NAIST-IS-DD9961202

Doctoral Dissertation

Translation Knowlegde Acquisition for Pattern-based Machine Translation

Mihoko Kitamura

November 1, 2004

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Mihoko Kitamura

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Shunsuke Uemura	(Member)
Professor Kiyohiro Shikano	(Member)

Translation Knowlegde Acquisition for Pattern-based Machine Translation[*]

Mihoko Kitamura

Abstract

The quality of machine translation is strongly dependent on the quantity and quality of the translation knowledge available to the system. Constructing translation knowledge by hand has inherent limitations, which begs for techniques to construct translation knowledge automatically or semi-automatically, and to integrate this translation knowledge easily.

This thesis deals with pattern-based machine translation and translation knowledge acquisition from parallel corpora, in order to fulfill the above demand.

The first work advocates the use of complex patterns in machine translation. In previous pattern-based machine translation, writing new patterns was difficult due to the lack of flexibility . We have built a pattern-based machine translation system with an emphasis on pattern readability. Patterns can be constructed by hand or automatically from parallel corpora.

The second work proposes a translation pattern extraction method that greedily extracts translation patterns based on co-occurrence of original and target word sequences in parallel corpora. This method can acquire translation patterns combining good coverage and accuracy, without any preliminary translation dictionary.

The third work extends the second work by combining it with extra linguistic resources, such as chunking information and translation dictionaries. Additionally we allow manual confirmation of extracted translation patterns. Experimental results show both higher accuracy and coverage. The above proposal is a

[*] Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD9961202, November 1, 2004.

statistical method, and it excels at extracting technical terms and proper nouns, but cannot extract complex patterns, such as discontinuous or idiomatic patterns and translation rules for word selection.

The last work proposes a method for automatic acquisition of more advanced translation rules from parallel corpora. The acquisition process uses both the similarity measure of word pairs obtained statistically from the parallel corpora, and the structural matching of the dependency trees obtained from parsed parallel sentences.

Fusion of these four techniques shall provide a practical machine translation system with the ability to extract translation knowledge.

Keywords:

machine translation, translation knowledge acquisition, pattern-based machine translation, parallel corpus, phrase alignment, structural alignment

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

As information distribution become global, the need for translation is going on increasing. In order to meet this need, machine translation systems have been attracting much attention.

It is almost fifty years since the first machine translation system started to be developed. After many technical changes, we see many systems for translation on the market today. However, these translation systems need pre-editing, post-editing and limiting the domain of a target text to get satisfying translation results. In addition, professionals familiar with the domain must provide translation knowledge extensively. Even then, there are limitations inherent to building up translation knowledge only by hand. For the above reason it is a high-priority issue to device a method for acquiring translation knowledge efficiently with high accuracy and utilize the translation knowledge in an effective manner.

"*Corpus-based machine translation*" is an approarch to solve the above problem.

There are other factors for the spread of the corpus-based approach. The research on statistical-based natural language processing [9] has been making steady progresses. An enormous amount of machine-readable documents has been accumulated [50], and many software tools for natural language processing, such as a statistical structural analyzer, are available for use.

Under these circumstances, we take advantage of the "corpus-based approach" and aim at acquiring translation knowledge automatically or semi-automatically from parallel corpora, and constructing a useful machine translation system which uses this translation knowledge.

To put it concretely, the following four research subjects are presented.

- A pattern-based machine translation system which can use translation knowledge from parallel corpora and handle features flexibly.

- Automatic translation pattern extraction using statistical methods without preliminary translation dictionary.

- Practical translation pattern extraction based on the above statistical method, which can be combined with manual confirmation and linguistic resources.

- Automatic acquisition of complex translation rules by structural matching of parallel sentences, based on a similarity measure statistically extracted from the same corpus.

1.2 Background

This section describes the background of corpus-based machine translation and translation knowledge acquisition from parallel corpora. These two technologies have a close connection, because translation knowledge acquisition can produce the translation dictionaries required for corpus-based machine translation.

1.2.1 Corpus-based Machine Translation

Machine translation, one of the important applications of natural language processing, has been studied for practical use. However, the longstanding major issues are the way to acquire the knowledge needed for translation, such as grammar rules and lexical translation dictionaries.

Corpus-based machine translation uses large corpora made of real sentences. It is hoped that corpus-based machine translation which translates using such corpora could solve the above issues.

The structure of corpus-based machine translation is depicted in Figure 1.1.

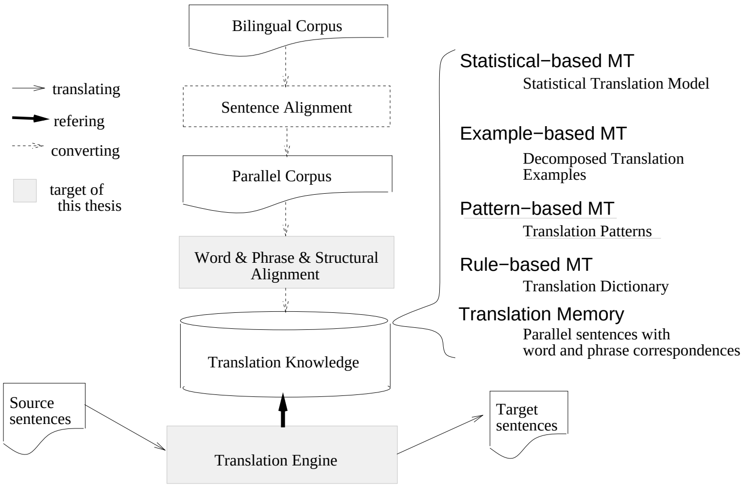


Figure 1.1 Corpus-based machine translation

Statistical-based machine translation and example-based machine translation are typical corpus-based approaches. The former uses statistical information which is extracted from a corpus and the latter uses translation examples directly without relying on explicit translation rules.

Other example of corpus-based machine translation are techniques for retrieving translation examples, aka Translation Memories (TM), or using translation knowledge extracted from a corpus in a traditional rule-based machine translation system.

Next we describe each technology and its background.

Statistical-based Machine Translation

Statistical-based machine translation has been proposed by Brown et al. [1, 6]. It attempts to estimate *"translation probabilities (translation model)"* between original and target sentences. The translation models are based on correspondences between original and target words, and word order of respectively original and target sentences. They applied this method to translation between English and French, and the percentage of the correct translation was about 40% [6].

One criticism of the translation model is that that does not model structural or syntactic aspects of the language. The translation model was only demonstrated for a grammatically similar language pair. It has been suspected that a language pair with very different word order such as English and Japanese would not be modeled well by these translation models.

One of the few researches which have applied the statistical approach to English-Japanese machine translation is by Yamada and Knight [56]. They used a statistical translation model for structurally analyzed sentences (that is to say a "syntax-based translation model"), and could thus overcome the problem of differences in word order.

Statistical-based approaches seem promising, but the gap between theory and practice is still large.

Example-based Machine Translation

The idea of example-based machine translation [44, 51] is to carry out translation by referring to translation examples that have the best similarity to the given sentence. The key technique is to define the similarity between examples and to identify the examples with the best similarity. There are, however, two bottlenecks in this approach. One is the "knowledge access bottleneck" for selecting the best similar examples from the database. The other is the "knowledge acquisition bottleneck," as it is not so easy to collect examples, e.g. parallel sentences must be syntactically aligned and analyzed in advance.

Some solutions to the "knowledge access bottleneck" have been proposed. For instance, Transfer-Driven Machine Translation (TDMT) introduced an example-based approach [13] to achieve efficient and robust translation. Pattern-based Machine Translation [52] can use directly translation knowledge extracted from a parallel corpus as translation dictionaries.

Figure 1.2 clarifies the difference between pattern-based machine translation and a traditional rule-based machine translation. As shown in Figure 1.2, the rule dictionaries for rule-based machine translation are of 3 types; (1) rules for parsing the original sentence, which are employed to convert word sequences of the original sentence into an interlingual structure, (2) translation rules, which are employed to convert the original interlingual structure to a target interlingual structure, and (3) rules for generating the target sentence, which are employed to

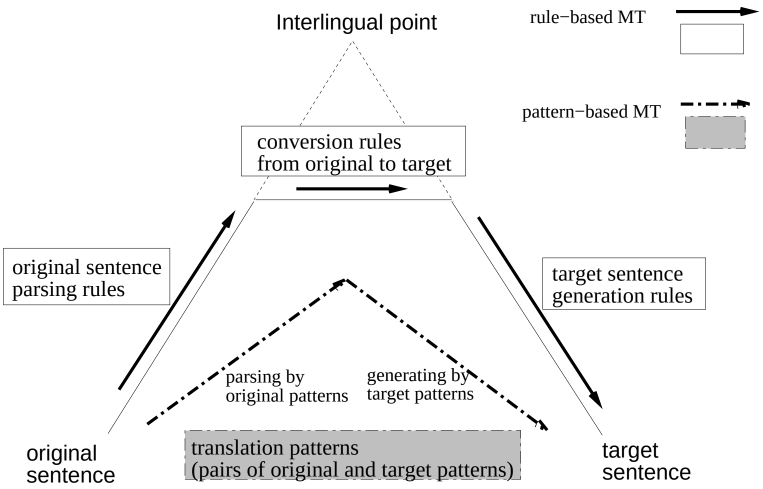


Figure 1.2 The differences of pattern-based and rule-based machine translation

convert the target interlingual structure to word sequences composing the target sentence. The translation result is obtained by applying these rules sequentially. In this case, later rules (ex. translation rules and rules for generating the target sentence) depend strongly on the output(specification) of former rules, so that it is difficult to modify them independently, as it might cause unwelcome side-effects. Moreover, when users want to find the cause of the failure, they need to go over intermediate results at each stage.

Pattern-based machine translation has only one kind of rules, i.e. translation patterns which are pairs of original and target patterns. Parsing is done by combining original patterns, and after parsing is finished, generation uses the target side of patterns. As there are no intermediate steps, side-effects caused by modifying rules are reduced. Moreover, when users probe the cause of the failure, they can find it by investigating the set of rules that were applied.

We describe such a system in Chapter 2. There are also researches for algorithms calculating efficiently the similarity between an input sentence and an example [41].

The problem of “knowledge acquisition bottleneck” can be solved by translation knowledge acquisition techniques described in Section 1.2.2. We describe also such techniques in Chapters 3, 4 and 5.

Other Corpus-based Machine Translation

There have been some attempts at using translation examples to improve the quality of translation systems. A practical approach would be to combine a machine translation system with a translation memory [5]. Sentences already stored in the Translation Memory (TM for short) can be translated directly by using the TM, while sentences not yet stored in the TM have to go through machine translation, and can be added to the TM after post- editing. This combination improves the translation system as it enables correct translation results to be reused. However, as accumulated translation examples are used only literally, they do not affect the quality of machine translation.

In recent years, there are trials for improving the translation quality of commercial rule-based machine translation systems by using translation knowledge extracted from parallel corpora [47]. The process works as follows; a parallel corpus which has the same domain as the target text is first prepared, and translation dictionaries are constructed from this parallel corpus. The machine translation system translates technical terms and domain-dependent expressions using the constructed dictionary, and translates other phrases using the system dictionary which has been developed manually and refined over time. That is, the corpus is only used when it is relevant. The above method has immediate effects for commercial rule-based machine translation systems, however constructing translation rules for a rule-based machine translation system is no easy feat, as the formulation of rules was defined with different goals in mind. For this reason it cannot overcome limitations inherent to rule-based systems.

1.2.2 Translation Knowledge Acquisition

There is a recent trend in natural language processing for linguistic knowledge extraction from corpora using statistical methods.

One of the causes for this trend is the recent availability of large-scale machine-readable linguistic resources [29], such as the world-wide web itself. The demand for text processing technologies to gather information from these resources, such as information retrieval, is growing steadily. A large variety of information is available through Internet, however finding the specific information one needs may be a difficult and time-consuming task. Natural language processing offers solutions to this problem. The rapid advances in computational power and establishment of statistical natural language processing make it possible to meet this demand.

In recent years the research on *parallel* corpora has advanced too. Parallel corpora are texts available in both the original and target languages, with information on how both versions are related. By comparing between both languages, we can acquire not only information for translation but also syntactic and semantic mono-lingual information.

As we can see in Figure 1.1, alignment technology is also an important technology for research in machine translation. For instance, sentence alignment is needed for Translation Memory, phrase alignment and structural alignment are needed for example-based machine translation.

Next, we review representative researches on sentence alignment, phrase alignment and structural alignment. Additionally we outline the similarity measures which are an important factor for the alignment, and a matching algorithm for structural alignment.

Sentence Alignment

DP-matching is an efficient technique for sentence alignment. First words of the original and target texts are matched using a translation dictionary, next original and target sentences are aligned by DP-matching based on the similarity between original and target words [54].

Variants of this method use statistical information from a parallel corpus in place of a translation dictionary [24], or use the number of words [14] or letters [7].

Kay & Röscheisen [24] presented an algorithm for aligning texts using only internal evidence. This process rests on the notion of which word in the original text corresponds to which word in the target text, using essentially the similarity of their distributions. It exploits a partial word alignment to induce a maximum likelihood for sentence alignment, which is in turn used, in the next iteration, to refine the word level estimate. The algorithm appears to converge to the correct sentence alignments in only a few iterations. The experiment using 214 sentences of an English article published in "*Scientific American*" and 162 sentences of the German version showed that the percentage of correct sentence alignments is 96% and the percentage of correct word alignments obtained as side products is close to 100%. This result shows that statistical methods can extract some kinds of translation knowledge with high accuracy.

Phrase Alignment

Though the above research only obtains word correspondences as side products, many methods directly aim at constructing translation dictionaries.

Kupiec [28] and Kumano [27] presented methods for alignment between word sequences such as compound nouns. Kupiec uses a NP recognizer for both English and French, and proposed a method to calculate the probabilities of correspondences using an iterative algorithm, the EM (Expectation Maximization) algorithm. He reported that among the 100 highest ranking correspondences the percentage of correct correspondences is 90%. The NP recognizer detected about 5000 distinct noun phrases in both languages, but the percentage of correct correspondences of the total data is not reported.

Kumano's objective is to obtain English translations of Japanese compound nouns (noun sequences) and unknown words using a statistical method similar to Brown's together with an ordinary Japanese-English dictionary. Japanese compound nouns and unknown words are detected by the morphological analysis stage and are determined before the later alignment processes. In an experiment with 2,000 sentence pairs, the percentage of correct compound noun pairs are 72.9% considering only the best correspondence for each compound noun of the source language and 83.8% considering the top three candidates. The percentage of correct unknown word pairs is 54.0% and 65.0% respectively.

Smadja proposes a method for finding translation patterns of fixed as well as flexible collocations[1] between English and French [49]. The method first extracts meaningful collocations in the source language in advance with the XTRACT

[1] When word order is variable or some optional elements can be inserted, it is called a *flexible* collocation.

system [48]. Then, aligned corpora are statistically analyzed for finding the corresponding collocation patterns in the target language. To avoid possible combinational explosion, some heuristics is introduced to filter implausible correspondences.

All these methods use some types of similarity measure in order to correlate translation units from different languages. Smadja and Kay use "Dice coefficient" [43], Brown and Kumano use "Mutual Information." We describe the similarity measures in Section 1.2.3.

Structural Alignment

Structural Alignment [15, 22, 30, 55] requires deeper analysis than phrase alignment and sentence alignment. The results of structural alignment as translation knowledge can be used directly for example-based machine translation. The advantage of structural alignment is that it provides correspondences between dependency relations.

Grishman [15] and Mayers [35] presented bottom-up methods for finding structural matchings. Grishman [15] used a beam search algorithm in a bottom-up way, and Mayers used a dynamic programming algorithm. Both algorithm do not resolve syntactic ambiguities, and the reported experiments only used simple English-Spanish parallel corpora.

Matsumoto [30] presented an algorithm for finding structural matchings between parallel sentences of two languages, such as Japanese and English. Parallel sentences are analyzed based on unification grammars, and structural matching is performed by making use of translatable word pairs of the two languages. Syntactic ambiguities are resolved simultaneously by the matching process. We outline this structural matching algorithm below, as our approach in Chapter 5 applies this algorithm.

1.2.3 Similarity Measures

Similarity measures are defined in order to correlate translation units from different languages. Several similarity measures have been proposed, and they are described in detail in [34].

Brown et al. [6] used Mutual Information to construct corresponding pairs of French and English words. Mutual Information MI(x, y) means the ratio of the probability that words x and y individually occur, $prob(x)prob(y)$, to the probability that words x and y co-occur, $prob(x, y)$. For a practical calculation, probabilities $prob(x, y)$, $prob(x)$ and $prob(y)$ are calculated from a parallel corpus by counting the co-occurrences f_{xy} and occurrences f_x, f_y of words x and y, and dividing them by the total number of sentences in the parallel corpus f_{all}. The logarithm of this ratio is used generally. In brief the Mutual Information is as follows:

$$MI(x, y) = \log_2 \frac{prob(x, y)}{prob(x)prob(y)} = \log_2 \frac{f_{xy} \cdot f_{all}}{f_x \cdot f_y}$$

Kay & Röscheisen [24] used the following Dice coefficient for calculating the similarity between English word x and French word y. In the formula, f_x, f_y represent the numbers of occurrences of x and y, and f_{xy} is the number of simultaneous occurrences of those words in corresponding sentences. The similarity of a pair of words x and y is defined by the numbers of their total occurrences and co-occurrences in the corpus.

$$Dice(x, y) = \frac{2f_{xy}}{f_x + f_y}$$

As an alternative, Melamed[36] considered the Log-Likelihood ratio [12], which is another measure for co-occurrence. It is defined as[2]

$$\begin{aligned} LogLike(x, y) &= \phi(f_{xy}) + \phi(f_x - f_{xy}) + \phi(f_y - f_{xy}) + \phi(f_{all} + f_{xy} - f_x - f_y) \\ &\quad - \phi(f_x) - \phi(f_y) - \phi(f_{all} - f_x) - \phi(f_{all} - f_y) + \phi(f_{all}) \\ \phi(f) &= f \cdot \log f \end{aligned}$$

The Log-Likelihood ratio measures the likelihood ratio of the hypothesis that the occurrence of word x does not depend on the occurrence of word y to the hypothesis that the occurrence of word x depends on the occurrence of word y,

[2] Dunning[12] actually uses probabilities rather than frequencies, i.e. $\phi(f) = \frac{f}{f_{all}} \cdot log \frac{f}{f_{all}}$, but as summands eliminate each other this formula just multiplies $LogLike(x, y)$ by a factor f_{all}.

in order to investigate how strongly the occurrence of word x depends on the occurrence of word y.

We experiment with the above three similarity, and investigate them in Sections 3.2 and 4.5.3.

1.2.4 Structural Matching

The following is the process for structural matching in Matsumoto[30].

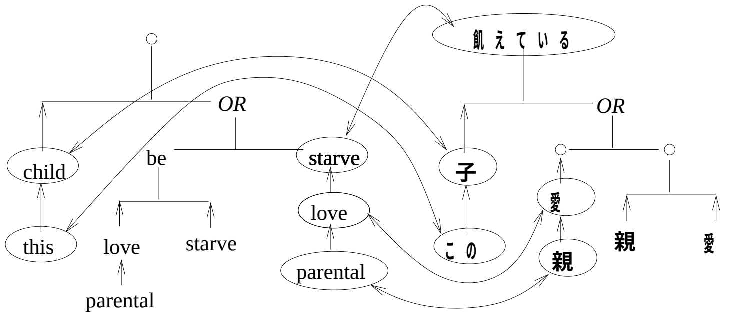
1. A pair of Japanese and English sentences are parsed independently into disjunctive feature structures.
2. Dependency structures are derived from their feature structures by removing unrelated features.
3. Structural matching is done based on the similarity between subgraphs that is defined from the similarity between word pairs.
4. The result of structural matching which has the maximum plausible score is selected.

When the matching of two dependency trees is performed top-down at Step 3., three types of nondeterminism arise:

- Selection of a top-most subgraph in both trees.
- Selection of which edge to follow in order to find a dependent subgraph in both of the dependency trees.
- Selection of a disjunct at an 'OR' node.

These nondeterminisms are resolved by a branch-and-bound algorithm. It looks for the answers with the best score. In each new step, it estimates the value of the maximum expected score along the current path, and compares it with the currently known best score. If the maximum expectation is less than the currently known best score, there is no chance to get better answers by pursuing the path. Then it backtracks to find other paths. The similarity between subgraphs is defined accordingly to the sets of content words appearing in them as follows.

(1) English: This child is starving for parental love.
Japanese: この 子-は 親-の 愛 -に 飢えている (*konoko-ha oya-no ai-ni ueteiru*)



(2) English: She has long hair.
Japanese: 彼女-の 髪-は 長い (*kanojo-no kami-ha nagai*)

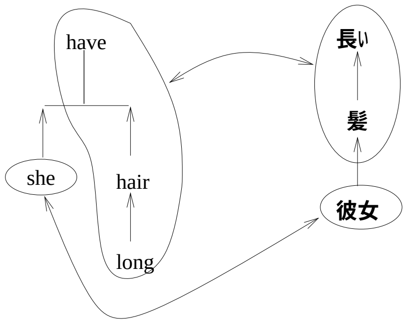


Figure 1.3 Samples of structural matching

Let s and t be subgraphs of the dependency graphs of Japanese and English sentences, and V_s and V_t be the sets of content words in s and t. Assume without loss of generality that the size of V_s is not larger than that of V_t. (We interchange V_s and V_t when $|V_s| > |V_t|$.) Taking an arbitrary injection p from V_s and V_t (p:$V_s \rightarrow V_t$), a set D of pairs is defined by each of such p's.

$$D = \{< w, p(w) > \mid w \in V_s\}$$

The similarity $f(s,t)$ between subgraphs is defined as

$$f(s,t) = \max_{p} \left\{ \sum_{d \in D} sim(d) \right\} \times 0.95^{|V_s|+|V_t|-2}$$

$sim(< w_1, w_2 >)$:the similarity between word pairs w_1, w_2

The value of similarity between word pairs is a discrete number from 1 to 6 which is defined by a translation dictionary or a thesaurus. 0.95 in this formula is the penalty for a larger subgraph.

Figure 1.3 shows examples of structural matching. For instance, the upper graph in Figure 1.3 shows that a possible translation of the word "飢えている (*ueteiru*) " is "starve" in this case. On the other hand, when the top subgraph contains more than one content word, it is regarded as an idiomatic expression. The lower example shows that the phrase "髪が (*kami-ga*) 長い (*nagai*)" can be translated into a English phrase "have long hair."

In Chapter 5, we will present an improved structural matching algorithm for extracting translation rules of a machine translation system.

1.3 Positioning and Objective

This section describes the positioning and the objective of this thesis, with respect to the above background.

1.3.1 Requirements for Practical machine translation

Though today some types of corpus-based machine translation are proposed and some corpus-based techniques are applied partially to conventional machine translation systems, complete corpus-based machine translation is not ready for the market.

The foremost reason is the translation quality. Additionally users are not satisfied that one cannot flexibly customize the translation dictionaries as desired. The translation knowledge used in statistical-based and example-based machine translation is not readable, and cannot be adjusted easily by users.

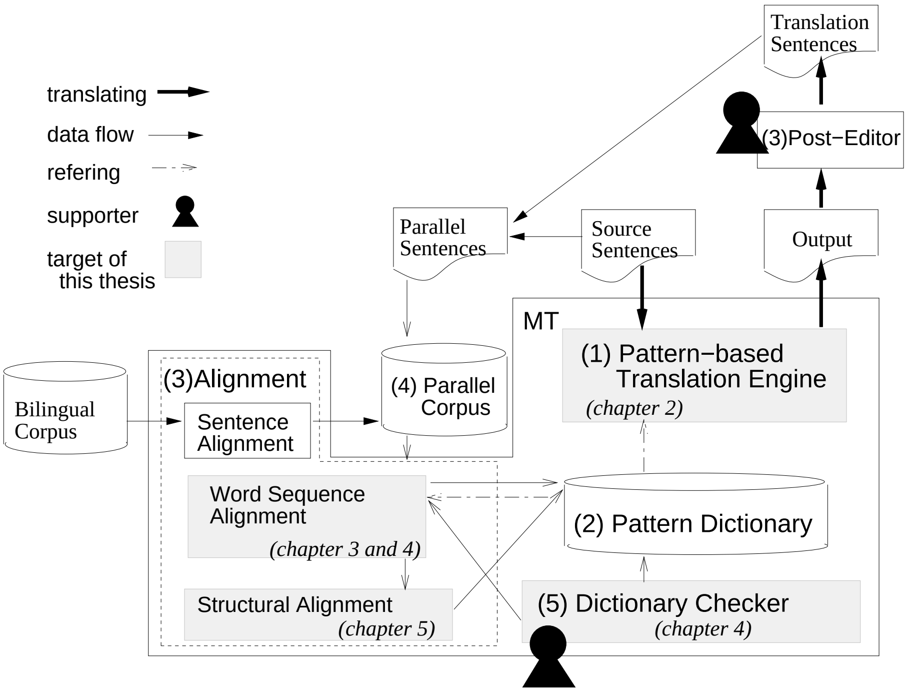


Figure 1.4. Pattern-based machine translation with translation knowledge acquisition

We present a pattern-based machine translation system with the ability to acquire translation knowledge; the system can acquire translation patterns from parallel corpora, and the translation patterns can be flexibly adjusted by users as they are brief and understandable. We outline it in the next section.

1.3.2 Pattern-based Machine Translation with Translation Knowledge Acquisition

Figure 1.4 is a pattern-based machine translation system able to use translation knowledge, realizing our goal. The features of this system are as follows. (The numbering corresponds to the figure.)

(1) We present a novel pattern-based machine translation system; its translation knowledge is understandable and can be repaired easily by users.

(2) This pattern-based machine translation system utilizes translation patterns created by decomposing translation examples as translation knowledge. Therefore we can utilize the result of phrase alignment directly as translation patterns.

(3) The system has modules for sentence, phrase and structural alignment. Sentence alignment is used to construct parallel corpora, phrase alignment is used to make translation patterns, and structure alignment is used to make advanced translation patterns such as word selection rules and flexible collocation.

(4) The alignment process and the translation process can reuse and accumulate translation results and translation dictionaries. This accumulation progressively improves both the alignment quality and the translation quality.

(5) Users can inspect intermediate results to ensure the quality of translation results. Human operation is introduced at important points.

1.4 Outline of this Thesis

This thesis studies fundamental techniques needed for the practical machine translation system described in Figure 1.4.

In Chapter 2, we propose a translation engine; a pattern-based machine translation system allowing complex patterns which are brief and understandable.

Chapter 3 describes a method to find correspondences of arbitrary length word sequences statistically in aligned parallel corpora. The main objective is to evaluate the quality of correspondences extracted without translation dictionary, and to improve the existing similarity measures.

Chapter 4 is a follow-up on Chapter 3. In order to make the method of Chapter 3 practical, we effectively combine this method with manual confirmation and linguistic resources, such as chunking information and translation dictionaries.

Chapter 5 presents a method to automatically extract more advanced translation rules from a parallel corpus by applying structural matching.

Chapter 6 recapitulates this thesis, discusses recent related works and compares them with our work, and presents future directions.

Chapter 2

Pattern-based Machine Translation allowing Complex Patterns

2.1 Introduction

In Chapter 1, we proposed the ideal translation environment we are aiming at. In this chapter, we propose the pattern-based machine translation that is the core of this translation system.

The pattern-based machine translation system we have developed simplifies the handling of features in patterns by allowing sharing constraints between non-terminal symbols, and implementing an automated scheme of feature inheritance between syntactic classes.

In the first section, we discuss the problems occurring with previous pattern-based machine translation and our solutions. Section 2.3 shows the outline of the pattern-based machine translation system we have developed. Section 2.4 describes the implementation and the evaluation. Section 2.5 describes Collaborative Translation Environment “Yakushite Net” which employs this translation engine. In the last section, we conclude.

2.2 Problems of Pattern-based Machine Translation

Pattern-based translation systems execute the parsing, transferring and generating processes by using only translation patterns, all the knowledge necessary for the translation being written in patterns. This provides good readability for all this knowledge, and what is more, it is easy for users to add new translation patterns. However, in previous pattern-based systems, writing new patterns was difficult due to the lack of flexibility in the way to describe constraints on the features associated with a non-terminal, requiring for instance a new non-terminal for each semantic condition, so that a deep understanding of the internals of the system was necessary in order to add new patterns[52].

We have built a pattern-based machine translation system with good readability by writing all the conditions, including semantics, gender and number of non-terminals and words as a combination of features, and making it possible to match, share and inherit features, but without full feature unification. Moreover, this system solves the problem of large computation times, by implementing feature inheritance through copying rather than unification, and by drastically reducing the number of candidates through the pruning of the features kept on each non-terminal symbol.

Rich expressiveness enables the user to enter accurate patterns, reducing potential conflicts with other patterns. The user need not know the details of how the pattern will be processed during translation. It is also possible to enter translation patterns acquired by statistical methods directly.

The system provides also priority control between patterns and between dictionaries in order to avoid an explosion of the number of candidates and reduce side effects caused by newly introduced patterns. Special cases where patterns cannot handle generation in the target language are processed by a Post Generator.

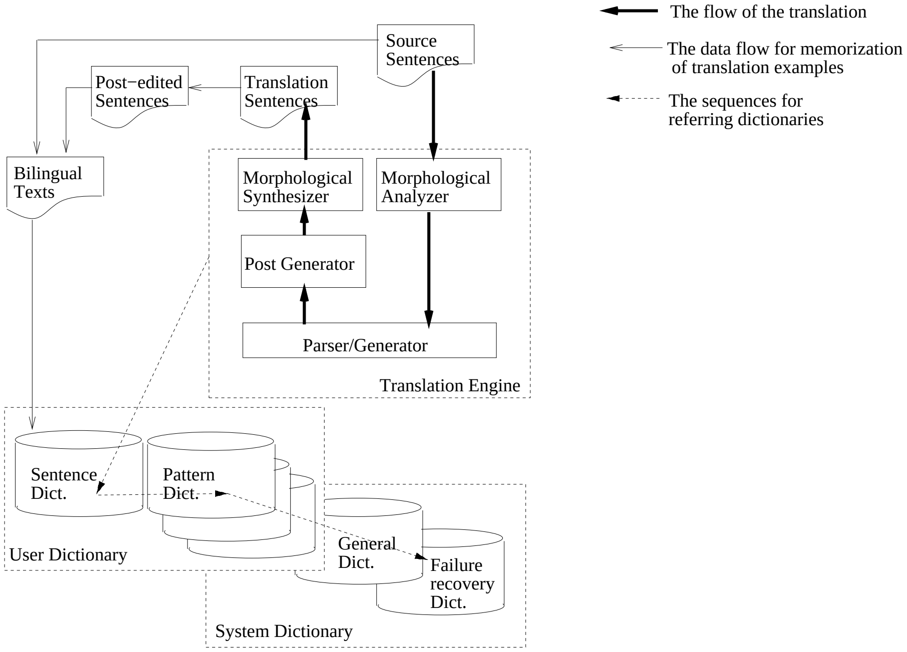


Figure 2.1 The architecture of pattern-based machine translation

2.3 Pattern-based Machine Translation allowing Complex Patterns

2.3.1 System Architecture

Figure 2.1 shows the architecture of our system. Thick arrows show the flow of the translation, thin arrows show the data flow for memorization of translation examples, and dotted lines show the sequence for referring dictionaries.

First, the source sentence is analyzed morphologically, normalizing words and decorating them with morphological features. This decorated sequence of words is then passed to the parser. The sentence is parsed by using the source side of translation patterns in the appropriate user and system dictionaries, and combining them bottom-up. When the sentence has been parsed successfully, the parse tree is translated by top-down generation of the parse tree of the target language, using the target side of patterns. Then, some features of the generated tree are handled by the Post Generator to produce refined sentences. Lastly, the morphological synthesizer adjusts inflection and conjugation, and the translated sentence is output.

Automatic feedback of correct post-edited translations and accumulation of translation examples improve the quality of future translations.

Morphological Analyzer

Morphological analysis uses a morphological dictionary, and associates to each surface form a normalized form, together with features specifying the part of speech, agreement, surface conjugation, and case. In most cases only one normalized form will be associated with a surface form, eventually with some of its features being multi-valued (for instance, for a verb in basic form, its agreement might be all persons except 3rd singular). In the special case of homonyms, the same surface form comes from two different dictionary words, and the result of the morphological analysis contains several different candidates for an input word. This is later handled by trying all these candidates in the parsing phase.

To simplify our presentation we will omit this case, and suppose in the following that the result of the morphological analysis is a linear sequence of words decorated with (eventually multi-valued) morphological features.

Parser and Generator

Figure 2.2 shows examples of translation patterns used in English to Japanese translation[1] . Examples (a)-(i) in Figure 2.2 are vocabulary patterns, (j)-(n) are grammatical patterns. In rule-based translation systems, vocabulary patterns would correspond to dictionaries, and grammatical patterns to grammar rules. As Figure 2.2 shows, patterns allow writing grammar rules and dictionaries in a united form without any specific distinction. All patterns are entered together in the system dictionary.

One can understand pattern (a) as the following CFG rules.

[1] Real patterns contain more features, but we omitted here features that are not required by our examples

```
(a) [en:VP:sSem=human play:pos=v:* [1:NP:sem=instrument]]
    [ja:VP [1:NP] を:pos=particle 弾く :pos=v:*];
(b) [en:VP:sSem!=human play:pos=v:*]
    [ja:VP 鳴る :pos=v::];
(c) [en:VP:sSem=human play:pos=v:* [1:NP:sem=sport|game]]
    [ja:VP [1:NP] を:pos=particle する :pos=v:*];
(d)+[en:VP:sSem=human play:pos=v:* ]
    [ja:VP [1:NP] 遊ぶ :pos=v:*];
(e) [en:N piano:pos=n:sem=instrument:*]
    [ja:N ピアノ :pos=n:*];
(f) [en:N tennis:pos=n:sem=sport:*]
    [ja:N テニス:pos=n:*];
(g) [en:N Ken:pos=n:sem=human:*]
    [ja:N 健:pos=n:*];
(h) [en:Adv never:pos=adv:*]
    [ja:Fs 決して :pos=n:*:postGen=neg];
(i) [en:SentenceSub when:pos=conj [1:Sentence:*]]
    [ja:SentenceSub [1:Sentence:sentenceType=sub:*] 時:pos=conj];
(j) [en:NP [1:N:*]]
    [ja:NP [1:N:*]];
(k) [en:S [1:Sentence:*]]
    [ja:S [1:SntenceType=main:*]];
(l) [en:Sentence [1:NP:sem={SEM}:personNum={NUM}][2:VP:sSem={SEM}:personNum={NUM}:*]]
    [ja:Sentence:sentenceType=main [1:NP] は :pos=particle [2:VP:*]];
(m)-[en:Sentence [1:NP:personNum={NUM}][2:VP:personNum={NUM}:*]]
    [ja:Sentence:sentenceType=main [1:NP] は :pos=particle [2:VP:*]];
(n) [en:Sentence [1:NP:sem={SEM}:personNum={NUM}][2:VP:sSem={SEM}:personNum={NUM}:*]]
    [ja:Sentence:sentenceType=sub [1:NP] が :pos=particle [2:VP:*]];
```

Figure 2.2 Examples of translation patterns

English: VP → play NP
Japanese: VP → NP を (*wo*) 弾く (*hiku*)

A pattern starts with the name of the language, and 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 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, marked by 'en' here. By applying patterns bottom-up, one can reduce word sequences to the corresponding left-hand side, and eventually reach the ’S’ non-terminal (the root of the parse tree), in pattern (k) of Figure 2.2.

Once the source parse tree has been completed, it is sufficient to convert each node using the corresponding target language pattern, marked by ’ja’ . 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.

Translation patterns can specify one or more features for both terminal and non-terminal symbols, such as ’*pos=verb*’ (the part of speech is verb), ’*personNum=3sg*’ (third person and singular), ’*sem=human*’ (the semantics is human). They can allow one or more values for one feature and also can specify negative information as in ’*pos!=verb*’ (the part of speech is not verb).

The features in the right-hand side of the source language patterns express conditions, either by requiring a specific value for a feature, or expressing a sharing constraint between two features, through unification variables (in curly brackets, like ’{SEM}’ or ’{NUM}’). Matching succeeds if all these conditions are satisfied. Corresponding words in the input sequence are then replaced by the non-terminal on the left-hand side, while the corresponding parse tree is built.

In order to ease the propagation of features inside the parse tree, one of the right-hand side pattern elements is designated as head, and marked by a “*” . Its features are inherited by the left-hand side non-terminal, except for those already defined in the left-hand side, which are ignored. Features on the left-hand side of source-language patterns, together with inherited features, appear in the newly replaced non-terminal, and they will be matched later by the right-hand side of other patterns.

Word selection in the target language is realized by checking features. In simple cases, the condition is directly applied to a symbol in the pattern. For instance, in patterns (a) and (c), “play” is associated with different semantic values according to whether its object is a music instrument, or either a sport or game; then it is translated into proper words in these different situations: play the piano gives “ピアノ (*piano*) を (*wo*) 弾く (*hiku*)” , but play tennis gives “テニス (*tenisu*) を (*wo*) する (*suru*)”.

More complex cases, like the difference between “a piano plays” and “Ken plays" , use sharing constraints and feature inheritance. Here the semantic features instrument and 'human' are inherited from both name patterns and verb phrases, and they are checked in the sentence construction pattern (l). Only agreeing subject and verb will be accepted, enabling the system to provide the proper translations "ピアノ (*piano*) は (*ha*) 鳴る (*naru*)" and "健 (*ken*) は (*ha*) 遊ぶ (*asobu*)".

In patterns (l), (m) and (n), sharing constraints are also a concise way of uniting person and number information.

In target language patterns, propagation works the other way round: features on the left-hand side of the target pattern act as constraints for the generation process, and features on the right-hand side are propagated to child nodes. Inheritance goes from the parent node to the head node, with the same overriding mechanism for features present in both.

The matching of the target language features makes it possible to provide proper translations in different grammatical situations. For example, differences such as the one between the subordinate clause "私 (*watashi*) が (*ga*) ピアノ (*piano*) を (*wo*) 弾く (*hiku*) 時 (*toki*)" (it means "when I play the piano") in pattern (n) and the complete sentence "私 (*watashi*) は (*ha*) ピアノ (*piano*) を (*wo*) 弾く (*hiku*)" (it means "I play the piano") in pattern (l) can be translated accurately.

Lastly, two decisions were taken to avoid multiplication of candidates. One is that the set of features each non-terminal symbol can have is limited according to a feature definition table as seen in Figure 2.3. For instance the CFG rule for S 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 symbols that have the same combination of feature values can be merged, and a 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: one can only choose between two patterns 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.

```
Sentence = { sentenceType };
VP       = { personNum
             conjugation
             subjSem      };
NP       = { personNum
             sem          };
```

Figure 2.3 The example of feature definition table

```
<Rule NAME=''postGen=neg''>
<StartLeaf>
  <Feature NAME=''postgen'' VALUE=''neg''/>
</StartLeaf>
<Scope TYPE=''NEAREST''>
  <Feature NAME=''category'' VALUE=''VP''/>
</Scope>
<OriginalLeaves>
  <OriginalLeaf ID=''1'' DIR=''LtoR''>
    <Feature NAME=''postgen'' VALUE=''neg''/>
  </OriginalLeaf>
  <OriginalLeaf ID=''2'' DIR=''RtoL''>
    <Feature NAME=''pos'' VALUE=''v''/>
  </OriginalLeaf>
</OriginalLeaves>
<EditedLeaves>
  <EditLeaf ID=''1'' COPYFROM=''1''/>
  <EditLeaf ID=''2'' COPYFROM=''2''/>
  <EditLeaf ID=''2'' DELTA=''1''>
    <Feature NAME=''pos'' VALUE=''aux''/>
    <Feature NAME=''baseForm'' VALUE=''ない''/>
  </EditLeaf>
</EditedLeaves>
</Rule>
```

Figure 2.4 The example of Post Generator rules

Post Generator

Generation using a synchronized grammar depends strongly on the structure of source language patterns, so 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. To take a simple example, although the Japanese translation for the English word never is "決して (*kesshite*) ... ない (*nai*)" , pattern (h) of Figure 2.2 cannot lexicalize "ない (*nai*)". Because the verb which never qualify cannot be identified when pattern (h) is applied.

The feature '*postGen=neg* within "決して (*kesshite*)" is matched by a Post Generator Rule which generates "ない (*nai*)" at the end of the verb phrase which includes "決して (*kesshite*)".

Figure 2.4 indicates an example of the rule for Post Generator. The rule means, if a word holds "*postGen=neg*, put "ない (*nai*)" backmost of VP(verb phrase) which includes the word. The rule is written in XML notation.

2.3.2 Our Approach to Search Space Control

The main problem pattern-based translation faces is that of effectively controlling the search space. If strict conditions are set for patterns, the translation is likely to end up in failure, however if patterns with very few conditions are used, too many patterns are applied, and the number of candidates increases explosively. To avoid this problem, we have introduced two priority control systems for patterns.

Control of Priority in a dictionary

Pattern (l) of Figure 2.2 shows a translation pattern in which the semantics of the subject is limited so that it can respond to different situations. However, if a user is not careful enough and does not give accurate semantics information in his/her pattern, it will not be matched and the translation will fail. To protect the system from such mistakes, translation patterns without limitation of meaning are also needed. However, when the strict pattern succeeds, the unlimited one will also succeed, and the number of candidates increases combinatorially. Even worse, unless one pattern is given preference, after the parsing process the system cannot judge which result is better and cannot choose a unique plausible translation.

To avoid these situations, the system provides a way to mark a pattern as being applicable only when patterns with more detailed conditions are not matched, by putting a - (minus) mark before it as in pattern (m). This avoids the situation where both patterns are applied. Experience showed us that we needed three priority levels. So there is also + (plus) in pattern (d) for higher priority patterns.

An additional criterion we use to select patterns is to choose a parse tree using a minimal number of patterns, as it will include patterns closer to the input sentence. This information is combined with the above priority of individual patterns to provide a comprehensive evaluation of parse trees.

Control of Priority between Dictionaries

Two problems may arise when users input a large number of patterns. One is a potential slowdown in translation speed, which is affected by the overall number of patterns. The other is that newly introduced patterns may conflict original ones and cause unstable translation behavior. We solve the two problems by developing a pruning mechanism, which would consider user patterns first, and then some dictionaries correlated with the user dictionary, and finally system dictionaries during translation. This pruning avoids an explosion of the number of candidates, and side effects caused by newly introduced user patterns are limited to this user dictionary.

Failure Recovery Dictionary

We have introduced the Failure Recovery Dictionary using the above pruning mechanism. Failure recovery dictionary is referred last among sub-dictionaries in the system dictionary. In other words, the failure recovery dictionary acts only when the normal parsing process using other dictionaries has failed.

The Failure Recovery Dictionary contains patterns with grammatical mistakes and patterns that help avoiding unsuccessful translation. For instance the following pattern allows the use of a subject and a verb for which agreement rules are not satisfied.

```
[en:Sentence [1:NP ] [2:VP:*]]
[ja:Sentence:setenceType=main [1:NP] は [2:VP:*]];
```

By default the system will work on a rigid translation that is grammatically correct, but does not consider rare phrase structures. This avoids slowing down translation of simple sentences. Whenever normal translation fails, the system tries again to translate with more patterns, which is slower but much more robust.

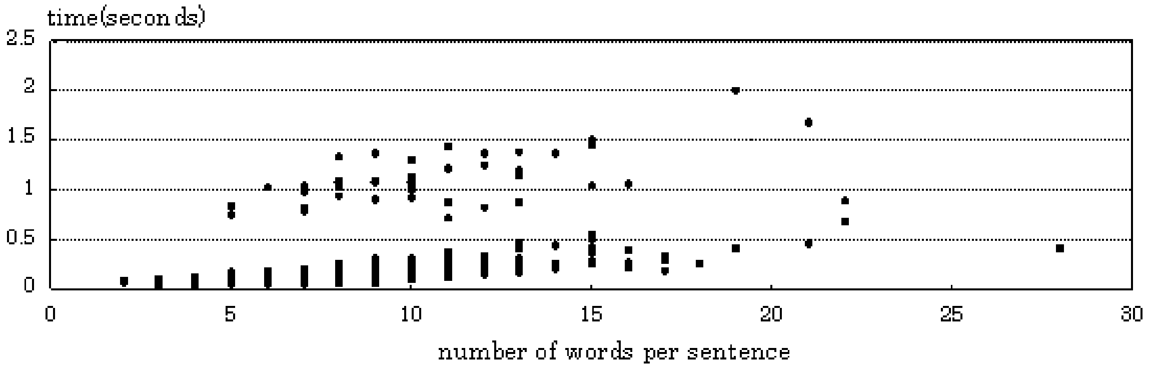


Figure 2.5 Translation speed

2.4 Implementation and Evaluation

2.4.1 Process for Development of Patterns

The number of grammatical patterns is about 2,000 and the vocabulary patterns are about 180,000. Vocabulary patterns were built based on dictionaries for a rule-based machine translation system which we had developed before.

Grammatical patterns newly was designed and developed to cover Collins' grammar [10]. For each item in the grammar we made an example and then created the corresponding pattern by hand. We also created various test examples for each item in the grammar and used them to check for conflicts in subsequent patterns.

The conflict rate is about 3% when we added new grammatical patterns. But our debugger, which can indicate visually the pattern selection process and the result of applied patterns, facilitated the detection of the cause of conflicts. Furthermore when we detected the cause, we could adjust patterns easily by refinement and addition of conditions.

2.4.2 System Specification and Evaluation

The above English-Japanese machine translation system has been implemented in Java.

The parser uses the *Earley* algorithm. At the time of this writing, the number of non-terminal symbols is about 80, and about 60 types of features are defined.

Most of the vocabulary patterns are managed in databases. The databases are converted into pattern format for entry into the dictionary. The number of rules for the Post Generator is about 280. Using only system dictionaries, we evaluated the translation quality using the JEIDA English-Japanese translation evaluation set [21], which is composed of 770 bilingual sentences. The failure recovery dictionary was referred by 9 sentences and the number of sentences that failed to parse is 10. The percentage of translations that were judged correct by professional translators was about 94 percent.

Moreover, the speed is acceptable and the translation time is roughly proportional to the length of sentences. Figure 2.5 shows processing time per sentence on a Pentium III machine at 933MHz. Translation times are noticeably slower when a sentence contains several structurally ambiguous constructions, such as coordination.

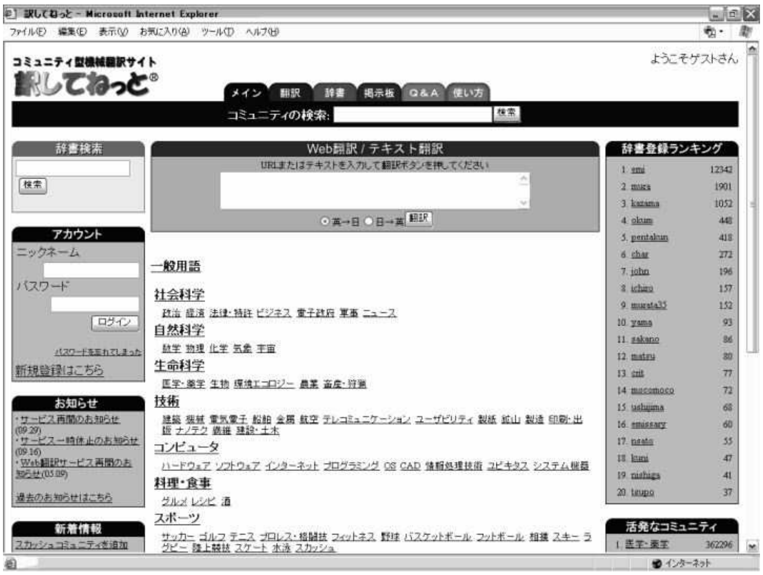
2.4.3 Comparison with rule-based machine translations

Now, we compare the translation patterns used in rule-based machine translation system and with those of our pattern-based machine translation system. ALT-J/E[17], is a transfer-based machine translation system employing transfer patterns as verbal word selection rules. Transfer patterns are similar to our patterns, as below

example (1): N1(subject) が (*ga*) LN2(permission) を (*wo*) 取る (*toru*)
→ N1 take N2 ex2:

example (2) N1(subject) が (*ga*) LN2(hotel) を (*wo*) 取る (*toru*)
→ N1 reserve N2

Transfer-based machine translation applies the patterns after the parsing completes and transfers the structure from source language to target language. Consequently it allows only particular patterns that have explicit parsing result, and cannot describe patterns as freely as our method. TDMT[23] could be described as pattern-based, it is however limited in a number of ways. First, each pattern,


訳してねっと - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
コミュニティ型機械翻訳サイト
訳してねっと®
ようこそゲストさん
メイン
翻訳
辞書
掲示板
Q&A
使い方
コミュニティの検索:
検索
辞書検索
検索
Web翻訳 / テキスト翻訳
URLまたはテキストを入力して翻訳ボタンを押してください
⊙英→日 ○日→英
翻訳
アカウント
ニックネーム
パスワード
ログイン
パスワードを忘れてしまった
新規登録はこちら
お知らせ
・サービス再開のお知らせ (09.29)
・サービス一時休止のお知らせ (09.16)
・Web翻訳サービス再開のお知らせ(05.09)
過去のお知らせはこちら
新着情報
スカッシュコミュニティを追加
一般用語
社会科学
政治 経済 法律・特許 ビジネス 電子政府 軍事 ニュース
自然科学
数学 物理 化学 気象 宇宙
生命科学
医学・薬学 生物 環境エコロジー 農業 畜産・狩猟
技術
建築 機械 電気電子 船舶 金属 航空 テレコミュニケーション ユーザビリティ 製紙 鉱山 製造 印刷・出版 ナノテク 繊維 建設・土木
コンピュータ
ハードウェア ソフトウェア インターネット プログラミング OS CAD 情報処理技術 ユビキタス システム機器
料理・食事
グルメ レシピ 酒
スポーツ
サッカー ゴルフ テニス プロレス・格闘技 フィットネス 野球 バスケットボール フットボール 相撲 スキー ラグビー 陸上競技 スケート 水泳 スカッシュ
辞書登録ランキング
活発なコミュニティ
インターネット

1.	emi	12342
2.	miura	1901
3.	kazama	1052
4.	okum	448
5.	pentakun	418
6.	char	272
7.	john	196
8.	ichiro	157
9.	murata35	152
10.	yama	93
11.	sakano	86
12.	matsu	80
13.	crit	77
14.	mocomoco	72
15.	ushijima	68
16.	emissary	60
17.	nsato	55
18.	kuni	47
19.	nishiga	41
20.	tsupo	37

1.	医学・薬学	362296

Figure 2.6 Top page of "Yakushite Net"

called transfer knowledge, must contain constituent boundary either a functional word or a special part-of-speech bigram marker, inserted by the morphological analyzer. Then, pattern features are very limited, allowing for semi-automatic acquisition, but precluding efficient generalization. These limitations mean that some complex phrase structures cannot be analyzed, and that even simple patterns must be given in lots of instances to overcome the absence of generalization.

2.5 Collaborative Translation Environment: "Yakushite Net"

Pattern-based translation systems get better as many users from various backgrounds use them, and enter lacking patterns, particularly technical words and idioms, which have an immediate impact on translation quality. For this purpose, we applied our system to the Collaborative Translation Environment "Yakushite Net" usable through Internet. Figure 2.6 shows the top page of "Yakushite Net".

This environment provides "community" dictionaries, which the user can se-

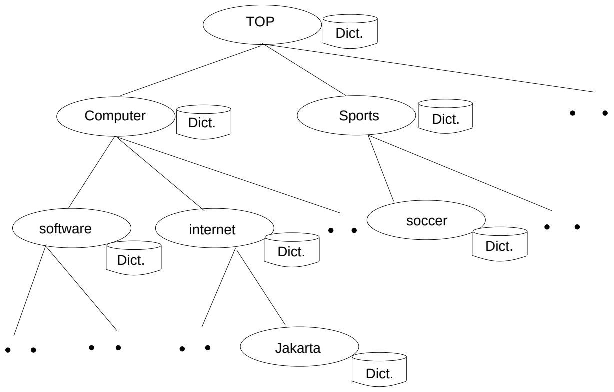


Figure 2.7 Tree structure of community dictionaries

lect according to his/her needs. These dictionaries can be improved by contributions from members of the community, distributed over the Internet. There are many communities, and their dictionaries are structured hierarchically, as shown in figure 2.7.

When translating in a certain community environment, the translation engine refers first to the community's own dictionary, and subsequently to broader dictionaries, starting with the parent community up to the top, with decreasing priority. These community dictionaries except the top one correspond to the user dictionaries in figure 2.1. The top dictionary is domain-independent, and corresponds to the system dictionary in figure 2.1.

Since "Yakushite Net" was opened to the public on 2004/09/29, the number of users has been steadily increasing, and there are 988 users at the time of this writing (2004/11/20). On an average daily basis, there 3,304 page views and 1,213 translation request a day. There are 235 kinds of community dictionaries on "Yakushite Net", and their total number of entries is 138,472, of which 6,235 were registered by general users.

We see the construction of well-targeted domain specific dictionaries, and their use according to the context, as the best solution to avoid unwieldy addition of user patterns.

2.6 Summary

The machine translation system we have developed has two major advantages.

1. The system is pattern-based, but it is possible to share constraints and inherit features between non-terminal symbols, simplifying the input of patterns.

2. The system has two priority control systems. One is the priority control among patterns in a dictionary. The other is the priority order between dictionaries using a pruning algorithm. The dictionary with the least priority is the failure recovery dictionary.

This machine translation system has already been available to users on Internet as the collaborative translation environment 'Yakushite Net'. We verified that the scalability, the usability and the translation quality are satisfactory.

Translation pattern dictionaries which this machine translation utilizes can be constructed by the word and phrase alignment technologies described in Chapter 4 and 5. Additionally more complex patterns such as a word selection rule and patterns containing non-terminal symbols can be constructed by the structural alignment technology described in Chapter 6.

Chapter 3

Translation Pattern Extraction based on Statistical Approach

3.1 Introduction

High quality translation dictionaries are indispensable for machine translation systems targeting good performance, especially for specific domains. Such dictionaries are effectively usable only for their own domains, and it would be extremely helpful if such a dictionary was obtained in an automatic way from a set of translation examples.

This section proposes a method to construct translation dictionaries that consist not only of word pairs, but also pairs of word sequences of arbitrary length. All pairs are extracted from a parallel corpus of a specific domain. The method is proposed and is evaluated with Japanese-English parallel corpora of three distinct domains.

In our method, the corresponding pairs are determined stepwise according to the similarity value. Iteration may be incorporated so that more plausible corresponding pairs are identified earlier, and a pair will be never reconsidered once it is fixed as the corresponding pair, and the remaining undergo the next iteration of recalculation of the similarity value[1] .

In the next section, we report the result of a preliminary experiment for various

[1] This algorithm is called "*greedy algorithm*" [34].

upper n words	MI	$Dice$	$Dice'$	
			$log_2 f_{je}$	f_{je}
100	64	74	99	69
500	60	81	90	60
1,000	54	75	77	50

Table 3.1 Comparison of Mutual Information and Dice coefficient

similarity measures between English and Japanese word sequences. Section 3.3 shows how to construct a translation dictionary in details. In Section 3.4, we describe an experiment and its evaluation, with 3 types of corpora. Section 3.5 summarizes.

3.2 Preliminary Experiment for Similarity

As described in chapter 1, Mutual Information (MI) and Dice coefficient are used conventionally as measures for plausible corresponding pairs.

3.2.1 Mutual Information and Dice Coefficient

To compare between MI and Dice coefficient, we calculated the similarity of Japanese and English words, using parallel corpora including 10,016 pairs of business sentences. The process is as follows: First Japanese and English texts are analyzed morphologically and all content words(nouns, verbs, adjectives and adverbs) are identified[2] . Next all content words with two or more occurrences are extracted, and then the two similarities of Japanese and English words, i.e. MI and Dice coefficient, are calculated.

Table 3.1 shows the percentage of correct pairings of Japanese and English words when confirming the upper n pairs manually ($n = 100$, 500, 1000).

[2] Japanese and English morphological analyzers of the Machine Translation System PENSÉE were used. PENSÉE is a trademark of Osaka Gas corporation, OGIS-RI, and Oki Electric Industry Co., Ltd.

The results in Table 3.1 show that Dice coefficient gives better correctness than MI. The reason is that MI gives abnormally high scores towards with few occurrences. As pointed in Dagan [11], Haruno et.al. [16] and Ohmori et.al.[39], MI requires that word pairs have a sufficient number of occurrences. However, in order to extract more word pairs, we calculate the similarity down to two occurrence. As a result, we think word pairs with few occurrences are extracted early, causing low for MT.

In a similar experiment, Ohmori [39] reported also on the comparison of MI and Dice coefficient. They applied both approaches to a French-English corpus of about one thousand sentence pairs. Since both methods show very inaccurate results for words with a single occurrence, only words with two or more occurrences were selected for inspection. The results show that though Dice coefficient gives a slightly better correctness, both methods do not generate satisfactory translation pairs.

Smadja [49] also reported that Dice coefficient is more effective. They compared the two formulas using their alignment algorithm. The results show that out of 43 extracted pairs, Dice coefficient could correctly extract 36, but MI only 26. They concluded that MI has not only problems with too few occurrences, but also that it is weak to the bias of the translation direction. For instance, while the Japanese word '**要点**(*yoten*)' is always translated by the English word "point", conversely "point" is translated by many words, such as '**点**(ten)', '**個所**(*kasho*)', '**ポイント** (*pointo*)' and '**先**(*saki*)', but this confuses MI. Such a phenomenon cannot be ignored.

Dice coefficient also has an important defect, which we will try to correct.

3.2.2 Weighted Dice Coefficient

The defect of Dice coefficient is that the similarity is decided only by the ratio of the co-occurrences to that of separate occurrences in both languages, independently their absolute value.

In Table 3.1, the upper 100 words are less correct than the upper 500 words. The reason is that the upper 100 words have more pairs whose co-occurrence is two times and as a result total correctness is lowered by the presence of these pairs.

In the formula of Dice coefficient, a perfect co-occurrence has similarity 1, whether it happens 2 times or 100 times.

But comparing results for both experiments, or is clear that the co-occurrence of 100 times is more correct. We need to adopt a formula that takes the absolute number of co-occurrences in consideration.

For the above reason, we weighted Dice coefficient with co-occurrence. The following formula defines the weighted Dice coefficient.

$$sim(w_J, w_E) = w(f_{je}) \cdot \frac{2f_{je}}{f_j + f_e} \tag{3.1}$$

w_J: a Japanese word sequence
w_E: an English word sequence
f_j: the number of occurrences of w_J
f_e: the number of occurrences of w_E
f_{je}: the co-occurrence of w_J and w_E
$w(f_{je})$: the weighting coefficient for f_{je}

In order to investigate an appropriate coefficient $w(f_{je})$, two values, $w(f_{je}) = log_2 f_{je}$ and $w(f_{je}) = f_{je}$, were tested in the same experiment as the above section. $Dice'$ in Table 3.1 shows the results.

$w(f_{je}) = f_{je}$ puts too much emphasis on the raw number of co-occurrences, and leads to lower correctness. The weight $w(f_{je}) = log_2 f_{je}$ leads to higher correctness, avoiding errors but not creating new ones.

Following these results, we choose $log_2 f_{je}$ as the weight coefficient $w(f_{je})$, in the extraction algorithm described in the next section.

3.3 Overview of the Method

Figure 3.1 shows the flow of the process to find the correspondences of Japanese and English word sequences. Both Japanese and English texts are analyzed morphologically.

We make use of two types of co-occurrences: Word co-occurrences within each language corpus and corresponding co-occurrences of those in the parallel corpus. In the current setting, all words and word sequences of two or more occurrences

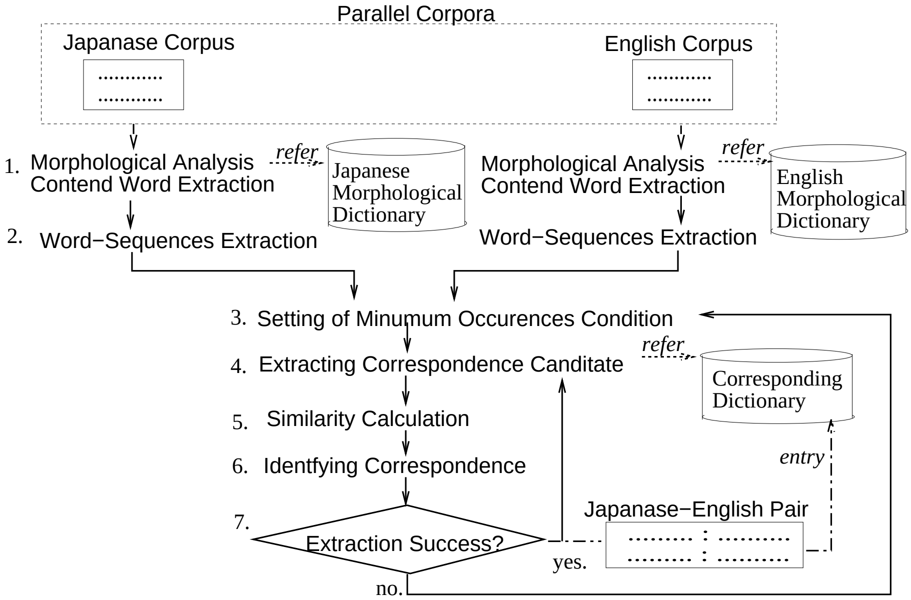

Parallel Corpora
Japanase Corpus
English Corpus
1. Morphological Analysis
Contend Word Extraction
refer
Japanese Morphological Dictionary
Morphological Analysis
Contend Word Extraction
refer
English Morphological Dictionary
2. Word−Sequences Extraction
Word−Sequences Extraction
3. Setting of Minumum Occurences Condition
4. Extracting Correspondence Canditate
refer
Corresponding Dictionary
5. Similarity Calculation
6. Identfying Correspondence
entry
Japanase−English Pair
7. Extraction Success?
yes.
no.

Figure 3.1 The flow of finding the correspondences of word sequences

are taken into account. Since frequent co-occurrence suggests higher plausibility of correspondence, we set a similarity measure that takes co-occurrence frequencies into consideration. Deciding the similarity measure in this way reduces the computational overhead in the later processes. If every possible correspondence of word sequences were to be calculated, the number of combinations would be huge. Since high similarity value depends on supported by high co-occurrence frequency, a gradual strategy can be taken by setting a threshold value for the similarity and by iteratively lowering it. Though our method does not assume any bilingual dictionary in advance, once words or word sequences are identified in an earlier stage, they are regarded as plausible entries of the translation dictionary. Such translation pairs are taken away from the co-occurrence data, then only the remaining word sequences need be taken into consideration in the subsequent iterative steps. Next section describes the details of the algorithm.

3.3.1 The Algorithm

The step numbering of the following procedure corresponds to the numbers appearing in Figure 3.1. In the current implementation, the “Corresponding Dictionary” is empty at the beginning.

(1) Japanese and English texts (E, J) are separately analyzed morphologically.

(2) For each sentence pair (ES, JS) of the analyzed texts, the sets EWS and JWS of phrase-like word sequences of length at most l_{max} (e.g. no more than 5) appearing in ES and JS are constructed, and the pair (EWS, JWS) is inserted in a database. The total number of occurrences of each word sequence is also kept separately.

(3) An initial threshold value f_{min} for the minimum number of occurrences is chosen appropriately according to the database.

(4) For every pair of word sequences occurring more than f_{min} times, the total number of bilingual co-occurrences in the database is counted.

(5) For each such pair, a correlation score is calculated. The most plausible correspondences are then identified using the correlation scores. The approved correspondences are registered in the “Correspondence Dictionary”.

(6) For each newly registered correspondence (ews, jws), and each pair (EWS, JWS) in the database such that EWS contains ews and JWS contains jws , all word sequences including ews (resp. jws) are removed from EWS (resp. JWS). The total number of occurrences for each word sequence is updated. The steps (4) through (6) are repeated until no new pair is approved.

(7) The threshold value f_{min} is lowered, and the steps (4) through (7) are repeated until f_{min} reaches a predetermined value f_{end} (e.g. $f_{end} = 2$).

This method has the advantage of being able to produce probable correspondences with high correlation score earlier in the process, and since approved correspondences are used to reduce the number of candidates, it also reduces incorrect candidates during ulterior steps.

corpus	sentence	single word				word seq.	
		total		occurrence$\geq$ 2		occurrence$\geq$ 2	
		Eng.	Jap.	Eng.	Jap.	Eng.	Jap.
business	10,016	2,300	3,739	2,218	3,568	73,026	72,574
science	9,792	7,254	9,415	6,764	8,856	27,329	37,258
computer	11,477	3,701	4,926	3,478	4,799	32,049	38,796

Table 3.2 Numbers of extracted words and word sequences

3.4 Experiments and Results

We experimented to extract translation patterns using parallel corpora of three distinct domains. This section describes the experimental settings, the results, and discuss then.

3.4.1 Experimental Settings

We used parallel corpora of three distinct domains: (1) a computer manual (9,792 sentence pairs)[3] , (2) a scientific journal (12,200 sentence pairs)[4] , and (3) business contract letters (10,016 sentence pairs) [20].

We call each corpus "business", "science", and "computer." All the Japanese and English sentences are aligned, and redundant parallel sentences are deleted. The final numbers of sentences are as follow; "business" has 10,016 sentences, "science" has 9,792 sentences, and "computer" has 11,477 sentences.

All sentences are morphologically analyzed[5] , and content words are extracted. The maximum length of the extracted word sequences is set at 10, and all content words of two or more occurrences are extracted. The initial value of f_{min} is set at the half of the highest number of occurrences of extracted word sequences and

[3] Online manuals of a computer made by Oki Electric Co., Ltd

[4] we OCRed articles in a year's "Scientific American" (English) and "Nikkei Science" (Japanese)

[5] Japanese and English morphological analyzers of Machine Translation System PENSÉE were used. PENSÉE is a trademark of Osaka Gas corporation, OGIS-RI, and Oki Electric Industry Co.,Ltd.

threshold	Num. of pairs	correct	near miss	correctness(+near)
1151	2	2	0	100(100)
575	3	3	0	100(100)
287	4	4	0	100(100)
143	12	12	0	100(100)
71	19	18	1	97.5(100)
35	48	48	0	98.9(100)
17	103	101	2	98.4(100)
10	164	155	8	96.6(99.7)
9	53	51	2	96.6(99.8)
8	67	63	4	96.2(99.8)
7	82	75	6	95.5(99.6)
6	134	114	20	93.5(99.7)
5	163	145	15	92.6(99.4)
4	318	257	50	89.4(98.6)
3	755	502	195	80.4(96.1)
2	1,975	(276)*	(169)*	67.7(92.5)
Total	3,902	$\sim 2,640$	~ 970	67.7(92.5)

Table 3.3 Results for “business” corpus

is lowered by dividing by two until it reaches to or under 10, then it is lowered by one in each iteration until 2.

3.4.2 Results and Discussion

Table 3.2 summarizes the numbers of sentences and word sequences extracted by Step 2 of our algorithm. For each corpus the table show the numbers of distinct content words, those of two or more occurrences, and the numbers of word sequences (of length between 1 and 10) of two or more occurrences.

“Business” and “computer” have a smaller number of content words than “science.” But concerning to the total number of word sequences, “science” has actually less sentences than the others. The reason is that the two former have many similar sentences since they are very homogeneous. Since the latter

threshold	Num. of pairs	correct	near miss	correctness(+near)
68	1	1	0	100(100)
34	21	19	1	90.9(95.5)
17	69	64	5	92.3(98.9)
10	142	133	8	93.1(99.1)
9	52	49	3	93.3(97.9)
8	69	69	0	94.6(99.2)
7	66	63	2	94.7(99.0)
6	105	99	6	94.7(99.2)
5	168	155	12	94.1(99.1)
4	292	263	25	92.9(99.0)
3	536	494	34	92.6(98.4)
2	1,307	(445)*	(46)*	90.4(98.6)
Total	2,828	$\sim 2,572$	~ 217	90.4(98.6)

Table 3.4 Results for “science” corpus

“science” is a collection of many papers written by different authors, the words which occur in that corpus are more varied.

Tables 3.3, 3.4 and 3.5 show the statistics obtained from the experiments. The columns specify the numbers of approved translation pairs. The correctness of the translation pairs are checked by a human inspector. A “near miss” means that the pair is not perfectly correct but some parts of the pair constitute the correct translation. The leftmost “threshold” means the number of occurrences to the corresponding threshold f_{min}. “Num. of pairs” means the number of pairs extracted in the step. We evaluate the results by their correctness and coverage. Correctness is evaluated according to the following 3 levels. .

correct: the result can be used as an entry in the translation dictionary.

near miss: at most one spurious word on one side of the correspondence.

mistake: otherwise.

We use this criterion in tables. ()* shows the number obtained by spot-checking some arbitrary 500 pairs. The rightmost “correctness” means the ratio

threshold	Num. of pairs	correct	near miss	correctness(+near)
209	1	1	0	100(100)
104	4	4	0	100(100)
52	19	19	0	100(100)
26	55	54	0	98.7(98.7)
13	145	140	5	97.3(99.6)
10	81	76	5	96.4(99.7)
9	58	55	2	96.1(99.4)
8	75	68	5	95.2(99.1)
7	106	99	7	94.9(99.3)
6	126	118	7	94.6(99.3)
5	214	198	13	94.1(99.1)
4	367	330	26	92.9(98.5)
3	629	519	97	89.4(98.3)
2	1,401	(395)*	(87)*	85.0(97.5)
Total	3,281	$\sim 2,788$	~ 411	85.0(97.5)

Table 3.5 Results for “computer” corpus

of correct ones to extracted ones at more than the threshold. () means the correctness including near miss.

It is noticeable that the pairs with high frequencies give very accurate translation in the cases of the computer manual and the business letters, whereas the scientific journal does not necessarily give high accuracy to highly frequent pairs. The reason is that the former two corpora are from homogeneous domains, while the corpus of scientific journal is a mix of distinct scientific fields. The former two corpora reveal a worse performance with the pairs with low frequency threshold. This is because those corpora frequently contain a number of lengthy fixed expression or particular collocations.

The science journal shows a stable accuracy of translation pair extraction. The accuracy exceeds 90% in all the stages. The reason would be that scientific papers do not repeat many fixed expressions.

Table 3.7 summarizes the combinations of the length of English and Japanese

	Japanese	English	Similarity
— 1. business —			
△	紛争 (、) 論争 (又は) 意見	dispute(,) controversy (or) difference (which may) arise	4.34
	営業 秘密	trade secret	3.72
	営業 時間	business hour	2.92
	確認 (付) 取消 不能 信用状	irrevocable confirm(ed) letter (of) credit	2.81
	技術 製造 ﾉ ｳ ﾊ ｳ	technique manufacture know-how	2.62
— 2. science —			
	出血 熱 ｳ イルス	hemorrhage fever virus	3.19
	ﾛ スアﾗ ﾓ ス 国立 研究所	Los Alamos national laboratory	2
△	ﾝ 型 ｼ ﾘ コｰ ﾝ	n type	1.78
*	ﾝ 型 ｼ ﾘ コｰ ﾝ	p type	1.78
	ｶ ﾘ ﾌ ｫ ルﾆ ア 大学 ﾃﾞ ｰ ﾋﾞ ス 校	university (of) California (at) Davis	1.58
— 3. computer —			
	イﾝ ﾀ ﾈ ｯ ﾄ	internet	5.25
	イﾝ ﾀ ﾈ ｯ ﾄ プﾛ ﾄ コル	internet protocol	1.66
	ﾈ ｰ ﾑ ﾄ ゥ アﾄﾞ レス ﾏ ｯ ﾋﾟ ﾝ グ	name (to) address map(ping)	1.58
*	操作 (が) す で に	attempt(on a)socket(on which a)connect operation already	1.36
	イﾝ ﾀ ﾈ ｯ ﾄ ｻ ｰ ﾋﾞ ス	internet service	1.45

△ means ”near miss”, and * means “mistake”

Table 3.6 Samples of Extracting Translation Patterns

word sequences. The fraction in each entry shows the number of correct pairs over the number of extracted pairs. This table indicates that translation pairs of lengthy or unbalanced sequences are almost always incorrect.

Tables 3.6 lists samples of translation pairs extracted from 3 types of corpora, and 3.8 lists ones from “business”. Table 3.6 lists some typical word sequence pairs. Many Japanese translations of English technical terms are automatically detected. Table 3.8 lists the top 30 pairs from the experiment on business contract letters.

This method is capable of getting interesting translation patterns. For example, “**営業秘密** (*eigyou himitsu*)” and “**営業時間** (*eigyou jikan*)” are found to correspond to “trade secret” and “business hour” respectively. Note that Japanese word “**営業** (*eigyou*)” is translated into different English words according to their occurrences with distinct words.

Table 3.9 shows the recall ratio based on the results of the experiments. The figures show the numbers of words that are included at least one extracted translation pairs. The recall rates are shown in parentheses, and indicate the proportion

Business		Length of Eng. Seq.									
		1	2	3	4	5	6	7	8	9	10
	1	823/843	43/58	0/6	0/1	0	0	0	0	0	0
	2	32/45	401/450	17/55	1/23	0/5	0/4	0/1	0	0/1	0
Length	3	0	79/122	72/90	7/23	0/8	0/4	0/4	0	0	0
of	4	0	6/21	29/45	15/23	2/5	1/2	0/1	0	0	0/1
Jap.	5	0	3/10	2/13	7/14	3/10	2/3	0	0/1	0/1	0/1
Seq.	6	0	0	2/4	2/3	0/1	0/2	0	0/1	0/1	0/2
	7	0	0/1	0	0/2	0	0/1	0	0/1	0	0
	8	0	0/1	0	0/1	0/1	0/1	0	0	0	0/1
	9	0	0	0	0	0	0	0	0	0	0
	10	0	0/1	0/1	0	0	1/1	0	0	0	0/6

Table 3.7 Length combination of word sequences and their accuracy “business”

of the words with two or more occurrences in the corpora that finally participated in at least one translation pair. For the “business” we consider the recall rate for $f_{min} >= 3$. The major reason that the recall rate is relatively low is that we decided to use a rather severe condition for selecting translation pairs in Step 6 of our algorithm. The condition may be loosen to get a better recall ratio though we may lose in precision. We have not yet tested our method with other conditions.

3.5 Summary

In this chapter, we proposed a method for obtaining translation dictionaries from parallel corpora, in which not only word-level correspondences but arbitrary length word sequence correspondences are extracted.

This work was originally motivated by the purpose of improving the performance of our translation pattern extraction from parallel corpora described in Chapter 5, in which translation patterns are extracted by syntactically analyzing both Japanese and English sentences and by structurally matching them.

During experiments, some discrepancy was caused by the poor quality of the translation dictionary. This is why we tried to pursue a way to obtain a

Japanese	English	Similarity	Pair.Freq.	Jap.Freq.	Eng.Freq.
— Freq.Stage 1151 —					
会社	company	10.73	3,952	4,081	4,720
ライセンシー	licensee	10.47	2,436	2,521	2,715
— Freq.Stage 575 —					
販売店	distributor	9.55	1,471	1,562	1,679
契約品	product	9.26	2,511	2,996	3,127
売り手	seller	9.24	999	1,039	1,116
— Freq.Stage 287 —					
買い手	buyer	8.92	940	970	1,112
当事者	party	8.84	1276	1,394	1,584
書面	writing	8.39	754	860	858
条	article	8.34	778	837	955
— Freq.Stage 143 —					
ｂ	b	8.07	332	345	344
ａ	a	8.01	324	335	340
ＡＢＣ	ABC	7.99	354	362	388

Table 3.8 Samples of top 12 translation patterns “business”

corpus	English	(recall)	Japanese	(recall)
business	867	(39.1%)	1,005	(28.2%)
science	2,240	(33.1%)	2,359	(26.6%)
computer	1,922	(55.3%)	2,224	(46.3%)

Table 3.9 Numbers of Words Extracted and Recall Rates

better translation dictionary from parallel corpora. We believe that the proposed method gives results of good performance compared with previous related works.

Chapter 4

Translation Pattern Extraction with Language Resources

4.1 Introduction

The automatic translation pattern extraction which was described in chapter 3 greedily extracted pairs of Japanese and English word sequences based on their frequency of co-occurrence in parallel corpora . While this method realized a highly precise extraction of translation patterns only by using a parallel corpus and a morphological analyzer, it also had weaknesses.

- Because precision and coverage relate inversely, precision is maintained by restricting coverage of patterns.
- The process is computationally heavy, which limits the size of the parallel corpora it can handle.

This chapter first presents a practical translation pattern extraction method, which is based on that method, combined with manual confirmation and linguistic resources such as chunking information and translation dictionaries. With this method, translation patterns are accurately extracted from corpora as small as 500 sentences. Secondly, we modify this method to extract translation patterns incrementally, starting from a small part of a parallel corpus and gradually enlarging the scope of extraction. The modified method can deal with relatively large parallel corpora, and takes less computing time while maintaining a high precision.

The next section explains weakness of the basic algorithm[1] described in Chapter 3 , and presents our new method, which can employ chunking information and translation dictionaries and manual confirmation of extracted pairs stepwise. Section 4.3 describes each improvement in detail. Section 4.5 describes our experimental results, and Section 4.6 discusses them. Finally Section 4.7 concludes this chapter.

4.2 New Extraction Algorithm for using Linguistic Resources

In this section, we analyze the problems of our previous method, and describe our improvements.

4.2.1 Weaknesses of the Basic Algorithm

We analyzed the translation pattern results which was described in chapter 3. The analysis showed the following mistakes.

(a) The correspondence is partly correct. However, the correspondence includes one or more unnecessary words.

(b) The correspondence is not correct.

(c) The extraction results contain many ambiguous correspondences.

(a) is caused by mistakes in the choice of the extent of word sequences. Correspondences extracted by the basic algorithm often ignore the sentence structure, because arbitrary word sequences are paired. In order to resolve this problem, the improved method uses input sentences already divided in chunks. (b) comes from the absence of semantic information. We tackle it by accessing dictionaries and accepting information from a human operator. For (c), we introduce an idea of "*divergence sensitivity*" into standard definitions for similarity.

[1] Hereinafter, the algorithm described in Chapter 3 is called "basic algorithm".

Meanwhile, the large number of correspondence candidates also tends to slow down the extraction process. To avoid having to handle all correspondence candidates from the beginning, we also split the corpus into evenly sized parts, and add them incrementally to the database.

4.2.2 Improved Algorithm

The following shows the flow of the process to extract correspondences of Japanese and English word sequences. The step numbering of the following procedure corresponds to the flow of the basic algorithm described in Chapter 3. The changes from the basic algorithm are described in *italic.* The step (5)-3 only applies if a translation dictionary is provided, and the step (5)-4 applies if a human operator is available.

(1) Japanese and English texts are separately analyzed morphologically *and structurally.*

(1)-2 *(E, J) is partitioned into $(E_1, J_1), (E_2, J_2), \cdots, (E_n, J_n)$. The current partition number i is set to* 1.

(2)-1 *For each sentence pair (ES, JS) in (E, J) , the sets EWS and JWS of phrase-like word sequences included in a segment appearing in ES and JS are constructed.* The pair (EWS, JWS) is inserted in the sentence database. The total number of occurrences of each word sequence is also kept separately.

(2)-2 *If $i > 1$, the sets EWS and JWS are filtered as described in step (6), using the "Correspondence Dictionary" produced by partitions* 1 *to* $i - 1$.

(2)-3 The pairs (EWS, JWS) inserted in the sentence database. The total number of occurrences of each word sequence is also kept separately.

(3) An initial threshold value f_{min} for the minimum number of occurrences is chosen appropriately according to the database.

(4) For every pair of word sequences occurring more than f_{min} times, the total number of bilingual co-occurrences in the database is counted.

(5)-1 For each pair of bilingual word sequences, a similarity is calculated. The most plausible correspondences are then identified using the similarity values. *The approved correspondences are kept as correspondence candidates, but they are not directly registered in the "Correspondence Dictionary" yet.*

(5)-2 *A correspondence candidate* (ews, jws) *is registered in the "Correspondence Dictionary" if at least one word of* ews *and one word of* jws *are associated in the translation dictionary.*

(5)-3 *Even if the condition of (5)-2 is not satisfied, correspondence candidates that have already been approved twice in the same* f_{min} *level are registered in the "Correspondence Dictionary".*

(5)-4 *A human operator can also be employed to check that correspondences satisfying the condition of (5)-3 are really correct. Incorrect correspondences are remembered so that they will not be presented twice to the operator.*

(6) For each newly registered correspondence (ews, jws), and each pair (EWS, JWS) in the database such that EWS contains ews and JWS contains jws, all word sequences including ews (resp. jws) are removed from EWS (resp. JWS). The total number of occurrences for each word sequence is updated. The steps (4) through (6) are repeated until no new pair is approved.

(7)-1 *If* $i = n$ *and* $f_{min} \leq f_{end}$*, or* $i < n$ *and* $f_{min} \leq f_{merge}$*, where* f_{merge} *is a fixed value larger than* f_{end}*, then extraction for partition finishes, otherwise steps (4) to (9) are repeated with* f_{min} *decreased by* 1.

(7)-2 *If* $i = n$*, extraction finishes. Otherwise, steps (2) to (9) are repeated, after advancing* i *by* 1.

4.3 Details of the Improvements

Our main improvements are the following five points: combining with chunking information, using translation dictionaries, allowing manual confirmation, text-splitting, and the improved similarity.

4.3.1 Chunking

During the step (1) of our improved algorithm, sentences are not only analyzed morphologically, but also divided into syntactically related non-overlapping segments (chunks). Examples of chunks are noun phrases, verb phrases and preposition phrases. The improved method limits the extraction of word sequences to the sub-sequences of a chunk. It is hoped that this process contributes to the reduction of translation patterns extracted that ignore the sentence structure. The availability of chunking information is indicated by Yamamoto et al. also. [59].

4.3.2 Translation Dictionary

The goal of the basic algorithm was to extract highly accurate correspondences from a parallel corpus without using any linguistic resources. But in practical use, it is important to enhance the quality of extracted correspondences by using efficiently existing linguistic resources. We introduce a mechanism to refer to translation dictionaries during extraction. Yet, we must be cautious that excessive dependence on translation dictionaries should not prevent the extraction of correspondences between unknown words.

In step (5)-2, correspondences that can be related to a dictionary entry are registered first. A pair of word sequences (ews, jws) is related to a dictionary entry (ed, jd) if either $ed \subset ews \wedge jd \cap jws \neq \emptyset$ or $jd \subset jws \wedge ed \cap ews \neq \emptyset$, that is if one side of the entry matches a part of one word sequence, and the other side of the entry contains a part of the other word sequence. Functional words like prepositions in English, and particles in Japanese, are ignored.

Next, in step (6), word sequences including one side of these correspondences are excluded from extraction candidates. Only correspondences that do not conflict with such highly probable correspondences may pass this filter, and be extracted in the next pass in step (5)-3.

Though our method uses translation dictionaries to avoid improbable translations, the extraction of correspondences unrelated to any dictionary is only delayed. This allows extracting technical terms and new words that do not exist in translation dictionaries.

4.3.3 Manual confirmation

The safest approach to determine correct correspondence candidates is still to have a human operator check it manually. This is however a costly process and human intervention should be limited to cases that cannot be evaluated automatically. We limit candidates presented to the operator to those selected in (5)-3, as they are statistically possible, but lack corroboration. Moreover the list to check is sorted in decreasing order of similarity and in alphabetic order, to speed up the selection of correct correspondences. Correspondences refused once are remembered to avoid querying the operator again.

4.3.4 Text-splitting for Large Corpus

The text-splitting method works as follows. The corpus is partitioned into several parts in step (1)-2. The extraction is repeated in one part until the level f_{merge} is reached. The level f_{merge} should be chosen so as to guarantee that no incorrect correspondences are generated during this first phase.

When the extraction results run out, another part is added. Then, the extraction is repeated in step (2)-2. When all parts are used up, the extraction goes on for the whole corpus until $f_{min} = f_{end}$.

By this mechanism, many candidate word sequences can be eliminated in step (2)-2, on the basis of correspondences extracted from previous parts, so that the number of candidates to consider decreases in later parts.

4.4 Divergence Sensitive Model for Similarity

We experimented using two representative similarity measures, Dice coefficient and Log-likelihood which are described in Chapter 1. These measures are shown briefly again. In addition we define a modified measure "Divergence Sensitive Model", in order to cover the shortcomings of these measures.

Dice Coefficient

In the basic method described in Chapter 3, the similarity of a translation pair (w_J, w_E) is calculated by the weighted Dice Coefficient defined as

$$sim(w_J, w_E) = \log_2 f_{je} \cdot \frac{2f_{je}}{f_j + f_e}$$

where f_j and f_e are the numbers of occurrences of w_J and w_E in Japanese and English corpora respectively, and f_{je} is the number of co-occurrences of w_J and w_E. By choosing to approve only pairs such that $sim(w_J, w_E) \geq \log_2 f_{min}$, we can ensure that no word sequence occurring less than f_{min} times (i.e. not yet considered at this step) can yield a greater similarity score than $sim(w_J, w_E)$.

Log-Likelihood

As an alternative, we also consider the Log-Likelihood [12, 34], which is another measure for co-occurrence. Supposing f_{all} is the number of sentences included in the parallel corpus, it is defined as

$$\begin{aligned} sim(w_J, w_E) &= \phi(f_{je}) + \phi(f_e - f_{je}) + \phi(f_j - f_{je}) + \phi(f_{all} + f_{je} - f_e - f_j) \\ &\quad - \phi(f_e) - \phi(f_j) - \phi(f_{all} - f_j) - \phi(f_{all} - f_e) + \phi(f_{all}) \\ \phi(f) &= f \cdot \log f \end{aligned}$$

Correspondingly, we should not approve pairs for which the Log-Likelihood is less than
$sim(w_J, w_E) \geq \phi(f_{all}) - \phi(f_{all} - f_{min}) - \phi(f_{min})$.

Divergence Sensitive Model

The extraction results of the basic algorithm contain many ambiguous correspondences. In other words, a word sequence in one language may be associated to many word sequences in the other. If the parallel corpus contains many similar sentences, the ambiguity cannot be resolved only by co-occurrence, and the output contains two or more correspondences for one sequence. On the other hand, when the correspondence is one to one, it is more likely that the correspondence is correct.

On these grounds, we define a modified similarity sensitive to divergence as

$$dsim(w_J, w_E) = \frac{sim(w_J, w_E)}{\log_2(fw_{JE} + fw_{EJ})}$$

where fw_{JE} (resp. fw_{EJ}) is the number of translation patterns of w_J (resp. w_E) under the current f_{min}. The above definition is such that $dsim(w_J, w_E) = sim(w_J, w_E)$ when w_J is in one to one correspondence with w_E, and lower otherwise.

4.5 Various Comparative Experiments

4.5.1 Experimental Settings

We used English-Japanese parallel corpora that are automatically generated from comparable corpora of major newspapers, the Yomiuri Shimbun and the Daily Yomiuri [53]. These corpora have in average 24 words per English sentence and 27 words per Japanese sentence. We used respectively Chasen [31] and CaboCha [26] for Japanese morphological and dependency analysis. For English analysis, we use the Charniak parser [8]. The translation dictionary combines the English to Japanese and Japanese to English system dictionaries of the machine translation system we are developing [25], and has 507,110 entries in total.

We evaluate the results by their accuracy and coverage. Accuracy is evaluated according to the following 3 levels. The global accuracy is evaluated as the ratio of the number of "correct" and "correct+near miss" to all extracted patterns. They are described as "correct(correct+near miss)" in tables.

correct: the result can be used as an entry in a translation dictionary.

near miss: at most one spurious word on one side of the correspondence.

mistake: otherwise.

The coverage is calculated by the following formula in respectively English and Japanese corpus, and evaluated as the average of the two corpora.

No.		1.	2.	3.	4.	5.
method		basic	chunk	dict	hand	hand(d-Log)
f_{end} = 2	total	3,033	3,057	2,981	2,847	3,527
	correct	2,686	2,808	2,832	2,770	3,447
	near miss	110	83	82	64	64
	accuracy	88(92)	91(94)	95(97)	97(99)	98(100)
	coverage	79(12)	80(15)	80(15)	80(15)	82(13)
f_{end} = 1	total	16,784	16,276	10,276	7,274	6,250
	correct	6,766	7,293	6,996	6,821	5,993
	near miss	434	391	335	221	151
	accuracy	40(42)	44(47)	68(71)	93(96)	96(98)
	coverage	85(16)	86(19)	86(19)	85(19)	85(19)
	time	11h47m	4h15m	4h49m	6h58m[2]	3h03m

Table 4.1 Comparison between the basic algorithm

$$coverage(\%) = (1 - \frac{non\ extracted\ wnum}{wnum}) \cdot 100$$

The non-extracted words of a sentence are the content words (i.e. words with an independent meaning) that are not included in any extracted correspondence applicable to this sentence. "*non extracted wnum*" is the total number of nonextracted words for a whole (monolingual) corpus. Respectively "*wnum*" is the total number of content words in a corpus. Additionally we provide another evaluation of coverage, based on the same formula, where the number of words is to be understood as the number of distinct words rather than their total number of occurrences. They are indicated as "total coverage (distinct coverage)" in tables.

Our method was implemented using Perl 5.8, and executed on a 3 GHz Xeon processor with 2GB of memory.

4.5.2 Comparison with the basic algorithm

[2] It took 1 hour 57 minutes to check candidates manually. We discuss this in section 4.6.

No.		1.	2.	3.	4.
measure		Dice	d-Dice	Log	d-Log
f_{end} $= 2$	total	2,981	2,455	3,830	2,905
	accuracy	95(97)	98(99)	94(98)	97(99)
	coverage	80(15)	78(13)	83(16)	81(12)
f_{end} $= 1$	total	10,276	9,957	6,412	6,432
	correct	6,996	7,040	5,740	5,750
	near miss	335	333	222	231
	accuracy	68(71)	70(74)	89(92)	89(92)
	coverage	86(19)	86(19)	85(19)	85(19)
	time	4h49m	4h44m	1h04m	1h05m

Table 4.2 Comparison between similarity measures

Our first experiment compares between the basic method and improved methods using 8,000 sentences from the above-described parallel corpus. The results are shown in Table 4.1. All cases use the Dice coefficient. No.1 is the basic algorithm, No.2 adds chunking, No.3 adds a translation dictionary to No.2, and No.4 adds human checking to No.3.

Experiments with the basic algorithm described in Chapter 3 had been limited to $f_{end} = 2$ in order to obtain good accuracy. Indeed the original method has 88% accuracy then, which is a good enough result, but our method shows up to 97% accuracy under the same conditions, which is close to perfect. If we lower f_{end} to 1, the gap widens, the basic algorithm having only 40% accuracy, while the new method can go up to 93% (with human help).

What is interesting is that in spite of a decrease in the number of extracted correspondences, the coverage does not degrade. All these methods allow ambiguous extraction, all correspondences of identical score being extracted simultaneously. The experimental results show that our method can eliminate improbable correspondences by a filtering based on linguistic resources.

similarity	No. 1.	No. 2.	similarity	No. 3.	similarity	No. 4.
	Dice	d-Dice		Loglike		d-Logllike
>= 1	95(97)	98(99)	>= 18.58	94(98)	>= 18.58	97(99)
>= 0.8	94(97)	98(99)	>= 16	94(97)	>= 15	97(99)
>= 0.6	94(97)	97(99)	>= 15	93(97)	>= 12	96(98)
>= 0.4	92(95)	96(98)	>= 14	93(96)	>= 9	94(97)
>= 0.2	88(92)	93(96)	>= 13	92(96)	>= 7	93(96)
>= 0.1	82(86)	90(94)	>= 12	91(95)	>= 6	92(95)
>= 0.05	73(87)	87(90)	>= 11	90(94)	>= 5	90(94)
> 0	68(71)	70(74)	>= 0	89(93)	>= 0	89(93)

Table 4.3. Relations of the similarity and the accuracy on each similarity measures

4.5.3 Comparison between similarity Measures

In our next experiment we compare the efficiency of various similarity measures. The results are shown in Table 4.2. No.1 in Table 4.2 is identical with chunking+dictionary method(Table 4.1. No.3), but we change the measure. No.2 uses “divergence sensitive Dice Coefficient”, No.3 uses “Log-Likelihood”, and No.4 uses “divergence sensitive Log-Likelihood”. Comparing Dice and Log-Likelihood, the accuracy for $f_{end} = 2$ differs only slightly, but the accuracy for $f_{end} = 1$ differs noticeably. Nevertheless, the coverage of Dice and Log-Likelihood is similar.

We can conclude that Log-Likelihood is more effective than Dice when we want to extract in level $f_{end} = 1$.

In both cases we can also see that making the measure divergence-sensitive improves the accuracy of the results. Correct correspondences are increased, and incorrect ones either decrease or are stable. We have also observed that divergence sensitive measures make clearer the correlation between score and accuracy in Table 4.3.

Consequently, we think that divergence sensitiveness is effective in acquiring more acceptable results.

No.		1.	2.	3.	4.
type		begin	middle	end	letters
size		8,000			9,045
en	w-num*1	175,768	193,284	200,707	158,652
	dw-num*2	7,687	8,866	9,355	2,746
ja	w-num	209,709	221,119	228,949	222,737
	dw-num	9,853	12,012	13,000	3,052
f_{end}	accuracy	97(99)	96(99)	96(99)	92(95)
$= 2$	cover	81(13)	76(12)	75(11)	92(22)
	total	6,452	6,588	6,676	4,631
	correct	5,764	5,640	5,613	3,113
f_{end}	near miss	236	335	351	271
$= 1$	accuracy	89(93)	86(91)	84(89)	67(73)
	cover	85(19)	81(16)	78(15)	94(34)
	dic-acc	97[97*3]	96	96	92
	total	4,054	4,000	3,917	1,798
	no-acc	77[71*3]	70	67	52
	total	2,398	2,588	2,759	2,833
	time	1h06m	4h58m	7h58m	0h26m

*1: "w-num" is total number of words for a whole corpus

*2: "dw-num" is total number of distinct words for a whole corpus

*3 is the result of the experiment without refering the translation dictionary

Table 4.4 Effect of the corpus-quality

4.5.4 Quality of the corpus

Table 4.4 shows the effect of changing corpus-quality. We used the divergence-sensitive Log-Likelihood measure, like in Table 4.2 No.4. Since the sentences in the newspaper corpus [53] are sorted by order of confidence by the sentence alignment program, we selected sub-corpora of identical size from the beginning, the middle, and the end of the corpus to see the effects of the quality of the parallel corpus. Table 4.4 No.1, 2 and 3 are the respective results from newspaper. Additionally, we also used business contract letters [20], which are not treated

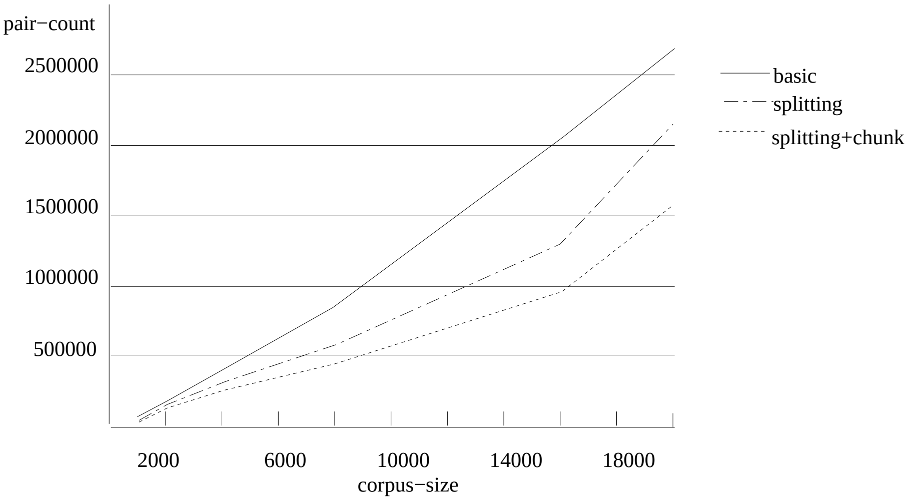


Figure 4.1 Number of candidates at $f_{merge} = 2$

with a sentence-alignment program but are aligned by hand. It is shown in Table 4.4 No.4.

Business contract letters is a technical document, and contains far fewer distinct words than the newspaper corpus. This negatively affects the accuracy, while increasing the coverage. Naturally, accuracy is very sensitive to the quality of the corpus. Here we consider “quality” for our own purpose, meaning that the translation is literal, and does not include many similar sentences. The result in Table 4.4 shows that “begin”, which offers the best quality, has the best accuracy, while “letters”, which is translated literally but has many similar sentences, has poor accuracy.

4.5.5 Verification of the Effect of Text-Splitting

Figure 4.1 shows the number of correspondence candidates in level $f_{min} = 2$, when experimenting several methods: “original” is the basic algorithm, “splitting” is the text-splitting method, and “splitting+chunk” is the text-splitting method combined with chunking. Both text-splitting and chunking contribute to reduce the number of candidates.

Table 4.5 shows the difference between using text-splitting or not using it.

No.		1.	2.	3.	4.
type		8,000	8,000-splitting	16,000	16,000-splitting
f_{end} $= 2$	accuracy	97(99)	97(99)	96(99)	96(98)
	coverage	81(13)	81(13)	81(12)	81(13)
f_{end} $= 1$	total	6,461	6,452	10,581	10,473
	correct	5,768	5,764	9,306	9,206
	near miss	231	236	439	437
	accuracy	89(93)	89(93)	88(92)	88(92)
	coverage	85(19)	85(19)	85(18)	85(18)
	time	01h00m	01h06m	09h04m	16h28m

Table 4.5 Effect of text-splitting

We found that text-splitting keeps the accuracy, and can speed up extraction to about twice as fast as the non-splitting case. While experimenting, we found that it is best to divide the corpus into four parts, and that splitting into too many parts causes a speed down, as many computations are repeated over and over.

4.5.6 Effect of the corpus size

Table 4.6 shows the effect of changing the corpus size. "Size" means the number of sentences from the newspaper corpus. We used the divergence-sensitive Log-Likelihood measure, like in Table 4.2 No.4. We selected sub-corpora of each size from the middle of the corpus to reduce the effects of the position in the newspaper corpus described in section 4.5.4. This table shows that coverage increases with the size of the corpus without loss of accuracy.

4.6 Discussion

In Table 4.7, we show examples of translation patterns extracted from the experiment of Table 4.2 No.4, using chunking, dictionary, and manual confirmation. This method can extract interesting correspondences. For example, though the

No.		1.	2.	3.	4.	5.	6.	7.
size		500	1,000	2,000	4,000	8,000	16,000	32,000
en	wn*1	12,231	24,381	48,681	97,720	193,284	38,5543	762,018
	dwn*2	2,451	3,481	4,872	6,665	8,866	11,641	15,002
ja	wn	14,206	27,818	55,754	111,787	221,119	441,867	880,525
	dwn	2,893	4,299	6,210	8,805	12,012	15,938	20,719
f_{end}	accuracy	94(96)	96(98)	96(99)	96(99)	96(99)	96(99)	96(98)
= 2	coverage	66(14)	68(13)	70(11)	73(12)	76(12)	79(12)	80(12)
	total	633	1,284	2,230	4,048	6,588	10,704	17,368
	correct	550	1,200	1,935	3,437	5,640	9,186	14,551
f_{end}	near miss	29	37	109	200	335	555	894
= 1	accuracy	87(91)	88(93)	87(92)	85(90)	86(91)	89(92)	84(89)
	coverage	72(18)	73(17)	76(16)	75(15)	81(16)	83(16)	84(17)
	time	0h01m	0h03m	0h06h	0h16m	4h58m	22h36m	52h35m

*1: "wn" is total number of words for a whole corpus

*2: "dwn" is total number of distinct words for a whole corpus

Table 4.6 Effect of the corpus-size

Japanese word sequence "**コモ/ン/ハウス**" was incorrectly analyzed morphologically[3] , it is correctly associated to "the common house" in the final results. The primary cause for near misses is incorrect or restrictive chunking. For example, the Japanese word "**冷戦**" means "the cold war", but it was incorrectly divided into "the cold" and "war" by chunking. We could recover these divided word sequences by referring to the original parallel sentences.

All examples of wrong correspondences are caused by mistaken application of the translation dictionary. The problem can be traced back to our rather lax notion of "related" entry, and to the overly broad coverage of used dictionaries. It would be safer to use a more adapted dictionary, and accept only matching entries. A next best method is to check more extensively for stop words (to be excluded from the lookup process). Alternative methods, like the usage of a

[3] As an output of the analysis, "**コモンハウス**" is correct, but it is divided in pieces like "**コモ/ン/ハウス**".

grade	Japanese	English	Score
correct	中 (center) ・ 東欧 (East Europe) 諸国 (countries)	the CEEs	11.48
	洗練 (refining) する (do) られる (passive)	sophisticated	7.61
	コモ (Como) ン (*unknown-word*) ハウス (house)	the common house	6.62
near miss	冷戦 (cold war)	war	48.45
	冷戦 (cold war)	the cold	44.14
	に従って (in accordance with)	in accordance	1.82
mistake	米国 (America)	Washington	32.12
	いまだに (still)	have yet	18.52
	休息 (rest) その他 (other) の (of)	other work	6.62

Table 4.7 Samples of corresponding word sequences

monolingual corpus, may help here too.

Our last concern is the workload required by manual confirmation. In the 4th experiment of Table 1, manually checking low-confidence candidates took about 2 hours for one operator. During this check, 1,900 correct correspondences could be extracted out of 5,500 candidates. An average time of 2 seconds by candidate was sufficient to dramatically improve the accuracy. In the 5th experiment of Table 1, the same experiment using Log-Likelihood combined with manual checking also achieved up to 96(98)% accuracy. The ratio of improvement to workload seems high enough to justify this semi-manual approach.

4.7 Summary

In this chapter, we proposed a practical method to extract translation patterns, which can be combined with manual confirmation and linguistic resources, such as word chunk information and translation dictionaries.

When the method was tested with a 8,000 sentence parallel corpus, it achieved 96% accuracy and 85% coverage, to compare to respectively 40% and 85% for the original algorithm. Without manual confirmation, it achieved 89% accuracy and 85% coverage.

Our method requires no language-dependent information such as cognate detection or transliteration; consequently we believe it is applicable to any language pair for which appropriate tools and data are available. One of the reasons that our approach is not automatic but semi-automatic is the prospect that the extraction results can be directly used as bilingual dictionaries in a machine translation system.

Chapter 5

Automatic Acquisition of Translation Rules for Structural Matching

5.1 Introduction

The translation knowledge extracted with the methods described in Chapters 3 and 4 produces translation patterns composed of fixed word sequences, such as technical terms or compound nouns. However, these methods cannot acquire more advanced translation patterns for phrases made of discontinuous word sequences, or translation rules for word selection.

This chapter presents a method for automatic extraction of more advanced translation rules from a parallel corpus by applying structural matching. In this chapter, translation rules are word selection rules and idiomatic translation rules composed of discrete words, which require information about correspondences between dependency structures.

The acquisition process applies the similarity of word-level pairs obtained from a machine readable bilingual dictionary and the one obtained statistically from parallel corpora. Pairs of translation examples are syntactically analyzed and are represented as disjunctive dependency graphs. Structural matching is done based on the similarity measure between subgraphs that is defined from the similarity between word pairs. Translation rules are constructed from matching results, by

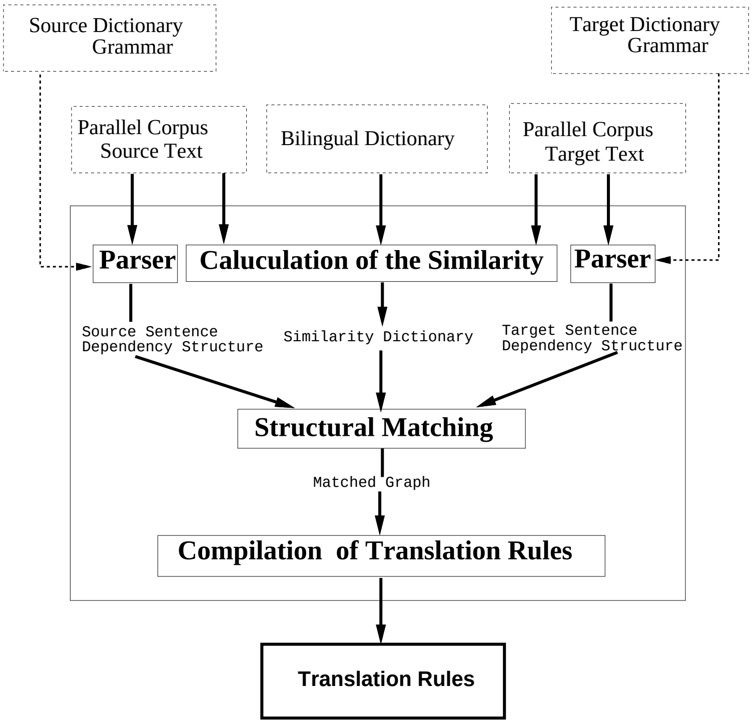


Figure 5.1 The flow of translation rules acquisition

using a thesaurus as a semantic classification. Two types of translation rules are then acquired according to the types of the matching results.

The organization of this chapter is as follows. In Section 5.2, we present a method for acquiring translation rules based on structural matching of parallel sentences. In Section 5.3, we present our experimental results. Additionally in Section 5.4, we actually apply the extracted translation rules to a simple LFG-based machine translation. In Section 5.5 we discuss the results of these experiments.

5.2 Acquisition Process for Structural Matching

Figure 5.1 shows the flow of the acquisition translation rules. Although the method is language independent, we deal with parallel texts of Japanese and English as we use a Japanese and English parallel corpus for the experiments. We assume the following three types of resources (surrounded by rectangles with broken line in Figure 5.1):

- Parallel corpora of the source and target languages.
- Grammars and dictionaries of the source and target languages.
- Machine readable bilingual dictionary.

The automatic acquisition of translation rules is composed of the following three processes:

Calculation of the Similarities Calculation of the similarities of word pairs of the source and target languages based on their co-occurrence frequencies in the parallel corpus.

Structural Matching Structural matching of the dependency structures obtained by parsing parallel sentences.

Acquisition of translation rules Acquisition of translation rules based on the structurally matched results.

We focus on a bilingual corpus of Japanese and English, and assume that sentence-level alignment has been done on the corpus. In case they are not aligned, we could have them aligned by one the sentence alignment algorithm described in Section 1.2.2 [7, 14, 24, 54].

5.2.1 Calculation of the Similarities

We define the similarity of a pair of Japanese and English words by numerical values between 0 and 1. We use the following two resources to obtain the similarity between a Japanese and English word pair:

- machine readable bilingual dictionaries
- parallel corpora

For the former, we assign a similarity of 1 to the translation pairs obtained from the machine readable dictionary. For pairs extracted from the parallel corpora, we use the modified similarity measure of Dice coefficient[43] described in Section 1.2.2. We preprocessed the corpus by analyzing them morphologically to obtain the base form of the words.

The similarity of a Japanese and English content word pair, is defined by

$$sim1(\langle w_J, w_E \rangle) = \frac{f_{je}}{f_j}$$

$$sim2(\langle w_J, w_E \rangle) = \frac{f_{je}}{f_e}$$

$$sim3(\langle w_J, w_E \rangle) = \frac{2f_{je}}{f_j + f_e}$$

where f_j and f_e are respectively the total numbers of the occurrences of the Japanese word w_J and English word w_E, and f_{je} means the total number of co-occurrence of w_J and w_E, that is, the number of occurrences they appear in the corresponding sentences.

We predetermine a following threshold value of the occurrences and the similarity, and select word pairs which have a greater similarity than the threshold. x is the threshold for selecting[1] .

$$f_{je} \geq 2 \bigwedge sim3(\langle w_J, w_E \rangle) \geq x$$

The similarity of a word pair is the maximum of $sim1$ and $sim2$. The reason why we do not apply simply $sim3$ is that $sim3$ may be low when a word in one language has diverse translations in the other language. For instance, in case of the similarity between an English word “provide” and the Japanese word ” 与える (*ataeru*)”, if “provide” is always translated to ” 与える (*ataeru*)”, it is desirable that the similarity between “provide” and ” 与える (*ataeru*)” gives a high value regardless of the diverse translations of ” 与える (*ataeru*)” (ex. “give”, “assign”, “grant”, etc...). In this case the similarity $sim2$ is used.

[1] The value of threshold depends on types of the corpus. For further description, see Section 5.3.2

5.2.2 Representation by Dependency Structure

Corresponding Japanese and English sentences in the parallel corpus are parsed with LFG-like grammars, resulting in feature structures. We do not use any semantic information in the current implementation. Only syntactic information is used for the analysis. When a sentence involves syntactic ambiguity, the result is represented as a disjunctive feature structure. This ambiguity is solved through the structural matching process described in Section 5.2.3.

A feature structure can be regarded as a directed acyclic graph (DAG). In the subsequent process of structural matching, we use the part of the DAG that relates with content words (such as nouns, verbs, adjectives and adverbs). The resulting DAG represents the dependency structure of the content words in the sentence.

Figure 5.2 shows examples of dependency structures extracted by this method. The upper arrow in a dependency structure indicates a dependency relation, and the term on the right of an upper arrow shows the case name. The *OR* in the English dependency structure (1) introduces a disjunctive structure.

5.2.3 Structural Matching

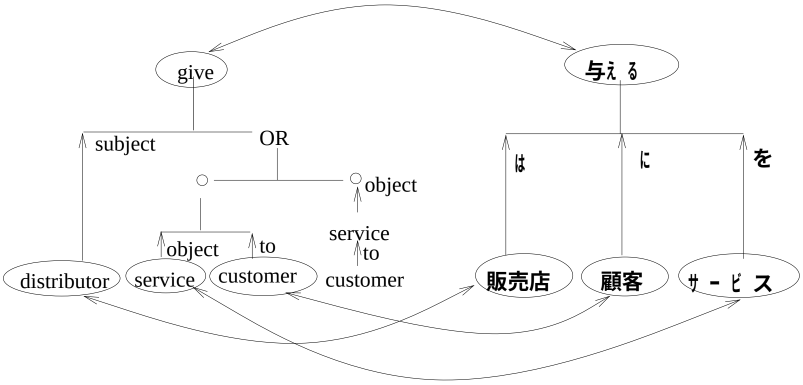
We start the matching process with a pair of disjunctive dependency graphs of Japanese and English sentences and find out the most plausible graph matching between them. The similarity of word pairs is extended to the similarity of subgraphs in the dependency structures. Though the basic definition and algorithm is taken from [30], which is reviewed in Section 1.2.4, we substantially redefine the similarity measure of words and subgraphs.

- We use the similarities described in Section 5.2.1 while Matsumoto et al. used integer numbers from 1 to 6 calculated from an existing thesaurus and a translation dictionary for their similarity measure.

- Matsumoto et. al specified a similarity which is inversely proportional to the size of the subgraph. But we rather pre-determine a threshold value [2] for the similarity. If a pair of words has a greater similarity than the

[2] In our experiment, the threshold was set at 0.15.

(1) English: Distributor gives service to customer.
Japanese: **販売店-は 顧客-に サービス-を 与える．**
(*hanbaiten-ha kokyoku-ni sabisu-wo ataeru*)
Best score = 3.945158



(2) English: Company compensates agent.
Japanese: **会社-は 代理店-に 報酬-を 与える．**
(*kaisha-ha dairiten-ni hoshu-wo ataeru*)
Best score = 1.55

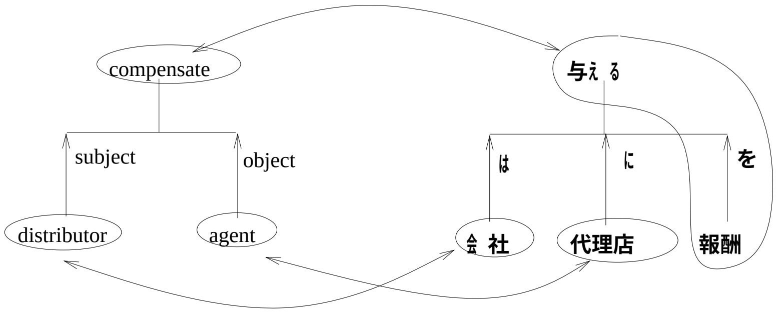


Figure 5.2 Results of structural matching

threshold, they are considered similar. On the contrary, if the similarity is less than the threshold, they are considered dissimilar, and can only be matched as part of a larger graph.

These changes have the following effect.

私は 母に めんどうをかけた. (*watashi-ha haha-ni mendou-wo kaketa*)
I gave my mother trouble.

The above example uses the idiomatic expression "めんどうを(*mendou-wo*)かける (*kakeru*) /give someone trouble". As our matching method prevents smaller subgraphs which have a low similarity from matching, "かける (*kakeru*)/give" and "めんどう (*mendou*)/trouble" are not matched individually, the similarity of each pair being low. Thus we can acquire correctly "[1]が (*ga*) [2]に (*ni*) めんどうを (*mendou-wo*)かける (*kakeru*)/[1] give [2] trouble".

Subgraph Similarity

The similarity between two subgraphs is defined by the sets of content words appearing in them. Let s and t be subgraphs of the dependency graphs of Japanese and English sentences, and V_s and V_t be the sets of content words in s and t. Assume without loss of generality that the size of V_s and V_t is not larger than that of V_t. We interchange V_s and V_t when $|V_s| > |V_t|$.

Taking an arbitrary injection p from V_s and V_t ($p : V_s \to V_t$)

$$D_p = \Big\{\langle a, p(a)\rangle \mid a \in A\Big\}$$

Then, we define the average similarity of words in the subgraphs as the average of the maximal value of possible matching between words in the subgraphs:

$$AverageSim = \frac{\max_p \left(\sum_{d \in D_p} sim(d)\right)}{\mid A \mid}$$

In the overall matching of the dependency graphs of corresponding Japanese and English sentences, words with high similarity should get a fine grained matching while words with low similarity should not get matched. In case a pair of words with low similarity are to be matched due to syntactic constraints, they should better get matched coarsely. The reason is that while a pair of words with low similarity are rarely matched at the word level they may be matched at the phrasal level, especially in idiomatic expressions.

To define the similarity between subgraphs, we predetermine a threshold value of similarity. Then, a pair of words with a similarity greater than the threshold is considered really similar, and one with a similarity less than the threshold is considered dissimilar. By referring to the threshold value as Th, the similarity between the subgraphs s and t is defined by the following expression:

$$f(s,t) = \left(Th + \frac{AverageSim - Th}{\mid A \mid + \mid B \mid -1}\right) \cdot \mid A \mid$$

A branch-and-bound algorithm is employed for searching the subgraph matching that gives the highest similarity value for the whole tree.

Figure 5.2 (1) shows examples of dependency structures and the results of the structural matching, in which the corresponding pairs are shown by arrows. Here the *best score* is the total similarity of the most plausible matching graph, produced by the algorithm. In case the total similarity of a graph is more than some threshold, the matching is regarded as successful. A threshold of 0.15 was found to give the best results.

5.2.4 Acquisition of Translation Rules

After accumulating structurally matched translation examples, the acquisition of translation rules is done through the following steps. We use categories from a thesaurus to denote the constraints on the instantiation of the acquired rules. Suppose we concentrate on a particular word or phrase in the source language graph that appears as a subgraph in matching graphs. We refer to the subgraph to be focused on as subgraph s.

1. Collect all the matched graphs that contain properly the subgraph s.

2. Extract the subgraph s and its children together with the corresponding parts of the target language graph. Some heuristics are applied in this process: corresponding pairs of pronouns are deleted, and zero personal pronouns in Japanese sentences are recovered.

3. The child elements are generalized to their categories in the thesaurus.

The system acquires two types of translation rules, word selection rules and translation templates, that represent word-level and phrase-level translation rules.

Word selection rules When the top subgraph consists of a single content word, we regard the corresponding subgraphs as possible translations of this word, and call it a word selection rule.

An example of word-level correspondence is shown in Figure 5.2 (1), "与える (*ataeru*)/give." In this case the child elements of "与える (*ataeru*)" indicate a condition on the applicability of this translate pattern.

Idiomatic translation rules When the top subgraph consists of more than one content word, we regard it as a phrasal expression, and call it a translation template.

Figure 5.2 (2) shows an example of phrase-level correspondence, "報酬-を (*houshuu-wo*) 与える (*ataeru*)/compensate".

Since we assume the translation is influenced by adjacent elements, i.e. the words that directly connect to the word in the subgraph, we generalize the information in the collected matches so as to identify the exact contexts in which the translation rule is applicable.

From the set of partial graphs sharing the same parent nodes, translation rules in the form of feature structures such as those in Figure 5.4, are obtained. This will be described in detail by the examples shown in Section 5.4.2.

5.3 Experiments and Results

We developed a system implementing the method described in Section 5.2, and performed the following two experiments.

- Calculation of word-level similarity between English and Japanese
- Acquisition of translation rules

The experiment of translation rules acquisition targets the Japanese verb "与える (*ataeru*)" which has many meanings and is difficult to translate.

5.3.1 Experimental Settings

The Japanese morphological analyzer used in this experiment is JUMAN[32], which knows about 120,000 Japanese morphologies. As an English morphological analyzer we used the machine translation system PENSÉE[3] which knows about 55,000 English morphologies.

We used SAX[33] as Japanese and English parser, and gave grammar rules described with DCG(Definite Clause Grammar) including 49 Japanese rules and 121 English rules.

We used the *Torihiki Jouken Hyougenhou Jiten*[20] composed of 36,560 sentences. *Torihiki Jouken Hyougenhou Jiten* is a collection of sample Japanese-English translated sentences for business contracts. We call this corpus “Business letters” here. All sentences including ” 与える (*ataeru*)” are divided into simple sentences manually, as the current structural matching system works only with simple declarative sentences.

The translation dictionary for calculating word-word similarities was a combination of EDICT 1994 [4] including 9,804 sentences and *Koudansha* Japanese-English dictionary [46] including 93,106 words.

We also used electronic versions of Bunrui-Goi-Hyo (BGH, a Japanese Thesaurus)[38] and Roget’s Thesaurus [42] for specifying the semantic classes of the child elements of the translated words.

The experimental process is as follows.

(1) Word-word similarities are calculated in two ways, with and without translation dictionary.

(2) All Japanese and English pair sentences including “与える (*ataeru*)” are extracted, and the dependency structures of the sentences are generated.

(3) Dependency structures of Japanese and English sentences generated in (2). are matched structurally. The threshold described in Section 5.2.3 is 0.15.

[3] PENSÉE is a trademark of Osaka Gas corporation, OGIS-RI, and Oki Electric Industry Co., Ltd.

[4] EDICT 1994 is obtainable through ftp via monu6.cc.monash.edu.au:pub/nihongo

Th	with dictionary		without dictionary	
	pair num	accuracy(%)	pair num	accuracy(%)
0.1	32,256	43	33,980	30
0.2	13,143	49	14,456	44
0.3	7,198	64	8,282	51
0.4	**4,643**	**72**	5,522	57
0.5	3,168	75	3,881	66
0.6	1,995	74	**2,543**	**69**
0.7	1,188	76	1,591	76
0.8	1,030	76	1,328	77
0.9	679	76	860	69
1.0	621	76	743	69
applied num	7,785(4,643+3,142)		2,543	
dictionary num	3,142(40%)		692(27%)	

Table 5.1 Statistics of word-word similarity (Business letters)

(4) Putting a subgraph including "**与える** (*ataeru*)" on above-mentioned subgraph s, two types of translation rules are acquired according to the above process.

In order to investigate domain-dependent translation rules, we additionally used 39,163 example sentences from *Koudansha* Japanese-English dictionary[46] and 18,656 sentences from a unix manual. We call the former "Dictionary examples" and the latter "Unix manual".

5.3.2 Experimental Results

Calculation of word pair similarities

Table 5.1 shows the results of the calculation of word-word similarity on "Business letters". "With dictonary" in the table means that a translation dictionary was used, and "without dictionary" means that only the parallel corpus was used. "Th" is the threshold given to $sim3$ described in Section 5.2.1. The "pair num"

English word	Japanese word	similarity	*sim*3
abridge	解する (*kaisuru*)	1	1
abridge	減殺 (*messatsu*)	1	0.8
abstract	抜粋 (*bassui*)	1	0.56
acceptance	検収 (*kenshu*)	0.9	0.67
accountant	公認会計士 (*kounin-kaikeishi*)	0.97	0.62
accountant	会計士 (*kaikeishi*)	1	0.69
administrator	管財人 (*kanzainin*)	1	–

Table 5.2. Examples of the bilingual dictionary with word-level similarity (Business letters)

is the number of extracted word pair greater than the threshold, and "accuracy" means the portion of correct results.

Though the number of the extracted word pairs does not include word pairs included which exists in the translation dictionary when the translation dictionary is used, it may include the word pairs when the translation dictionary is not used. The accuracy is the number of correct word pairs out of arbitrary chosen 100 word pairs.

In Table 5.1 we can find that the best accuracy is about 70%. By this result, we chose a threshold value going an accuracy of around 70%. The threshold of "with dictionary" is 0.4, and the one of "without dictionary" is 0.6.

The "applied num" means the number of pairs used in structural matching. The "dictionary num" means the number of pairs already existing in a translation dictionary, and the percentage of total is between parentheses.

Some examples of similarities obtained in the experiment are shown in Table 5.2. The "similarity" in this table means the maximum among *sim*1, *sim*2 and *sim*3, and the "–" for *sim*3 indicates pairs already existing in a translation dictionary. In Table 5.2, we find that although the values for *sim*3 of "accountant/公認会計士 (*kounin kaikeisi*)" and "accountant/会計士 (*kaikeishi*)" are low due to the ambiguity, we use high similarities for structural matching by applying *sim*2 and *sim*1.

This method remarkably can extract 2,543 pairs with an accuracy of about

Th	Dictionary examples		Unix manual	
	pair num	accuracy(%)	pair num	accuracy(%)
0.1	7,427	67	8,565	27
0.2	**3,381**	**83**	3,349	38
0.3	1,812	90	1,929	56
0.4	1,179	93	**1,241**	**68**
0.5	772	95	855	73
0.6	448	97	539	82
0.7	279	96	359	85
0.8	234	96	280	78
0.9	96	98	124	73
1.0	86	98	97	67
applied num	9,742(3,381+6,361)		2,607(1,241+1,320)	
dictionary num	6,361(65%)		1,320(52%)	

Table 5.3. Statistics of word-word similarity (Dictionary examples and Unix manual)

70% from a "Business letters" including 36,000 sentences, without a using any translation dictionary.

We get a number of domain specific terms about "Business letters", such as "agent/**代理店** (*dairiten*)" and "accountant/**会計士** (*kaikeishi*)," which are not found in ordinary bilingual dictionaries.

We calculated word pair similarities also in "Dictionary examples " and "Unix manual" to compare different domains. The results are shown in Table 5.3. Those results use a translation dictionary. We also find that our method can acquire many domain-dependent translations. For example, "administrator" is translated as "**管財人** (*kanzainin*)" in the "Business letters", and it is translated as"**システム管理者** (*sisutemukanrisha*)" in the "Unix manual".

In this table we find that we must define the threshold of the similarity used in structural matching depending on the corpus. For instance, dictionary examples have high accuracy (83%) even when the threshold is 0.2, because "Dictionary examples" are composed of simple sentences with little ambiguity. Conversely

	sentence	parsing	word-level	phrase-level
Business letters	427	416	253(5)	163(14)
Dictionary examples	48	40	27(0)	13(1)
Unix manual	218	185	153(13)	32(6)

*() indicates the number of incorrect results

Table 5.4 Statistics of matching results “与える (*ataeru*)”

“Business letters” and “Unix manual” have an accuracy of about 70% even when the threshold is 0.4, and the accuracy does not change much even if the threshold is higher. The reason is that “Business letters” and “Unix manual” include many similar sentences and expressions and have a number of complex and compound sentences which are composed of many words.

Since extracted pairs are used only as seeds for structural matching, we need not acquire perfect pairs. However, we could improve this method by applying the method described in Chapters 3 and 4 in the future.

Acquisition of Translation rules

We selected the Japanese verb ” 与える (*ataeru*)” which occurs frequently occurrence, and collected structurally matched results in three types of documents. The result is in Table 5.4. For example, there were 427 sentences including Japanese verb “与える (*ataeru*)”. in “Business letters”, and 416 sentences were successfully parsed. Of that number, 253 sentences gave a top subgraph with a single content word, and 163 sentences gave a top subgraph with more than one content word.

Table 5.4 shows that our method can acquire the results of structural matching with a high degree of accuracy in either corpus. We think that the integration of statistical and structural methods has a synergy effect, leading to higher accuracy.

To acquire word selection rules, the results are classified into groups according to the translated target words. A word selection rule is acquired from each target word by generalizing the child nouns. The word selection rules for Japanese verb “与える (*ataeru*)” are summarized in Table 5.5. Tables 5.5 and 5.6 show all the rules for English words occurring more than once.

English (occurrence)	が (*ga*)	を (*wo*)	に (*ni*
give(176)	[仕手 (player)] [位置 (position)] [終始 (start end)] [範囲 (area)] [整数 (integer number)] [一二三 (number)] [専門職 (specialist)] [社寺 (temple)] [店 (store)] [同盟 (association)] [呈示 (display)]	[仕手 (player)] [存在 (substance)] [開始 (start)] [時機 (chance)] [過去 (past)] [点 (point)] [範囲 (area)] [数 (number)] [伝達 (communication)] [感情 (feeling)] [承認 (acceptance)] [方法 (method)] [言語 (language)][表 (table)] [権利 (rights)] [教育 (education)] [救護 (aid)][給与 (salary)] [本体 (body)][賛成 (approval)]	[様相 (aspect)] [時機 (chance)] [終始 (start end)] [範囲 (area)] [相手 (party)] [社寺 (temple)] [店 (store)] [同盟 (association)] [原理 (principle)] [仕手 (player)]
provide(14)	[仕手 (player)] [社寺 (temple)] [同盟 (association)]	[単位 (unit)][練習 (practice)] [証明 (proof)][呈示 (display)] [救護 (aid)]	[範囲 (area)] [社寺 (temple)] [店 (store)]
	[事項 (item)]		[同盟 (association)]
assign(11)	[程度 (degree)]	[意味 (meaning)]	
afford(10)	[仕手 (player)][点 (point)] [整数 (integer number)] [芸術 (art)]	[時機 (chance)][権利 (rights)] [救護 (aid)]	[被告 (defendant)] [同盟 (association)]
render(10)	[仕手 (player)] [社寺 (temple)] [同盟 (association)]	[力 (power)] [成員 (member)][解決 (solution)] [救護 (aid)]	[範囲 (area)]
furnish(6)	[仕手 (player)][社寺 (temple)] [店 (store)]	[感情 (feeling)][呈示 (display)] [救護 (aid)][製造 (production)]	[仕手 (player))][範囲 (are)] [社寺 (temple)]
confer(5)	[店 (store)]	[権利 (rights)]	
grant(4)	[事務所 (office)]	[貸借 (lease)]	
grant(4)	[社寺 (temple)]	[承認 (acceptance)]	
extend(3)	[範囲 (area)]	[救護 (aid)]	[仕手 (player)]
issue(3)	[店 (store)]	[命令 (order)]	[相手 (party)]
authorize(2)			[専門職 (specialist)]

Table 5.5 Word selection rules for "与える (*ataeru*)" (Business letters)

For instance, the table specifies that "与える (*ataeru*)" is translated into "give" when its subject is one of the semantic classes, *player*, *position*, *start end* and so on, and its object is one of the classes *player*, *substance*, *start* and so on.

Phrasal translation rules are treated in the same way. Such examples of "与え

occurrence	translation pattern rules (upper:Japanese pattern, lower:Englsh pattern)
41	[1:[名 (name)][変化 (change)][開始 (start)]] が [2:[注意 (attention)][価値 (value)][権利 (rights)][性質 (character)][義務 (duty)][能力 (ability)]] に (悪)* 影響を 与える [1] affect (adversely)* [2]
13	[1:[同盟 (association)][事務所 (office)][仕手 (player)]] が [2:[事務所 (office)][仕手 (player)][店]] に { 損害 /害 } を 与えない [1] hold harmless [2]
8	[1:[事務所 (office)][仕手 (player)][相手 (party)][役員 (officer)]] が [2:[物品 (goods)][貨幣 (money)][因果 (cause and effect)]] に { 権利/権能 } を 与える [1] entitle [2]
5	[1:[事務所 (office)][仕手 (player)][店 (store)] が [2:[[専門職 (specialist)][事務所 (office)]] に 権限を 与える [1] authorize [2]
4	[1:[店 (store)][役員 (officer)] が [2:[事務所 (office)]] に 助言を 与える [1] advise [2]
2	[1:[仕手 (player)]] が [2:[因果 (cause and effect)]] に 同意を 与える [1] assent to [2]
2	[1:[事務所 (office)]] が [2:[店 (store)]] に 同意を 与える [1] give a consent to [2]

()* indicates an arbitrary word. {x/y} indicates that both x word and y word can be applied.

Table 5.6 Translation patterns for "与える (*ataeru*)" (Business letters)

る (*ataeru*)" are shown in the lower part of Table 5.6. For instance, the Japanese phrase " X が (*ga*) Y に (*ni*) 報酬-を (*houshuu-wo*) 与える (*ataeru*)" is translated into " X compensate Y ", if X and Y satisfy the semantic constraints attached to the translation rule.

Table 5.7 shows examples of the translation "与える (*ataeru*)" depending on the parallel corpus. In order to compare with an existing translation dictionary we show also the translations of "与える (*ataeru*)" in "*Koudansha* Japanese-English Dictionary". In Table 5.7, we find some translations of "与える (ataeru)" which we cannot find in the *Kodansha* dictionary and which are domain-specific. For example, "render" (meaning "give a bill") in "Business letters", and "display" (meaning "give a display") in "Unix manual" are typical of domain-specific words.

If we give translation rules generated from the results described in Table 5.5 to a machine translation system, the machine translation system will then be able to translate "与える (*ataeru*)" correctly according to its local context.

In Table 5.6 we can see translation patterns which we cannot find in existing

	extracted translations*		
Koudansha	Dictionary examples	Business letters	Unix manual
give	give	give	give
provide	provide	provide	provide
cause	cause	assign	affect
allow	allow	afford	have
assign	assign	render	deny
bestow	bestow	furnish	include
serve	have	confer	display
disposal	inspire	grant	seed
	do	offer	grant
	leave	issue	
	tell	extend	
	grant	authorize	

*:descending order of the number of occurrences

Table 5.7. Examples of translations of "与える (*ataeru*)" classified by type of parallel corpus

translation dictionaries. For instance, "影響を (*eikyo-wo*) 与える (*ataeru*)" corresponds to "affect", and "悪影響を (*akueikyo-wo*) 与える (*ataeru*)" corresponds to "affect adversely". Additionally the negative sentence "損害を (*songai-wo*) 与えない (*ataenai*)" corresponds with the affirmative sentence "hold harmless".

5.4 Application to a Machine Translation System

We present the overview of a machine translation system using the translation rules acquired with the above experiments.

This system achieves an integration of rule-based, example-based, and statistical-based methods, in that the preliminary mechanism follows transfer rule-based approach while the granularity of translation rules is determined by the sample

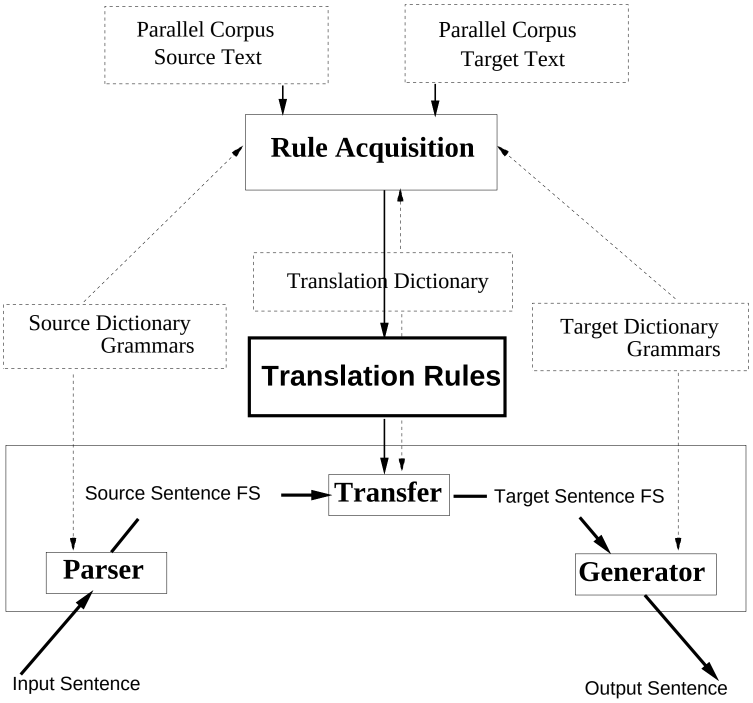


Figure 5.3 Architecture of the machine translation system

translated sentences using statistical information.

5.4.1 System Architecture

Figure 5.3 shows the configuration of the system, which comprises the following four parts.

RULE ACQUISITION This module acquires the translation rules based on the procedure shown in the preceding section. Translation rules for all the words appearing in the corpus are acquired in this process.

PARSER The input sentence is parsed with a LFG-like grammar and the dictionary, resulting in a feature structure of the source language sentence.

The grammar and the dictionary are the same as the ones that are used in the acquisition process. All the possible parses are obtained when the input sentence is syntactically ambiguous.

TRANSFER The feature structure is transferred into another feature structure of the target language by applying the translation rules.

In case no translation rule is applicable, the system use the bilingual dictionary as the default.

At present, phrase-level translation rules are assumed to have priority on word-level translation rules, and translation rules with more occurrence are assumed to have higher priority. In case the input sentence is ambiguous and is represented as a disjunctive feature structure, unifiable translation rules with higher priority are applied eagerly until the overall unification result is obtained.

GENERATOR Finally, the target sentence is generated from the feature structure of the target language.

Again, the grammar and the dictionary are the same as the ones used in the analysis of the target sample sentences.

The details of the procedure will be described in Section 5.4.2.

5.4.2 Translation Rules

The translation rules acquired in the experiments described in Section 5.3 are converted into the following data structure in our machine translation system.

```
tr_dict( index,
         source feature structure,
         target feature structure,
         condition).
```

index: The index word of the translation rule.

source feature structure: The source language feature structure.

```
(1) 与える
tr_dict(与える, [ pred:与える (verb), が:X, を:[pred:同意(noun)], に:Z ],
                [ pred:assent(verb), subj:X,  to:Z ],
                true ).
tr_dict(与える, [ pred:与える (verb), が:X, を:[pred:影響(noun)], に:Z ],
                [ pred:affect(verb), subj:X,  obj1:Z ],
                true ).
tr_dict(与える, [ pred:与える (verb), が:X,  を:Y,  に:Z ],
                [ pred:give(verb), subj:X,obj1:Y, obj2:Z ],
                ( checksem(X,[11000,11040,11600,...]),
                  checksem(Y,[11642,11910,13004,...]),
                  checksem(Z,[11000,11040,12630,...]) ) ).
tr_dict(与える, [ pred:与える (verb),  が:X,  を:Y,  に:Z ],
                [ pred:affect(verb), subj:X, obj1:Y, obj2:Z ],
                ( checksem(X,[11501,...]),
                  checksem(Y,[11112,...]),
                  checksem(Z,[13761,...]) ) ).
tr_dict(与える, [ pred:与える (verb) ],
                [ pred:give(verb) ],
                true ),

(2) 委託
tr_dict(委託, [ pred:[委託(noun), 者(suffix)] ],
              [ pred:consignor(noun) ],
               true ).
tr_dict(委託, [ pred:委託(noun) ],
              [ pred:reference(noun) ],
               true ).

(3) 条件
tr_dict(条件, [ pred:[支払い (noun), 条件(noun)] ],
              [ pred:[payment(noun),term(noun)] ],
               true ).
tr_dict(条件, [ pred:条件(noun) ],
              [ pred:condition(noun) ],
               true ).

(4) その
tr_dict(その, [ pred:その (determiner) ],
              [ detform:the ],
               true ).
```

Figure 5.4 Examples of translation rules

feature	meaning	feature	meaning	feature	meaning
pred	predicate	modif	modifier	が	*ga*-case
detform	determiner	to	to-case	は	*ha*-case
voice	voice	on	on-case	を	*wo*-case
subj	subject	in	in-case	に	*ni*-case
obj1	direct object	by	by-case	の	*no*-case
obj2	indirect object			で	*de*-case

Table 5.8 Feature names used in the translation rules

target feature structure: The target language feature structure.

condition: The semantic condition for the translation rule. It is described by sets of semantic classes for the variables appearing in the source feature structure.

In the condition, `checksem/2` is a prolog predicate for checking the semantic classes of the variables (semantics classes are expressed by the class numbers in the thesaurus).

Identifying the most suitable semantic classes in the thesaurus is by no means an easy task. In the current implementation, we use the semantic classes at the lowest level of the Japanese thesaurus BGH[38], which has 6 layers.

This leads the description of the semantic condition to be a list of lowest level semantic classes. Therefore, in our current implementation the translation rules compiled with few translation examples are far from complete. Some of the final form of translation rules are shown in Figure 5.4. The meaning of the feature names are summarized in Table 5.8.

5.4.3 The translation procedure

We now describe the translation process with a sample translation of the following Japanese sentence:

Japanese: 委託者が その 条件に 同意を与える．

(*itakusha-ga sonojouken-ni doui-wo ataeru*)

In the transfer process, translation rules are applied to the feature structures of the input sentence in a top-down manner. The translation starts from the root node and proceeds recursively in the feature structures.

1. The parsing process of the source sentence produces the following feature structure.

   ```
   [ pred:与える (verb)),
      が : [ pred: [ 委託 (noun), 者 (suffix) ] ],
      を: [ pred: 同意 (noun) ],
      に : [ pred: 条件 (noun),
            mnp: [ pred: その (determiner) ] ],
      tense: present ]
   ```

2. In the transfer module, the top-most content word "与える" is used as the index word and the translation rules with this index word are regarded as applicable rules. Out of the translation rules in Figure 5.4 (1), the final rule covers the largest part in the above feature structure. It then produces the following feature structure.

   ```
   [ pred:assent(verb),
     subj: [ pred:[委託 (noun), 者 (suffix)] ],
     to(prep): [ mnp:[pred:その (determiner) ],
                 pred: 条件 (noun) ],
     tense: present ]
   ```

3. The residual parts recursively undergo the transfer process as far as any unification failure does not stop the process. In a case of unification failure other possibilities are sought by backtracking.

   The current example produces the following final feature structure.

   ```
   [ pred: assent(verb),
     subj: [ pred:consignor(noun) ],
     to(prep): [ pred:condition(noun),
                 det:[detform:the(determiner)]],
     tense: present ]
   ```

Japanese: **会社-が 販売店-に 全て-の 権利-を 与える**

```
tr_dict(与える,
        [ pred: 与える (verb),
          ga: X,
          wo: Y,
          ni: Z ],
        [ pred: confer(verb),
          subj: X,
          obj1: Y,
          on: Z ],
        ( checksem(X,[12630,12650]),
          checksem(Y,[11343]) ) ).
```

English: A company confers all rights on a distributor.

Japanese: **会社-は 代理店-に 報酬-を 与える**

```
tr_dict(与える,
        [ pred: 与える (verb),
          ga: X,
          wo: [pred:houshu(noun)],
          ni: Z ],
        [ pred: compensate(verb),
          subj: X,
          obj1: Y ],
        ( checksem(X,[12650]),
          checksem(Y,[12630]) ) ).
```

English: A company compensates an agent.

Figure 5.5 Sample translations

4. The generator produces the following target sentence. The generator employs the semantic-head-driven generation algorithm[45].

   English: Consignor assents to the condition.

Figure 5.5 shows 2 other examples of translations.

No.	cause of failure	sentences
(1)	Correspondance between a word and a subgraph	7
(2)	Word pair similarities	7
(3)	Dependency structures	4
(4)	Correspondance between a content word and a function word	1
	total	19

Table 5.9 The causes of failures in structural matching

5.5 Discussion

5.5.1 Causes for Failure in Structural Matching

The accuracy of structural matching influences the performance of the machine translation system. Our method failed to match structurally for 19 sentences out of 416 in “Business letters”. This section analyzes the causes for the failure of structural matching. We summarized these causes and the number of error sentences in Table 5.9. In the following we give details for each item.

(1) It is difficult to match a word with a subgraph composed of several words.

> Xは **他の 者に 供給する 承認を与える 権利を留保する**
> (*x-ha hoka-no mono-ni kyokyusuru shounin-wo ataeru kenri-wo ryuhosuru*)
> X reserves the right to authorize others to supply.

From the above example our method acquires the incorrect result of “**他の** (*hoka-no*)/others” and “**者に** (*mono-ni*) **承認を** (*shounin-wo*) **与える** (*ataeru*)/authorize”, where as “**他の 者に** (*hoka-no mono-ni*)” should not be divided.

The cause of the failure is as follows: the similarity of ” **承認を** (*shounin-wo*) **与える** (*ataeru*)/authorize” is lower than the similarity of “**他の** (*hoka-no*)/others”, with the result that ” **者に** (*mono-ni*)” is merged into a subgraph with “**承認を** (*shounin-wo*) **与える** (*ataeru*)/authorize” incorrectly.

A similar failure is found for the compound noun “**試運転** (*shiunten*)/test run”, where “test” and “run” are divided incorrectly.

We think that we can solve this problem by applying not only word similarity but also the similarity of dependency relations through structural matching. For instance, in the above example “others” and “authorize” are in *object* case relation. According to this relation, the dependency relation similarity of ” に (*ni*):(*object*)” is higher than the one of “の (*no*):(*object*)”, and ” 他の者(*hoka-no mono*)に (*ni*)” shall correspond to “other(*object*)” correctly.

(2) Necessary word pair similarities for a structural matching could not be acquired.

This failure is caused by a defect in the word pair similarity measure we applied. For instance, the word pair similarity measure could not acquire pairs of “増加(*zouka*)/extra” and “援助(*enjo*)/guidance” because of the polysemy. Structural matching can cover the shortcomings of the word pair similarity measure. However, when the word pair similarity is insufficient or the dependency structures are complicated, it fails.

Improving the word pair similarity measure should solve this problem. We argue this improvement in Section 6.3 again.

(3) Extracted dependency structures were not adequate.

This failure is caused by incorrect parsing. We must improve the grammars and dictionaries applied in parsing. Alternatively it would be effective to use a statistical dependency parser such as CaboCha[26] or the Charniak parser[8].

(4) A content word corresponds to a function word.

> Xはリードタイムを与えてYに注文を出す．
> X places any order with Y with lead time.

In the above example, “与えて (*ataete*)” corresponds actually to the function word “with” actually. However “with” functions as a case relation of “time” in the dependency structure. In consequence “与えて (*ataete*)” is merged with “出す (*dasu*)”, and ” 与えて (*ataete*)出す *dasu*)” corresponds to “place” incorrectly.

We think to apply surface word order in a sentence to calculation of the similarity for subgraph. Concretely adjacent surface words should give increased priority to a subgraph. For the above example, the possible subgraphs including ”与えて (*ataete*)” are “タイムを(*taimu-wo*) 与えて (*ataete*)” and “与えて (*ataete*) 出す (*dasu*)”. If the surface word order in the sentence were to be considered, “ タイムを(*taimu-wo*) 与えて (*ataete*)” would have priority, and we could acquire the pair ”タイムを(*taimu-wo*) 与えて (*ataete*)/time(*with*)”.

5.5.2 Features of Our Machine Translation System

Our machine translation system has the following characteristics: The system uniformly deals with word selection rules such as “与える (*ataeru*)/confer” and phrasal translation rules such as “ X が (*ga*) Y に (*ni*) 報酬-を(*houshuu-wo*) 与える (*ataeru*)/ X compensate Y .” Even if there are no translation rules to apply, the system uses the bilingual dictionary as the default. Translation pairs appearing in the bilingual dictionary are regarded as word selection rules with no condition.

The larger the size of the parallel corpus, the more translation rules are acquired, thus ensuring gradual improvement in the quality of translation. Since all the translation rules are acquired from translation examples, manual compilation of translation rules is made minimal. Also, since the structural matching results used to obtain the translation rules are symmetric, both English-Japanese and Japanese-English translation rules can be acquired, making two-way translation possible.

Another important characteristic is that ambiguities (ambiguous translations caused by multiple applicable translation rules and ambiguous structural analyses) are solved by raising the priority of the translation rules with more specific information. The frequency information of translation pairs is also used for deciding the priority among the translation options.

Since the parsing phase and the generation phase share the grammars and dictionaries that are used in the acquisition phase of the translation rules, contradiction among the parsing, generation, and translation rules does not occur.

On the other hand, the following issues should be considered. The quality of the translation rules depends on the applicability of the used thesaurus. In our experiment, we used the lowest classes of the Japanese Thesaurus, BGH[38] for describing the conditions of rule selection, and for the English word selection rules, we used the lowest classes of the Roget’s Thesaurus. Suitability of the decision will be verified by larger-scale experiments.

Some inadmissible word selection and phrasal rules were acquired in the experiment. For example, the word selection rule, “ X[human] に (*ni*) Y[problem] を (*wo*) 唱える (*tonaeru*)[5] ” was paired with “make Y[problem] to X[human],” which is not a good translation rule. Rather, “make an objection to X[human]/ X[human] に (*ni*) 異議を (*igi-wo*) 唱える (*tonaeru*)” should be considered as an appropriate idiomatic expression. Idiomatic expressions like this example should be distinguished from normal word selection rules.

The proposed method is suitable to domains with comparatively formal sentences. Experiments with colloquial expressions reveal much more difficulties in acquiring “good” translation rules. Moreover, the current method cannot cope with expressions affected by contextual information.

The method should be augmented so as to deal with complex sentences. We do not think that a direct augmentation of the structure matching algorithm is applicable to complex sentences. Some two-level technique should be developed, one for finding appropriate decomposition of complex sentences and the other for finding the detailed matching of the decomposed components.

5.6 Summary

In this chapter, we proposed a method for acquiring translation rules based on the structural matching of parallel sentences. Parallel corpora are first used to acquire the similarity of word-level pairs between two language. Then, parallel sentences are structurally matched according to this similarity. Translation rules are constructed from the matched results by using a thesaurus as a semantic classification.

The experimental results using 36,000 sentences show that this method is useful for obtaining translation rules. Additionally we verified its usefulness with a machine translation system applying the extracted translation rules.

Generally, translation may differ depending on the domain. Our translation acquisition method provides an easy and effective way to develop translation rules according to the domain of the corpus, and the machine translation system also

[5] “唱える (*tonaeru*)” means advocate.

can easily adapt to any domain, provided that parallel corpora in that domain are accumulated.

Chapter 6

Conclusion

This chapter recapitulates this thesis, discusses recent related works, and presents future directions.

6.1 Summary

This thesis presents four works towards a practical corpus-based machine translation system, which can acquire translation knowledge from large parallel corpora and translate using this knowledge as translation dictionary.

The first work proposes pattern-based machine translation allowing complex patterns. The pattern-based machine translation system simplifies the handling of features by allowing sharing constraints between non-terminal symbols, and implementing an automated scheme of feature inheritance between syntactic classes. To avoid conflicts inherent to the pattern-based approach the system has priority control between patterns and between dictionaries. This approach proved its scalability in our web-based collaborative translation environment.

The second work proposes a method finding correspondences between word sequences greedily in aligned parallel corpora. Translation candidates of word sequences are evaluated by a similarity measure between the sequences defined by co-occurrence frequency and independent frequency of the word sequence. The similarity measure is an extension of Dice coefficient. A greedy method with gradual threshold lowering is proposed for getting a high quality translation dictionary. The method was tested with parallel corpora of three distinct domains and achieved over 80% accuracy.

The third work uses linguistic resources and manual confirmation to make the second work practical. Additionally we improve performance for large corpora by extracting gradually from smaller slices of the corpus, and using the extracted patterns to eliminate less probable patterns in following slices. This method was tested with a 8,000 sentence parallel corpus, and achieved an accuracy of 96% for a coverage of 85%, to compare to an accuracy of 40% for a coverage of 79% with the original algorithm of the second work. While accuracy still depends on the quality of the corpus, its size does not matter. As a result we were able to acquire results of high accuracy for large corpora.

The fourth work proposes a method for automatic acquisition of translation rules from a parallel corpus. The acquisition is composed of mainly three process; calculation of the similarities of word pairs obtained statistically from parallel corpora, structural matching of the dependency structures obtained from parsed parallel sentences, and acquisition of translation rules based on the structural matched results. Translation rules are acquired according to the types of the matching results. The rule types are word selection rules, and fused or idiomatic translation patterns. Experimental results using three kinds of Japanese and English parallel corpora for practical use show that this method performs very well at obtaining rules and patterns for machine translation.

6.2 Recent Related Works

This section picks up some recent studies of corpus-based machine translation and translation dictionary acquisition. First we describe phrase alignment and structural alignment techniques for the translation dictionary acquisition. Next we focus on two works applying the results of structural alignment to machine translation, which are similar with our approach and goal.

6.2.1 Phrase Alignment

There are two approaches to automatically extract translation patterns from parallel corpora: one is to look for the translations of a predetermined set of phrases

	researches with heuristics: heuristics types	researches without heuristics
asymmetric	Moore(2003): Positional, Capitarized Al-Onaizan et.al.(2002): MRD Smadja et.al.(1993): POS, Positional	—
symmetric	**Kitamura(Chapter 4)**: MRD, Chunk, Confirm Kupiec(1993): POS Kumano et.al.(1994): MRD, POS Melamed(1995): Cognate, POS, Syntactic	**Kitamura(Chapter 3)** Haruno et.al.(1996)

Table 6.1 Classification of phrase alignment technique

of the source language in the sentences of a parallel corpus, and the other is to extract patterns as exhaustively as possible without such a predetermined set.

Moore[37] called the former “asymmetrical approach” and the later “symmetrical approach”. We can classify also according to whether heuristics are used or not. In Table 6.1, we categorize recent related works and representative past related works from these two perspectives.

In recent year researches for phrase alignment have favored the asymmetric approach. Moore[37] adopted the former approach, using a hierarchy of 3 learning models, each one refining the accuracy. In order to enhance the accuracy, his method requires sentences to be perfect translations, where all words have a translated counterpart, and uses language-dependent information such as capitalization and punctuation. It thus achieves good precision even extracting translation patterns that occur only once in a document. This method was targeted at extraction of technical terms in computer software manuals. Technical terms contain many unknown words, therefore existing linguistic knowledge such as dictionaries and parsing results do not help. For this kind of materials, good results can be obtained by combining superficial analysis with statistical methods, but the applicability of such an approach to more general contents is questionable. Our method effectively uses dictionaries and statistical information too, and is able to extract many proper nouns.

Al-Onaizan and Kevin[2] belong also to the first approach. Their method is again geared towards technical terms. They use a large monolingual corpus of data such as Web pages for the target language (English), and transliteration information for the source language (Arabic). Their approach to deal with two languages of different families, by using effectively existing linguistic information, is similar to ours. Some of their ideas could be incorporated into our method. For instance, when manually checking an expression, it could be searched on the Web to confirm whether it is correct, and transliteration of English words into Japanese could be used.

Meanwhile, our methods in Chapters 3 and 4 belong to the second approach. There are only few symmetrical approaches without heuristics, like the one described in Chapter 3. It is challenging to acquire translation patterns with very little linguistic information. One of these few researches is by Haruno et.al. [16]. But their method first prepares wordlists of both languages using another statistical method, and then extracts correspondences between these wordlists. Our method is remarkable for its ability to exhaustively extract patterns from parallel corpora directly. Kupiec [28] and Kumano [27] also predetermine sets of meaningful phrases of both source and target language.

Melamed [36] proposed a method similar to that of Chapter 4. Our method in Chapter 4 uses three kinds of information, chunking information, translation dictionaries and manual confirmation, to improve the accuracy of extracted correspondences. Correspondingly, Melamed carefully selects accurate translation pairs by filtering candidates using four kinds of information: translation dictionaries, part of speech, cognate detection and syntactic information. This looks similar to our approach, but the difference is that our method targets extraction of translation candidates consisting of multiple word sequences, while Melamed limits his target to single word correspondences. In that case, the process is not constrained by computing power when using linguistic resources, as there is only a limited number of combinations. When dealing with word sequences of arbitrary length, usage of language resources has to be balanced with computational complexity.

6.2.2 Structural Alignment

We summarize representative researches on structural alignment between Japanese and English in Table 6.2.

A pioneer work on acquiring translation rules from parallel corpora is presented in Kaji et.al. [22]. Sentences in both languages are analyzed according to

	Word Alignment	Structure Type	Matching Method
Kaji (1992)	MRD	PS (phrase structure)	bottom-up matching between PSs by MRD
Kitamura (Chapter 5)	MRD, statistics (with similarity)	DS (dependence structure)	top-down matching between DSs by on word similarities
Yamamoto (2000,2003)	—	DS	complete statistical method
Imamura (2002)	unspecified	partial parse tree	bottom-up matching between partial trees by word-link and syntactic categories
Aramaki (2003)	MRD	DS by *basic-phrase*	(1) matching between DSs by MRD (2) matching in remaining phrases

Table 6.2 Summary of representative structural alignment techniques

their phrase structure by the CYK method, producing a triangular matrix describing all the possible parses. Then the potential phrase structures are matched bottom-up, starting from dictionnary correspondences between individual words. Matched phrases are required at each level to contain only corresponding words, or words without correspondence, which is sufficient to resolve some ambiguities. Finally templates are extracted from matched sentences by abstracting some matched phrases as variables. Although this method seems to work well for template extraction, the absence of explicit dependencies in the phrase structure makes it difficult to generalize, as one cannot easily allow the insertion of extra phrases in a phrase structure. Another difference compared with ours is that they do not attempt to find correspondences between words that were not related by the dictionnary, i.e. the only knowledge they extract is in the form of rather large templates.

Imamura [18] refines Kaji's approach by considering word categories and various heuristics to improve the accuracy of structural matching. Like us, it uses similarities to account for potentially wrong or superfluous dictionnary entries. Using these it is able to find correspondences between “*remaining*” phrases, containing no word-level correspondences. However it stops at the phrase-level, and again does not try to extract new word-level correspondences.

Aramaki et.al. [4] use a more deterministic approach. First, sentences are parsed to non-ambiguous dependency structures. Then basic phrases are aligned bottom-up, starting from dictionary correspondences. For phrases containing no word-level correspondences, the dependency structures and various heuristics are applied. Again alignment stops at the phrase level.

Meanwhile our method is composed of two types of alignment, statistical word-level alignment and structural phrase-level alignment. The method of Yamamoto et.al. [58] is purely statistical. Their method is very similar to our greedy extracting method described in Chapter 3, though they extract structure-level correspondences.

Additionally they proposed an algorithm avoiding combinatorial explosion, based on the PrefixSpan algorithm of sequential pattern mining [57]. This method is very fast, and has allowed exhaustive translation pattern extraction from a parallel corpus including 150,000 sentences, while keeping a precision of 56-84%. Our method in Chapter 4 has successfully dealt with parallel corpora up to 32,000 sentences, but we have not conducted further experiments. Since increasing the number of sentences does not enhance but reduces the precision of our method, as shown in the experiments, dividing bilingual documents according to some possible indicators (e.g. topic area) and extracting separately shall bring out better results in terms of both precision and coverage.

Yamamoto's method can find correspondences through a single statistical framework. It is attractive because of its simplicity, as it does not require any particular measure for phrasal alignment. But in practice, grammatical information such as syntactic categories and some heuristics will have to be employed to cover the weakness of a purely statistical method.

We plan to integrate the method of Chapter 4 and the one of Chapter 5 in the near future. They have complementary abilities; the method of Chapter 4 first extracts small but precise phrase-level correspondences with their similarities, and next the method of Chapter 5 extracts structural correspondences using these similarities. We think that it is possible to extend the behavior of the similarity employed in Chapter 4 to that expected by the method of Chapter 5. In order to achieve this aim, we have to solve some problems described in the next Section.

6.2.3 Machine Translation with Translation Knowledge Acquisition

Imamura and Aramaki experimented with machine translation systems using translation knowledge extracted by their methods [3, 19]. The theme of the above two researches is solving the “knowledge access bottleneck” described in Section 1.2.1. Their experiment is highly suggestive in constructing our machine translation system.

Imamura [19] proposed a method using translation literalness to select appropriate rules from a great number of extracted translation rules. The Translation Correctness Rate (TCR) is defined as the measure of translation literalness. Based on the TCR, a bilingual sentence is divided into literal parts and the other parts.

From the literal parts, small translation rules which have loose conditions are extracted, and they are applicable to many input sentences. On the other hand, from non-literal parts, large translation rules as extracted with strict conditions. They evaluated the effects on their machine translation system, and about 8.6% of machine translation results were improved.

The approach described in Chapter 5 is similar to Imamura’s approach in the sense that our extraction method adjusts the granularity of structural matching based on the similarity of subgraphs. As a result translation rules extracted by our method already contain suitable conditions.

Imamura’s method decomposes a parallel sentence into many phrase-level translation rules, and then gives applicability conditions to each translation rules based on TCR. Meanwhile Aramaki et al.’s method has a database composed of Translation Examples (TEs) aligned at phrase-level. A phrase-level pair is called a Fragment Translation Example (FTE). When an input sentence is given, their system searches for fitting a set of FTEs based on both *monolingual similarity* and *translation confidence*, and generates the translation by combining the set of FTEs [3]. “monolingual similarity” is the similarity between an input sentence and an original sentence of TE. “translation confidence” is a measure of the degree of correlation of a translation unit (FTE). Their experiment on translation selection showed an accuracy of 82.7%, and demonstrated the basic feasibility of their approach.

We think that “monolingual similarity” is an important factor to select appropriate translation rules. We have a considering it for our pattern-based machine translation system. Translation rules extracted by our method could keep a reference to the original sentence which it was derived from. Then it would be possible to use it as contextual information to evaluate the appropriateness of this pattern when we use it to translate a sentence.

6.3 Future Work

This thesis proposes fundamental techniques for developing a practical corpus-based machine translation system, however we are yet to develop the complete system described in Figure 1.4.

In the near future we must integrate these works and develop a complete system. There are some interesting topics of research for realizing this system.

Removing various restrictions with structural matching. The structural matching algorithm described in Chapter 5 has the following restrictions.

1. The current structural matching system works only with simple declarative sentences. As a result we have to divide complex sentences into simple sentences manually.

2. Since the unit of similarity which this algorithm uses is not the phrase but the word, this algorithm cannot apply directly the similarity of word sequences described in Chapters 3 and 4.

We must improve this structural matching algorithm so that this algorithm can use directly complex sentences and word sequence similarity. Specifically it needs the following improvements.

- After deleting a bare possible dependency structure by applying a statistical dependency parser such as Charniak parser[8], structural matching is processed.

- Structural matching proceeds based on correspondence between word sequence pairs extracted by statistical methods described in Chapters 3 and 4.

These improvements should prevent combinatorial explosion so that the above restrictions could be removed. Additionally these improvements will give more accurate results for structural matching because the statistical method of Chapter 4 is highly accurate.

Diversification of the type of translation rules extracted by the structural matching method. Though our structural matching method can extract idiomatic translation patterns and word selection rules for some words, it cannot extract various types of rules regarding arbitrary words as it cannot determine which part of the dependency structure is useful. We could extract various types of translation knowledge by integrating both the statistical and structural matching methods. Useful matching parts might be selected by applying statistical and structural information extracted with our techniques.

Reuse of the result of post-edit by users. Users almost always post-edit the results of a machine translation system. Our system can extract a translation dictionary from the original sentences and modified target sentences according to the method described in Chapter 3 and 4. However it is more efficient if our system detects only the modified parts in a target sentence, and acquires translation patterns only for the modified parts.

We will apply the method of structural matching for detecting the modified part in a target sentence. The new method could match structurally between target sentences before and after modification, and detect structural-level differences between them. Then it would extract the structures of the original sentence corresponding to the differences, and convert from the original and modified target structures to a translation pattern.

Evaluation of machine translation systems. Research for automatic evaluation method of machine translation systems makes exciting progress.

We should evaluate on how the extracted translation patterns contribute to the accuracy of the machine translation system by automatic evaluation methods such as BLEU[40].

Multilingualization. Our interest in multilingual translation stems from the language independence of our parser and generator. It shall be possible to build a machine translation system for a new language using just decomposed translation examples as pattern dictionary, without deep knowledge of the language itself. Then our translation pattern extraction tool would allow to extract patterns from translation examples of the new language, and ideally to build a system from examples alone.

References

[1] Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, D. Melamed, F.-J. Och, D. Purdy, N. A. Smith, and D. Yarowsky. Statistical machine translation, final report, jhu workshop 1999. Technical report, CLSP/JHU, 1999. http://www.clsp.jhu.edu/ws99/projects/mt/.

[2] Y. Al-Onaizan and Kevin K. Translating named entities using monolingual and bilingual resources. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pages 400–408, 2002.

[3] E. Aramaki, S. Kurohashi, H. Kahioka, and H. Tanaka. Word selection for ebmt based on monolingual similarity and translation confidence. In *Proceedings of HLT-NAACL 2003 Workshop*, pages 57–64, 2003.

[4] E. Aramaki, S. Kurohashi, S. Sato, and H. Watanabe. Finding translation correspondences from parallel parsed corpus for example-based translation. In *Proceedings of Machine Translation Summit VIII*, pages 27–32, 2001.

[5] ATLAS. FUJITSU LIMITED. http://software.fujitsu.com/en/atlas.

[6] P. Brown, J Cocke, S.D. Pietra, V.J.D. Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, 1990.

[7] P.F. Brown, J.C. Lai, and R.L. Mercer. Aligning sentences in parallel corpora. In *Proceedings of 29th Annual Meeting of the Association for Computational Linguistics (ACL-91)*, pages 169–176, 1991.

[8] E. Charniak. A maximum-entropy-inspired parser. In *Proceedings of 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 132–139, 2000. http://ftp.cs.brown.edu/pub/nlparser.

[9] K.W. Church and R.L. Mercer. Introduction to the Special Issue on Computational Linguistics Using Large Corpora. *Computational Linguistics*, 19(1):1–24, 1993.

[10] Collins. *Collins Cobuild English Grammar.* 16(1). COBUILD.London, 1990.

[11] K.W. Dagan, I.and Church and W.A. Gale. Robust Bilingual Word Alignment for Machine Aided Translation. In *Proceedings of 2nd Annual Workshop on Very Large Corpora (WVLC-93)*, pages 1–8, 1993.

[12] T. Dunning. Accurate methods for statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1991.

[13] O. Furuse and H. Iida. Constituent boundary parsing for example-based machine translation. In *Proceedings of 17th International Conference on Computational Linguistics (COLING-94)*, 1994.

[14] W. Gale and Church K. Identifying word correspondences in parallel texts. In *Proceedings of 4th DARPA Speech and Natural Language Workshop*, pages 152–157, 1991.

[15] R. Grishman. Iterative Alignment of Syntactic Structures for a Bilingual Corpus. In *Proceedings of 2nd Annual Workshop on Very Large Corpora (WVLC-94)*, pages 57–68, 1994.

[16] M. Haruno, S. Ikehara, and T. Yamazaki. Learning bilingual collocations by word-level sorting. In *Proceedings of 19th International Conference on Computational Linguistics (COLING-96)*, pages 525–530, 1996.

[17] M. Hayashi, S. Yamada, A. Kataoka, and A. Yokoo. ALT-J/C A Prototype Japanese-to-Chinese Automatic Language Translation System. In *Proceedings of Machine Translation Summit VIII*, 2001.

[18] K. Imamura. Hierarchical phrase alignment harmonized with parsing. In *Proceedings of 6th Natural Language Processing Pacific Rim Symposium (NLPRS-2001)*, pages 377–384, 2001.

[19] K. Imamura, E. Sumita, and Y. Matsumoto. Automatic construction of machine translation knowledge using translation literalness. *Natural Language Processing (Japan)*, 11(2):85–99, 2004. In Japanese.

[20] S. Ishigami. *Business Contract Letter Dictionary, E-book Version, No.1 Sale of Goods*. IBD Corporation, 1992. In Japanese.

[21] JEIDA. *Evaluation Standards for Machine Translation Systems 95-COMP-17.* the Japan Electronic Industry Development Association (JEIDA), 1995. In Japanese.

[22] H. Kaji, Y. Kida, and Y. Morimoto. Learning translation templates from bilingual text. In *Proceedings of 15th International Conference on Computational Linguistics (COLING-92)*, pages 672–678, 1992.

[23] H. Kashioka and H. Ohta. Applying TDMT to Abstracts on Science and Technology. In *Proceedings of Machine Translation Summit VII*, 1999.

[24] M. Kay and M. Röscheisen. Text-Translation Alignment. *Computational Linguistics*, 19(1):121–142, 1993.

[25] M. Kitamura and T. Murata. Practical machine translation system allowing complex patterns. In *Proceedings of MT Summit IX*, pages 232–239, 2003.

[26] T. Kudo and Y. Matsumoto. Applying cascaded chunking to japanese dependency structure analysis. In *Proceedings of JSAI Technical Report, SIG-NL-142*, pages 97–104, 2001. http://cl.aist-nara.ac.jp/˜taku-ku/software/cabocha, In Japanese.

[27] A. Kumano and H. Hirakawa. Building an mt dictionary from parallel texts based on linguistic and statistical information. In *Proceedings of 17th International Conference on Computational Linguistics (COLING-94)*, pages 76–81, 1994.

[28] J. Kupiec. An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of 31st Annual Meeting of the Association for Computational Linguistics (ACL-93)*, pages 23–30, 1993.

[29] M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a Large Annotated Corpus of English:The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[30] Y. Matsumoto, H. Ishimoto, and T. Utsuro. Structural Matching of Parallel Texts. In *Proceedings of 31st Annual Meeting of the Association for Computational Linguistics (ACL-93)*, pages 23–30, 1993.

[31] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, K. Takaoka, and M. Asahara. Japanese morphological analysis system chasen version 2.2.1. Technical report, Graduate School of Information Science Nara Institute of Science and Technology, 2000. http://chasen.aist-nara.ac.jp.

[32] Y. Matsumoto, S. Kurohashi, T. Utsuro, H. Myouki, and M. Nagao. Japanese morphological analysis system juman version 2.0. INFORMATION SCIENCE TECHNICAL REPORT NAIST-IS-TR94025, Graduate School of Information Science Nara Institute of Science and Technology, 1994. http://www.kc.t.u-tokyo.ac.jp/nl-resource/juman.html, In Japanese.

[33] Y. Matsumoto and R. Sugimura. Sax; a parsing system based on logic programming languages. *Computer Software*, 3(4):308–315, 1986. http://chasen.naist.jp/sax.html, In Japanese.

[34] Y. Matsumoto and T. Utsuro. Lexical knowledge acquisition. In *Handbook of Natural Language Processing*, pages 563–610. Marcel Dekker, 2000.

[35] A. Mayers, R. Yangarber, and R. Grishman. Alignment of Shared Forests for Bilingual Corpora. In *Proceedings of 19th International Conference on Computational Linguistics (COLING-96)*, volume 1, pages 460–465, 1996.

[36] I.D. Melamed. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In *Proceedings of 3rd Annual Workshop on Very Large Corpora (WVLC-95)*, pages 184–198, 1995.

[37] R.C. Moore. Learning translations of named-entity phrases from parallel corpora. In *Proceedings of 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2003)*, pages 259–266, 2003.

[38] National Language Research Institute NLRI. *Word List by Semantic Principles, 'Bunrui-Goi-Hyo'*. Syuei Syuppan, 1994. In Japanese.

[39] K. Ohmori, J. Tsutsumi, and Nakanishi. M. Building bilingual word dictionary based on statistical information. In *Proceedings of 2nd Annual Meeting of The Association for Natural Language Processing*, pages 49–52, 1996. In Japanese.

[40] K. Papineni, S. Roukos, T. Ward, and Zhu W.J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pages 311–318, 2002.

[41] E. Planas and O. Furuse. Multi-level similar segment matching algorithm for translation methods and example-based machine translation. In *Proceedings of 23rd International Conference on Computational Linguistics (COLING-2000)*, 2000.

[42] S.R Roget. *Roget's Thesaurus.* Crowell Co., 1911.

[43] G. Salton and MJ McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[44] S. Sato. MBT2: A Method for Combining Fragments of Examples in Example-Based Translation. *Artificial Intelligence*, 75(1), 1995.

[45] M. Shieber and C.N. Pereira. Semantic-head-driven generation. In *Computational Linguistics*, pages 30–42, 1990.

[46] M. Shimizu and N. Narita. *Japanese-English Dictionary.* KODANSHA BUNKO, 1979. In Japanese.

[47] S. Shimohata. An empirical method for identifying and translating technical terminology. In *Proceedings of 23rd International Conference on Computational Linguistics (COLING-2000)*, 2000.

[48] F. Smadja. Retrieving collocation from text: Xtract. *Computational Linguistics*, 19(1):143–177, 1993.

[49] F.A. Smadja and K.R. McKeown. Translating Collocations for Bilingual Lexicons: A Statistical Approach. *Computational Linguistics*, 22(1):1–38, 1996.

[50] SNLR (International Workshop on Sharable Natural Language Resource). NAIST, Nara, 1994.

[51] E. Sumita, H. Iida, and H. Kohiyama. Translating with Examples: A New Approach to Machine Translation. In *Proceedings of 1st Theoretical and Methodological Issues in Machine Translation (TMI-90)*, 1990.

[52] K. Takeda. Pattern-based context-free grammars for machine translation. In *Proceedings of 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, pages 144–151, 1996.

[53] M. Utiyama and H. Isahara. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, pages 72–79, 2003.

[54] T. Utsuro, H. Ikeda, M. Yamane, Y. Matsumoto, and M. Nagao. Bilingual Text Matching Using Bilingual Dictionary and Statistics. *Proceedings of 17th International Conference on Computational Linguistics (COLING-94)*, 2:1076–1082, 1994.

[55] H. Watanabe, S. Kurohashi, and E. Aramaki. Finding structural correspondences from bilingual parsed corpus for corpus-based translation. In *Proceedings of 23rd International Conference on Computational Linguistics (COLING-2000)*, pages 906–912, 2000.

[56] K. Yamada and K. Knight. A syntax-based statistical translation model. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, pages 523–530, 2001.

[57] K. Yamamoto, T. Kudo, Y. Tsuboi, and Y. Matsumoto. Learning sequence-to-sequence correspondences from parallel corpora via sequential pattern mining. In *Proceedings of HLT-NAACL 2003*, pages 73–80, 2003.

[58] K. Yamamoto and Y. Matsumoto. Acquisition of phrase-level bilingual correspondence using dependency structure. In *Proceedings of 23rd International Conference on Computational Linguistics (COLING-2000)*, pages 933–939, 2000.

[59] K. Yamamoto and Y. Matsumoto. Extracting translation knowledge from parallel corpora. In M. Carl and A. Way, editors, *Recent Advances in*

Example-Based Machine Translation, pages 365–395. Kluwer Academic Publishers, 2003.

List of Publications

Major Publications

Journal Articles

[1] Mihoko Kitamura and Yuji Matsumoto, "Automatic Acquisition of Translation Rules from Parallel Corpora," In *IPSJ Journal*, vol.37, no.6, June. 1996, pp.1030–1040 (in Japanese).

[2] Mihoko Kitamura and Yuji Matsumoto, "Automatic Extraction of Translation Patterns in Parallel Corpora," In *IPSJ Journal*, vol.38, no.9, Apr. 1997 pp.727–736 (in Japanese).

International Conferences(Reviewed)

[1] Mihoko Kitamura and Yuji Matsumoto, "A Machine Translation System based on Translation Rules Acquired from Parallel Corpora," In *Proceedings of Recent Advances in Natural Language Processing(RANLP-95)*, September 1995, pp 27–44.

[2] Mihoko Kitamura and Yuji Matsumoto, "Automatic Extraction of Word Sequence Correspondences in Parallel Corpora," In *Proceedings of Forth Annual Workshop on Very Large Corpora(WVLC-96)*, September 1996, pp.79–87.

[3] Mihoko Kitamura and Yamamoto, H., "Intelligent Retrieval system for creating sentences in foreign language", In *Proceedings of Artificial Intelligence in Education(AIED-97)*, August 1997, pp.612–614.

[4] Mihoko Kitamura and Murata, T., "Practical Machine Translation System allowing Complex Patterns," In *Proceedings of Machine Translation Summit IX*, September 2003, pp.232–239.

[5] Mihoko Kitamura and Yuji Matsumoto, "Practical Translation Pattern Acquisition from Combined Language Resources," In *Proceedings of The First*

International Joint Conference on Natural Language Processing(IJCNLP-04), March 2004, pp 652–659.

[6] Mihoko Kitamura, Nakagawa, T., Kim, S. and Murata, T., "Development of Chinese-Japanese MT System based on Language-Independent Translation Engine," In *Proceedings of Asian Symposium on Natural Language Processing to Overcome Language Barriers*, March 2004, pp 39–45.

Local Workshops

[1] Mihoko Kitamura, Kai, K., Okada, K., and Nagata, J., "A Extensible Japanese to English Machine Translation System", In *IEICE Technical Report*, NLC91-24, 1991, pp63–70. In Japanese

[2] Mihoko Kitamura, and Yuji Matsumoto, "Automatic Acquisition of Translation Knowledge from Parallel Bilingual Corpus", In *IEICE Technical Report*, NLC94-2, 1994, pp 9–16. In Japanese.

[3] Mihoko Kitamura, and Yuji Matsumoto, "Automatic Acquisition of Translation Knowledge from Parallel Corpus", In *Proceedings of the 8th Annual Conference of JSAI*, 1994, pp645–648. In Japanese.

[4] Mihoko Kitamura, and Yuji Matsumoto, "Automatic Construction of Translation Dictionary using Parallel Texts", In *Proceedings of Symposium on Learning for Natural Language Processing*, 1994, pp150–157. In Japanese

[5] Mihoko Kitamura, and Yuji Matsumoto, "Translation Knowledge Acquisition from Parallel Corpora and Application to Machine Translation System", In *Proceedings of first Annual Meeting of Natural Language Processing(NLP 95)*, 1995, pp289–292. In Japanese.

[6] Mihoko Kitamura, and Yuji Matsumoto, "Domain-Independent Translation Rules Extraction using Parallel Text", In *Proceedings of 2nd Annual Meeting of Natural Language Processing(NLP 96)*, 1996, pp1–4. In Japanese.

[7] Mihoko Kitamura, and Yuji Matsumoto, “Automatic Extraction of Translation Patterns in Parallel Corpora”, In *IEICE Technical Report*, NLC96-19, 1996, pp 69–76. In Japanese.

[8] Mihoko Kitamura, and Yamamoto, H., “Translation Examples Retrieval System using Sentence and Word Alignment Technique”, In *Proceedings of the 53tn IPSJ Annual Meetings*, (2)2H-1, 1996, pp385–386. In Japanese

[9] Mihoko Kitamura, and Yuji Matsumoto, “English Writing Assistant System on Knowledge Acquisition Methods from Translations”, In *JSAI Technical Report*, SIG-J-9602, 1996, pp46–51. In Japanese.

[10] Mihoko Kitamura, Toshiki Murata, Tatsuya Sukehiro, Sayori Shimohata, Miki Sasaki, Toshihiko Matsunaga and Tetsuji Nakagawa, “Basic Technology and Development of Community Type Machine Translation Sites *Yakushite-Net*”, In *Proceedings of the 65th IPSJ Annual Meetings*, Vol.5, March 2003, pp.319–322. In Japanese.

[11] Mihoko Kitamura, “Semiautomatic Translation Patterns Extraction from Small-scale Parallel Texts”, In *Proceedings of the 2nd Forum on Information Technology(FIT 2003)*, Vol.E, September 2003, pp.87–88. In Japanese.

Other Publications

Journal Articles

[1] Yuji Matsumoto and Mihoko Kitamura, “Acquisition of Translation Rules from Parallel Corpora”, *Recent Advances in Natural Language Processing*, John Benjamins Publishing Company, 1997, pp405–416.

[2] Hideki Yamamoto and Mihoko Kitamura, “Corpus based natural language processing and an education system using it”, *Journal of Japanese Society for Information and Systems in Education(JSISE)*, Vol.16, No.1, 1999, pp43–50. In Japanese.

International Conferences(Reviewed)

[1] Kaoru Yamamoto, Yuji Matsumoto, and Mihoko Kitamura, “A Comparative Study on Translation Unit for Bilingual Lexicon Extraction”, In *Proceedings of the 39th Annual Meeting and 10th Conference of the European Chapter of the Association for Computer Linguistics, Data-driven MT Workshop*, 2001, pp87–94.

[2] Sayori Shimohata, Mihoko Kitamura, Tatsuya Sukehiro and Toshiki Murata, “Collaborative Translation Environment on the Web”, In *Proceedings of Machine Translation Summit VIII*, September 2000, pp331–334.

[3] Toshiki Murata, Mihoko Kitamura, Tsuyoshi Fukui, and Tatsuya Sukehiro, “Implementation of collaborative translation environment: YakushiteNet”, In *Proceedings of Machine Translation Summit IX*, September 2003, pp.479-482. In Japanese.

Local Workshops

[1] Yuji Matsumoto and Mihoko Kitamura, “Acquisition of Translation Rules from Parallel Corpus”, In *NAIST Technical Report*, NAIST-IS-TR94014, June 1994. In Japanese

[2] Yukitaka Nakatsuka, Takehito Utsuro, Mihoko Kitamura, and Yuji Matsumoto, “Graphical User Interface of Structural Matching of Parallel Sentences: V-Para”, In *NAIST Technical Report*, NAIST-IS-TR96023, December 1996.

[3] Toshihiko Matsunaga, Mihoko Kitamura, and Toshiki Murata, “Sentence Matching Algorithm of Revised Documents with Considering Context Information”, In *IEICE Technical Report*, NLC2003-22, August 2003, pp.43–48. In Japanese

[4] Miki Sasaki, Mihoko Kitamura, Sayori Shimohata., Tetsuji Nakagawa, “Automatic Domain Determination using *Core-Word*”, In *Proceedings of the*

2nd Forum on Information Technology(FIT 2003), Vol.E, September 2003, pp.171–172. In Japanese.

Award

[1] 1996 IPSJ Best Paper Award
Mihoko Kitamura and Yuji Matsumoto, “Automatic Acquisition of Translation Rules from Parallel Corpora,” In *IPSJ Journal*, vol.37, no.6, June. 1996, pp.1030–1040. In Japanese.

Patents

[1] Mihoko Kitamura, “Translation Patterns Construction Method and Translation Patterns Construction Apparatus” *Japanese Patent No.3305953.*, 2002

[2] Mihoko Kitamura, Hideki Yamamoto and Mitsuo Shimohata, “Dictionary Registration Apparatus and Machine Translation System” *Japanese Patent No.3429612.*, 2003

Press Releases

[1] Nihon Keizai Shinbun, 1996/8/26, page 15 (Science and Technology): "辞書用語簡単に追加 OKI 専門的文書に効果" (OKI allows easy addition to dictionaries; effective for specialized documents)

[2] Nikkei Sangyo Shinbun, 2004/10/01, page 6: "利用者参加型の翻訳サイト" (A translation web-site where users can participate)

Acknowledgements

I would like to thank many people on the completion of this thesis.

First of all I would like to express my sincere gratitude to Professor Yuji Matsumoto for supervising this thesis. He taught me the excitement of research in natural language processing, and trained me as a researcher. Without his continuous encouragement, guidance and support, I would not have been able to complete the doctoral thesis.

I am also grateful to the members of my thesis committee: Professor Shunsuke Uemura and Professor Kiyoshiro Shikano for useful suggestions and comments.

A part of this work was supported by a grant from National Institute of Information and Communications Technology(NICT).

The major part of the work reported in this thesis was done at Oki Electric Co., Ltd. I am grateful to all previous and current members of Ubiquitous System Laboratory, especially Mr. Toshiki Murata, who not only cooperated with me in some part of the research reported in this thesis but always cheered me up. My special thanks go to Mr. Hideki Yamamoto, Mr. Junji Nagata, Mr. Toshihisa Nakai, Mr. Harushige Sugimoto, Mr. Hideya Hayashi and Mr Isamu Nose for providing me with the chance to research to machine translation; and to Mrs. Sayori Shimohata, Mr. Tetsuji Nakagawa, Mr. Tatsuya Sukehiro, Mr.Atsushi Ikeno, Mr. Mitsuo Shimohata, Ms. Miki Sasaki, Mr. Takahiro Yamazaki and Mr. Junichi Fukumoto for their stimulating discussion and valuable advice.

I also would like to express my gratitude to all previous and current members at Matsumoto laboratory. Dr. Kaoru Yamamoto gave me valuable comments and advice through productive discussion. Discussions with her filled me with the willingness to try a new way. Mrs. Sanae Fujita had a lot in common with me, combining studies, work and motherhood, and we could support each other.

It is no doubt that I could concentrate this work with the help of many in my daily life, such as my family, my friends and the kindergarten “Rokuman Hoikuen” for my children. I would like to thank to my parents, Kiyoshi Kitamura and Yumiko Kitamura for their consistent encouragement and generous support. I thank my best friends, Mrs. Mamiko Ohta and Sonoko Fujibayashi, who gave their continuous encouragement and advice for my life. My much-loved little

sons, Serge Garrigue and Hugo Garrigue provided emotional support to keep my work.

Last but not least, I wish to eternally thank my husband, Jacques Garrigue, for his continuous encouragement and for a valuable help and criticism as a researcher. Furthermore he has corrected patiently many of my very poor English writings including this thesis.

The birth of my baby son, Hugo, provided me with a last chance to complete this doctoral thesis. Avec toute ma reconnaissance.