Chapter 13

EXTRACTING TRANSLATION KNOWLEDGE FROM PARALLEL CORPORA

Kaoru Yamamoto
Graduate School of Information Sciences,
Nara Institute of Science and Technology
kaoru-ya@is.aist-nara.ac.jp

Yuji Matsumoto
Graduate School of Information Sciences,
Nara Institute of Science and Technology
matsu@is.aist-nara.ac.jp

Abstract

This chapter describes two studies concerning automatic extraction of translation knowledge from parallel corpora.

In the first study, we use statistically probable dependency relations to acquire word and phrasal correspondences. We obtained 90% precision using an English-Japanese parallel corpus of 9268 sentences in the business domain. The result showed that statistically probable dependency relations are effective in translation knowledge acquisition even for language pairs with different word ordering.

The second study compares three models of translation units, each of which uses different linguistic information: word segmentation, chunk boundaries, and word dependencies. The study investigates the relationship between the linguistic clues applied and translation knowledge extracted. We found that chunk boundaries are useful linguistic clues in extracting compound NPs which will be effective for extracting bilingual lexicons in the new domain. Furthermore, word dependencies are also useful for longer translation pairs such as idiomatic expressions.

We demonstrate that using statistical NLP tools, in particular, statistical dependency parsers, offers robustness to the approach as well as valuable linguistic clues that work effectively in extracting translation knowledge from parallel corpora of languages from different families.

Keywords: parallel corpora, lexical knowledge acquisition, models of translation units

1. Introduction

Recent advances in natural language processing (NLP) owe a lot 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 huge amount of linguistic data using computationally expensive statistical methods that could not have been employed before. We are now in the 21st century, and corpus-based NLP can be seen in every aspect of NLP, including POS tagging and parsing.

Machine translation (MT) is no exception to the above trend, with data-driven approaches to MT exemplified by Example-based MT (EBMT) and Statistical MT (SMT). The important assumption in these paradigms is that lexicons and rules for translation are somehow acquired from corpora which can then be compiled into an MT engine. As the translation process becomes more complicated, manual preparation of translation lexicons and rules becomes too cumbersome, so that it is desirable to extract translation knowledge automatically.

This chapter focuses on extracting translation knowledge from parallel corpora. Figure 13.1 illustrates how our work is placed in the view of EBMT.

Our work takes *parallel corpora* as input where each sentence pair are translations of each other. Parallel corpora can be prepared manually or can be obtained from parallel texts using sentence alignment methods such as Haruno et al. (1996).

The output of our work is *translation knowledge* expressed in word- or phrase-equivalent units. The bilingual lexicographers can check the output before using them for MT systems. The workload of preparing a bilingual lexicon will be reduced dramatically.

This chapter reports two research efforts conducted by the authors in the recent past with different objectives in mind. The first study uses statistically probable dependency relations obtained from parsers to acquire word and phrasal correspondences. The objective is to examine the use of dependency relations in translation knowledge acquisition even for language pairs with different word orders. The second study compares three models of translation units, each of which uses different

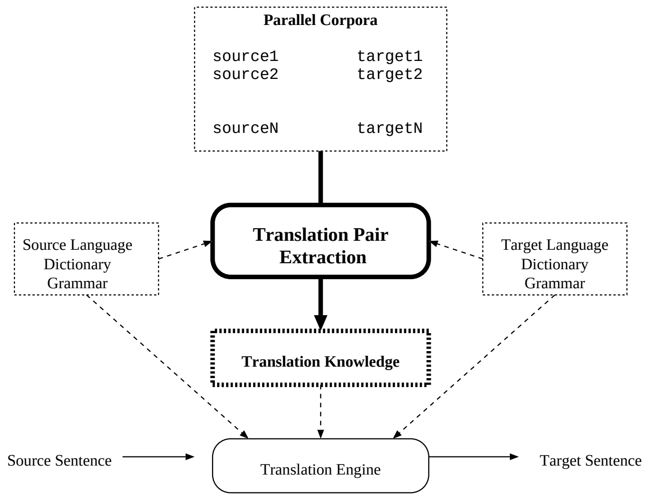


Figure 13.1. EBMT and Translation Knowledge Acquisition

linguistic information: one with only word segmentation, one with chunk boundaries, and finally one with word dependencies. It aims to investigate the relationship between the linguistic clues applied and translation knowledge extracted.

From these pieces of research, we demonstrate an automatic extraction of translation knowledge from parallel corpora by effective use of linguistic clues that the state-of-the-art statistical NLP tools offer.

The outline of this chapter is as follows. In Section 2, we review previous work that use parallel corpora to acquire translation knowledge and describe our approach to the problem. Then, we present research aiming to acquire knowledge of phrasal correspondence using dependency relations in Section 3 followed by a, comparative study on translation units, in Section 4. Our motivations, descriptions of the approaches taken, along with experimental results obtained are given in both sections. In Section 5, we compare our research with related work. Finally, in Section 6, we end this chapter with concluding remarks.

2. Background

There are two distinct approaches to acquiring translation knowledge 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 the form of words, noun phrases, collocations or constituents.

Alignment is the process by which correspondences of sub-components within the paired sentences are found. The goal is to find as many correspondences as possible for *some* substructures in the parallel corpus. Extraction, as the name suggests, focuses on extracting subcomponents that correspond by processing the entire parallel corpus. The goal is to find correspondences for *some* substructures in the parallel corpus.

It is vital to distinguish between approaches which are often confused. Alignment concentrates on completeness (or coverage) by aligning all corresponding subcomponents, but the suitability (or precision) of the resultant alignments as possible entries in a bilingual lexicon is compromised. Extraction acquires as many meaningful correspondences as possible with high precision, but fails to find all such correspondences in the entire parallel corpora.

Below, we review previous approaches taken in alignment and in extraction. Based on these observations, we describe our approach to finding translational correspondences from parallel corpora.

2.1 Alignment

Alignment has been investigated at word level as well as at structural levels. Historically, word alignment was seen as a natural extension to sentence alignment. Such a view was shared by Dagan et al. (1993) who employed a dynamic programming formulation similar to that of sentence alignment with finer granularity and with different slope constraints. Dagan et al. reported that for a 160,000-word excerpt of the Canadian Hansards, an alignment was produced in which about 55% of the words were correctly aligned, 73% were within one word of the correct position, and 84% 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 the paired sentences is parsed independently in advance with the algorithm finding matching constituents by measuring similarities. Much research has been reported in this area, including Kaji et al. (1992), Matsumoto et al. (1993), Grishman (1994), Matsumoto et al. (1995), and Meyers et al. (1996). These methods not only acquire more linguistically interesting alignments but also resolve ambiguity which arises from monolingual parsing. Wu (1995) employed a bi-parsing approach based on generative transduction grammar.

Although research results show some promise, there are some problems associated with alignment. First of all, word alignment is robust but exploits 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 as translation rules in EBMT. However, these methods are often hampered by the performance of parsers used in both languages. Most methods use rule-based parsers which often fail to handle complex sentences that include subordinate clauses and conjunctions, thereby sacrificing robustness.

2.2 Extraction

As with alignment, extraction has been investigated at various level of correspondences from words to collocations. In early work, word correspondences were searched for (e.g. Gale & Church (1991); Melamed (1995). However, as texts are not translated literally, there is an obvious shortcoming in word-for-word models. These methods fail 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.

Kupiec (1993) extracted English-French noun phrase (NP) correspondences. 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 & Hirakara (1994) also find NP correspondences between English and Japanese.

Smadja et al. (1996) extracted English-French collocations with their Champollion system, which is based on the Xtract system (Smadja, 1993). The system uses a statistical method to successfully extract rigid collocations as well as flexible ones. First, it identifies English collocations and then gradually expands French translation words that statistically co-occur with the English collocations. The method reports that 78% 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 & Matsumoto (1996) extracted rigid collocations of unrestricted length from English-Japanese parallel corpora. Translation units are strings of content words (e.g. nouns, adjectives, verbs, adverbs) in both languages. The method was tested 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 the structural dependencies which are embedded in sentences.

2.3 Our Approach

Our work uses extraction, but pays special attention to linguistic clues obtainable from statistical NLP tools. Figure 13.2 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 clues for each half of the paired sentences. These translation units are collected into candidate sets at the end of the monolingual step. Pair extractions then take two candidate sets from each language and correspondences based on co-occurrences in the parallel corpora are found.

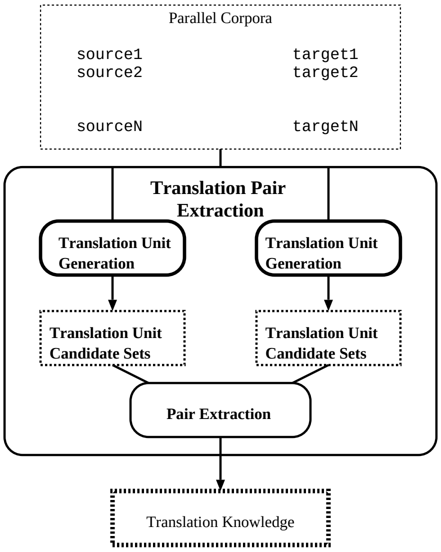


Figure 13.2. Overview of Translation Knowledge Extraction

The distinct characteristics of our approach are twofold: First, we use statistical NLP for preprocessing parallel corpora to obtain linguistic clues, thereby maintaining a high level of robustness that statistical NLP approaches enjoy. Second, we apply linguistically meaningful clues (namely chunk boundaries and word dependencies) to automatic extraction of translation knowledge. Phrasal correspondences as well as word correspondence can be extracted with our approach.

Our goal is to achieve the robustness of the extraction method while incorporating the linguistic notions used in structural alignment.

3. Phrasal Correspondence Acquisition Using Dependency Relations

This section describes a method to acquire phrase correspondences from sentence-aligned parallel corpora using statistically probable *dependency relations*, i.e. modifier-modifiee relations in a sentence.

The organisation of this section is as follows: in Section 3.1, we explain our rationale of applying dependency relations obtained from statistical parsers to the task of translation knowledge acquisition. In Section 3.2, we present how translation units are generated using dependency relations. Then, the pair extraction algorithm is described in Section 3.3 and experimental results are given in Section 3.4. Finally, we discuss the effectiveness of using dependency relations for the acquisition of phrase correspondences.

3.1 Support for Statistically Probable Dependency Relations

There are two reasons for using dependency structures estimated from statistical parsers. One is a linguistic requirement: dependency relations seem an appropriate level of abstraction for language pairs where the two languages belong to different language families. The other concerns technical advancement: recent developments in statistical parsing have led to performance accuracy of over 90 %, rendering such techniques practical. Below, we describe our justification of adopting statistically probable dependency relations in translation knowledge extraction.

3.1.1 Linguistic Requirement. Melamed (1995) conducted a comprehensive experiment on use of word position, alignment and cognates as linguistic heuristics for the extraction of bilingual lexicons. However, some heuristics such as cognate and alignment are not applicable to our language pairs. Even for word position heuristics, the effectiveness is limited to compound nouns as in "Natural/*shizen* Language/*gengo* Processing/*syori*".

Matsumoto et al. (1993) argue that dependency structures are preferable to phrase structures, as they abstract away from the vagaries of word order. Japanese accomodates relatively free word ordering. For example, two sentences "*kaigi-ga 2-ka nara-de kaisai-sareru*" and "*nara-de 2-ka kaigi-ga kaisai-sareru*" convey the identical meaning that "The conference is held on the second in Nara". The surface word orders are different, but they share the same dependency relations as illustrated in Figure 13.3.

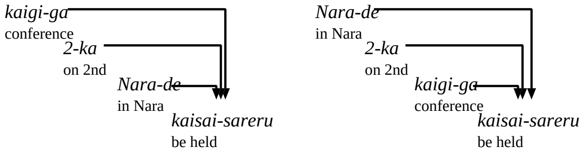


Figure 13.3. Different Word Orders, but Same Dependency Relations

Our justification of employing dependency relations as linguistic clues stems from the observation that the relative positions of source and target words may not necessarily coincide between the two languages, but the dependency structures between such words 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.

3.1.2 Technical Advancement. The other reason for using statistically probable dependency relations is due to recent advances in statistical parsing. Recently, many statistical parsing models have been proposed for English, including Collins (1997), Ratnaparkhi (1997), and Charniak (2000), and likewise for Japanese, including Fujio & Matsumoto (1998), and Kudo & Matsumoto (2001). Statistical parsing has an advantage over its rule-based counterparts owing to its wider coverage and reduced workload for grammar maintenance. More importantly, however, 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.

As a consequence, by using statistically probable dependency relations in the generation of translation units, our approach uses richer linguistic clues and improves coverage and robustness over previous methods.

3.2 Translation Unit Generation Using Dependency Relations

The process of generating translation units in each language is shown in Figure 13.4. The source half of the paired sentences in the parallel corpus is fed into the *morphological analysis* component to tokenize into

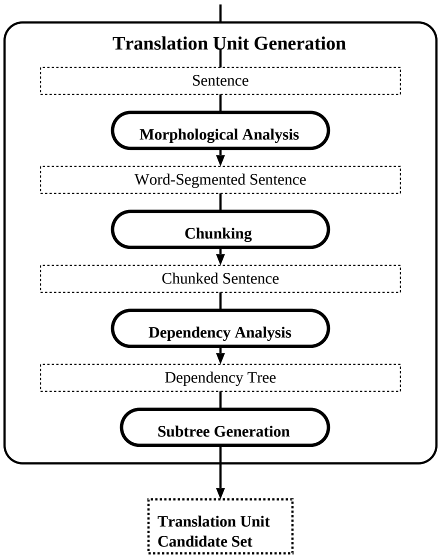


Figure 13.4. Translation Unit Generation

words each annotated with its corresponding part-of-speech tag. The word-segmented sentence proceeds to the *chunking* phase. Either 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 the *dependency analysis* phase so that a dependency tree of the sentence may be constructed. Finally, the *subtree generation* component produces a set of subtrees of the dependency tree, each of which corresponds to a translation unit where dependencies are preserved. The process is repeated for every sentence, and all the generated subtrees form a translation unit candidate set of the language.

We use a maximum likelihood model proposed in Fujio & Matsumoto (1998) as the basis 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 segments in chunking:

- English
    - a base NP: none of its child consituents are NP;
    - an NP with any preceding preposition or conjunction (e.g. ([with/IN] [an/DT example/NN]) is regarded as a chunk);

  - main verbs with preceding 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 the problem of ambiguity in that the correct dependency relation for every segment may not be determined. Even with state-of-the-art statistical parsers, sentence precision is only 50%.

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

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

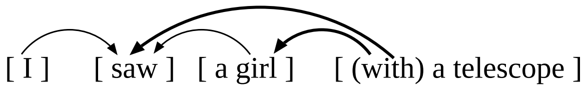
3.2.1 Model A: Best-one. The best-one model uses only the most likely (statistically best) dependency relations. At most one dependency is allowed for each segment. Translation units generated by the best-one model correspond to subtrees in sentences.

3.2.2 Model B: Ambiguous. The ambiguous model uses dependency relations above a certain confidence score. Figure 13.5 shows translation units generated by the ambiguous model. Dependency relations for "with a telescope" are ambiguous and both probable parses are considered in the generation of translation units. For translation units comprising 3 segments, two different subtrees may be generated. In order to distinguish an isomorphic subtree from a non-isomorphic one, different labels ("<T>" and "<L>") are annotated (cf. Figure 13.5).

The $(k+1)$th dependency relation for $k \geq 1$ (i.e. statistically non-best dependency relation) is also included if

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

The parameter 2 is chosen arbitarily. Multiple dependencies may be considered for each segment. Translation units generated by the ambiguous



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 13.5. Ambiguous Model

model also correspond to subtrees in sentences. In fact, the ambiguous model is a superset of the best-one model.

In this model, more likely dependency relations will appear more frequently in 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 whether ambiguity in dependency analysis will be resolved in pair extraction by supplying alternative parses.

3.2.3 Model C: Adjacent. The adjacent model uses only adjacency relations between segments, i.e. all segments are adjacent to the previous segment. Unlike the previous two models, translation units generated by the adjacent model correspond to substrings of sentences. Here, structural clues are ignored and word ordering constraints are the dominant influencing factor by far. This model is taken as the baseline case and is included for comparison purposes.

The sets of translation unit candidates thus generated become the input for the pair matching phase of our approach, which is described in the next subsection.

3.3 Pair Extraction Algorithm

Our strategy is to allow positive overlaps where ambiguity exists in sets of translation unit candidates. This is balanced by a rather more conservative discovery of correspondences in the pair extraction algorithm.

The pair extraction algorithm is based on Kitamura & Matsumoto (1996). 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 often than the threshold. Then, it calculates the correlation of all combinations in the given candidate sets and extracts statistically correlated pairs. When no more pairs are found, the threshold is lowered to gradually expand the number of translation units under consideration. The same extraction process is repeated until the minimum threshold is reached. The full algorithm is stated below:

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

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

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

   (a) Collect translation units that appear at least f_{curr} times in English candidate sets and in Japanese candidate sets and extract the corresponding pairs as follows:

   - For an English translation unit p_e, obtain the plausible candidate set PJ = { p_{j1}, p_{j2}, ..., p_{jn} } such that sim(p_e,p_{jk}) $> log_2\ f_{curr}$ for all k. Find a Japanese translation unit p_j in PJ whose correlation score is the maximum $sim(p_e, p_{jk})$ in PJ.

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

   - For the Japanese translation unit p_j, obtain the plausible candidate set PE = { p_{e1}, p_{e2}, ..., p_{en} } such that sim(p_{ek},p_j) $> log_2\ f_{curr}$ for all k. Find an English translation unit p'_e in PE whose correlation score is the maximum $sim(p_{ek}, p_j)$ in PE.

$$p'_e \in \underset{p_{ek} \in PE}{\texttt{argmax}}\ sim(p_{ek}, p_j)$$

   - Extract (p_e,p_j) as a translation pair if p_e is equal to p'_e. The correlation score of (p_e, p_j) is the highest among PJ for p_e and PE for p_j.

(b) For the translation pair p_e, p_j, and their overlapping translation units, filter out the sentence positions where both p_e and p_j appear in parallel corpora.

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

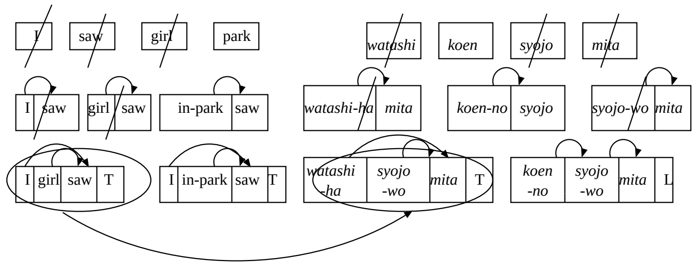


Figure 13.6. Filtering: (I,watashi),(saw,mi-ta),(girl,syojyo),(park,koen)

Several techniques are incorporated to overcome the problem of combinatorial explosion. First, we employ 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 from higher scores to lower scores.

Secondly, filtering for competing translation units is incorporated in step (b) of the algorithm. Figure 13.6 illustrates filtering for a sentence pair "I saw a girl in the park/*watashi-ha syojyo-wo koen-de mita*". A set of translation units derived from English is depicted on the left, with the Japanese equivalents depicted on the right. Once a pair "I_girl_saw(T) / *watashi_syojyo-wo_mita*(T)" is determined as a translation pair, the algorithm assumes that "*watashi_syojyo-wo_mita*(T)" will not be paired with translation units that are overlapped with "I_girl_saw(T)" (cancelled by diagonal lines in Figure 13.6) for the sentence pair. The operation effectively discards the found pairs and causes correlation scores in the preceding iterations to be recalculated.

The algorithm effectively extracts translation pairs from overlapping sets of candidates.

3.4 Experimental Results

We use 9268 paired sentences of English-Japanese business letter samples (Takubo & Hashimoto, 1999). The NLP tools used to obtain candidates are summarised in Table 13.1.

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

Table 13.1. NLP tools used in this experiment

Parameter setting is performed 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 until it reaches the minimum threshold f_{min} 2. In other words, translation pairs that co-occured only once in the parallel corpora will not be extracted. All parameters are 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 is judged by a bilingual speaker of English and Japanese (Japanese native). Precision for each model is summarised in Table 13.2. 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 expand correspondence candidate sets PE and PJ so that translation units with the correlation score $\geq log_2\ 2$ (≥ 1) are also considered. (Note, frequency for those translation units are $f_{ej} = f_e = f_j = f_{min} = 2$) These are marked by asterisks "*" in Tables.

3.5 Discussion

Table 13.2 shows that both the best-one and the ambiguous models achieve better performance than the adjacent (baseline) model. This implies that applying statistically probable dependency relations to the generation of translation units (subtrees) is effective in this task. Our goal of extracting phrasal correspondences without losing robustness has been achieved by the effective use of statistical dependency parsers.

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)

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)

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 13.2. Precision: best-one model(top), ambigious model(middle), adjacent model(below)

English	Japanese
thank+you	*arigatou*
consultations+include	*kyogi_ni_ha+fukmeru*
apply+for_the_position	*syoku_ni+ouho_itasu*
thank+you+in_advance	*mae-motte+onegai+moushi-ageru*
not+hesitate+to_contact	*enryo-naku+go-renraku*
be+enclosed+a_copy	*1_bu_doufu_itasu*
be_writing+to_let+know	*syojyo_wo-motte+oshirase_itasu*
applications+include	*yoto_ni_ha+aru*
upcoming_borard+of_director_s´_meeting	*jikai_no+torishimari-yaku_kai*
will_have+to_cancel	*chushi_se_zaru_wo+e_naku+naru*
have+high_hope	*ooini+kitai_suru*
business+is_expanded	*syobai_ha+hatten_suru*
we+have_learned+from_your_fax	*ki_fakkusu_de+shiru*
leaving+in+about_ten_days	*yaku_10_nichi_go+syuppatsu*
get+you+in_close_business_relationship	*kinmitsu_na+torihiki_kankei_wo+kizuku*
we+are_inquiring+regarding	*ni-kanshi+otazune_itasu*
pay+special_attention	*tokubetsu_no+chui_wo+harau*

Table 13.3. Random samples of correct translation patterns in best–one model.

English	Japanese
(have_been_pleased)+to_serve+as_thier_main_banker	*syuryoku_ginkou_to+naru*
[be_held]+at_hotel_new_ohtani	*hoteru_nyuohtani_de+kaisai_suru*
assets_position+(in_good_shape)	*shisan_jyotai*
(have_been_placed)+into_our_file	*watashidomo_no+fairu*
(put)+one_month_limit	*1_kagetsu_no+kigen*
[passed]+on_past_tuesday	*kayobi_ni+nakunara_reru*

Table 13.4. Randam samples of near–correct translation patterns where score is 1.000.

However, our method did not fulfil the expectation that ambiguity in dependency analysis will be resolved in pair extraction by supplying alternative parses. The number of extracted translation pairs decreased in the ambiguous model from the best-one model, despite the fact that candidate sets in the ambiguous model constitute a superset of those in the best-one model.

The expected effect in the ambiguous model is that more partially correct subtrees are generated in candidate sets which will boost the correlation score in our pair extraction algorithm. In reality, allowing redundant parsing introduces 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.

Allowing redundant parses leads to 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 is reduced.

This section used statistically probable dependency relations to extract word and phrasal correspondences. With 9268 paired sentences, we achieved nearly 90% precision. The experiment shows that although statistical parsers are prone to errors, dependency relations serve as effective linguistic clues in translation knowledge extraction. Although we allow aggressive overlaps in translation unit candidate sets where ambiguity exists, this was not effectively balanced by the conservative discovery of correspondences in the pair extraction algorithm.

4. A Comparative Study of Translation Units

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

This section compares three models of translation units, each of which uses different linguistic information: one with only word segmentation, one with chunk boundaries, and one with word dependencies. We use NLP tools to annotate such linguistic information. Table 13.5 shows some publicly available robust NLP tools, most of which use statistical techniques that are trained directly from annotated corpora. Although state-of-the-art NLP tools do not offer 100% precision, they enable many partially correct answers to be found.

tool	usage	language	technique	performance
TnT	POS tagging	English	HMM	97 %
MXPOST	POS tagging	English	ME	96 %
ChaSen	POS tagging	Japanese	HMM	98 %
YamCha	chuker	English	SVM	94 %
YamCha	chuker	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 13.5. Preprocessing Tools

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

The organization of this section is as follows: in Section 4.1, we describe three models used to generate translation units. In Section 4.3, we present our experimental results. Finally, Section 4.4 analyzes the characteristics of each model.

4.1 Linguistic Heuristics for Translation Units

In this work, we focus on three kinds of linguistic clues obtainable from NLP tool, namely word segmentation (spaces), chunk boundaries (square brackets), and word dependencies (arrows), as shown in Figure 13.7.

```
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] .
```

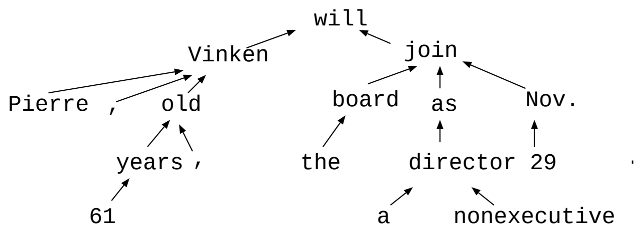


Figure 13.7. Linguistic Clues: Word Segmentation (top), Chunk Boundary (middle) and Word Dependency (bottom)

NLP tools have reached a level where they have become practical, 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 allow overlaps which arise from real ambiguity, but which eliminate impossible overlaps.

4.2 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 the Plain N-gram and Chunk-bound N-gram models, translation units are built using only content (open-class) words. This is because functional (closed-class) words such as prepositions alone will usually act as noise and so they are filtered out in advance.

A word is classified as a functional word if it matches one of the following 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) "*meishi-daimeishi*" (noun-pronoun), "*meishi-kazu*" (noun-number), "*meishi-hijiritu*" (noun-dependent), "*meishi-tokusyu*" (noun-specific), "*meishi-setsubi-jyodoshigokan*" (noun-suffix-postparticle), "*meishi-setsubi-fukushikano*" (noun-suffix-adverbial), "*meishi-setsubi-jyodoshi*" (noun-suffix-model), "*settoushi*" (prefix), "*doushi-setsubi*" (verb-suffix), "*doushi-hijiritsu*" (verb-dependent), "*jyoshi*" (postposition), "*jyodoshi*" (postparticle), "*keiyoshi-hijiritsu*" (adjective-dependent), "*keiyoshi-setsubi*" (adjective-suffix), "*kigo*" (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 the Dependency-linked N-gram model, translation units are built not only with content words but also function words. This is because inclusion of function words seems natural when considering word dependency. However, invalid translation units such as those consisting solely of function words are eliminated in advance.

4.2.1 Model 1: Plain N-gram. The Plain N-gram model was first proposed in Kitamura & Matsumoto (1996). The translation units generated in this model are word sequences from unigram 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.

4.2.2 Model 2: Chunk-bound N-gram. The Chunk-bound N-gram model assumes prior knowledge of chunk boundaries. The definition of "chunk" has room for discussion. In our experiment, the definition for the English chunking task complies with the CoNLL-2000 text chunking task,[1] while the definition of Japanese chunks is based on "bunsetsu" in the Kyoto University Corpus (1997).[2]

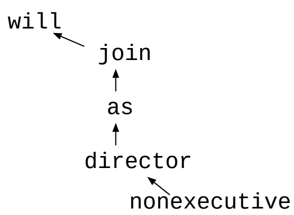
[1] http://lcg-www.uia.ac.be/conll2000/chunking/
[2] http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/corpus.html

Unlike the Plain N-gram model, the Chunk-bound N-gram model will not extend beyond the chunk boundaries. N varies depending on the number of words in a chunk[3].

4.2.3 Model 3: Dependency-linked N-gram. In the Dependency-linked N-gram model, 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 Section 3), we only use the statistically best parse in the Dependency-linked N-gram model. We treat each branch of the word dependency tree as word sequences, and generate translation units from unigrams to N-grams 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.

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

Furthermore, the Dependency-linked N-gram model can generate 'non-contiguous' N-grams which cannot be generated by the Plain N-gram model for any N. Figure 13.8 shows a set of N-grams generated from a branch of a dependency tree in Figure 13.7. Bolded N-grams are ones that cannot be generated by Plain N-gram.



nonexecutive
nonexecutive-director
nonexecutive-director-as
nonexecutive-director-as-join
nonexecutive-director-as-join-will
director
director-as
director-as-join
director-as-join-will
as-join
as-join-will
join
join-will

Figure 13.8. Our Dependency-linked N-gram Model

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

4.3 Experimental Results

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

We used a corpus of 5000 aligned sentences consisting of English-Japanese business expressions. 4000 sentences pairs are used for training and the remaining 1000 sentences are used for evaluation.

The NLP tools used in this experiment are ChaSen, YamCha, CaboCha for Japanese text processing and TnT, YamCha, and Collins' parsers for English text processing. The choice of NLP tools was made based on performance criteria (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 13.6 shows the number of translation units found in 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 13.6. 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 measured by 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 percentage correct respectively in Melamed (1995). 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 Table 13.7. "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}", as judged by an English-Japanese bilingual speaker. "acc" lists accuracy, namely 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 Table 13.8. As before, "f_{curr}" indicates the threshold. The brackets indicate language: "E" for

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**

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**

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 13.7. Precision: Plain N-gram (top), Chunk-bound N-gram (middle), Dependency-linked N-gram (bottom)

English and "J" for Japanese. "fnd" is the number of content tokens matched with correct translation pairs. "idl" 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 corpora[4]. "cvr" indicates 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 a testset of 1000 parallel sentences, there are 14422 tokens in the English half and 18998 tokens in the Japanese half. "idl" increases as the threshold is lowered, while the total remains consistent.

4.4 Discussion

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

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 13.9 shows the Venn diagram of translation pairs extracted by each model. An interesting observation can be made in the distribution

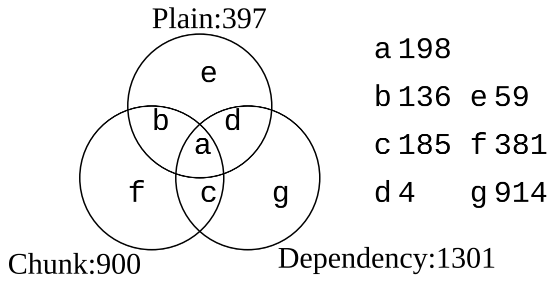


Figure 13.9. Distribution of Extracted Translation Pairs

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

f_{curr}	fnd(E)	idl(E)	i_cvr(E)	t_cvr(E)	fnd(J)	idl(J)	i_cvr(J)	t_cvr(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

f_{curr}	fnd(E)	idl(E)	i_cvr(E)	t_cvr(E)	fnd(J)	idl(J)	i_cvr(J)	t_cvr(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

f_{curr}	fnd(E)	idl(E)	i_cvr(E)	t_cvr(E)	fnd(J)	idl(J)	i_cvr(J)	t_cvr(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 13.8. Coverage: Plain N-gram (top), Chunk-bound N-gram (middle), Dependency-linked N-gram (bottom)

model	English	Japanese
Plain	U.S.-Japan	*nichibei*
Plain	look forward (to) visiting	*homon(-wo-)tanoshimi*
Plain	give information	*shiryo-teikyo*
Chunk	Hong Kong	*honkon*
Chunk	San Diego	*sandiego*
Dependency	apply for position	*syoku-ni-obo-suru*
Dependency	be at your service	*yomei-ni-shitagai-masu*
Dependency	checking into matter	*ken-wo-chosa*
Dependency	tell about matter	*ken-nitsuite-oshirase*
Dependency	free of charge	*muryo*
Dependency	out of question	*mondai-gai*
Dependency	out of print	*zeppan*

Table 13.9. Correct translation pairs

of the Plain N-gram model. 84% of the translation pairs extracted by the Plain N-gram model can also be extracted by the Chunk-bound N-gram model. If we ignore the intersection of all three models, 34% of the units in the Plain N-gram model are shared by the Chunk-bound N-gram model. In contrast, only 1% of the translations of the Plain N-gram model are common to the Dependency-linked N-gram with, not including the intersection of all three models. Translation pairs extracted only by the Plain N-gram model accounts for just 14% of the total found. From these, we could conclude that translation pairs extracted by the Plain N-gram model can nearly be found by the Chunk-bound N-gram model.

Table 13.9 lists samples of correct translation pairs that are unique to each model.

The Plain N-gram model seems to extract longer translation pairs that co-occur coincidently. For example, the data contains many instances where 'look forward to visiting' and '*homon-wo-tanoshimi*' co-occur. The Plain N-gram model generates sequences of content words, such as 'look_forward_visit' and '*homon_tanoshimi*', and all instances are counted for calculating similarity. As for the Chunk-bound N-gram model, the translation pair will not be extracted due to the presence of the function words 'to' and '*wo*'. In the Dependency-linked N-gram model, 'look forward/*tanoshimi*' and 'visit/*houmon*' were extracted separately. A close examination of parsed results reveals that, in some sentences, 'look forward' and 'to' were not dependency-linked.

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

The Dependency-linked N-gram model managed to extract translation pairs which are useful for structural disambiguation (e.g. 'checking **into** matter/*ken*-**wo**-*chosa*' and 'tell **about** matter/*ken*-**nitsuite**-*oshirase*') Such extractions become possible mainly because function words are included in translation units in the Dependency-linked N-gram model.

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

5. Related Works

Our view on EBMT is close to advocates of Template-driven EMBT in this volume, where phrase-equivalent correspondences seem to be an agreed unit of correspondences for transfer rules. Our first piece of research focused exclusively on extraction of phrasal correspondences which closely matches the aims of Template-driven EBMT.

Support for dependency trees as a starting point to acquire translation correspondences is also shared by Hideo Watanabe et al. (cf. Chapter 14) and Arul Menezes and Steven Richardson (cf. Chapter 15). Both works are noteworthy in that they demonstrate not only acquiring transfer rules by means of structural alignment but also applying these rules directly to EBMT.

Confining our discussion to the acquisition phase, our work is still unique in a number of ways. First, Watanabe et al. and Menezes & Richardson use alignment while our work involves extraction by our classification in Section 2. Second, we employ publicly available statistical dependency parsers, not in-house parsers. This not only ensures that we are able to take advantage of the robustness that statistical parsers offer, but also that our approach is portable. Finally, but most important of all, we do not require a seed bilingual dictionary to find phrasal correspondences. Moreover, the algorithm does not distinguish word correspondences from phrasal correspondences, bur rather they are uniformly treated as translation units in the pair extraction algorithm. This means that bilingual resources can be incorporated into our method without modification thanks to the incremental nature of our greedy algorithm. Word correspondences that we can be sure of from bilingual dictionaries can be safely eliminated from sets of translation candidates before the pair extraction algorithm operates, and the same procedure can be taken to find further phrasal correspondences.

6. Conclusion

This chapter described two pieces of research concerning automatic extraction of translation knowledge from parallel corpora.

The first piece of research, "Phrasal Correspondence Acquisition Using Dependency Relations", 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 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 arising from dependency analysis. We allow aggressive overlaps in translation unit candidate sets where ambiguity exists. However, this was not sufficiently balanced by a conservative discovery of correspondences in the pair extraction algorithm. Although the ambiguous model is a superset of the best-one model, it is not the optimal superset where only the real syntactic ambiguities for the parser used are considered. The problem of syntactic disambiguation still remains in our research agenda.

The second work, "A Comparative Study on Translation Units", compares three models of translation units, each of which uses different linguistic information: one with word segmentation only, one with chunk boundaries, and one with word dependencies. 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 boundaries or with word dependencies outperformed the previous baseline model, namely 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 lexicons for a new domain. However, longer translation pairs such as idiomatic expressions are better handled by word dependency.

The two independent but related works demonstrate that use of statistical NLP tools, in particular, statistical dependency parsers, offers robustness to the approach as well as valuable linguistic clues that work effectively in extracting translation knowledge from parallel corpora where the two languages come from different families.

Acknowledgement We thank Mihoko Kitamura for fruitful discussion. We also thank Nikkei Shimbun for providing us with a pre-aligned parallel corpus. Furthermore, we thank Masakazu Fujio, Taku Kudo, the ChaSen members, Michael Collins, and Thorsten Brants for making their NLP tools publically available.

References

Collins, M. (1997). *Three Generative Lexicalised Models for Statistical Parsing*, ACL-97: 35th Annual Meeting of the Association for Computational Linguistics, 16–23.

ChaSen and IPADIC Users Manual (2001). http://chasen.aist-nara.ac.jp/

Charniak, E. (2000). *A Maximum Entropy Inspired Parser*, NAACL-2000: 1st Meeting of the North American Chapter of the Association for Computational Linguistics, 132–139.

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

Fujio, M. and Matsumoto, Y. (1998). *Japanese Dependency Structure Analysis based on Lexicalized Statistics*, Proc. of 3rd Conf. of Empirical Methods in Natural Language Processing, 88-96.

Gale, W. and Church K (1991). *Identifying Word Correspondences in Parallel Texts*, Proceedings of the 4th DARPA Speech and Natural Language Workshop, 152–157.

Grishman, R. (1994). *Iterative alignment of syntactic structures for a bilingual corpus*, Proc. of 2nd Annual Workshop on Very Large Corpora, 57–68.

Haruno, M. and Yamazaki, T. (1996). *High-performance bilingual text alignement using statistical and dictionary information*, ACL-34: The 34th Annual Meeting of the Association for Compuational Linguistics, 131–138.

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

Kitamura, M. and Matsumoto, Y. (1996). *Automatic Extraction of Word Sequence Correspondences in Parallel Corpora*, Proc. of 4th Workshop on Very Large Corpora, 79–87.

Kudo, T. and Matsumoto, Y. (2001). *Applying Cascaded Chunking to Japanese Dependency Structure Analysis*, IPSJ SIG-NL-142, 97–104, (In Japanese).

Kumano, A. and Hirakawa, H. (1994). *Building an MT dictionary from parallel texts based on linguistic and statistical information*, COLING-94: The 15th International Conference on Computational Linguistics, 76–81.

Kupiec, J. (1993). *An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora*, ACL-93: 31st Annual Meeting of the Association for Computational Linguistics, 23–30.

Matsumoto, Y., Ishimoto, H. and Utsuro, T. (1993). *Structural Matching of Parallel Texts*, 31st Annual Meeting of the Association for Computational Linguistics, 23–30.

Matsumoto, Y. and Kitamura, M.(1995). *Acquisition of Translation Rules from Parallel Corpora*, Recent Advances in Natural Language Processing, Current Issues in Linguistic Theory 136, 406–416.

Melamed, I.D. (1995). *Automatic Evaluation and Uniform Filter Cascades for Inducing N-best translation lexicons*, Proceedings of the Third Workshop on Very Large Corpora, 184–198.

Meyers, A., Yangarber, R. and Grishman, R. (1996). *Alignment of Shared Forests for Bilingual Corpora*, The 16th International Conference on Computational Linguistics, 460–465.

Takubo, K. and Hashimoto, M. (1999). *A Dictionary of English Business Letter Expressions*, Nihon Keizai Shimbun, Inc.

Santorini, B. (1991). *Part-of-Speech Tagging Guidelines for the Penn Treebank Project*, LDC.

Smaja, F. (1993). *Retrieving collocation from text: Xtract*, Computational Linguistics 19, 143–177.

Smadja, F., McKeown, K.R. and Hatzuvassiloglou, V. (1996). *Translating Collocations for Billingual Lexicons: A Statistical Approach*, Computational Linguistics 22(1), 1–38.

Ratnaparkhi, A. (1997). *A linear observed time statistical parser based on maximum entropy models*, Proc. of 2nd Conf. on Empirical Methods in Natural Language Processing, 1–10.

Wu, D.(1995), *An algorithm for simultaneously bracketing parallel texts by aligning words*, Proceedings of the 33rd Annual Meeting of ACL, 244-251.