NAIST-IS-DT9961030

Doctor's Thesis

Extracting Translation Knowledge from Parallel Corpora

Kaoru Yamamoto

February 5, 2002

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

Doctor's Thesis
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

Kaoru Yamamoto

Thesis committee: Yuji Matsumoto, Professor
Kiyoshiro Shikano, Professor
Katsumasa Watanabe, Professor

Extracting Translation Knowledge from Parallel Corpora*

Kaoru Yamamoto

Abstract

This thesis deals with extracting *translation knowledge* from parallel corpora. The aim is to acquire meaningful translation knowledge which can be used to build translation knowledge base, aiding translators or language learners. We present three works on this topic.

The first work uses statistically probable dependency relations obtained from parsers to acquire word and phrasal correspondences. The result showed that statistically probable dependency relations are effective in translation knowledge acquisition even for language pairs with different word ordering.

The second work compares three models of translation units each of which uses different linguistic information: word segmentation, chunk boundary, and word dependency. We found that chunk boundaries are useful linguistic clues in extracting compound noun phases which will be effective for extracting bilingual lexicons in the new domain. Furthermore, word dependency are also useful for longer translation pairs such as idiomatic expressions.

The final work proposes a data mining approach to extracting bilingual lexicon from parallel corpora. The task is viewed as sequential pattern mining and the PrefixSpan algorithm is applied for counting co-occurrence and independent frequencies efficiently. This method uniformly extracts rigid compounds as well as non-contiguous collocations and offers useful resources for translation aids.

We demonstrate effective application of natural language processing and data mining techniques to extracting translation knowledge from parallel corpora of different language family.

*Doctor's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT9961030, February 5, 2002.

Keywords:

translation knowledge acquisition, corpus-based approach

Contents

List of Figures

Chapter 1

Introduction

As an application of natural language processing (NLP), machine translation (MT) can claim to be one of the oldest fields of study. Even with such a long tradition, MT has not yet met users' full expectations and still has much room for improvement. The difficulty in MT lies in the fact that it has to deal with almost every aspect of computational linguistics, in at least two languages.

Recent advances in NLP owe greatly to corpus-based or statistical approaches. In the 1990s, large-scale machine-readable linguistic resources became available and computational power increased. These factors allowed us to process a voluminous amount of linguistic data using computationally expensive statistical methods that could not have been employed before. We are now in the 21st century, corpus-based NLP can be seen in every aspect of NLP, including POS tagging and parsing.

MT is no exception to the above trend; data-driven approaches exemplified by Example-based MT (EBMT) and Statistical MT (SMT) gain focus of attention. The important assumption in these paradigms is that lexicons and rules for translation are somehow acquired from corpora which can then compiled into an MT engine. As the MT gets complicated, manual preparation of translation lexicons and rules become cumbersome. With the availability of parallel corpora and increased computational power, automatic acquisition of translation knowledge is desired, and this is precisely the topic of this thesis.

1.1 Goals and Objectives

This thesis deals with extracting *translation knowledge* from linguistic corpora using NLP techniques. The goal is to demonstrate automatic methods of translation knowledge extraction which will be useful for constructing translation knowledge base. Such a knowledge base can aid translators or language learners. Translation knowledge can take many forms: paragraphs, sentences, phrases, and words. However, we restrict our attention to translation knowledge expressed in phrase-equivalent units, since methods to identify phrase-equivalent units have not been addressed properly in previous works.

Figure 1.1 illustrates the overview of our work in context of translation aids.

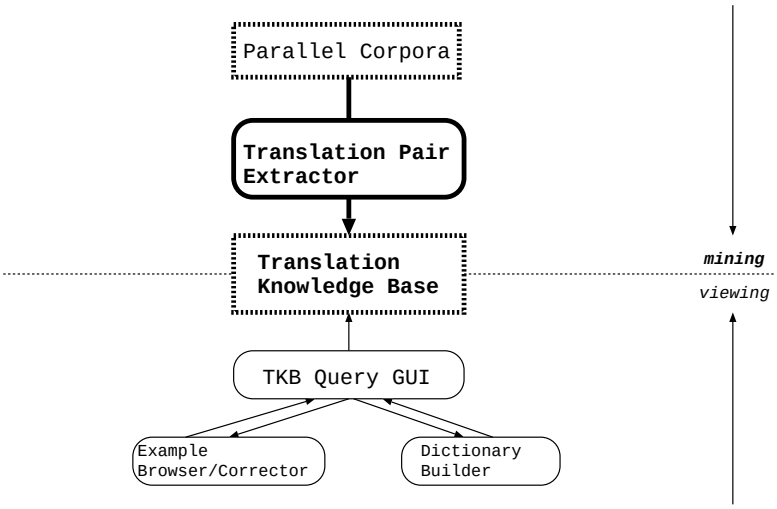


Figure 1.1. Translation Aid Framework

Our work will concentrate mainly on the mining phase of translation knowledge base management. We propose three approaches for acquiring translation knowledge from parallel corpora.

The translation pair extraction takes bilingual parallel corpora as input where each sentence pair are translations of each other. The output is translation knowledge expressed in word- or phrase-equivalent units.

Most previous works, for example Gale [17], addressed primarily on translational correspondences at word-level. However, a word-for-word translation model

cannot handle idiomatic expressions. In order to deal with such situations, we need a phrase-level unit of correspondences.

Our research goal is to propose methods to extract translational correspondences not only of word-level but also of phrase-level. The task of translation knowledge extraction comprises of two subtasks:

1. **generation and counting of tentative candidates**
2. design of extraction algorithm with a suitable similarity measure

In this thesis, we exclusively focus on (1). First, we investigate how far linguistic clues obtained from NLP tools such as chunk boundary and word dependency are effective in generation of tentative candidates. Second, we focus on efficient methods to count tentative candidates by adopting data mining approach.

1.2 The Outline of the Thesis

The thesis is organised as follows.

Chapter 2 is devoted to the background of extracting bilingual lexicons. A typical problem setting of translation knowledge extraction and basic preliminaries are also given.

The next three chapters describe our original work on translation knowledge extraction.

In Chapter 3, we apply statistically probable dependency relations to acquire word and phrasal correspondences. The approach is based on the observation that phrase dependencies are preserved across language pairs even if they have different word ordering constraints. We use corpus-based dependency parsers to obtain statistically probable dependency relations. The objective is to show effectiveness of dependency relations in translation knowledge acquisition even for languages with different word ordering constraints.

Chapter 4 is a follow-up on Chapter 3. This study compares three models of translation units, each of which uses different linguistic information, namely, one with only word segmentation, one with chunk boundary, and one with word dependency. The aim is to investigate relationship between the linguistic clues applied and the translation knowledge extracted.

In Chapter 5, we propose a data mining approach to extracting bilingual lexicon from parallel corpora. The task is viewed as sequential pattern mining from transactions of paired bilingual sentences and apply the PrefixSpan algorithm to find frequently appearing sequential patterns in parallel corpora. We have achieved an efficient generation and counting of bilingual multiword expressions by adopting the PrefixSpan algorithm.

Chapter 6 concludes our work and present future directions.

Chapter 2

Background

This chapter gives a basic background of data-driven MT, which has been the main driving force for extracting bilingual lexicons. Then, we describe a typical problem setting and some preliminaries in the translation knowledge extraction.

2.1 Data-Driven MT

The recent availability of large-scale machine-readable linguistic resources and the rapid advances in computational power lead us to the era of corpus-based approaches in NLP.

MT is no exception to the above trend; data-driven approaches such as Statistical MT (SMT) and Example-based MT (EBMT) gain focus of attention. In these paradigm, lexicons and rules for translation are somehow acquired from corpora which can then compiled into MT engine.

Figure 2.1 illustrates the general framework of data-driven MT.

As the MT gets complicated, manual preparation of translation lexicons and rules are cumbersome and automatic acquisition of translation knowledge is strongly desired.

Before we describe translation knowledge acquisition, which is the main theme of the thesis, we briefly look at data-driven MT approaches which are another major motivation for translation knowledge acquisition.

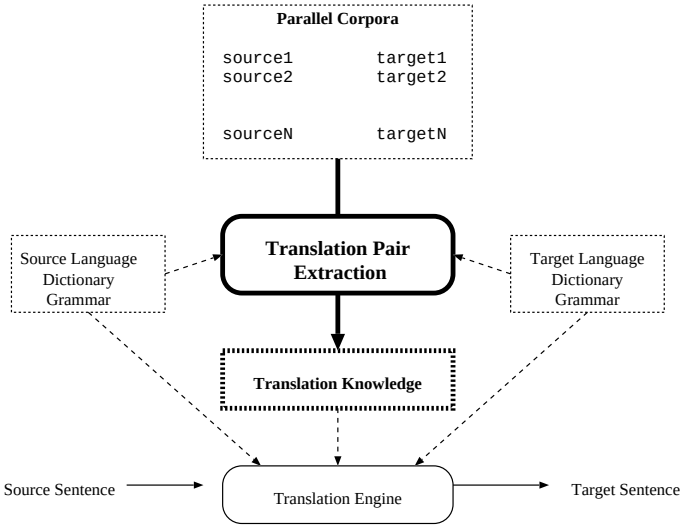


Figure 2.1. Data-driven MT

2.1.1 Statistical MT

The SMT framework has been proposed by Brown et al. in late 1980s where the translation probabilities between English and French are estimated using the EM algorithm [7][6]. The model is based on Shannon's noisy channel model which is illustrated in Figure 2.2. It receives an English sentence e, transforms it into a French sentence f, and sends the French sentence f to a decoder. The decoder then determines the English sentence $\hat{e}$ that f is most likely to have arisen from.

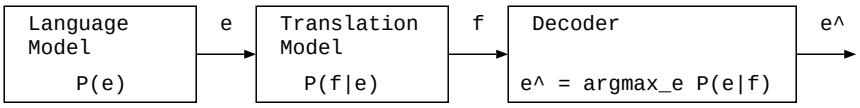


Figure 2.2. Noisy Channel Model in Statistical MT

There are three components for translation from French to English: a language model, a translation model and a decoder. Since the parameters of the model, called the translation probabilities is closely related to lexical acquisition, we will look at it in detail. Below, a translation model based on word alignment is described.

We use the same notations in Brown et al.: e is the English sentence; l is the length of e in words; f is the French sentence; m is the length of f; f_j is the word j in f; a_j is the position in e that f_j is aligned with; e_{a_j} is the word in e that f_j is aligned with; $P(w_f|w_e)$ is the translation probability, the probability that we will seen word w_f in the French sentence given that we see word w_e in the English sentence; Z is a normalization constant.

We compute $P(f|e)$ by summing the probabilities of all alignments. For each alignment, we make two simplifying assumptions. Each French word is generated by exactly one or no English word, and the generation of each French word is independent of the generation of all other French words in the sentence.

The EM algorithm is used to estimate the translation probability starting with a random initialization. In the estimation step, we compute the expected number of times we will find w_f in the French sentence given that we have w_e in the English sentence.

$$Z_{w_f,w_e} = \sum_{(e,f)s.t.w_e \in e, w_f \in f} P(w_f|w_e)$$

where the summation ranges over all pairs of aligned sentences such that the English sentence contains w_e and the French sentence contains w_f. The maximization step re-estimates the translation probabilities from the following expectations.

$$P(w_f|w_e) = \frac{Z_{w_f,w_e}}{\sum_v Z_{w_f,v}}$$

where the summation ranges over all English words v.

Various extensions to penalize implausible alignments have been proposed. For example, the *alignment probability* $a(i|j,m,l)$ is the probability that the jth word in a French sentence of length m is the translation of the ith word in an English sentence of length l. This is to implement the heuristic that distortion in the positions of the two aligned words will decrease the probability of the alignment. Similarly, a notion of *fertility* is introduced for each English word which tells us how many French words it usually generates.

The model had a tremendous impact in the MT research community in the early 1990s. However, Brown pointed a problem that the lack of linguistic knowl-

edge encoded in the system causes many translation failures. In particular, the model has no notion of phrase and non-local dependencies that are difficult to capture.

Despite such problems, research in SMT has revived notably by the USC ISI Group and the RWTH Aachen Group [4]. The focus of their attention is to incorporate linguistic knowledge into the model, while retaining robustness where statistical MT is superior.

SMT is less popular in English-Japanese MT, only Yamada and Knight [45] have applied. A primary reason is due to the syntactic difference between the two languages which one cannot ignore when modelling translation equivalence.

2.1.2 Example-based MT

The original idea for the EBMT framework dates back to Nagao's 1984 paper [33]. The essence of EBMT is captured by his much quoted statement:

> Man does not translate a simple sentence by doing deep linguistic analysis, rather, Man does translation, first, by properly decomposing an input sentence into certain fragmental phrases [...], then by translating these phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference.

As he stated, there are three main components of EBMT: matching fragments against a database of real examples, identifying the corresponding translation fragments, and then recombining these to give the target text.

As an example, we illustrate how an English sentence "He took a cup of milk" is translated into a Japanese sentence "彼は一杯のミルクを飲んだ". We parse the English sentence into a suitable (tree) representation and match against real examples in the database. Suppose there is an example ⟨ He took a cup of coffee, 彼は一杯のコーヒーを飲んだ ⟩ in the database, a thesaurus stating the similar relation between "coffee" and "milk", a dictionary which confirms a bilingual correspondence between "milk" and "ミルク". By applying such resources, we will

obtain a suitable (tree) representation of the translation. This will be generated into a translated surface string "彼は一杯のミルクを飲んだ"

Unlike the SMT framework, there is no firm definition and theory for EBMT, and this has led to some confusion with the Translation Memory. However, there have been many works that claimed to be EBMT, including Sato and Sumita [39], [43].

We leave details to the recent review by Somers [42]. A notable point is that EBMT has compensated a rule-based MT paradigm, especially in the domain where a powerful but complex grammar cannot be prepared easily.

2.2 Translation Pair Acquisition

Previous work in acquiring bilingual correspondences used parallel corpora as well as from non-parallel corpora. The former parallel corpora contain paired sentences of the original and its translation. On the other hand, non-parallel corpora are a pair of monolingual corpora in the same or similar field. By definition, sentences in non-parallel corpora are not aligned.

Our works assume parallel corpora. Hence, we primarily pay our attention to acquiring bilingual correspondences from parallel corpora in this section.

There are two distinct approaches in acquiring bilingual correspondence from sentence-aligned parallel corpora. One is alignment and the other is extraction. Both aim to find correspondences of smaller components in the sentences that may take form of word, noun phrase, collocation or constituents.

Alignment is the process by which correspondences of subcomponents within the paired sentence are found. The goal is to find correspondence for *all* subcomponents in the sentence. Extraction, as its name stands, focuses on extracting subcomponents that correspond by processing the entire parallel corpora. The goal is to find correspondences for *some* substructures in the parallel corpus.

It is vital to distinguish between often confused approaches. Alignment concentrates on completeness (or coverage) by aligning all corresponding subcomponents, however, suitability (or precision) of the resulted alignments as an entry for bilingual lexicon is sacrificed. Extraction acquires meaningful correspondences with high precision, but fails to find all such correspondences in the entire parallel

translation units	preprocessing
Word	tokenization, POS tagging
Base NP	tokenization, POS tagging, chunking
Collocations	tokenization, POS tagging, chunking
Dependency Structure	tokenization, POS tagging, chunking, dependency analysis

Table 2.1. Relationship between Translation Units and Preprocessing

corpora.

Below, we provide preliminaries in translation pair acquisition with respect to unit of translation, corpora, similarity measure, type of algorithm, and evaluation. Then, we review previous approaches taken in alignment and in extraction from parallel corpora.

2.2.1 Preliminaries

Translation Units

Various level of translation units can be defined given a raw bilingual text. Examples are words, noun phrase (NP) compounds[1], collocations, phrase/dependency structure. The more complex a desired translation unit is, the deeper a degree of language processing it will require. Table 2.1 summarises require NLP tools for the extraction of four types of translation units.

The preprocessing required in each stage can be obtained using rule-based systems based on regular expressions or finite state automaton. Recent works use corpus-based tools that are trained directly from annotated corpus using statistical methods such as Hidden Markov Model or machine learning techniques such as Decision Tree, Maximum Entropy, Support Vector Machines. It is important to point out that such a preprocessing will contain errors and unresolved ambiguities irrespective of whether they are rule-based or corpus-based.

Early works sought word-for-word correspondences. However, there are a number of attempts to extend correspondences to NP phrases or collocations typically found in technical documents. Kupiec extracted English-French NP corre-

[1]They are mostly base NP, in other words, NP which do not embed NP inside.

spondences [26]. The method used HMM-based part-of-speech (POS) taggers and finite-state NP recognizers to extract NP in each language and applied an EM-like algorithm to find NP correspondences. Smadja et al. extracted English-French collocations with Champollion system [40] which is based on Smadja's Xtract system [41]. Smadja and colleagues reported that 73% of the French translations of valid English collocations were judged to be correct by three evaluators.

Corpora

We compare two kinds of corpora used for translation pair acquisition: Parallel and Non-parallel.

Parallel corpora are bilingual texts where each sentence in one language has a correspondent translated sentence in the other language. The main assumptions are that the majority of essential information has not been lost through translation and that the order of the sentences is retained. As such, the bilingual correspondence can be estimated by co-occurrence frequency and the position of the translation units in the parallel corpora. Earlier works used the Canadian Hansard [1] for English-French translation pairs [7].

Non-parallel corpora[2] is a pair of monolingual corpora in the same or similar field. The assumptions used in parallel corpus no longer hold, in other words, neither frequency nor position of occurrence are comparable. Instead, bilingual lexicon extraction from non-parallel corpora uses the following characteristics: a) semantically similar terms appear in similar contexts and b) words in the same domain and the same time period have comparable usage patterns. Fung and Rapp have independently investigated this task in English-Chinese, and English-German respectively [15][35].

Parallel corpora are an expensive resources, but various levels of translation correspondences from simple ones (e.g. words) to complex ones (e.g. phrases) can be extracted. In contrast, a vast amount of non-parallel corpora can be obtained at much cheaper cost with the explosive boom of the Internet. However, the extracted translation pairs are mostly limited to word correspondence. This is because contextual similarity is calculated based on the bag-of-word distance model such as Euclidean distance where sequential or dependency relations among

[2]Non-parallel corpora are sometimes referred as comparable corpus or noisy parallel corpus

	y	¬ y	total
x	freq(x,y) = a	freq(x,¬y) = b	freq(x)
¬ x	freq(¬x,y) = c	freq(¬x,¬y) = d	freq(¬x)
total	freq(y)	freq(¬y)	

Table 2.2. Contingency Table

words are completely lost.

Similarity

Similarity measures are defined in order to correlate translation units from different language. Several similarity measures have been proposed, and a detailed discussion can be found in [30].

Most methods for estimating translation pairs from parallel corpora start with the following intuition: Words that are translations of each other are more likely to appear in corresponding sentences (i.e. to co-occur) than pairs of unrelated words. Co-occurrence based similarity measures can be defined in terms of a contingency table.

The formulae below are defined in terms of the contingency table in Table 2.2 and $N = a + b + c + d$.

- Mutual Information

$$\log_2 \frac{aN}{(a+b)(a+c)}$$

- Weighted Mutual Information[16]

$$(\frac{a}{N}) \log_2 \frac{aN}{(a+b)(a+c)}$$

- Dice Coefficient[37]

$$\frac{2a}{2a+b+c}$$

- Weighted Dice Coefficient[23]

$$(\log_2 a)\frac{2a}{2a+b+c}$$

- ϕ^2 statistics[17]

$$\frac{(ad-bc)^2}{(a+b)(a+c)(b+d)(c+d)}$$

- Log-likelihood[13]

$$\begin{aligned} a\log a + b\log b + c\log c + d\log d - \\ (a+b)\log(a+b) - (a+c)\log(a+c) - \\ (b+d)\log(b+d) - (c+d)\log(c+d) + \\ (a+b+c+d)\log(a+b+c+d) \end{aligned}$$

For non-parallel corpora, the contextual similarity is used for measuring the similarity between terminology. Fung defined a two-dimensional matrix for contextual similarity, one dimension for a seed bilingual lexicon and the other for unknown words [16]. For each unknown word, word relation vectors are calculated by weighted mutual information. This is an operation on a monolingual half of non-parallel corpora. The bilingual correspondence between two monolingual word relation vectors was calculated with Euclidean distance and Cosine measure.

Algorithm

Two types of algorithms have been used in the previous literature: greedy determination, and iterative estimation. The former ranks the translation pairs according to their association score using a similarity measure. Then, an algorithm extracts pairs that are more than a given threshold and the remaining pairs undergo the next iteration with a lower threshold. Once pairs are extracted, they are never re-examined.

The iterative estimation uses the Estimation Maximization algorithm (EM algorithm). The initial translation probability (translation model parameters)

is estimated using frequency and an algorithm refines the translation model parameters using the expectation maximization algorithm. Unlike in greedy determination, the likelihood of correspondences are considered for all pairs at each iteration, and the last iteration determines the final correspondence.

A number of heuristics are employed to reduce the number of combinations to be considered. Examples include the use of MRD (machine readable dictionary), cognates, parts-of-speech and position (alignment) of the word in the sentence. Melamed compared different filters and demonstrated that such linguistic knowledge sources are effective in augmenting the induction of translation probabilities in statistical MT [31].

Evaluation

Extracted translation pairs are normally evaluated in terms of precision and recall. Precision is defined as the ratio of "correct" pairs to the extracted. Definition of correctness depends on a parallel corpus and usually relies on human judges. Recall is the ratio of extracted pairs to what is supposed to be extracted in the original parallel corpus. Generally, recall is difficult to calculate, since what should have been extracted beforehand is not well-defined.

Melamed proposed an automatic evaluation of word-based translation lexicon called BiBLE (Bitext-Based Lexicon Evaluation)[31]. BiBLE uses "precision" and "percent correct" and takes F-measure of the two. The intuition behind it is that a better lexicon will find more correspondences in a parallel corpus.

BiBLE works well with a large parallel corpus (e.g. 100,000 paired sentences in the Canadian Hansard), but infrequent but correct translation pairs do not obtain much credit. Moreover, it is not easy to extend his method to evaluate bilingual lexicons that are more than just word correspondences. Based on these reasons, BiBLE is not much used despite its automatic features, and subjective human judgment is still a popular evaluation method.

2.2.2 Alignment

The goal of alignment is to find correspondence for *all* subcomponents in the sentence. Alignment has been investigated in word level as well as structure

levels.

Historically, word alignment was seen as a natural extension from sentence alignment. Such view was shared by Dagan and Church who employed a dynamic programming formulation similar to that of sentence alignment with finer granularity and with different slope constraints [12] [9]. Dagan et al. [12] reported that for 160,000-word excerpt of the Canadian Hansards produced an alignment in which about 55% of the words were correctly aligned, 73% were within one word of the correct position, and 84% words were within three words of the correct position.

A structural alignment produces an alignment between constituents (or sentence substructures) within the sentence pairs of parallel corpora. The task is usually conducted in a parse-parse-match manner, in that each half of paired sentences is parsed independently in advance and then the algorithm finds matching of constituents by measuring similarities. Many works have been reported in this line, including Kaji et al. [21], Matsumoto et al. [29], Grishman et al. [18], Kitamura et al. [22], Meyer et al. [32]. These methods not only acquire richer alignment but also resolve ambiguity arisen from monolingual parsing.

Although research results show promising results, there are some problems associated with alignment. First of all, word alignment is robust but exploits the positional heuristics which does not provide good approximates to languages with relaxed word ordering constraints such as Japanese. On the other hand, structural alignment finds meaningful correspondences that can be used for translation rules in EBMT. However, these methods are often hampered by the performance of parsers used in both languages. Most methods used rule-based parsers which often failed to handle complex sentences that include subordinate clauses and conjunctions, thereby sacrificing robustness.

2.2.3 Extraction

The goal of extraction is to find correspondences for *some* substructures in the parallel corpus.

Early works sought for word correspondences such Gale [17] and Melamed [31]. However, as texts are not translated literally, there is an obvious shortcoming in the word-for-word model. These methods failed to account for compounding

nouns, collocations or idiomatic expressions typically found in technical documents. In order to overcome this, some research has been conducted to extract longer correspondences than words.

Kupeic extracted English-French noun phrase (NP) correspondences [26]. The method used an HMM-based POS tagger and a finite-state NP recognizer to extract NPs in each language and apply an EM-like algorithm to find correspondences. Kumano et al. also find NP correspondences between English and Japanese.

Smadja et al. extracted English-French collocations with Champollion system [40] which is based on Smadja's Xtract system [41]. The system uses a statistical method to find rigid collocation as well as flexible collocations are successfully extracted. First, it identifies English collocations and then gradually expands French translation words that statistically co-occur with the English collocations. The method has been reported that 73% of the French translations of valid English collocations were judged to be good by three evaluators. The problem with this approach is that French collocations are not found exhaustively.

Kitamura et al. extracted rigid collocations of unrestricted length from English-Japanese parallel corpora [23]. Translation units are strings of content words (e.g. nouns, adjectives, verbs, adverbs) in both languages. The method is experimented with three different domains, achieving over 90% precision. The problem with this approach is that flexible collocations are not addressed.

Despite higher precision and robustness compared with alignment, translation knowledge by extraction does not exploit linguistic information, in particular structural dependency embedded in sentences.

2.3 Summary

This chapter reviewed a basic background in translation knowledge acquisition. We gave an overview of a data-driven MT paradigm and preliminaries, and explained the difference between alignment and extraction in the task of translation knowledge acquisition.

Chapter 3

Using Dependency Structures to Extract Phrase Correspondence

3.1 Introduction

In this chapter and the next chapter, we aim to demonstrate robust extractions of translation knowledge from parallel corpora by effective use of linguistic clues offered by the state-of-the-art statistical NLP tools. According to the classification presented in Chapter 2, our work is on **extraction**, but pays special attention to linguistic clues obtainable from statistical NLP tools. Our goal is to achieve robustness that extraction has and to incorporate richer linguistic notions that structural alignment use.

Figure 3.1 shows the framework of our approach to translation knowledge extraction. It consists of a monolingual step and a bilingual step. In the monolingual step, we generate possible translation units using linguistic clue for each half of paired sentence. These translation units are collected into candidate sets at the end of monolingual step. A pair extraction module takes two candidate sets from each language and finds correspondences based on co-occurrences in the parallel corpora.

This chapter describes a method to acquire phrase correspondence from sentence-aligned parallel corpora using statistically probable *dependency relations*, in other words, modifier-modifiee relations in a sentence. The distinct characteristics of our approach is that by the use of the statistical NLP for pre-

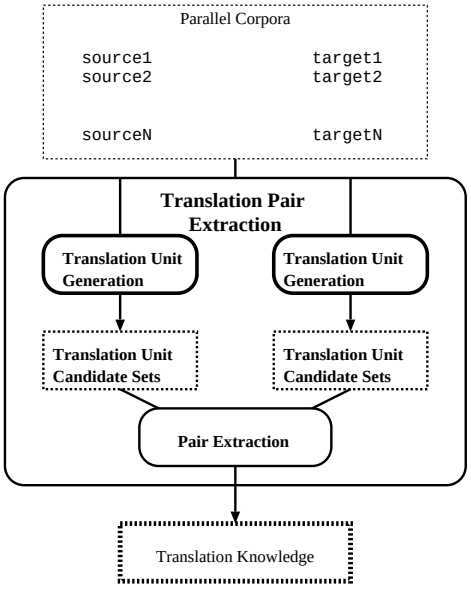


Figure 3.1. Overview of Translation Knowledge Extraction

processing parallel corpora to obtain linguistic clues, we maintain a high level of robustness that statistical NLP approaches enjoy.

The organisation of this chapter is as follows. In Section 3.2, we argue why dependency structures obtained from statistical parsers are suitable linguistic clues for language pairs that do not share the similar alphabets nor word ordering. In Section 3.3, we present translation unit generation using dependency relations. Then, the pair extraction algorithm is described in Section 3.4. Finally, we discuss the effectiveness of dependency relations for bilingual correspondence extraction in Section 3.6.

3.2 Statistically Probable Dependency Relations

We use statistical dependency parsers which are trained from syntactically annotated corpora to obtain statistical probable dependency relations. They are used as our linguistic information for translation unit generation.

There are two factors favouring the use of statistical probable dependency relations for phrasal translation pairs. One is a linguistic requirement: dependency relations seem an appropriate level of abstraction for languages pairs that do not belong to the same language families. The other is a technical advancement: the performance of statistical parsing has progressed rapidly and is now reaching nearly 90 % precision. The high perfomance opens up an opportunity to be used to tackle other NLP problems such as the translation pair acquisition.

3.2.1 Linguistic Requirement

Dependency relations can be obtained from dependency analysis based on dependency grammar which focuses on individual dependencies between words and phrases [20]. In this framework, every phrase is regarded as consisting of a governor and dependants, where dependants may be optionally classified further. The syntactically dominating word is selected as the governor, with modifiers and complements acting as dependants. Dependency structures are depicted as a directed acyclic graph, where arrows direct from dependants to governors.

As we saw in Chapter 2, most approaches are limited to word correspondences using primitive linguistic information obtainable from tokenization and POS tagging. The exception is Melamed's work where he conducted a comprehensive experiment on use of word position, alignment, cognates as linguistic heuristics [31]. However, in the case of English-Japanese pairs, some heuristics such as cognate and alignment are not applicable. Even for word position heuristics, the effectiveness is limited to compounding nouns as in "Natural/自然 Language/言語 Processing/処理".

Matsumoto and colleagues argued that dependency structures are preferable to phrase structures, as they abstract word ordering away [29]. Japanese language accomodates relatively free word ordering. For example, two sentences "会議が 2日 奈良で 開催される" and "2日 奈良で会議が 開催される" convey the identical meaning. The surface word orders are different, but they share the same dependency relations as illustrated in Figure 3.2.

Our justification of employing dependency relations as linguistic clues stems from the observation that the word ordering and positions may not necessarily coincide between the two languages, but the dependency structure between words

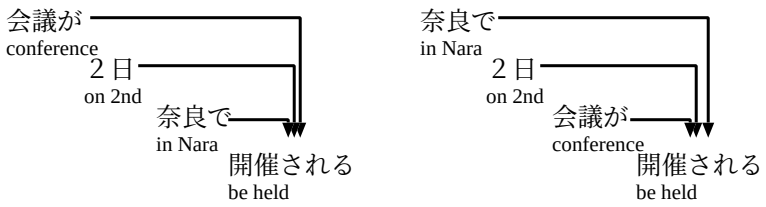


Figure 3.2. Different Word Orders, but Same Dependency Relations

will be preserved. We believe that dependency relations offer structural linguistic clues (syntactic information) and are effective for language pairs with different word ordering constraints.

3.2.2 Technical Advancement

Recently many statistical parsing models have been proposed for English [11], [36], [8] and for Japanese [14], [24]. Statistical parsing has an advantage over its rule-based counterpart owing to its wider coverage and reduced workload for grammar maintenance. But more importantly, it produces an output with some probabilistic confidence even for long or complex sentences.

Although statistical techniques cannot always provide a complete and correct parse for a sentence, they nevertheless produce many valid partial parsing results which are useful for acquiring phrase correspondences. With the use of statistical dependency parsers, we can handle sentences that are rejected by rule-based parsers, and use partial dependency trees of those sentences to acquire phrase correspondences.

3.3 Translation Unit Generation Using Dependency Relations

The process of generating translation units in each language is shown in Figure 3.3. A monolingual half of sentence in parallel corpora are fed into the *morpological analysis* to tokenize into words each annotated with the corresponding part-of-speech tag. The word-segmented sentence proceeds to the *chunking*. Ei-

ther a set of rules or a statistical model is used to determine whether or not a group of words are chunked into a segment. The chunked sentence is passed on to *dependency analysis* to construct a dependency tree of the sentence. Finally, the *subtree generation* produces a set of subtrees of the dependency tree each corresponding to a dependency-preserving translation unit. The process is reapeated for every sentence, and all the generated subtrees form a translation unit candidate set of the language.

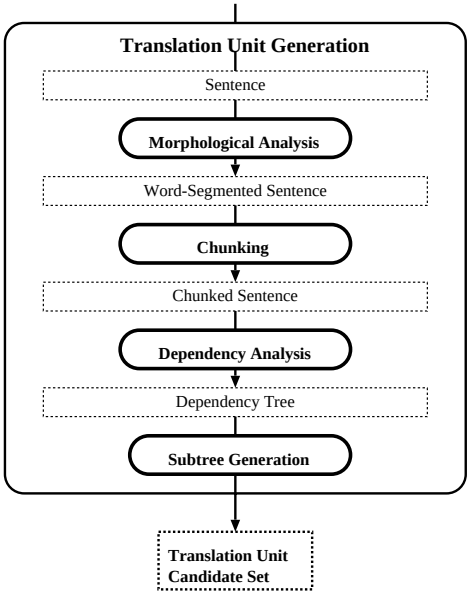


Figure 3.3. Translation Unit Generation

The originality of our work is the use of dependency analysis in the translation unit generation. We use the maximum likelihood model proposed in [14] for a statistical dependency parser. The model calculates the dependency probability between segments based on their co-occurrence and distance.

As the parser works for dependency relations between segments, we treat the following as a segment unit in chunking:

- English
  - a base NP: none of its child constituents are NP

  - a preposition or conjunction with the succeeding base NP
  - main verbs with preceeding auxiliary verbs

- Japanese
  - a bunsetsu: one or more content words optionally followed by function words

The goal of subtree generation is to generate meaningful translation units from a dependency tree. However, dependency analysis faces with the problem of ambiguity in that the correct dependency relation for every segment may not be determined. Even with the state-of-the-art statistical parses, the sentence precision is only 50%.

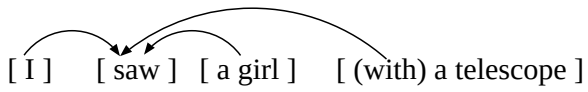
However, in our problem setting, we do not require completely correct parses for all paired sentences unlike structural alignment. Rather, we desire to generate as many correct partial subtrees as possible from parallel corpora. Hence, we deliberately supply multiple parses for ambiguous dependency relations. This will hopefully increase the number of partially correct subtrees in the translation unit candidate sets.

We generate translation units using the three models described below. The purpose is to apply statistically probable dependency relations to translation unit generation and to examine to what extent statistically probable relations are useful.

3.3.1 Model A: Best-one

The best-one model uses only the most likely (i.e. statistically best) dependency relations obtained from the statistical dependency parser. At most one dependency is allowed for each segment.

Figure 3.4 shows translation units generated by the best-one model. The translation units generated by the best-one model correspond to substrees of sentences. We build the translation units from a single segment (i.e. nodes of dependency tree) to a subtree composed of at most 3 segments. For translation units comprised of 3 segments, there are two different subtree types and "<T>" and "<L>" are annotated to distinguish the difference.



size 1) {I, saw, girl, telescope}

size 2) {I_saw, girl_saw, with-telescope_saw}

size 3) {I_girl_saw<T>,
I_with-telescope_saw<T>,
girl_with-telescope_saw<T>}

Figure 3.4. Best-one Model

We retain functional words in the translation unit if the segment is not the governor of the dependency relation. In the example, the segment `with a telescope` is a daughtor to the segment `saw`, and thus the translation unit contains the functional word `with`. This is based on the linguistic intuition that the difference is notable in `saw sb/sth` **with** `a telescope` and `saw sb/sth` **at** `the station`, but not so in **at** `the station of Shinkansen line` and **in** `the station of Shinkansen line`.

The translation unit are expressed as a daughtor-governor relation. Hence, the word ordering in the original sentences are not necessarily followed like `girl_saw` in the example.

3.3.2 Model B: Ambiguous

The ambiguous model uses dependency relations above the confidence score of 2. Figure 3.5 shows translation units generated by the ambiguous model. In this example, dependency relations for `with a telescope` are ambiguous and both probable parses are considered in translation unit generation.

The $(k+1)$th dependency relation for $k \geq 1$ is also included if

$$\frac{\texttt{prob(kth ranked dependency)}}{\texttt{prob((k + 1)th ranked dependency)}} \leq 2$$

Multiple dependencies may be considered for each segment. The translation units generated by the ambiguous model also correspond to substrees of sentences. In fact, the ambiguous model is a superset of the best-one model.

size 1) {I, saw, girl, telescope}

size 2) {I_saw, girl_saw, **with-telescope_saw**, **with-telescope_girl**}

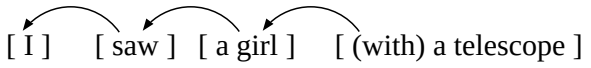
size 3) {I_girl_saw<T>,
I_with-telescope_saw<T>,
girl_**with-telescope_saw**<T>,
with-telescope_girl_saw<L>}

Figure 3.5. Ambiguous Model

In this model, more likely dependency relations will appear more frequently given a large corpus, which, in turn, has an effect of boosting the correlation score in translation pair extraction. The purpose of this model is to examine if ambiguity in dependency analysis will be resolved in pair extraction by supplying alternative parses.

3.3.3 Model C: Adjacency

The adjacency model uses only adjacency relations between segments. Thus, for any segment, its only dependency is the immediately preceeding segment as illustrated in Figure 3.6.



size 1) {I, saw, girl, telescope}

size 2) {saw_I, girl_saw, with-telescope_girl }

size 3) {girl_saw_I<L>,
with-telescope_girl_saw<L>}

Figure 3.6. Adjacency Model

Unlike the best-one model and the ambiguous model, the translation units

generated by the adjacency model correspond to substrings of sentences. In the adjacency model, structural clues are ignored and word ordering constraints is the dominant influencing factor. This model is included for comparative purposes.

The ranslation unit candidate sets thus generated become the inputs for pair matching described in the next subsection.

3.4 Translation Pair Extraction

Our strategy is to allow aggressive overlaps in translation unit candidate sets where ambiguity exists which is balanced by conservative correspondence discovery in the pair extraction algorithm.

The pair extraction algorithm is based on Kitamura and Matsumoto [23]. Pair matching of translation units in candidate sets is a combinatorial problem. The algorithm is controlled by a threshold. It first collects translation units from candidate sets that occur more than the threshold times. Then, it calculates the correlation between all possible pairs from the two candidate sets and extracts pairs starting with the higher score. When no more pairs are found, the threshold is lowered gradually to expand translation units under consideration. The same extraction process is repeated until the minimum threshold.

The threshold in the pair extraction algorithm has two roles. One is a threshold for the co-occurrence frequency. By setting the minimum threshold, the algorithm does not find translation pairs either of which independent frequency is less than the pre-defined minimum threshold. The other role is a threshold for the correlation score. By lowering the threshold in a stepwise manner, translation pairs are extracted from most correlated ones to least correlated ones.

The correlation of two words can be estimated by the positional distribution of the translation units in the parallel corpora (cf. Chapter 2). We choose the weighted Dice coefficient proposed by Kitamura and Matsumoto [23] which is defined as:

$$sim(p_e, p_j) = (\log_2 a)\frac{2a}{(a+b)+(a+c)}$$

where $(a+b)$ and $(a+c)$ are the number of occurrences in Japanese and English corpora respectively and a is the number of co-occurrences. It is calculated by

the number of times that appeared independently in the corpus and the number of times that co-occur in the parallel corpus.

Matsumoto give a detailed analysis on several correlation measures used in translation pair acquisition from parallel corpora [30]. Our choice of the weighted Dice coefficient as a similarity measure is based on the results reported in the literature.

According to the literature, Church and colleagues used mutual information and reported that it was a good estimate for relatively frequent events but suffered from overestimation for infrequent events [10]. Smadja et al. also pointed out the problem and used an alternative similarity measure namely, a Dice coefficient which ranges betwenn 0 and 1 [40]. Kitamura and Matsumoto note that the same correlation score will be given regardless of the co-occured frequency of the pair in the parallel corpora. For example, the score 0.6667 will be calculated for a frequent pair ($a = 200$, $b = 100$, $c = 100$, $d = 199,600$) and an infrequent pair ($a = 2$, $b = 1$, $c = 1$, $d = 199,996$). In order to discriminate frequent pairs from infrequent ones, they add the promoting weights for frequent pairs, giving the correlation score 5.0959 for the frequent pair while 0.6667 for the infrequent pair.

We now describe the pair extraction algorithm below.

1. For each English translation unit p_e in the English candidate set, store sentence positions in which p_e is found[1]. Delete any English translation unit p_e from the English candidate set that appears less than the predefined minimum threshold f_{min}.

2. Apply the above operation for each Japanese translation unit p_j in the Japanese candidate set.

3. Repeat the following until the current threshold f_{curr} reaches the predefined minimum threshold f_{min}.

   (a) For each pair of an English translation unit p_e and a Japanese translation unit p_j appearing at least f_{curr} times independently, identify the most likely correspondences according to the correlation scores.

[1]These sentence positions are used to calculate co-occurrence and independent occurrence between the English translation unit p_e and the Japanese translation unit p_j.

- For an English translation unit p_e, obtain the plausible candidate subset PJ = { p_{j1}, p_{j2}, ..., p_{jn} } such that sim(p_e,p_{jk}) > $\log_2 f_{curr}$ for all k. Similarly, obtain the plausible candidate subset PE for a Japanese translation unit p_j.
- Register (p_e,p_j) as a translation pair if

$$p_j = \underset{p_{jk} \in PJ}{\texttt{argmax}} \; sim(p_e, p_{jk})$$
$$p_e = \underset{p_{ek} \in PE}{\texttt{argmax}} \; sim(p_j, p_{ek})$$

  The correlation score of (p_e,p_j) is the highest among PJ for p_e and PE for p_j.

(b) Filter out the co-occurred sentence positions for p_e, p_j, and their overlapped translation units.

(c) Lower f_{curr} if no more pairs are found.

Several tactics are incorporated to overcome the combinatorial explosion. First, it is a greedy algorithm which means that a translation pair determined in the early stage of the algorithm will never be considered again. With a stepwise lowering of the threshold, the algorithm extracts translation pairs with higher score to lower ones.

Secondly, a filtering process is incorporated. Figure 3.7 illustrates filtering for a sentence pair (`I saw a girl in the park`, 私は公園の少女を見た). A set of candidates derived from English is depicted on the left, while that from Japanese is depicted on the right. Once two candidates (e.g. `I_girl_saw(T)` and /私_少女を_見た `(T)`) are designated as a translation pair, the the subcomponents of one translation unit (e.g. `I`, `I_saw`) are not be paired up with the subcomponents of the translation unit in the other language (e.g. 私, 私_見た); See Figure 3.7, where discarded subcomponents are marked with a dotted line. The operation effectively discards the matched pairs and causes the recalculation of the correlation scores in the proceeding iterations.

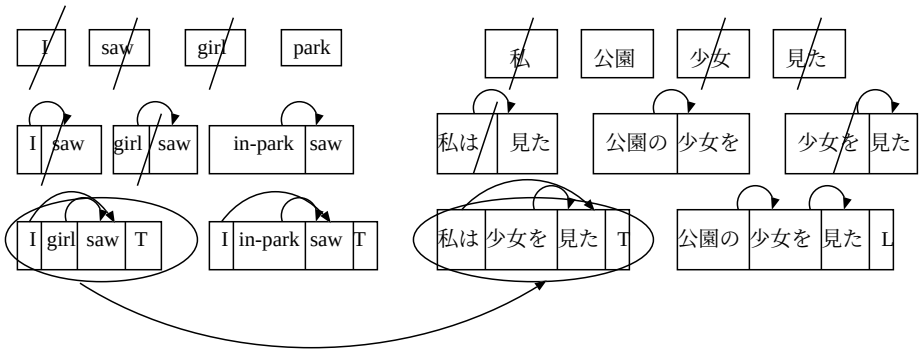


Figure 3.7. Filtering: (I, 私),(saw, 見た),(girl, 少女),(park, 公園)

3.5 Experiment and Results

We used 9268 sentences from English–Japanese business letter samples which were already aligned. The NLP tools used to obtain candidates are summarised in Table 3.1.

Task	Tool	Reported precision
POS(E)	ChaSen2.0	96%
POS(J)	ChaSen2.0	97%
chunking(E)	SNPlex1.0	rule-based
chunking(J)	Unit	rule-based
dependency(E)	edep	trial system
dependency(J)	jdep	85–87 %

Table 3.1. NLP tools used in this experiment

Parameter setting were as follows: The threshold of occurrence is adjusted according to the equations below. The threshold f_{curr} is initially set to 100 and is gradually lowered down until it reaches the minimum threshold f_{min} 2. All parameters were empirically chosen.

$$f_{curr} = \begin{cases} f_{curr}/2 & (f_{curr} > 20) \\ 10 & (20 \geq f_{curr} > 10) \\ f_{curr} - 1 & (10 \geq f_{curr} \geq 2) \end{cases}$$

Results are evaluated in terms of precision. The correctness of acquired phrase correspondence was judged by a speaker of English and Japanese (Japanese native). Precision for each model is summarised in Tables 3.2, 3.3 and 3.4. f_{curr} stands for the threshold. **e** is the number of extracted phrase correspondences found at f_{curr}. **c** is the number of correct correspondences found at f_{curr}. **acc** is the ratio of correct ones to extracted ones at f_{curr}. The accumulated results for **e**, **c**, and **acc** are indicated by **'**.

To examine the characteristics of each model, we relax constraints to include equality for the plausible candidate sets PJ such that $\mathrm{sim}(p_e,p_{jk}) \geq \log_2 2$ for all k, and PE such that $\mathrm{sim}(p_{ek},p_j) \geq \log_2 2$ for all k are also considered. This means that tentative translation pairs with the correlation score of 1 (i.e. $\log_2 2$) is also taken into account . The results of this relaxation are marked by asterisks "*" in the tables.

f_{curr}	**e**	**c**	**acc**	**e'**	**c'**	**acc'**
25	6	6	100.00	6	6	100.00
12	7	7	100.00	13	13	95.00
10	7	6	85.71	20	19	95.83
9	4	4	100.00	24	23	92.30
8	13	13	100.00	37	36	97.29
7	13	10	76.92	50	46	92.00
6	20	19	95.00	70	65	92.85
5	29	29	100.00	99	94	94.94
4	72	67	93.05	171	161	94.15
3	164	150	91.46	335	311	92.83
2	461	414	89.80	796	725	**91.08**
(*2	474	264	55.69	1270	989	77.93)

Table 3.2. Precision: Best–one model

Random samples of correct and near–correct translation pairs are shown in Table 3.5, Table 3.6 respectively. Extracted translation pairs were matched against the original corpora to restore their word ordering. This restoration is done manually this time, but can be automated with a small modification in our algorithm. In the tables, "+" indicates a segment-separator and "_" indicates a

f_{curr}	**e**	**c**	**acc**	**e'**	**c'**	**acc'**
25	6	6	100.00	6	6	100.00
12	7	7	100.00	13	13	100.00
10	7	6	85.71	20	19	95.00
9	4	4	100.00	24	23	95.83
8	13	13	100.00	37	36	97.29
7	13	11	84.61	50	47	94.00
6	19	18	94.73	69	65	94.20
5	29	29	100.00	98	94	95.91
4	73	68	93.15	171	162	94.73
3	126	118	93.65	297	280	94.27
2	468	432	91.50	765	712	**93.07**
(*2	759	256	33.72	1524	968	63.51)

Table 3.3. Precision: Ambiguous model

f_{curr}	**e**	**c**	**acc**	**e'**	**c'**	**acc'**
25	6	6	100.00	6	6	100.00
12	7	7	100.00	13	13	100.00
10	7	6	85.71	20	19	95.00
9	4	4	100.00	24	23	95.83
8	13	13	100.00	37	36	97.29
7	13	10	84.61	50	46	92.00
6	19	18	94.73	69	64	92.75
5	29	29	100.00	98	93	94.89
4	73	68	93.15	171	161	94.15
3	126	114	93.65	297	275	92.59
2	484	419	86.57	781	694	**88.86**
(*2	496	280	56.45	1277	974	76.27)

Table 3.4. Precision: Adjacency model

English	Japanese	score
thank+you	ありがとう	4.7037
consultations+include	協議_に_は+含める	2.3219
apply+for_the_position	職_に+応募_いたす	2.2157
thank+you+in_advance	前もって+お願い+申し上げる	1.6000
not+hesitate+to_contact	遠慮なく+ご連絡	1.6000
be+enclosed+a_copy	1_部_同封_いたす	1.0566
be_writing+to_let+know	書状_をもって+お知らせ_いたす	1.0566
applications+include	用途_に_は+ある	1.0000
upcoming_borard+of_director_s´_meeting	次回_の+取締役_会	1.0000
will_have+to_cancel	中止_せ_ざる_を+得_なく+なる	1.0000
have+high_hope	大いに+期待_する	1.0000
business+is_expanded	商売_は+発展_する	1.0000
we+have_learned+from_your_fax	貴_ファックス_で+知る	1.0000
leaving+in+about_ten_days	約_1_0_日_後+出発	1.0000
get+you+in_close_business_relationship	緊密_な+取引_関係_を+築く	1.0000
we+are_inquiring+regarding	に関し+お尋ね_いたす	1.0000
pay+special_attention	特別_の+注意_を+払う	1.0000

Table 3.5. Random samples of correct translation pairs in the best–one model.

morpheme-separator. Moreover, segments to be *deleted* in order to become a correct translation pair are written in *italic*, while segments to be **added** in order to become a correct translation pair are written in **bold**.

3.6 Discussion

The Tables 3.2, 3.3 and 3.4 show that both the best-one model (91.08%) and the ambiguous model (93.07 %) achieve better performance than the baseline adjacent model (88.86 %). Although the difference is marginal, the result implies that

English	Japanese
have_been_pleased+to_serve+as_thier_main_banker	主力_銀行_と+なる
be_held+at_hotel_new_ohtani	ホテル_ニューオータニ_で+開催_する
assets_position+*in_good_shape*	資産_状態
have_been_placed+into_our_file	私ども_の+ファイル
put+one_month_limit	1_ヶ月_の+期限
passed+on_past_tuesday	火曜日_に+亡くなら_れる

Table 3.6. Randam samples of near–correct translation pairs where score is 1.000.

applying statistically probable dependency relations to translation unit (subtree) generation is effective in translation pair extraction.

The difference between the dependency models (i.e. best-one and ambiguous) and the adjacency model increases when the threshold f_{curr} reaches 3. Translation pairs which are not found in the adjacency models are extracted in the dependency models, which we conjecture that dependency relations do not come to effect until relatively low thresholds.

Many partially variant sentences such as "遠慮なく+ご連絡 (not hesitate + to contact)", "遠慮なく+私に+ご連絡 (not hesitate + us + to contact)" and "遠慮なく+折り返し+当方に+ご連絡 (not hesitate + immediately + us + to contact)" are concentrated at relatively low thresholds. In the adjacent model, translation units preserve surface word ordering. Hence, translation units generated from such variant sentences are collected as distinct entities. On the other hand, dependency models generate dependency-preserving translation units. Thus, dependency models have advantageous for variant sentences which differ in surface word ordering but the same in linguistic dependency structure. In the above case, the common subtree "遠慮なく+ ご連絡 (not hesitate + to contact)" will be counted three times.

Based on the above observation, the dependency models have a boosting effect in collecting statistics of translation units, which in turn increases the similarity score and chances to be extracted as translation pairs.

However, the effect of supplying alternative parses in translation unit generation is questionable. Experimental results (Tables 3.2 and 3.3) did not fulfil our initial expectation that ambiguity in dependency analysis will be resolved in pair

extraction by supplying alternative parses. Although the precision of the ambiguity model slightly improved over the best-one model, the number of extracted translation pairs decreased even though candidate sets in the ambiguous model is a superset of the best-one model.

Our general strategy is to allow aggressive overlaps in translation unit candidate sets where ambiguity exists. This is hopefully balanced by conservative correspondence discovery in the pair extraction algorithm. The expected scenario in the ambiguous model is that more partially correct subtrees are generated in candidate sets which will boost the correlation score in pair extraction algorithm. In reality, however, supplying alternative parses leads to a rapid increase in the number of translation units. For example, the number of English translation units jumped from 10892 in the best-one model to 26333 in the ambiguous model, of which, 72% of translation units appear only twice in the parallel corpora. The statistically redundant parsing causes an increase of new translation units comprised of ambiguous dependency relations. The net result is that the size of candidate sets drastically increases at low thresholds, finding correspondences becomes more difficult, and the number of extracted translation pairs reduced.

Two issues are related to the structural ambiguity resolution stemmed from dependency analysis. The first issue is the accuracy of dependency parsing. The better accuracy a statistical parser achieves, the number of partially correct subtrees will increase in the best-one model. The relative merit of supplying alternative parses will be reduced. The performance of dependency parser used for the experiment achieved 85-7% (see Table 3.1), but the state-of-the-art statistical parsers at the time of writing is over 90 % [8]. The performance improvement of 5% may provide a different outlook on the effect of redundant parsing: the structural ambiguity may or may not be an issue to be taken up. The other issue is how far we should allow redundancy in dependency analysis. In the experiment, we picked a parameter of the statistical dependency parser arbitrarily for redundant parsing. Although the ambiguous model is a superset of the best-one model, it is not the *optimal* ambiguous model where only the real ambiguous dependency relations are allowed. The remaining research issue is a tuning the parameter of redundant parsing in order to see the full effect of alternative parses to tackle the structural ambiguity resolution.

3.7 Summary

In this chapter, we applied statistically probable dependency relations to extract word and phrasal correspondences. We achieved nearly 90 % precision with experiments of 9268 paired sentences. The experiments show that although statistical parsers are prone to some error, dependency relations serve as effective linguistic clues in translation knowledge extraction even for language pairs with different word ordering constraints. The unaccomplished goal is structural disambiguation arisen from dependency analysis. We allow aggressive overlaps in translation unit candidate sets where ambiguity exists. However, this was not sufficiently balanced by conservative correspondence discovery in the pair extraction algorithm. The problem of ambiguity still remains our research agenda.

Chapter 4

A Comparative Study on Candidate Generation

4.1 Introduction

Parallel corpora do not have any other linguistic information other than sentences being aligned. We used NLP tools to annotate linguistic clues from which translation units can be generated.

This chapter compares three models of translation units each of which uses different linguistic information: one with only word segmentation, one with chunk boundary, and one with word dependency. We use NLP tools to annotate such linguistic information. Table 4.1 shows robust NLP tools publicly available, most of which uses statistical techniques that are trained directly from annotated corpus. Although the state-of-the-art NLP tools do not offer 100% precision, there are many partially correct answers.

The purpose of this study is to examine effectiveness of linguistic clues obtainable from NLP tools, and to investigate the relationship between linguistic clues applied and translation knowledge extracted.

The organization of this chapter is as follows: in Section 4.2, we describe three models used to generate translation units. In Section 4.4, we present our experimental results. Finally, Section 4.5 analyzes characteristics of each model.

tool	usage	language	technique	performance
TnT	POS tagging	English	HMM	97 %
MXPOST	POS tagging	English	ME	96 %
ChaSen	POS tagging	Japanese	HMM	98 %
YamCha	chunker	English	SVM	94 %
YamCha	chunker	Japanese	SVM	96 %
Collins	parser	English	probabilistic	86 %
Charniak	parser	English	ME-like	90 %
Jdep	parser	Japanese	probabilistic	86 %
CaboCha	parser	Japanese	SVM	90 %

Table 4.1. Preprocessing Tools

4.2 Linguistic Heuristics for Translation Units

In this work, we focus on three kinds of linguistic clues obtainable from NLP tools. They are word segmentation (spaces), chunk boundary (squared brackets), and word dependency (arrows) shown in Figure 4.1.

NLP tools have reached to a practical level, but they never guarantee 100% precision. Moreover, they propagate ambiguities or errors to translation unit generation. For example, a morphological analyzer may produce an inconsistent word segmentation or a dependency parser may give unintended parses. In the next section, we propose three generative models of translation units that attempt to allow overlaps arisen from real ambiguity but to eliminate impossible overlaps.

4.3 Models of Translation Units

Three N-gram models of generating translation units, namely Plain N-gram, Chunk-bound N-gram, and Dependency-linked N-gram are compared. In Plain N-gram and Chunk-bound N-gram, translation units are built using only content (open-class) words. This is because functional (closed-class) words such as prepositions are insignificant in contiguous counponding words.

A word is classified as a functional word if it matches one of the following

```
 Pierre Vinken  ,  61 years   old  ,  will join   the board
 as   a nonexecutive director   Nov. 29  .

[Pierre Vinken] , [61 years] [old] , [will join] [the board]
[as] [a nonexecutive director] [Nov. 29] .
```

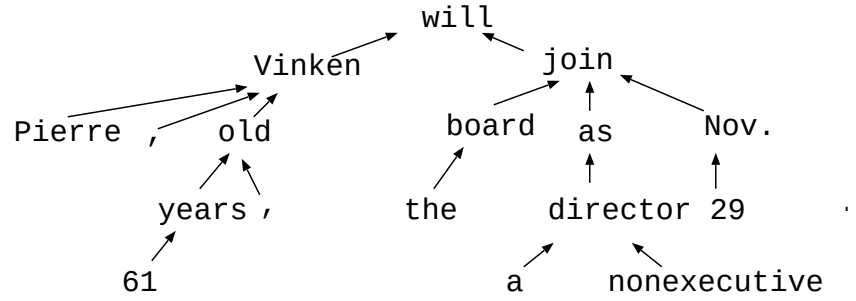


Figure 4.1. Linguistic Clues: Word Segmentation (top), Chunk Boundary (middle) and Word Dependency (bottom)

conditions. (The Penn Treebank (1991) part-of-speech tag set is used for English, whereas the ChaSen (2001) part-of-speech tag set is used for Japanese.)

part-of-speech(J) "名詞-代名詞" (noun-pronoun), "名詞-数" (noun-number), "名詞-非自立"(noun-dependent), "名詞-特殊" (noun-specific), "名詞-接尾-助動詞語幹"(noun-suffix-postparticle), "名詞-接尾-副詞可能" (noun-suffix-adverbial), "名詞-接尾-助動詞"(noun-suffix-model), "接頭詞" (prefix), "動詞-接尾"(verb-suffix), "動詞-非自立" (verb-dependent), "助詞"(postposition), "助動詞"(postparticle), "形容詞-非自立" (adjective-dependent), "形容詞-接尾"(adjective-suffix), "記号" (symbol)

part-of-speech(E) "CC", "CD", "DT", "EX", "FW", "IN", "LS", "MD", "PDT", "PR", "PRS", "TO", "WDT", "WD", "WP"

stemmed-form(E) "be"

symbols punctuations and brackets

In Dependency-linked N-gram, translation units are built not only with content words but also with functional words. This is because inclusion of function

words seems natural when considering word dependency. However, invalid translation units such as ones consisting solely of functional words are eliminated in advance.

4.3.1 Model 1: Plain N-gram

Plain N-gram was first proposed in Kitamura et al. [22]. The translation units generated in this model are word sequences from uni-gram to a given N-gram. Linguistic information used in this model is kept to a minimum, and is used as the baseline model in our work. The upper bound for N is fixed to 5 in our experiment.

4.3.2 Model 2: Chunk-bound N-gram

Chunk-bound N-gram is an extended version of plain N-gram which assumes prior knowledge of chunk boundaries. The definition of "chunk" has rooms for discussion. In our experiment, the definition for English chunk task complies with the CoNLL-2000 text chunking tasks [2] and the definition for Japanese chunk is based on "bunsetsu" in the Kyoto University Corpus [27].

Unlike Plain N-gram, Chunk-bound N-gram will not extend beyond the chunk boundaries. N varies depending on the number of words in a chunk[1].

4.3.3 Model 3: Dependency-linked N-gram

In Dependency-linked N-gram, a sentence is parsed into a word dependency tree. Since there was only a marginal advantage of the ambigious model over the best-one model (See Chapter 3), we only use the statistically best parse in Dependency-linked N-gram. We treat each branch of the word dependency tree as word sequences, and generate translation units from uni-gram to N-gram where N is the number of words in each branch of the word dependency tree. In our experiment, we used the probabilistically best parse result generated from the parser.

[1]The average number of words in English and Japanese chunks are 2.1 and 3.4 respectively for our parallel corpus.

Dependency-linked N-gram is distinct from the previous work. The granularity of translation units in Dependency-linked N-gram is finer, since it is based on word dependency instead of phrase dependency. We conjecture that the data sparseness problem will be resolved by focusing on word dependency.

Furthermore, Dependency-linked N-gram can generate 'non-contiguous' N-gram which cannot be generated from Plain N-gram with any N. Figure 4.2 shows a set of N-grams generated from a branch of a dependency tree in Figure 4.1. Bolded N-grams are ones that cannot be generated by Plain N-gram.

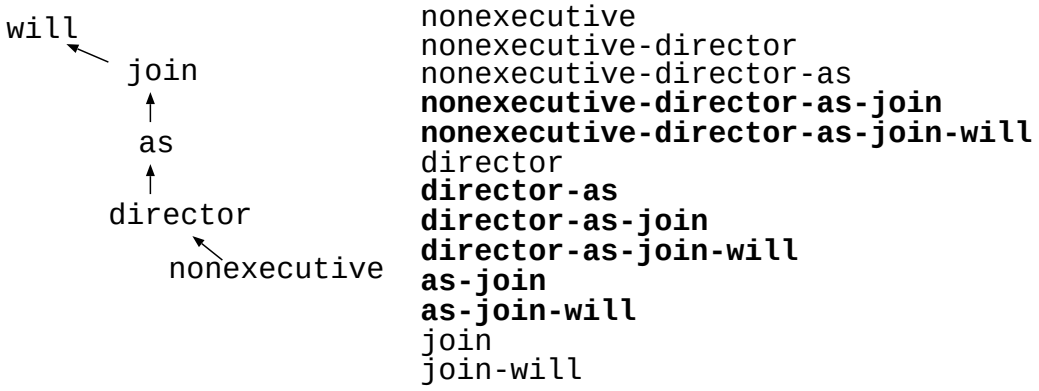


Figure 4.2. Dependency-linked N-gram

4.4 Experimental Results

We apply the same pair extraction algorithm described in the previous chapter with the same threshold lowering schedule, since our aim was to examine the effectiveness of each model in extraction.

Data for our experiment is 5000 sentence-aligned corpus from English-Japanese business expressions. 4000 sentences pairs are used for training and the remaining 1000 sentences are used for evaluation.

NLP tools used in this experiment are ChaSen, YamCha, CaboCha for Japanese text processing and TnT, YamCha, Collins parsers for English text processing. The choice of NLP tools was made based on performance (greater

than 85% accuracy) as well as ease of use.

Translation units that appear at least twice are considered to be in the candidate sets for the translation pair extraction algorithm. Table 4.2 shows the number of translation units found by each model. Note that translation units are counted not by token but by type.

model	English	Japanese
Plain	4286	5817
Chunk-bound	2942	3526
Dependency-linked	15888	10229

Table 4.2. Number of Translation Units

The result is evaluated in terms of accuracy and coverage. Accuracy is the number of correct translation pairs over the extracted translation pairs in the algorithm. This is calculated by type. Coverage measures applicability of the correct translation pairs for unseen test data. It is the number of tokens matched by the correct translation pairs over the number of tokens in the unseen test data. Accuracy and coverage roughly correspond to precision and percent correct respectively in Melamed (1995) [31]. Accuracy is calculated on the training data (4000 sentences) manually, whereas coverage is calculated on the test data (1000 sentences) automatically.

Stepwise accuracy for each model is listed in Tables 4.3, 4.4 and 4.5. f_{curr} indicates the threshold, i.e. stages in the algorithm. **e** is the number of translation pairs found at stage f_{curr}, and **c** is the number of correct ones found at stage f_{curr}. The correctness is judged by an English-Japanese bilingual speaker. **acc** lists accuracy, the fraction of correct ones over extracted ones by type. The accumulated results for **e**, **c** and **acc** are indicated by **'**.

Stepwise coverage for each model is listed in Tables 4.6 4.7 and 4.8. As before, f_{curr} indicates the threshold. The brackets indicate language: **E** for English and **J** for Japanese. **found** is the number of content tokens matched with correct translation pairs. **ideal** is the upper bound of content tokens that may be found by the algorithm; it is the total number of content tokens in the translation units whose co-occurrence frequency is at least f_{curr} times in the original parallel

f_{curr}	**e**	**c**	**acc**	**e'**	**c'**	**acc'**
100.0	0	0	n/a	0	0	n/a
50.0	0	0	n/a	0	0	n/a
25.0	1	1	1.000	1	1	**1.000**
12.0	2	2	1.000	3	3	**1.000**
10.0	5	5	1.000	8	8	**1.000**
9.0	4	4	1.000	12	12	**1.000**
8.0	3	3	1.000	15	15	**1.000**
7.0	6	6	1.000	21	21	**1.000**
6.0	9	9	1.000	30	30	**1.000**
5.0	17	16	0.941	47	46	**0.979**
4.0	31	31	1.000	78	77	**0.988**
3.0	64	64	1.000	142	141	**0.993**
2.0	349	256	0.733	491	397	**0.809**

Table 4.3. Precision: Plain N-gram

f_{curr}	**e**	**c**	**acc**	**e'**	**c'**	**acc'**
100.0	2	2	1.000	2	2	**1.000**
50.0	2	2	1.000	4	4	**1.000**
25.0	10	10	1.000	14	14	**1.000**
12.0	32	32	1.000	46	46	**1.000**
10.0	9	9	1.000	55	55	**1.000**
9.0	14	14	1.000	69	69	**1.000**
8.0	21	21	1.000	90	90	**1.000**
7.0	17	16	0.941	107	106	**0.991**
6.0	18	16	0.888	125	122	**0.976**
5.0	38	35	0.921	163	157	**0.963**
4.0	93	91	0.978	256	248	**0.969**
3.0	138	134	0.971	394	382	**0.967**
2.0	547	518	0.946	941	900	**0.956**

Table 4.4. Precision: Chunk-bound N-gram

f_{curr}	**e**	**c**	**acc**	**e'**	**c'**	**acc'**
100.0	1	1	1.000	1	1	**1.000**
50.0	5	5	1.000	6	6	**1.000**
25.0	11	10	0.909	17	16	**0.941**
12.0	27	26	0.962	44	42	**0.955**
10.0	17	15	0.882	61	57	**0.934**
9.0	12	12	0.882	73	69	**0.945**
8.0	25	25	1.000	98	94	**0.959**
7.0	35	34	0.971	133	128	**0.962**
6.0	32	31	0.968	165	159	**0.964**
5.0	49	48	0.979	214	207	**0.967**
4.0	96	92	0.958	310	299	**0.965**
3.0	189	184	0.973	499	483	**0.968**
2.0	1003	818	0.815	1502	1301	**0.866**

Table 4.5. Precision: Dependency-linked N-gram

corpora[2]. **cover** lists coverage. The prefix **i_** is the fraction of found tokens over ideal tokens and the prefix **t_** is the fraction of found tokens over the total number of both content and functional tokens in the data. For 1000 test parallel sentences, there are 14422 tokens in the English half and 18998 tokens in the Japanese half. **ideal** increases as the threshold is lowered, while **total** remains consistent.

4.5 Discussion

Chunk-bound N-gram and Dependency-linked N-gram obtained better results than the baseline Plain N-gram. The result indicates that chunk boundaries and word dependencies are useful linguistic clues in the task of translation knowledge extraction.

[2]Plain N-gram and Chunk-bound N-gram have content words only, while Dependency-linked N-gram also include function words. The reason for calculating "ideal" is that it is unfair to evaluate coverage of content-words-only models where function words are counted.

f_{curr}	found(E)	ideal(E)	i_cover(E)	t_cover(E)	found(J)	ideal(J)	i_cover(J)	t_cover(J)
100.0	0	445	0	0	0	486	0	0
50.0	0	1182	0	0	0	1274	0	0
25.0	46	2562	0.018	0.0015	46	2564	0.018	0.0011
12.0	156	4275	0.036	0.0051	146	4407	0.033	0.0037
10.0	344	4743	0.073	0.0113	334	4935	0.068	0.0086
9.0	465	4952	0.094	0.0153	455	5247	0.087	0.0117
8.0	511	5242	0.097	0.0168	501	5593	0.090	0.0129
7.0	577	5590	0.103	0.0190	567	5991	0.095	0.0146
6.0	744	5944	0.125	0.0245	734	6398	0.115	0.0189
5.0	899	6350	0.142	0.0297	891	6894	0.129	0.0229
4.0	1193	6865	0.174	0.0394	1195	7477	0.160	0.0307
3.0	1547	7418	0.209	0.0511	1549	8257	0.188	0.0398
2.0	2594	8128	0.319	0.0857	2617	9249	0.283	0.0674

Table 4.6. Coverage: Plain N-gram

f_{curr}	found(E)	ideal(E)	i_cover(E)	t_cover(E)	found(J)	ideal(J)	i_cover(J)	t_cover(J)
100.0	92	253	0.364	0.0072	92	328	0.280	0.0092
50.0	122	764	0.160	0.0095	122	746	0.164	0.0122
25.0	243	1510	0.161	0.0191	236	1423	0.166	0.0236
12.0	439	2590	0.169	0.0345	432	2515	0.172	0.0432
10.0	483	2829	0.171	0.0379	472	2739	0.172	0.0472
9.0	540	3009	0.179	0.0424	526	2911	0.181	0.0526
8.0	629	3168	0.199	0.0494	623	3086	0.202	0.0623
7.0	687	3348	0.205	0.0540	681	3256	0.209	0.0681
6.0	760	3539	0.213	0.0597	754	3464	0.218	0.0754
5.0	871	3803	0.229	0.0685	864	3748	0.231	0.0864
4.0	1076	4091	0.263	0.0846	1070	4059	0.264	0.1070
3.0	1402	4409	0.318	0.1102	1391	4423	0.314	0.1391
2.0	2007	4803	0.418	0.1578	2004	4865	0.412	0.2004

Table 4.7. Coverage: Chunk-bound N-gram

f_{curr}	found(E)	ideal(E)	i_cover(E)	t_cover(E)	found(J)	ideal(J)	i_cover(J)	t_cover(J)
100.0	78	1454	0.054	0.0061	78	1957	0.040	0.0078
50.0	170	2495	0.068	0.0133	170	2715	0.063	0.0170
25.0	264	3787	0.070	0.0207	278	3606	0.077	0.0278
12.0	394	5470	0.072	0.0309	408	4465	0.091	0.0408
10.0	503	5947	0.085	0.0395	515	4709	0.109	0.0515
9.0	558	6192	0.090	0.0438	570	4837	0.118	0.0570
8.0	665	6456	0.103	0.0523	680	4967	0.137	0.0680
7.0	801	6788	0.118	0.0629	814	5123	0.159	0.0814
6.0	900	7110	0.127	0.0707	911	5274	0.173	0.0911
5.0	1043	7520	0.139	0.0820	1065	5449	0.195	0.1065
4.0	1249	8055	0.155	0.0982	1274	5674	0.225	0.1274
3.0	1690	8690	0.194	0.1329	1686	5992	0.281	0.1686
2.0	2665	9664	0.276	0.2095	2703	6531	0.414	0.2703

Table 4.8. Coverage: Dependency-linked N-gram

Translation pairs extracted from Chunk-bound N-grams and those extracted from Dependency-linked N-grams seem to be in complementary relation. Chunk-bound N-grams extract locally contiguous translation pairs (rigid compounding words) with high precision, while Dependency-linked N-grams extract longer, sometimes non-contiguous, translation pairs where functional words are included.

Figure 4.3 shows the Venn diagram of translation pairs extracted by each model.

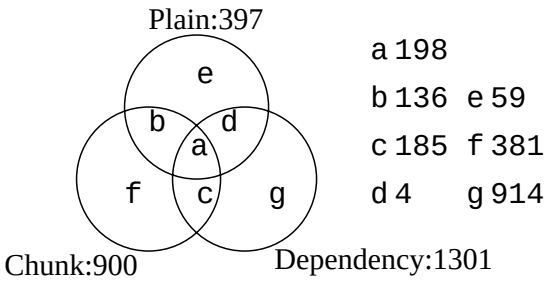


Figure 4.3. Distribution of Extracted Translation Pairs

model	English	Japanese
Plain	U.S.-Japan	日米
Plain	look forward (to) visiting	訪問 (-を-) 楽しみ
Plain	give information	資料提供
Chunk	Hong Kong	香港
Chunk	San Diego	サンディエゴ
Dependency	apply for position	職-に-応募-する
Dependency	be at your service	用命-に-従い-ます
Dependency	checking into matter	件-を-調査
Dependency	tell about matter	件-について-お知らせ
Dependency	free of charge	無料
Dependency	out of question	問題-外
Dependency	out of print	絶版

Table 4.9. Correct translation pairs

An interesting observation can be made in the distribution of Plain N-gram. 84% of Plain N-gram can be extracted by Chunk-bound N-gram. If we ignore the intersection of all three models, 34% of Plain N-gram are shared by Chunk-bound N-grams. In contrast, only 1% of Plain N-gram are in common with Dependency-linked N-gram exempting the intersection of all three models. Translation pairs extracted only by Plain N-gram is just 14 %. From these, we could conclude that translation pairs extracted by Plain N-gram can nearly be found by Chunk-bound N-gram.

Table 4.9 lists samples of correct translation pairs that are unique to each model.

Plain N-gram seems to extract longer translation pairs that are coincidently co-occurred. For example, there are many instances that 'look forward to visiting' and '訪問を楽しみ' co-occur. Plain N-gram generates word sequences of content words 'look_forward_visit' and '訪問_楽しみ' and all instances are counted for calculating similarity. As for Chunk-bound N-gram, the translation pair will not be extracted due to functional words 'to' and 'を'. In Dependency-linked N-gram, 'look forward/楽しみ' and 'visit/ 訪問' were extracted separately. A close

examination of parsed results reveals that, in some sentences, 'look forward' and 'to' were not dependency-linked.

Chunk-bound N-gram mostly extracts compounding NP (including named entity) that are of one-to-one correspondence. The result complies with our intuition, as translation units are enclosed by chunked boundaries. A reason why the other two models failed to extract those shown in the table is due to unnecessary generation of overlapped candidates. The filtering process in the extraction algorithm may not work effectively if too many overlapping candidates are generated.

Dependency-linked N-gram managed to extract translation pairs useful for tranlsation variation (e.g. 'checking **into** matter/件-**を**-調査' and 'tell **about** matter/ 件-**について**- お知らせ') Such extraction become possible mainly because function words are included in translation units of Dependency-linked N-gram.

From the above discussion, we see that chunk boundaries are useful linguistic clues especially in extracting compound NPs. This will be effective in preparing bilingual lexicon for a new domain. However, if we aim for longer translation pairs such as idiomatic expressions, word dependency plays an important role.

4.6 Summary

This chapter compares three models of translation units each of which uses different linguistic information: one with word segmentation only, one with chunk boundary, and one with word dependency. We use NLP tools to annotate such linguistic information which never guarantee 100% precision. Instead, we apply partially correct results that NLP tools give to generate meaningful translation units. Translation units with chunk boundary or with word dependency outperformed the previous baseline model, one with word segmentation only. Further analysis reveals that chunk boundaries are useful linguistic clues especially in extracting compound NPs. This will be effective in preparing bilingual lexicon for a new domain. However, longer translation pairs such as idiomatic expressions are better handled with by word dependency.

Chapter 5

A Data Mining Approach to Bilingual Lexicon Extraction

5.1 Introduction

In the last two chapters, we investigate how linguistic clues obtainable from statistical NLP tools such as chunk boundary and word dependency are effective in bilingual lexicon extractions.

In this chapter, we propose a data mining approach to extracting bilingual lexicon from parallel corpora. Like previous chapters, we aim to extract single word correspondences as well as bilingual collocations. The kind of bilingual collocations we consider are multiword expressions that appear frequently in a given domain, including light verb (e.g. have a question), proper names (e.g. New York), and terminological expressions.

Smadja et al. classified the types of bilingual collocations into rigid compounding collocations and flexible collocations [40] . The former rigid compounding collocations refer to consecutive word sequences. Proper names and some terminological expressions fall into this category. Haruno et al. argued that rigid compounding collocations may appear trivial, but more than half of useful collocations belong to this class [19]. The other category, flexible collocations, refer to multiword expressions with some intervening words. By nature, they are more difficult to extract from parallel corpora.

There are number of attempts to extract bilingual multiword expressions from

parallel corpus in the past. Early works such as [26] and [25] restricted their target to noun phrase correspondences. Later, the research community has extended its target to accommodate a more general notion of bilingual multiword expressions [40], [22] and [19]. Our work also aims to cover bilingual collocations more than just noun phrase correspondences.

Our goal is two-fold: first to extract single word correspondences and rigid compounding collocations with high accuracy, second to extract flexible collocations in a scalable manner. We adopt a data mining approach to achieve efficient generation and counting of bilingual multiword expressions in parallel corpora. Having generated bilingual candidates exhaustively, we extract single word correspondences, rigid compounding collocations and flexible collocations in greedy manner.

5.2 Related Works

In this section, we review these works from which our method is motivated.

Smadja et al. proposed a method to find both rigid compounding collocations as well as flexible collocations from the English-French Hansard corpus [40]. The method first identified English noun-noun, verb-noun, and adjective-noun collocations using their collocation extractor called Xtract [41]. Then it searches for correlated words in French by ranking the Dice similarity between an English collocation and a French word. They tested with three years of the Hansard corpus, and reported 73 % accuracy on average.

Kitamura et al. extracted English-Japanese word sequences of arbitrary length [22]. The method first generates from uni-gram to 10-gram of content words in each half of parallel corpora, and gradually finds correspondences by setting a threshold on a weighted Dice similarity. Experiment results with three distinct domains, each with approximately 10000 parallel sentences, showed that 80-90 % correctness.

Haruno et al. presented learning both rigid compounding collocations and flexible collocations from English-Japanese parallel corpora of stock market bulletin [19]. The method identifies useful rigid compounding collocations in each half of parallel corpora. This is achieved by first generation of monolingual chunks

through word-level sorting and then combining monolingual chunks that result a high (pointwise) mutual information score. Monolingual rigid compounding collocations thus found are used to construct bilingual rigid compounding and flexible collocations, again using mutual information. They experimented with 20000 parallel sentences and reported 70 % of rigid compounding collocations and 35 % of flexible collocations are considered to be correct.

Kitamura et al. achieved high performance but excluded flexible collocations. Smadja et al. selected English collocations of mid-range frequency first then determine corresponding French collocations. Haruno et al. also followed a two-steged extraction, first finding monolingual collocations then forming bilingual collocations. The primary reason for limiting a set of tentative flexible collocations in the monolingual stage is to avoid combinatorial explosion in generating and counting candidates for bilingual collocations.

In general, it is unrealistic to count non-contiguous n-gram (corresponding to flexible collocations in our task) independently in each half of parallel corpora and find correspondences between them. To illustrate this point, we count all contiguous n-grams (corresponding to rigid compounding collocations) and all non-contiguous n-grams, appearing at least twice in the corpora using the PrefixSpan algorithm [34] (See Section 5.3). Data is from our English-Japanese parallel corpora, containing 144743 English words and 186470 Japanese words.

Table 5.1 shows English and Japanese results. *freq* stands for the frequency of n-grams in the parallel corpora. For example, the first row stands for the number of n-grams that appear 10 times in the monolingual corpus. *contiguous only* means the number of contiguous n-grams whose frequency is *freq* and *non-contiguous included* means the number of contiguous and non-contiguous n-grams whose frequency is *freq*. As we see from the table, by including non-contiguous n-gram, the number of possible bilingual combination increases ($15040 \cdot 12913 \rightarrow 167100 \cdot 74403$), leading to a combinatorial explosion easily.

In this chapter, we adopt a data mining approach to achieve an efficient generation and counting for bilingual lexicon extraction. We view bilingual lexicon extraction from parallel corpora as sequential pattern mining from a large database, and apply the PrefixSpan algorithm to find a complete set of sequential patterns.

freq	*contiguous only (en)*	*non-contiguous included (en)*	*contiguous only (ja)*	*non-contiguous included (ja)*
10	139	234	131	309
9	155	282	165	382
8	190	370	244	503
7	246	563	292	757
6	351	807	422	1067
5	569	1401	579	1747
4	912	2624	912	3198
3	1871	6809	1972	7776
2	9193	152239	6805	56640
total	15040	167100	12913	74403

Table 5.1. Contiguous and Non-contiguous N-grams

5.3 PrefixSpan: Sequential Pattern Mining

In this section, we describe our analogy between bilingual lexicon extraction and sequential pattern mining. First we describe what we need to count in bilingual lexicon extraction. Then, we formally define the sequential pattern mining problem [3] and introduce the PrefixSpan algorithm [34] which we apply in this chapter.

5.3.1 Co-occurrence based Similarity

Most methods for estimating bilingual correspondences from parallel corpora start with the following intuition: words that are translations of each other are more likely to appear in corresponding sentences (i.e. to co-occur) than other pairs of unrelated words. This co-occurrence based similarity can be calculated by co-occurrence and independent frequencies of linguistically meaningful expressions in parallel corpora.

The relationship between co-occurrence and independent frequencies can be seen in a contingency table shown in Table 5.2. In the table, $C(x, y)$ means the number of times that x and y both appears in the parallel corpora. The table shows the co-occurrence frequency a and independent frequencies $a+b$ and $a+c$ of

two expressions e and j. The total number of sentence N is given by $a+b+c+d$.

	j	$\neg j$
e	$C(e,j) = \mathbf{a}$	$C(e,\neg j) = b$
$\neg e$	$C(\neg e,j) = c$	$C(\neg e,\neg j) = d$

Table 5.2. Contingency table

In the task of bilingual lexicon extraction, we need to count these values effectively. In order to accompolish this, we formulate the task as sequential pattern mining and apply the PrefixSpan algorithm for efficient generation and counting of bilingual lexicon candidates.

5.3.2 Sequential Pattern Mining

The sequential pattern mining problem was first introduced in [3] and is stated as follows.

Let $I = i_1, i_2, ..., i_n$ be a set of all **items**. An **element** is a subset of items, denoted as $(x_1 x_2 ... x_m)$, where x_k is an item. If an element has only one item, then the brackets are omitted, hence, (x) is written as x. A **sequence** s is denoted by $\langle s_1, s_2, ..., s_n \rangle$, where s_j is an element. An item can occur at most once in an element of a sequence, but can occur multiple times in different elements of a sequence.

The number of instances of items in a sequence is called the **length** of the sequence. A sequence $\alpha = \langle a_1, a_2, ..., a_n \rangle$ is called a **subsequence** of another sequence $\beta = \langle b_1, b_2, ... b_m \rangle$ and β is a **super sequence** of α, denoted as $\alpha \sqsubseteq \beta$, if there exist integers $1 \leq j_1 < j_2 \ ... \ < j_n \leq m$ such that $a_1 \subseteq b_{j_1}$, $a_2 \subseteq b_{j_2}$, ..., $a_n \subseteq b_{j_n}$.

A **sequence database** S is a set of tuples $\langle sid, s \rangle$, where sid is a **sequence_id** and s is a sequence. A tuple $\langle sid, s \rangle$ is said to contain a sequence α, if α is a subsequence of s, i.e., $\alpha \sqsubseteq s$. The **support** of a sequence α in a sequence database S is the number of tuples in the database containing α, i.e. $support_S(\alpha) = \big|\{\langle sid, s \rangle | (\langle sid, s \rangle \in S) \wedge (a \sqsubseteq s)\}\big|$. Given a positive integer ξ as the **support threshold**, a sequence α is called a ξ-frequent **sequential pattern**

in a sequential database S if the sequence is contained by at least ξ tuples in the database, i.e. $support_S(\alpha) \geq \xi$.

Given a sequence database and a support threshold ξ, the problem of **sequential pattern mining** is to find the complete set of sequential patterns in the database that appear at least ξ times.

The task of bilingual lexicon extraction (BLE) from sentence-aligned parallel corpora can be mapped straightforwardly to sequential pattern mining (SPM). Table 5.3 summarises our analogy between the two problems. We concatinate a English string and a Japanese string to form a bilingual sequence representing a parallel sentence in parallel corpora. Then, extracting bilingual lexicons is to mine sequential patterns from bilingual sequences.

SPM	BLE
sequential pattern	bilingual lexicon
sequence database	parallel corpora
sequence	bilingual sequence of words
element	words
item	features of words

Table 5.3. Analogy between SPM and BLE

There are some variations as to what is to be regarded as an element. For example, an element may have a lexical entry only. This implies a sequence will be of form $\langle l_1 l_2 ... l_n \rangle$ where l_i is a lexical entry of a word. Alternatively, we can have an element containing morphological features of the word such as part-of-speech and stemmed form. A resulting sequence will be of the form $\langle (l_1 p_1 s_1)(l_2 p_2 s_2)...(l_n p_n s_n) \rangle$ where l_i, p_i, and s_i are the lexical entry, the part-of-speech and the stemmed form of the word respectively.

We note a vital difference of patterns through the sequential pattern mining from contiguous N-gram generation. Although N-grams are simple to generate and count, they are at best local substrings in a sentence and fail to account for non-contiguous patterns that may not be consecutive but co-occur more than a chance. Furthermore, we need to predefine the value for N so that the majority of linguistically interesting patterns are included. The sequential pattern mining

overcomes the limitation, since it can cope with any number of intervening items in a sequence. It works by a simple frequency heuristic and no need to set the value for N. This is somewhat desirable for tentative bilingual lexicon generation, since the algorithm can cover rigid compounding collocations as well as flexible collocations without any restriction on the length of a sequence.

5.3.3 PrefixSpan

The PrefixSpan algorithm [34] employs a divide-and-conquire approach which is based on the observation that the frequent sequences are grown from frequent prefix subsequences. It uses frequent prefixes to partition the database into a set of smaller and disjoint databases each sharing the prefix and recursively grow subsequence fragment in each divided database. Fitst, we quote definitions of **prefix**, **projection**, **postfix**, and **projected database** ciritical to understanding of the PrefixSpan algorithm. In the discussion below, we assume that all items in an element are sorted alphabetically.

Definition (Prefix) Given a sequence $\alpha = \langle e_1 e_2 ... e_n \rangle$, a sequence $\beta = \langle e'_1 e'_2 ... e'_m \rangle$ $(m \leq n)$ is caled a **prefix** of α if (1) $e'_i = e_i$ for $i \leq m-1$; (2) $e'_m \subseteq e_m$; and (3) all the items in $(e_m - e'_m)$ are alphabetically after those in e_m'.
Definition (Projection) Suppose β is a subsequence of α $(\beta \sqsubseteq \alpha)$. A subsequence α' of sequence α $(\alpha' \sqsubseteq \alpha)$ is called a **projection** of α with respect to prefix β if (1) α' has a prefix β and (2) there exists no proper supersequence α'' if α' such that α'' is a subsequence of α and also has prefix β.
Definition (Postfix) Let $\alpha' = \langle e_1 e_2 ... e_n \rangle$ be the projection of α with respect to prefix $\beta = \langle e_1 e_2 ... e_{m-1}, e'_m \rangle$ $(m \leq n)$. A sequence $\gamma = \langle e''_m e_{m+1} ... e_n$ is called the **postfix** of α with respect to prefix β, denoted as $\gamma = \alpha/\beta$, where $e''_m = (e_m - e'_m)$. We also denote $\alpha = \beta \cdot \gamma$ or $\beta\gamma$
Definition (Projected database) Let α be a sequential pattern in sequence database S. The α-**projected database**, denoted as $S|_\alpha$, is the collection of postfixes of sequences in S with respect to prefix α.

We now introduce the lemma on the projected database in the original paper.

Lemma (Projected database) Let α, β and γ be sequential patterns in sequence database S and b and c are items such that $\beta = \alpha b$ and $\gamma = \alpha c$.

1. $S|_\beta = (S|_\alpha)|_b$
2. $support_S(\gamma) = support_{S|_\alpha}(c)$
3. The size of α-projected database cannot exceed that of S.

Now, the PrefixSpan algorithm takes input of a sequence database S and the minimum support threshold ξ and outputs a complete set of ξ-frequent sequential pattens. It calls *PrefixSpan*([], S) where *PrefixSpan* is stated as follows:

Function *PrefixSpan*(α, $S|_\alpha$)

- Parameters:
  a prefix α, and α-projected database $S|_\alpha$
- Method:
  1. Find a set of item B whose element b is such that $support_{S|_\alpha}(b) \geq \xi$
     (a) Append b to α to form an extended sequential pattern αb and output it
     (b) Construct the αb-projected database $(S|_\alpha)|_b$ for each αb and call *PrefixSpan*(αb, $(S|_\alpha)|_b$)

Figure 5.1 illustrates the relationship between sequential patterns (prefixes) and their projected databases. Suppose the algorithm has found $\langle a \rangle$ as a sequential pattern and its projected database is drawn as a subregion under the node $\langle a \rangle$ (shaded area). By scanning the $\langle a \rangle$-projected database, we find frequent items b_0, b_1, ..., b_n. So, we adjoin b_i to $\langle a \rangle$ to form a longer sequential pattern (prefix) and recursively mine smaller $\langle ab_i \rangle$-projected database.

We apply the PrefixSpan algorithm to count these co-occurrence frequency a and independent frequencies $a+b$, and $a+c$ efficiently. The relationship between the PrefixSpan and contengency table is illustrated in Figure 5.2. The count from a branch $\langle e_0 j_0 \rangle$ representing a bilingual pattern adds up to co-occurrence frequency a. The count from a branch $\langle e_0 \rangle$ representing an English pattern adds

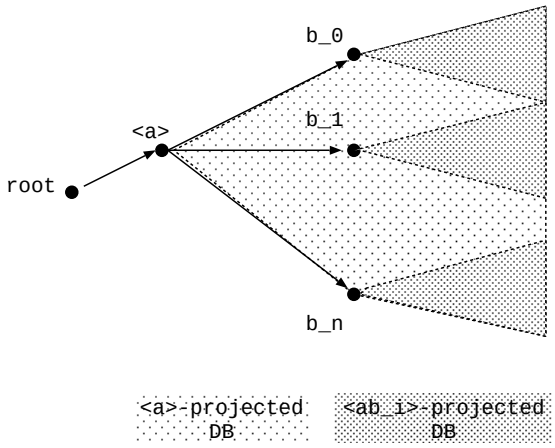


Figure 5.1. Prefix and Projected Database

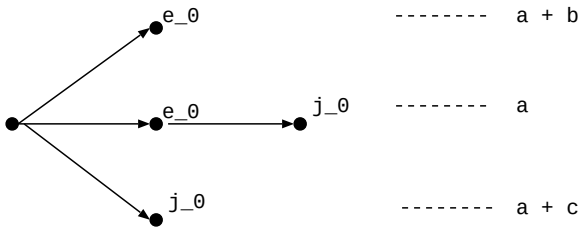


Figure 5.2. Contingency Table and PrefixSpan

up to independent frequency $a + b$, and similary, the count from a branch $\langle j_0 \rangle$ representing a Japanese pattern adds up to independent frequency $a + c$.

When the minimum support is k, the PrefixSpan algorithm mines all sequential patterns that appear at least k times. The logic of the PrefixSpan leads to the fact that if a bilingual sequential pattern $\langle e_0 j_0 \rangle$ is found with the minimum support k, then the constituent monolingual sequential patterns $\langle e_0 \rangle$ and $\langle j_0 \rangle$ should also be found, with their frequencies being at least k.

In effect, formulating the bilingual collocation extraction as above, the PrefixSpan algorithm offers a direct method of counting co-occurrence frequency (a) and independent frequencies ($a + b$ and $a + c$) simultaneously. Therefore, we can achieve a scalable and exhaustive counting of tentative bilingual collocations, rigid and flexible, that appear at least a predefined threshold.

5.4 Bilingual Lexicon Extraction

In this section, we describe our procedure of extracting bilingual lexicons using the PrefixSpan algorithm. Our method is illustrated in Figure 5.3 and comprises of three steps: (1) prepare bilingual sequences for sequential pattern mining, (2) generate tentative candidate sequential patterns and count their co-occurrence and independent frequencies by the PrefixSpan algorithm, and (3) greedy extraction of bilingual lexicons based on a weighted Dice coefficient measure. Unlike the previous pair extraction algorithm used in Chapters 3 and 4, bilingual tentative translation units are generated through sequential pattern mining.

5.4.1 Bilingual Sequences

Figure 5.4 illustrates preparation of a bilingual sequence from a parallel sentence. First, a morphological analyzer or a POS tagger are applied to each sentence in parallel corpora. Then, we use part-of-speech information to filter out functional words, primarily to reduce the number of elements in a sequence yet retaining the majority of translated information.

In our preliminary experiments, we focus on bilingual collocations of content words only, though arguably, interesting collocations such as phrasal verbs include

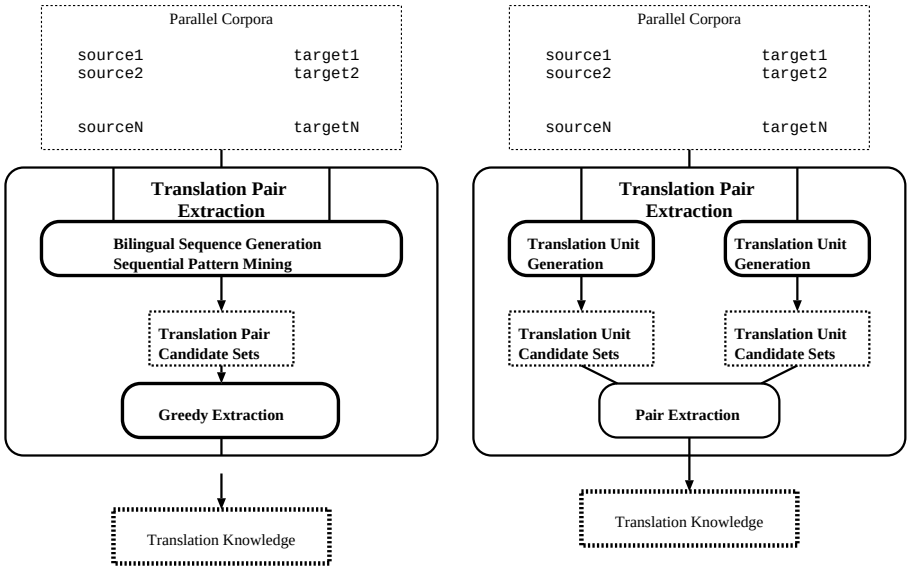


Figure 5.3. Proposed Method(Left), Kitamura et al.(Right)

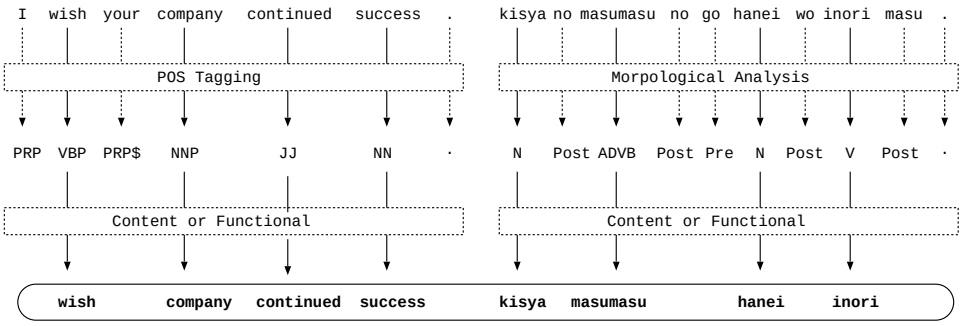


Figure 5.4. Bilingual Sequence

functional words (e.g. prepositions). Since it intends to be a preliminary experiment, the objective is to examine whether our proposed method is applicable or not.

We experimentally removed functional words to obtain the following practical effects. We observe that the likelihood of mining bilingual patterns consisting only of functional words is far greater than the chances of finding idiomatic expresions or phrasal verbs, and the length of bilingual sequences become much shorted. Even we pay the price for removing functional words, we still can obtain collocations of light verb, proper names and some domain-specific terminology, the goals we set initially.

We preprocess bilingual sequences, however, this operation can be handled with by the extended PrefixSpan described below.

5.4.2 Linguistic Extensions to PrefixSpan

The PrefixSpan algorithm efficiently generates and counts tentative bilingual collocations, but it tends to overgenerate sequential patterns most of which are not linguistically meaningful. In order to account for high frequency as well as linguistically meaningfulness in tentative candidate generation, we have a simple linguistic extension to the PrefixSpan algorithm.

We introduce a linguistic predicate $P(\alpha, b)$ to determine if b should be included in a set of items B in the step (1) of the algorithm (See Section 5.3.3). It now becomes:

1. Find a set of item B whose element b is such that $support_{S|_\alpha}(b) \geq \xi$ and $P(\alpha, b)$ is true.

Here are some possible extensions for $P(\alpha, b)$:

- *contiguous n-gram*: true if α and b forms a consecutive sequence
- *content word pattern*: true if b is a considered to be a content word
- *length bounded pattern*: true if length of α is less than a predefined length
- *clause bounded pattern*: true if no clause boundary (e.g. semicolons) between α and b in a sequence

We used *contigious n-gram* for Table 5.1, and the same bilingual patterns in the previous subsection 5.4.1 can be generated using *content word pattern* extension.

5.4.3 Greedy Extraction

Our procedure to extract bilingual lexicons in a greedy mannar is as follows. By lowering a threshold gradually, bilingual lexicons with higher ranked similarity can be extracted first.

1. Generate and count tentative bilingual lexicons using the PrefixSpan algorithm whose occurrence is at least k
2. For each bilingual pattern $\langle e_0 j_0 \rangle$, calculate a weighted Dice coefficient [22] defined as:

$$sim = \log_2 a \frac{2a}{(a+b)+(a+c)}$$

   where a is the co-occurence frequency of $\langle e_0 j_0 \rangle$, $(a+b)$ is the independent frequency of $\langle e_0 \rangle$ and $(a+c)$ is the independent frequency of $\langle j_0 \rangle$. Exclude a bilingual pattern $\langle e_0 j_0 \rangle$ that satisfy either of conditions below:

   - $b = 0$ and $c = 0$
   - $sim < 1.5$

3. Initialise a bilingual dictionary *dic*
4. Initialise a high threshold th and repeat below
   (a) Open a file from step 2
   (b) For each bilingual pattern $\langle e_0 j_0 \rangle$,
      i. Ignore if both English pattern $\langle e_0 \rangle$ and Japanese pattern $\langle j_0 \rangle$ have stronger correspondences in *dic*
      ii. Insert $\langle e_0, j_0 \rangle$ to *dic*
   (c) Close the file

(d) Terminate if $th = k$, else adjust th to a lower value and repeat the process

5. Output a bilingual dictionary *dic*

Reasons for excluding some tentative bilingual correspondences are as follows. First, the bilingual pattern $\langle e_0 j_0 \rangle$ whose b and c are zero means that all the parallel sentence in which $\langle e_0 j_0 \rangle$ exists must have both $\langle e_0 \rangle$ and $\langle j_0 \rangle$ and the other parallel sentences will have neither $\langle e_0 \rangle$ nor $\langle j_0 \rangle$. In such a case, many combinations of subsequences from the parallel sentences satisfying the minimum support k will be generated all having the identical similarity score. A longest match heuristic could be used, but an internal experiment shows that it degrades the output accuracy. The second condition is empirically imposed. An internal experiment shows that a high proportion of incorrect bilingual collocations $\langle e_0 j_0 \rangle$ has similarity scores lower than 1.5.

5.5 Experimental Results

Our experiment data is 9268 parallel sentences in business domain [44]. For preparing bilingual sequences, we used TnT for English POS tagging and ChaSen for Japanese morphological analysis. The average length of bilingual sequences are 14.4 after removing functional words. The PrefixSpan is implemented in C++, executed on Linux (Pentium III 1GHz) and took 52 mins to find all frequent sequences with the minimum support 2. They extracted 167100 English patterns, 74403 Japanese patterns and 69034794 bilingual patterns. The similarity calculation took 111 mins, and resulted 7,888 bilingual patterns remained. Finally, a greedy algorithm terminated in 10 mins, extracting 2,511 bilingual collocations in total.

Table 5.4 shows the performance of data mining approach. e, c, and *acc* stand for the number of extracted pairs, the number of correct pairs, and the accuracy respectively. $'$ are accumulated results for the extracted, the correct and the accuracy. We found that there are 274 pairs that are extracted at similarity between 1.5 and 2.0, of which only 83 are correct. By not extracting pairs with similarity below 2.0, the performance will boost to 66 %.

f_{curr}	e	c	acc	e'	c'	acc'
100	6	6	100.0	6	6	**1.000**
50	13	13	100.0	19	19	**1.000**
10	167	146	87.42	279	253	**0.9068**
9	27	22	81.48	306	275	**0.8986**
8	56	40	71.42	362	315	**0.8701**
7	57	52	91.22	419	367	**0.8758**
6	94	87	92.55	513	454	**0.8849**
5	102	89	87.25	615	543	**0.8829**
4	206	164	79.61	821	707	**0.8611**
3	428	258	60.28	1249	965	**0.7726**
2	1248	593	47.51	2497	1558	**0.6239**

Table 5.4. Accuracy: Data Mining Approach

f_{curr}	e	c	acc	e'	c'	acc'
100	0	0	n/a	0	0	n/a
50	0	0	n/a	0	0	n/a
10	9	9	1.0000	20	19	**0.9500**
9	7	7	1.0000	27	26	**0.9629**
8	10	10	1.0000	37	36	**0.9729**
7	12	11	0.9166	49	47	**0.9591**
6	25	25	1.0000	74	72	**0.9729**
5	29	28	0.9655	103	100	**0.9708**
4	70	68	0.9714	173	168	**0.9710**
3	114	109	0.9561	287	277	**0.9651**
2	646	490	0.7585	933	767	**0.8220**

Table 5.5. Accuracy of Kitamura et al.

	1	2	3	4	5	6
1	896/961	200/289	6/ 25	0/ 2	0/0	0/0
2	94/157	242/532	47/181	6/49	0/7	0/1
3	6/ 15	24/102	24/ 86	4/30	0/6	0/0
4	0/ 0	2/ 21	4/ 21	1/ 1	1/1	0/0
5	0/ 0	0/ 3	1/ 6	0/ 0	0/0	0/0
6	0/ 0	0/ 0	0/ 1	0/ 0	0/0	0/0

Table 5.6. Length of Correct/Extracted Patterns (data mining approach)

	1	2	3	4	5	6
1	636/662	55/75	2/4	0/0	0/0	0/0
2	25/42	35/76	5/23	0/3	0/0	0/0
3	1/ 5	5/19	1/ 7	0/3	0/0	0/0
4	0/ 1	1/ 8	0/ 1	0/2	0/0	0/0
5	0/ 0	0/ 2	0/ 0	0/0	0/0	0/0
6	0/ 0	0/ 0	0/ 0	0/0	0/0	0/0

Table 5.7. Length of Correct/Extracted Patterns (Kitamura et al.)

As a comparative purpose, we have conducted the experiment on Plain N-gram model (in this chapter, referred to as Kitamura et al.) on the same data. Table 5.5 shows the results. Although our data mining approach results in a lower accuracy than Kitamura et al., the number of extracted and correct bilingual extraction is far greater: at a threshold 3, our proposed method extracted 1249 with 77 % accuracy, while there are only 287 extracted though with 96 % accuracy in Kitamura et al.

Table 5.7 shows the distribution of translation pairs according to its length. The X direction corresponds to the number of words in English half of a translation pair, while the Y direction corresponds to the number of words in Japanese half of a translation pair. c/e means that the there are c correct out of e extracted. Kitamura's method extracted 636 single word correspondences with the accuracy of 96 % as well as 130 bilingual collocations with the accuracy of 48 %.

type	English	Japanese	similarity
rigid	look forward	楽しみ	33.6329
rigid	Los Angeles	ロサンゼルス	22.3214
rigid	foreign exchange	外国 為替	16.6154
rigid	enclosed envelope	同封 封筒	7.71429
rigid	annual report	年次 報告	7.11111
rigid	earliest convenience	都合 つき 次第	4.26667
flexible	wish ... success	成功 お祈り	13.6226
flexible	let ... know	お知らせ	40.209
flexible	thank ... letter	手紙 ... ありがとう	11.5714

Table 5.8. Sample Bilingual Collocations

On the other hand, our method extracted for 896 single word correspondences with the accuracy of 93 % as well as 662 bilingual collocations with the accuracy of 43 %.

Comparing results between the data mining approach and Kitamura et al., it is one win, one lose. The advantage of the data mining approach is the increased coverage, in that the the number of extracted and correct translation pairs and the proportion of multiword expressions are greater. On the other hand, the drawback is a degrade in accuracy. This is due to a simple greedy method we employ. The extraction still gets possible combinations of subsequences with the same score, even though translation pairs with zero independent frequencies are empirically excluded. In order to discriminate such cases, a heuristic, other than the longest match heuristic, may be incorporated.

Finally, Table 5.8 lists some samples of extracted collocations. Even dropping functional words, our method managed to extract rigid collocations as well as flexible collocations from parallel corpora.

5.6 Disccussion

Table 5.9 summarizes the results of the data mining approach against three translation units in Chapter 4 with the same 9268 parallel sentences. *acc* means the

	Data Mining	Plain	Chunk-bound	Dependency-linked
acc ($f_{curr} = 3$)	965/1249 (77%)	277/287 (96%)	860/898 (96%)	854/905 (94%)
acc ($f_{curr} = 2$)	1558/2497 (62%)	767/933 (82%)	1711/1873 (91%)	1844/2009 (92%)
single word	896/961 (93%)	636/662 (96%)	1494/1587 (94%)	1455/1526 (95%)
multiple word	662/1536 (43%)	130/271 (48%)	217/286 (76%)	389/483 (81%)
rigid	584	130	217	366
flexible	78	–	–	23

Table 5.9. Performance Comparison

overall accuracy, the same result presented in the previous section. *single word* is accuracy for single word correspondence only, while *multiple word* is accuracy for multi word expressions only. Out of correct *multi word*, the number of *rigid* compounding collocations and *flexible* collocations are listed. *Plain* N-gram (or Kitamura et al.), *Chunk-bound* N-gram and *Dependency-linked* N-gram uses the pair extraction algorithm described in Chapter 3, which is different from the extraction algorithm described in this chapter. Moreover, Dependency-linked N-gram does not include functional words (cf. Chapter 4) in order to set the same condition.

The trend is similar in the last section, low in accuracy but better in the quantity of extracted translation pairs. It should be noted that both Chunk-bound N-gram and Dependency-linked N-gram also have high proportion in single word correspondences: 87.3% and 78.9% respectively. Moreover, the proportion of flexible collocations in correct multiple word is greater in the data mining approach ($78/662 = 11.8\%$) than Dependency-linked N-gram ($23/389 = 5.9\%$). A possible explanation for better results in multi word expression is that data mining approach exhaustively generates co-occurred rigid and flexible multi word expressions and thus is robust against preprocessing (parsing) errors. However, this characteristic also leads to poor accuracy, in that it picks up too many 'near-miss' translation pairs. A careful design to incorporate linguistic constraints will be required so as to improve accuracy as well as retaining simplicity and exhaustiveness of the approaches.

Table 5.10 shows the time taken to generate tentative translation units. All

	Data Mining	Plain	Chunk-bound	Dependency-linked
Morphlogical Analysis (J)	7s	7s	7s	7s
POS Tagging (E)	9s	9s	9s	9s
Bunsetsu Identification(J)	–		2h46m	
Chunking (E)	–		2h21m	
Dependency Analysis (J)	–		–	2h46m
Parsing (E)	–		–	6h16m
PrefixSpan	52m	–	–	–
Extraction(Chapter 4)	–	3h32m	2h13m	3h56m
Extraction(Chapter 5)	2h1m	–		

Table 5.10. Preprocessing Time

preprocessing is executed on Linux (Pentium III 1GHz). As we see from the table, the data mining approach is more scalable to the other methods proposed in Chapter 4.

We conclude our discussion by describing possible usages of the approach. The approach is useful where manual check can be assumed. The method achieves a wider coverage for translation pairs, including near-miss ones which human can correct. It reduces the workload in identifying correspondences in parallel corpora.

5.7 Summary

In this chapter, we have adopted a data mining approach to bilingual lexicon extraction. We aim to extract single word correspondences as well as bilingual collocations. As for single word correspondences, we obtained a high performance of 93 % accuracy (896/961). Furthermore, our method has extracted a number of interesting bilingual collocations that co-occur frequently in the parallel corpora. The main point in this work is that by formulating the problem as a sequential pattern mining, we have achieved an efficient generation and counting of tentative bilingual lexicon which could not have been done before due to a combinatorial explosion.

Chapter 6

Conclusion

6.1 Summary

This thesis focuses on extracting *translation knowledge* from parallel corpora using NLP techniques. We demonstrate automatic methods of translation knowledge extraction which can aid translators or language learners. We present three works on this topic.

The first work uses statistically probable dependency relations obtained from parsers to acquire word and phrasal correspondences. The result showed that statistically probable dependency relations are effective in translation knowledge acquisition even for language pairs with different word ordering.

The second work compares three models of translation units each of which uses different linguistic information: word segmentation, chunk boundary, and word dependency. We found that chunk boundaries are useful linguistic clues in extracting compound noun phases which will be effective for extracting bilingual lexicons in the new domain. Furthermore, word dependency are also useful for longer translation pairs such as idiomatic expressions.

The final work proposes a data mining approach to extracting bilingual lexicons from parallel corpora. The task is viewed as sequential pattern mining and the PrefixSpan algorithm is applied for counting co-occurrence and independent frequencies efficiently. We demonstrated that a data mining approach can be applicable to our problem, especially suitable for extracting bilingual collocations that are often reported as difficult in previous work.

The contribution of our work to the research community using parallel corpora, and in general data-driven MT, can be summarized by comparing with the other related works.

First, our work is the first of its kind to apply the state-of-the-art statistical parsers publicly available in the NLP community to acquire translation knowledge from parallel corpora. The use of statistical parsers gives robustness to the approach. It is also applicable to the structural alignment where parse failures stemmed from rule-based parsers pose the major problem. Furthermore, we believe that our work is an interesting application for statistical parsing, and a better interface with statistical parsing in the treatment of syntactic ambiguity and possibility of redundant parsing may lead to an improvement on overall performance of translation knowledge acquisition.

Second, we propose practical methods for acquiring single- and multi- word correspondences. Early work sought mainly for single word correspondences [31]. However, for a non-segmented language such as Japanese and Chinese, word segmentation itself has ambiguity. Our methods, as with Kitamura et al. [23] addressed the word segmentation ambiguity by allowing aggressive overlaps in generation of tentative candidates which is then balanced by conservative correspondence extraction.

Finally, our work is unique in many respects from the other works which also addressed the problem of acquiring multiword correspondences from parallel corpora. Different from Kupeic [26], we do not limit our target to noun phrases. Our work does not depend on the word ordering and surface word distances, different from Smadja et al. [40]. This is important for English-Japanese pairs that do not share the same word ordering. Different from Haruno et al [19], our work performs an exhautive generation of tentative candidates in both languages but still avoids a combinatorial explosion with the greedy extraction of correspondences.

6.2 Future Directions

We conclude this thesis with the list of possible future research directions.

- Construct a translation aid system

As mentioned in introduction, we have concentrated on the acquisition part of translation knowledge base, and did not investigate the usefulness of such acquired translation knowledge in the real translation aid system. In order to examine the usefulness of our work in the real application setting, we need to construct a translation aid system.

- Construct an MT system that works by applying the translation pairs acquired from parallel corpora

  Similar to the previous point, our extracted translation pairs can be used for data-driven MT. Kitamura et al. [22] proposed a prototypical model, but they cried out for a better acquisition of translation lexicon and rules from parallel corpora. Since we improved the acquisition part a great deal in quantity and some in quality, it would be interesting to incorporate the improved translation pairs to examine its effect.

- Another view on resolving structural disambiguation

  Our work in Chapter 3 did not succeed in resolving structural disambiguation. No satisfactory answer can be given at this stage. If this still remains an issue to tackle in MT even with an improvement on parsing, then just supplying alternative parses is not sufficient and the other mechanism to allow essential parses only will be required.

- A fuller examination on bilingual collocation extraction using the PrefixSpan algorithm

  Chapter 5 presented a promising direction for bilingual collocation extraction using the PrefixSpan algorithm. However, there are a number of points to consider before we can claim its usefulness. The apparent shortcoming in our preliminary experiment is dropping functional words in generation of bilingual sequences. Interesting bilingual collocations, especially flexible collocations, do include functional words. We need to tackle this problem more so as to account for frequent as well as linguistically meaningful bilingual collocations.

- A further investigation into data mining approaches to translation knowledge extraction

In Chapter 5, we have adopted a data mining approach to bilingual lexicon extraction. Although the extracted translation patterns are limited to sequential patterns, we have demonstrated a good coupling between translation knowledge extraction and data mining. Recently, there are a number of attempts to mine structured patterns such as trees or graphs from a large semi-structured data [5]. Since richer translation rules are often represented in a tree or a graph, we would like to apply those techniques to directly extract translation rules from parallel corpora.

References

[1] Aligned hansards of the 36th parliament of canada release 2001-1a. Natural Language Group of the USC Information Sciences Institute. http://www.isi.edu/natural-language/download/hansard/index.html.

[2] *Text Chunking, CoNLL-2000 Shared Task Description.* http://lcg-www.uia.ac.be/conll2000/chunking/, 2000.

[3] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. 1995 International Conference of Very Large DataBases (VLDB'95)*, pp. 3–14, 1995.

[4] 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/.

[5] T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Sakamoto, and Arikawa S. Efficient substructure discovery from large semi-structured data. Technical report, Department of Informatics, Kyushu University, 2001.

[6] P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, pp. 263–311, 1993.

[7] P.F. Brown, J.C. Lai, and R.L. Mercer. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of ACL*, pp. 169–176, 1991.

[8] E. Charniak. A maximum-entropy-inspired parser. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 132–139, 2000.

[9] K. Church. Char_align: a program for aligning parallel texts at the character level. In *31st Annual Meeting of the Association for Computational Linguistics*, pp. 1–8, 1993.

[10] K. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *Computational Linguistics*, Vol. 16, pp. 22–29, 1990.

[11] M.J. Collins. Three generative, lexicalised models for statistical parsing. In *35th Annual Meeting of the Association for Computational Linguistics*, pp. 16–23, 1997.

[12] I. Dagan, K. Church, and W.A. Gale. Robust bilingual word alignment for machine aided translation. In *Proc. of Workshop on Very Large Corpora*, pp. 1–8, 1993.

[13] T. Dunning. Accurate methods for statistics of surprise and coincidence. In *Computational Linguistics*, Vol. 19, pp. 61–74, 1991.

[14] M. Fujio and Y. Matsumoto. Dependency analysis based on lexical collocation probability. In *IPSJ*, Vol. 40(12), pp. 4201–4212, 1999.

[15] P. Fung. A statistical view on bilingual lexicon extraction: from parallel corpora to non-parallel corpora. In *Lecture Notes in Artificial Intelligence*, Vol. 1529, pp. 1–17. Springer Publisher, 2000.

[16] P. Fung and K. McKeown. Finding terminology translations from non-parallel corpora. In *Proceedings of the 5th Workshop on Very Large Corpora*, pp. 192–202, 1997.

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

[18] R. Grishman. Iterative alignment of syntactic structures for a bilingual corpus. In *Proc. of 2nd Annual Workshop on Very Large Corpora*, pp. 57–68, 1994.

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

[20] R. Hudson. *Word Grammar*. Blackwell, 1984.

[21] H. Kaji, Y. Kida, and Y Morimoto. Learning translation templates from bilingual text. In *COLING-92: The 15th International Conference on Computational Linguistics*, pp. 672–678, 1992.

[22] M. Kitamura and M. Matsumoto. Automatic acquisition of translation rules from parallel copora. In *IPSJ*, Vol. 37(6), pp. 78–88, June 1996. In Japanese.

[23] M. Kitamura and M. Matsumoto. Automatic extraction of translation patterns in parallel corpora. In *IPSJ*, Vol. 38(4), pp. 108–117, April 1997. In Japanese.

[24] T. Kudo and Y. Matsumoto. Applying cascaded chunking to japanese dependency structure analysis (in japanese). In *SIG-NL-142*, pp. 97–104, 2001.

[25] A. Kumano and H. Hirakawa. Building an MT Dictionary from Parallel Texts Based on Linguistic and Statistical Information. In *COLING-94:The 15th International Conference on Computational Linguistics*, pp. 76–81, 1994.

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

[27] S. Kurohashi and M. Nagao. Kyoto university text corpus project(in japanese). In *NLP(Japan)*, pp. 115–118, 1997.

[28] Y. Matsumoto and M. Asahara. Ipadic users manual. Technical report, Nara Institute of Science and Technology, 2001.

[29] Y. Matsumoto, H. Ishimoto, and T. Utsuro. Structural matching of parallel texts. In *31st Annual Meeting of the Association for Computational Linguistics*, pp. 23–30, 1993.

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

[31] I.D. Melamed. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In *Proceedings of the Third Workshop on Very Large Corpora*, pp. 184–198, 1995.

[32] A. Meyers, R. Yangarber, and R. Grishman. Alignment of shared forests for bilingual corpora. In *The 16th International Conference on Computational Linguistics*, pp. 460–465, 1996.

[33] M. Nagao. A framework of a mechanical translation between japanese and english by analogy principle. In A. Elithorn and Banerji R., editors, *Artificial and Human Intelligence*, pp. 173–180, 1984.

[34] J. Pei, B. Han, J. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hau. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. of International Conference of Data Engineering (ICDE2001)*, pp. 215–224, 2001.

[35] R. Rapp. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting of ACL*, pp. 320–322, 1995.

[36] A. Ratnaparkhi. A linear observed time statistical parser based on maximum entropy models. In *Proc of 2nd Conf. on Emperical Methods in Natural Language Processing*, pp. 1–10, 1997.

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

[38] B. Santorini. Part-of-speech tagging guidelines for the penn treebank project. Technical report, LDC, 1991.

[39] S. Sato and M. Nagao. Toward memory-based translation. In *COLING-90: 13th International Conference on Computational Linguistics*, pp. 247–252, 1990.

[40] F. Smadja, K.R. McKeown, and V. Hatzuvassiloglou. Translating collocations for billingual lexicons: A statistical approach. In *Computational Linguistics*, Vol. 22(1), pp. 1–38, 1996.

[41] F. Smaja. Retrieving collocation from text: Xtract. In *Computational Linguistics*, Vol. 19, pp. 143–177, 1993.

[42] H. Somers. Example-based machine translation. In *Handbook of Natural Language Processing*, pp. 611–627. Marcel Dekker, 2000.

[43] H. Sumita and H. Iida. Experiments and prospects of example-based machine translation. In *ACL-91: 29th Annual Meeting of the Association for Computational Linguistics*, pp. 185–192, 1991.

[44] K. Takubo and M. Hashimoto, editors. *A Dictionary of English Bussiness Letter Expressions*. Nihon Keizai Shimbun, Inc, 1995.

[45] K. Yamada and K. Knight. A syntax-based statistical translation model. In *39th Annual Meeting of the Association for Computational Linguistics*, pp. 523–530, 2001.

Appendix

A Nikkei Business Letter Corpus

The parallel corpus used is a collection of business letter specimens and has already been aligned at sentence level[44]. There are 9268 paired sentences in total[1]. The sample translation of "balance" are selected below:

> In the selection of these two gentlemen, I am confident we have created a strong and well-balanced top management team.
>
> 両氏の任命により、当社は強力でバランスのとれた最高経営陣を作りあげたものと確信しています。
>
> They carry accounts with our Nihonbashi Branch, keeping a deposit balance of substantial size.
>
> 当該先は当行の日本橋支店に複数口座を持ち、かなりの額の預金残高があります。

As we see from the samples, the same word is used in multiple senses; the word "balance" is used in the business sense (e.g. "deposite balance/預金残高") as well as in the ordinary sense (e.g. "well-balanced/バランスのとれた")[2]

I treat tokens segmented by tokenizer (for English) and morphological analyzer (for Japanese) are words. Table 6.1 shows the average sentence length and a histogram of sentence length.

The number of words counted by token and by type, the distribution of part-of-speech is summarized in Table 6.2. English and Japanese POS tag annotation schemes are based on the Penn Treebank and IPADIC respectively[38][28].

[1] The original corpus contains 13577 paired sentences and 1610 paired terms. I have removed paired sentences which are not one-to-one correspondence.

[2] There are 28 sentences where the word "balance" appeared. Of which, only 2 sentences use it in the ordinary sense, and the remaining 26 sentences use it in the business sense.

length	English	Japanese
average	15.62	20.12
0-9	1182	455
10-19	5940	4286
20-29	2009	3490
30-39	132	939
40-	5	98

Table 6.1. Sentence Length

	English	Japanese
token	144743	186470
content tokens	67801	84984
function tokens	76942	101486
type	8715	8462

Table 6.2. Number of Words

List of Publications

Major Publications

Journal Paper

[1] Yamamoto, K. and Matsumoto, Y., "Translation Pattern Acquisition Using Dependency Structures," Journal of Information Processing Society Japan, vol.42, no.9, pp.2239–2247, September 2001 (in Japanese).

Book Chapter

[1] Yamamoto, K. and Matsumoto, Y., "Extracting Translation Knowledge From Parallel Corpora," Recent Advances in Example-based Machine Translation, the Kluwer series, Text, Speech and Language Technology, Kluwer Academic Publishers (to appear).

International Conferences (Reviewed)

[1] Yamamoto, K. and Matsumoto, Y., "Acquisition of Phrase-level Bilingual Correspondence using Dependency Structures," The 18th International Conference on Computational Linguistics, Saarbruken, Germany, July 2000, pp.933-939.

[2] Yamamoto, K. Matsumoto, Y. and Kitamura, M., "A Comparative Study on Translation Units for Bilingual Lexicon Extraction," 39th Annual Meeting and 10th Conference of the European Chapter of the Association for Computational Linguistics, Data-driven MT Workshop, Toulouse, France, July 2001, pp. 87-94.

Local Workshops

[4] Yamamoto, K. and Matsumoto, Y., "Translation Pattern Acqusition using Dependency Structures," IPSJ Technical Report, SIG-NL-137, June 2000, pp. 15-22 (in Japanese).

[5] Yamamoto, K. and Matsumoto, Y., "A Comparative Study on Translation Units," IPSJ Technical Report, SIG-NL-145, September 2001, pp. 15-22 (in Japanese).

Other Publications

International Conferences (Reviewed)

1. Takamura, H., Yamada, H., Kudoh, T., Yamamoto, K. and Matsumoto, Y. "Ensembling based on Feature Space Restructuring with Application to WSD," 6th Natural Language Processing Pacific Rim Symposium, Tokyo, Japan, November 2001, pp. 41-48.

Local Workshops

[4] Miyata, T., Yamamoto, K. and Matsumoto, Y., "English Dependency Parsing Based on Support Vector Machine," IPS Technical Report, SIG-NL-140, November 2000, pp. 135-142 (in Japanese).

[5] Takamura, H., Yamada, H., Kudoh, T., Yamamoto, K. and Matsumoto, Y. "Ensembling based on Feature Space Restructuring with Application to WSD," IPS Technical Report, SIG-NL-144, July 2001, pp. 83-90 (in Japanese).

Acknowledgments

Many people have kindly helped me in the three-year life at NAIST.

The one person to whom I am most indebted is my supervisor, Professor Yuji Matsumoto. He has introduced me the fundamentals of natural language processing, guided me the way to conduct research and joys of research. Without his constant encouragement, infinite patience and generous support, I would not have been able to complete the doctoral thesis.

I would like to express my gratitude to the members of my thesis committee: Prof. Kiyoshiro Shikano and Prof. Katsumasa Watanabe for their helpful suggestions and comments.

My special thanks also go to the past and the present staff members at Matsumoto laboratory. Dr. Edson T. Miyamoto has kindly corrected many of my poor English writings including the draft of this thesis. Dr. Masashi Shimbo has taken the burden of correcting my final presentation. Dr. Takashi Miyata has instructed me the basic NLP programming skills. Ms. Toshie Nobori has not only supported administrative matters but cheered me up in many occasions.

It is no doubt that my research has been supported by many colleagues at Matsumoto laboratory. Most work is inspired by fruitful discussion with Ms. Mihoko Kitamura. She has also been a supportive adviser of life in general. My last work owes greatly to Taku Kudo and Yuta Tsuboi who introduced me the field of data mining. I would like to thank Masahiko Fujio, Tatsuo Yamashita, Satoru Takabayashi, Kazuma Takaoka and ChaSen-members for offering various NLP tools, and machine administrative members to maintain PCs and Workstations at Matsumoto Laboratory. Furthermore, I thank Hiroyaru Yamada and Hiroya Takamura who explained me many of mysteries in machine learning. Last but not least, my appreciation goes to Masayuki Asahara for his valuable support

and criticism.

I would like to express my appreciation to Prof. Hiroto Yasuura of Kyusyu University, my former research director at ISIT/Kyushu. He has been a cupid for me to meet Prof. Yuji Matsumoto and is still a trusted mentor.

Finally, I would like to thank my parents, Katsuji Yamamoto and Ayako Yamamoto for their consistent encouragement and support.