	record-possible-is
Manual:	This output is **recordable**	
Machine:	This output power is **to be able to record**	

These translations differ significantly. In this case, the system can only generate the manual translation if both the rules and dictionaries are modified.

4 Identifying and Classifying Mismatches

We described three types of mismatches in the previous section. In this section, we describe how to identify mismatches from pairs of manual and machine parse trees and classify them into

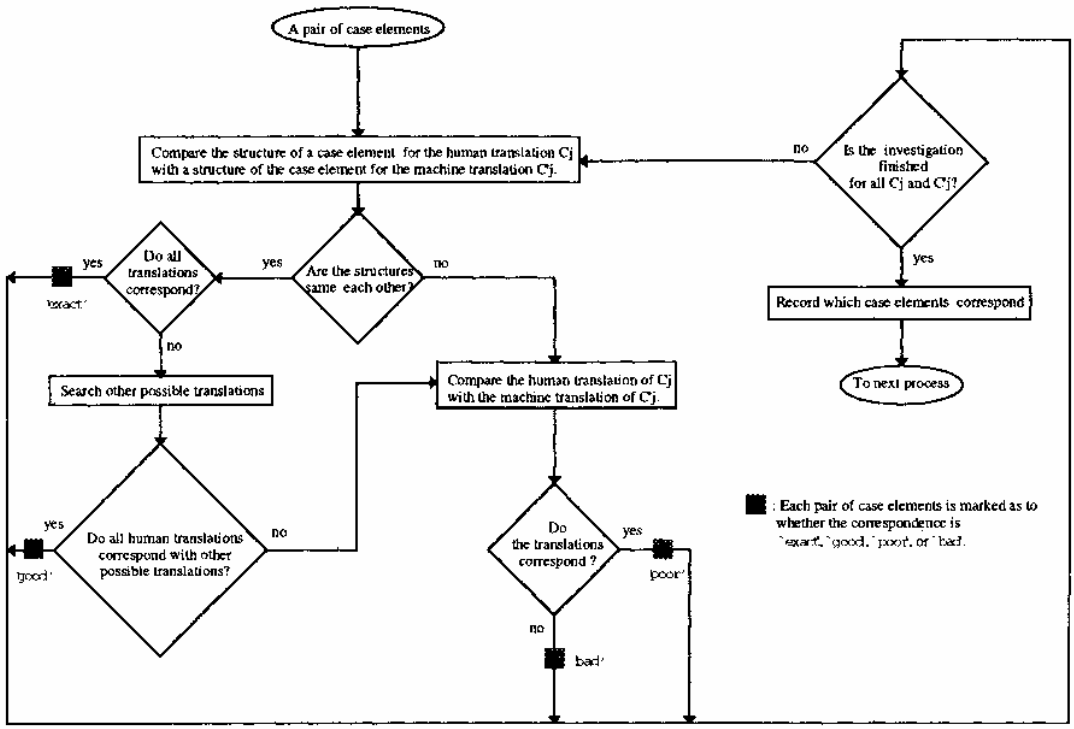

Figure 2: Process for case element

three types of mismatches. This classification can them be used to find appropriate solution in the MT system.

4.1 Method of Aligning Case Elements

As a first step, this paper considers only simple sentence, i.e. containing only one clause. We introduce a method that investigates the degree of correspondence for case elements in Fig. 2. It outputs the degree of correspondence for each case element by investigating each human generated case element and each machine generated case element. Note that a case element might have the plural number of words. It is necessary to consider the following two points:

- Whether the structures of the machine generated case element correspond with the structures of the human generated case element.
- Whether the manual translation in a case element is the same as one of the other possible translations.

Here, we use 'exact' if the human parse tree is exactly the same as the machine parse tree, 'good' if the human parse tree is the same as the structure of the machine parse tree and has some other possible translations, 'poor' if the manual translation in a case element only corresponds to the machine translation, and 'bad' for all other cases. The degree of correspondence degrades in the order 'exact' 'good', 'poor', 'bad'.

We describes how to align case elements using example (A) (word level mismatch) in the previous section. In this example, human generated case elements are 'I', 'sometimes', and 'software'. Machine generated case elements are 'I', 'sometimes', and 'soft hat'. It is assumed that 'soft hat' is included in MT system's word dictionary as the other possible translation. The process shown in Fig. 2 investigates each case element completely and extracts the result shown in Table 1. It estimates pair of 'software' 'soft hat' to be 'poor'. Because their structures

are not same but their translations correspond. If ‘soft hat’ be not included in MT system’s dictionary, then it would estimate pair of them to be ‘bad’.

We can find pairs of case elements which correspond to each other from the results. In this case, pairs of case elements are ‘I’ ‘I’, ‘sometimes’ ‘sometimes’, and ‘software’ ‘soft hat’.

Table 1: Result of corresponding for case element 1

	‘I’	‘sometimes’	‘software’
‘I’	exact	bad	bad
‘sometimes’	bad	exact	bad
‘soft hat’	bad	bad	poor

4.2 Method of Classifying Mismatches

We propose a method of classifying word, predicate, and sentence mismatches from poorly translated expressions automatically by comparing manual and machine parse trees. We introduce degree of correspondence as a new measure to classify mismatches. This degree of correspondence is decided according to the following criteria:

[match] Structure and translation are exact matches. This is the case of 'exact' in the previous subsection.

[equivalent] The manual translation can be generated from the source language by the machine translation system, (i.e. other English entries for the same Japanese word which the MT system does not select (other possible translations) are the same as the manual translation.) This is the case of ‘good’ or ‘poor’ in the previous subsection.

[mismatch] The manual translation can not be generated from the source language by the machine translation system. This is the case of ‘bad’ in the previous subsection.

If the degree of correspondence is [match] or [equivalent], we say the structure of the machine translation corresponds to the structure of the manual translation. Table 2 classifies matches using the degree of correspondence.

The degree of correspondence for the predicate part is evaluated by investigating both it's structure and translation. The degree of correspondence for case elements are evaluated with methods as well as the case of predicate part. Note that it is necessary to investigate all case elements, because simple sentence may include several case elements.

Table 2: Classification of matches

Part of predicate	Part of case elements	Mismatch level		
match	match	PERFECT MATCH		
match	equivalent			word
match	mismatch	sentence		(word)
equivalent	match		predicate	
equivalent	equivalent		predicate	word
equivalent	mismatch	sentence		(word)
mismatch	match		predicate	
mismatch	equivalent		predicate	word
mismatch	mismatch	sentence		(word)

Mismatches and matches are classified by the degree of correspondence above as follows:

- Perfect matches are classified if both predicate parts and case elements correspond to criterion [match] above. That is, when each part of the structure in the machine parse tree is same with each part of the human parse tree respectively.
- Word mismatches are classified if case elements correspond to the above criterion [equivalent]. That is, when machine translation of case elements are not the same with the manual translations but correspond to each other.
- Predicate mismatches are classified if the machine translation of predicate part is not, same with the manual translation, but they correspond to each other.
- Any other sentence mismatches are classified in the case except above situation.

Note that if a poorly translated expression is classified into sentence mismatch, it may also include word mismatches. For example, see example (C) (sentence level mismatch) in the previous section. This sentence includes the word mismatch pair of 'output' and 'output power'. In this case, we show '(word)' in Table 2.

Each mismatch is classified and extracted by following way. First, it investigates the degree of correspondence for case elements, then classifies word matches as pairs of case elements. Second, it investigates the degree of correspondence for the predicate parts, then classifies predicate matches as pairs of simple sentence structures. Finally, it classifies sentence mismatches.

5 Word Mismatches Extraction Prototype

We have made a prototype which extracts word matches based on the method shown by Fig. 2 using **ALT-J/E**. We evaluated this prototype for 577 simple sentences in a set of technical manual sentences. This technical manual has many restricted special expressions in the domain. Here, its recall is the number of pairs of case elements extracted by this prototype divided by the number of pairs of case elements extracted by a human. As a result, the recall was 45.2% and the precision was 100%. This level of recall can be raised as follows:

1. Search general electronic dictionaries not held in **ALT-J/E**.
2. Align the machine case element with the human case element if there exists both only one machine mistranslated word and only one human case element.
3. Align the machine case element with the human case element by using English dictionaries if only the suffix differs: for example derivative, 'ing' form between machine translation word and manual translation word.

A recall of 83.8% can be achieved by using the above. The precision can achieve 100%), because extraction conditions are strong.

6 Conclusion

In this paper, we proposed a method for automatically adapting rule-based machine translation systems to new domains. The method uses examples (a bilingual corpora that consists of source language and translations in the target language). This method classifies translation pairs (produced by the MT system and manually) according to the degree of correspondence for case elements and predicate parts. It assigns each pair to one of three levels of mismatches. According to the level of mismatches, word dictionaries and pattern transfer dictionaries, or rules are created. This ensures that the rule based translation system is adapted to the target domain efficiently. Preliminary testing extracted word mismatches only. The prototype achieved a recall rate of 45.2%) and precision rate of 100%. Moreover, techniques were introduced that would raise the recall to 83.8%.

In the future, we intend to consider other methods of raising the recall further and examine all levels of mismatches in detail.

References

[1] Satoshi Shirai, Satoru Ikehara, and Tsukasa Kawaoka. Effects of automatic rewriting of source language within a Japanese to English MT system. In *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-93),* 1993.

[2] M. Miura, M. Hirata, and N. Hoshino. Learning mechanism in machine translation system "PIVOT". In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92),* pp. 693-699, August 1992.

[3] H. Nomiyama. Lexical selection mechanism using target language. *Technical Reports of SIG on Natural Language Processing,* Vol. NL86-8, 1991. (in Japanese).

[4] S. Doi and K. Muraki. Translation ambiguity resolution based on text corpora of source and target languages. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92),* pp. 525-531, August 1992.

[5] H. Kaji, Y. Kida, and Y. Morimoto. Learning translation templates from bilingual text. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92),* pp. 672-678, August 1992.

[6] T. Utsuro, Y. Matsumoto, and M. Nagao. Lexical knowledge acquisition from bilingual corpora. In *Proceedings of the 14th International Conference: on Computational Linguistics (COLING-92),* pp. 581-587, Nantes, France, 1992.

[7] Jing-Shin Chang and Keh-Yih Su. A corpus-based statistics-oriented transfer and generation model for machine translation. In *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-93).* 1993.

[8] O. Furuse, E. Sumita, and H. Iida. A method for realizing transfer-driven machine translation. *Technical Reports of SIG on Natural Language Processing,* Vol. NL80-8, 1990. (in Japanese).

[9] Hideo Watanabe. A similarity-driven transfer system. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92),* pp. 770-776, August 1992.

[10] Hideo Watanabe. A system for finding translation patterns by comparing an MT result and its correction. *Journal of Natural Language Processing,* Vol. 1, No. 1, pp. 59-75. October 1994.

[11] Satoru Ikehara, Satoshi Shirai, Akio Yokoo, and Hiromi Nakaiwa. Toward an MT system without pre-editing - effects of new methods in ALT-J/E. In *Proceedings of MT Summit III,* pp. 101-106, 1991.

[12] R. Wilhelm. A modified tree-to-tree correction problem. *Information Processing Letters,* Vol. 12, pp. 127 132, 1981.

[13] E. Tanaka. Structural distances and similarities. *Journal of Information Processing Society of Japan,* Vol. 31, No. 9, pp. 1270-1279, 1990. (in Japanese).

[14] Setsuo Yamada, Hiromi Nakaiwa, Kentaro Ogura, and Satoru Ikehara. Automatically adapting a MT system to a domain using examples. *Technical Reports of SIG on Natural Language Processing,* Vol. NL104-13, 1994. (in Japanese).