Learning Sequence-to-Sequence Correspondences from Parallel Corpora via Sequential Pattern Mining

Kaoru Yamamoto† and **Taku Kudo**‡ and **Yuta Tsuboi**§ and **Yuji Matsumoto**‡
†Genomic Sciences Center, The Institute of Physical and Chemical Research
1-7-22-E209, Suehiro-cho, Tsurumi-ku, Yokohama, 230-0045 Japan
kaorux@gsc.riken.go.jp
‡Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara, 630-0192 Japan
taku-ku@is.aist-nara.ac.jp, matsu@is.aist-nara.ac.jp
§Tokyo Research Laboratory, IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken, 242-8502 Japan
yutat@jp.ibm.com

Abstract

We present an unsupervised extraction of sequence-to-sequence correspondences from parallel corpora by sequential pattern mining. The main characteristics of our method are two-fold. First, we propose a systematic way to enumerate all possible translation pair candidates of rigid and gapped sequences without falling into combinatorial explosion. Second, our method uses an efficient data structure and algorithm for calculating frequencies in a contingency table for each translation pair candidate. Our method is empirically evaluated using English-Japanese parallel corpora of 6 million words. Results indicate that it works well for multi-word translations, giving 56-84% accuracy at 19% token coverage and 11% type coverage.

1 Introduction

This paper addresses the problem of identifying "multi-word" (sequence-to-sequence) translation correspondences from parallel corpora. It is well-known that translation does not always proceed by word-for-word. This highlights the need for finding multi-word translation correspondences.

Previous works that focus on multi-word translation correspondences from parallel corpora include noun phrase correspondences (Kupiec, 1993), fixed/flexible collocations (Smadja et al., 1996), n-gram word sequences of arbitrary length (Kitamura and Matsumoto, 1996), non-compositional compounds (Melamed, 2001), captoids (Moore, 2001), and named entities [1].

In all of these approaches, a common problem seems to be an identification of meaningful multi-word translation units. There are a number of factors which make handling of multi-word units more complicated than it appears. First, it is a many-to-many mapping which potentially leads to a combinatorial explosion. Second, multi-word translation units are not necessarily contiguous, so an algorithm should not be hampered by the word adjacency constraint. Third, word segmentation itself is ambiguous for non-segmented languages such as Chinese or Japanese. We need to resolve such ambiguity as well.

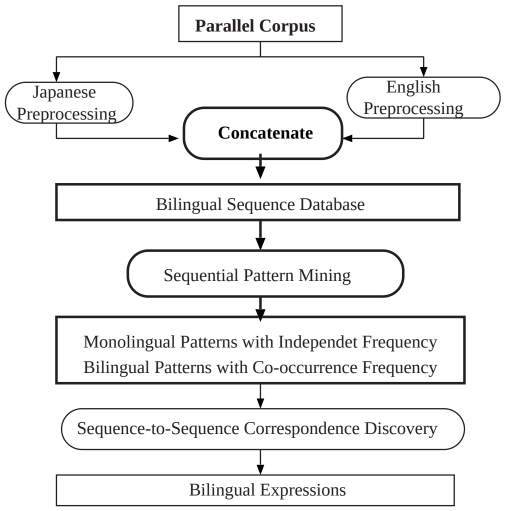
In this paper, we apply sequential pattern mining to solve the problem. First, the method effectively avoids an inherent combinatorial explosion by concatenating pairs of parallel sentences into single bilingual sequences and applying a pattern mining algorithm on those sequences. Second, it covers both rigid (gap-less) and gapped sequences. Third, it achieves a systematic way of enumerating all possible translation pair candidates, single- or multi-word. Note that some are overlapped to account for word segmentation ambiguity. Our method is balanced by a conservative discovery of translation correspondences with the rationale that direct associations will win over indirect ones, thereby resolving the ambiguity.

2 Our Basic Idea

Our approach is illustrated in Figure 1. We concatenate corresponding parallel sentences into bilingual sequences to which sequential pattern mining is applied. By doing so, we obtain the following effects:

- It exhaustively generates all possible translation can-

[1] As of this writing, we learn that Moore will present his results on named entity at EACL 2003.



Parallel Corpus
Japanese Preprocessing
English Preprocessing
Concatenate
Bilingual Sequence Database
Sequential Pattern Mining
Monolingual Patterns with Independet Frequency
Bilingual Patterns with Co-occurrence Frequency
Sequence-to-Sequence Correspondence Discovery
Bilingual Expressions

Figure 1: Our Approach

didates, *both rigid and gapped* sequences, yet *avoiding combinatorial explosion*.

- It achieves an efficient calculation of a contingency table in a *single* running of sequential pattern mining.

In what follows, we describe sequential pattern mining and each module in Figure 1.

2.1 Sequential Pattern Mining

Sequential pattern mining discovers frequent subsequences as patterns in a sequence database (Agrawal and Srikant, 1995). Here, a subsequence is an order-preserving item sequence where some gaps between items are allowed. In this paper, we write the support of subsequence s in sequence database S as $support_S(s)$, meaning the occurrence frequency of s in S. The problem is defined as follows:

Given a set of sequences S, where each sequence consists of items, and a given a user-specified minimum support ξ, **sequential pattern mining** is to find all of the subsequences whose occurrence frequency in the set S is no less than ξ.

A sequential pattern is different from N-gram pattern in that the former includes a pattern with and without gaps and does not impose any limit on its length. These characteristics in sequential pattern mining leads us to the idea of concatenating corresponding parallel sentences into a bilingual sequence database from which *bilingual* sequential patterns are mined efficiently.

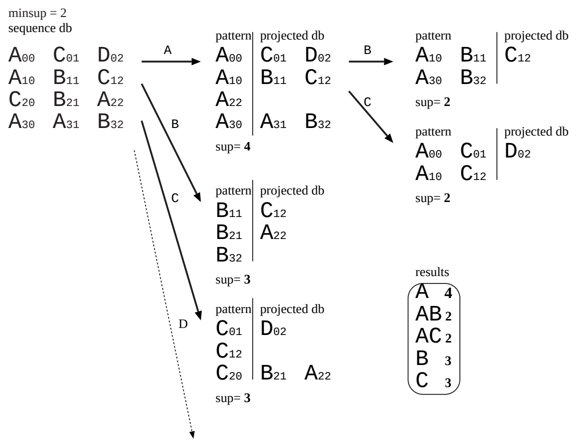


Figure 2: A Sample Execution of PrefixSpan

2.2 Bilingual Lexicon Extraction

2.2.1 Bilingual Sequence Database

For each parallel sentence, we undergo language-dependent preprocessing, such as word segmentation and part-of-speech tagging. Then we concatenate the monolingual sequences into a single bilingual sequence, and a collection of bilingual sequences becomes a sequence database S.

2.2.2 Sequential Pattern Mining

A single run of sequential pattern mining takes care of identifying and counting translation candidate patterns – rigid and gapped, some of which are overlapped – in the bilingual sequence database. All English subsequences satisfying the minimum support ξ will be generated (e.g., "e_1", "$e_1 e_2$", "$e_1 e_3$" $\cdots$, indicated by E_i). Similarly, all Japanese and bilingual subsequences with support $\geq \xi$ will be generated (indicated by J_j and $E_i J_j$ respectively). It is important to point out that for any bilingual pattern $E_i J_j$, corresponding English pattern E_i and Japanese pattern J_j that form constituents of the bilingual pattern are always recognized and counted.

PrefixSpan

In order to realize sequential pattern mining, we use PrefixSpan algorithm (Pei et al., 2001). The general idea is to divide the sequence database by frequent *prefix* and to grow the prefix-spanning patterns in depth-first search fashion.

We introduce some concepts. Let α be a sequential pattern in the sequence database S. Then, we refer to the α-**projected database**, $S|_\alpha$, as the collection of postfixes of sequences in S w.r.t prefix α.

A running example of PrefixSpan with the minimum support $\xi = 2$ (i.e., mining of sequential patterns with frequency ≥ 2) is shown in Figure 2. Each item in a se-

```
function PrefixSpan $(\alpha, S|_\alpha)$
begin
  $B \leftarrow \{b | (s \in S|_\alpha, b \in s)$
          $\wedge (support_S|_\alpha(\langle b \rangle) \geq \xi)$
          $\wedge (\mathbf{projectable}(\alpha, b))\}$
  foreach $b \in B$
  begin
    $(S|_\alpha)|b \leftarrow \{\langle i, s' \rangle | (\langle i, s \rangle \in S|_\alpha)$
                $\wedge (s' = postfix(s, b))\}$

    call PrefixSpan $(\alpha b, (S|_\alpha)|_b)$
  end
end
```

Figure 3: Pseudo Code of PrefixSpan

quence database is indicated by e_{ij} where e is an item, i is a sequence id, j is the offset for the $postfix$ of sequence id i. First, frequent sequential patterns with length 1 are selected. This gives A, B and C. The support of D is less than the minimum support 2, so D-projected database will *not* be created. For projections drawn with bold lines in Figure 2, we proceed with a frequent prefix A. Since A satisfies the minimum support 2, it creates a A-projected database derived from the sequence database S ($S|_A$). From $S|_A$, frequent items B, C are identified, subsequently forming prefix patterns AB and AC, and corresponding projected databases of the postfixes $S|_{AB}$, $S|_{AC}$. We continue with projection recursively to mine all sequential patterns satisfying the minimum support count 2.

PrefixSpan is described in Figure 3. The predicate **projectable** is designed to encode if a projection is feasible in an application domain. The original PrefixSpan gives a predicate that always returns true.

There are a number of possibilities for projectable to reflect linguistic constraints. A default projectable predicate covers both rigid and gapped sequences satisfying the minimum support. If we care for word adjacency, the projectable should return true only when the last item of the mined pattern and the first item of a postfix sequence in the projected database are contiguous. Another possibility is to prevent a certain class of words from being an item of a sequence. For example, we may wish to find a sequence consisting only of content words. In such a case, we should disallow projections involving functional word item.

2.2.3 Sequence-to-Sequence Correspondence

The effect of sequential pattern mining from bilingual sequence database can better be seen in a contingency table shown in Table 1. Frequencies of a bilingual pattern E_iJ_j, an English pattern E_i, and a Japanese pattern J_j correspond to a, $a+b$, and $a+c$ respectively. Since we know the total number of bilingual sequences $N = a+b+c+d$, values of b, c and d can be calculated immediately.

Table 1: Contingency Table

	J_j	$\neg J_j$	
E_i	**a**	b	$\mathbf{a+b}$
$\neg E_i$	c	d	
	$\mathbf{a+c}$		**N**

The contingency table is used for calculating a similarity (or association) score between E_i and J_j. For this present work, we use Dunning's log-likelihood ratio statistics (Dunning, 1993) defined as follows:

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

For each bilingual pattern E_iJ_j, we compute its similarity score and qualify it as a bilingual sequence-to-sequence correspondence if no equally strong or stronger association for monolingual constituent is found. This step is conservative and the same as step 5 in Moore (2001) or step 6(b) in Kitamura and Matsumoto (1996). Our implementation uses a digital trie structure called Double Array for efficient storage and retrieval of sequential patterns (Aoe, 1989).

For non-segmented language, a word unit depends on results of morphological analysis. In case of Japanese morphological analysis, ChaSen (Matsumoto et al., 2000) tends to over-segment words, while JUMAN (Kurohashi et al., 1994) tends to under-segment words. It is difficult to define units of correspondences only consulting the Japanese half of parallel corpora. A parallel sentence-pair may resolve some Japanese word segmentation ambiguity, however, we have no way to rank for word units with the same degree of segmentation ambiguity. Instead, we assume that frequently co-occurred sequence-to-sequence pairs in the entire parallel corpora are translation pairs. Using the global frequency of monolingual and bilingual sequences in the entire parallel corpora, we have better chance to rank for the ties, thereby resolving ambiguity in the monolingual half. To follow this intuition, we generate overlapped translation candidates where ambiguity exists, and extract ones with high association scores.

Sequential pattern mining takes care of translation candidate generation as well as efficient counting of the generated candidates. This characteristic is well-suited for our purpose in generating overlapped translation candidates of which frequencies are efficiently counted.

3 Experimental Results

3.1 Data

We use the English-Japanese parallel corpora that are automatically aligned from comparable corpora of the news wires (Utiyama and Isahara, 2002). There are 150,000 parallel sentences which satisfy their proposed sentence similarity. We use TnT (Brants, 2000) for English POS tagging and ChaSen (Matsumoto et al., 2000) for Japanese morphological analysis, and label each token to either content or functional depending on its part-of-speech.

Table 2: Statistics of 150,000 parallel sentences

	Japanese	English
content (token)	2,039,656	2,257,806
content (type)	47,316	57,666
functional (token)	2,660,855	1,704,189
functional (type)	1,811	386

3.2 Evaluation Criteria

We evaluate our sequence-to-sequence correspondence by accuracy and coverage, which we believe, similar criteria to (Moore, 2001) and (Melamed, 2001) [2]. Let C_{seq} be the set of correct bilingual sequences by a human judge, S_{seq} be the set of bilingual sequences identified by our system, C_{token} be the multiset of items covered by C_{seq}, T_{token} be the multiset of items in the bilingual sequence database, C_{type} be the set of items covered by C_{seq}, and T_{type} be the set of items in the bilingual sequence database. Then, our evaluation metrics are given by:

$$accuracy = \frac{|C_{seq}|}{|S_{seq}|}$$

$$token\ coverage = \frac{|C_{token}|}{|T_{token}|}$$

$$type\ coverage = \frac{|C_{type}|}{|T_{type}|}$$

In order to calculate accuracy, each translation pair is compared against the EDR (Dictionary, 1995). All the entries appeared in the dictionary were assumed to be correct. The remaining list was checked by hand. A human judge was asked to decide “correct”, “nearmiss”, or “incorrect” for each proposed translation pair without any reference to the surrounding context. Distinction between “nearmiss” and “incorrect” is that the former includes translation pairs that are partially correct[3]. In Tables 3, 4, and 5, accuracy is given as a range from a combination of “correct” and “nearmiss” to a combination of “nearmiss” and “incorrect”. Having calculated the total accuracy, accuracies for single-word translation pairs only and for multi-word translation pairs only are calculated accordingly.

3.3 Results

Our method is implemented in C++, and executed on a 2.20 GHz Penntium IV processor with 2GB memory. For each experiment, we set the minimum support ($minsup$) and the maximum length ($maxpat$) of patterns. All experiments target bilingual sequences of content words only, since we feel that functional word correspondences are better dealt with by consulting the surrounding contexts in the parallel corpora[4]. An execution of bilingual sequence databases compiled from 150,000 sentences, takes less than 5 mins with $minsup = 3$ and $maxpat = 3$, inferring 14312 translation pairs.

Given different language pair, different genre of text, different evaluation criteria, we find it difficult to directly compare our result with previous high-accuracy approaches such as (Moore, 2001). Below, we give an approximate comparison of our empirical results.

3.3.1 Rigid Sequences

Table 3 shows a detailed result of rigid sequences with $minsup = 3$, $maxpat = 3$. In total, we obtain 14312 translation pairs, out of which we have 6567 single-word

[2]We would like to examine how many distinct translation pairs are correctly identified (accuracy) and how well the identified subsequences can be used for partial sequence alignment in the original parallel corpora (coverage). Since all the correct translation pairs in our parallel corpora are not annotated, the sum of true positives and false negatives remain unknown. For this reason, we avoid to use evaluation terms precision and recall to emphasize the difference. There are many variations of evaluation criteria used in the literature. At first, we try to use Moore’s criteria to present a direct comparison. Unfortunately, we are unclear about frequency for multi-words in the parallel corpora, which seems to require for the denominator of his coverage formula. Further, we also did not split train/test corpus for cross-validation. Our method is an unsupervised learning, and the learning does not involve tuning parameters of a probabilistic model for unseen events. So we believe results using entire parallel corpora give indicative material for evaluation.

[3]We include “not sure” ones for a single-word translation. Those are entries which are correct in some context, but debatable to include in a dictionary by itself. As for multi-word translation, we include pairs that can become “correct” in at most 2 rewriting steps.

[4]Inclusion of functional word items in bilingual sequences is debatable. We have conducted an preliminary experiment of approx 10,000 sentences taken from a English–Japanese dictionary. As sentences are shorter and more instructive, we get grammatical collocations such as “impressed with / *ni kanmei* ” and “apologize for / *koto owabi*” or phrasal expressions such as “for your information / *go sanko*” and “on behalf of / *wo daihyo shi*”. However, we felt that it was not practical to include functional words in this work, since the parallel corpora is large-scale and interesting translation pairs in newspaper are named entities comprised of mostly content words.

Table 3: Result of Rigid Sequence Only with minsup = 3, maxpat = 3. Accuracy is given as a range from a combination of "correct" and "nearmiss" to a combination of "nearmiss" and "incorrect". The left side of slash gives a tigher evaluation and the right side of slash gives a looser evaluation.

minsup	maxpat	extracted sequence	correct sequence	total accuracy	single-word accuracy	multi-word accuracy	token coverage	type coverage
3	3	1000	927 / 988	0.927 / 0.988	0.942 / 0.988	0.824 / 0.984	0.142	0.018
3	3	2000	1836 / 1969	0.918 / 0.986	0.953 / 0.992	0.742 / 0.945	0.164	0.035
3	3	3000	2723 / 2932	0.908 / 0.977	0.951 / 0.991	0.732 / 0.923	0.174	0.050
3	3	4000	3563 / 3882	0.891 / 0.971	0.951 / 0.990	0.695 / 0.909	0.179	0.064
3	3	5000	4330 / 4825	0.866 / 0.965	0.948 / 0.989	0.656 / 0.903	0.182	0.076
3	3	6000	5052 / 5752	0.842 / 0.959	0.945 / 0.990	0.618 / 0.891	0.184	0.087
3	3	7000	5776 / 6656	0.825 / 0.951	0.941 / 0.989	0.607 / 0.879	0.186	0.098
3	3	8000	6350 / 7463	0.794 / 0.933	0.938 / 0.987	0.568 / 0.848	0.187	0.104
3	3	9000	7034 / 8345	0.782 / 0.927	0.935 / 0.985	0.562 / 0.844	0.188	0.113

Table 4: Result of Rigid Sequences Only with minsup = 10 and minsup = 5.

minsup	maxpat	extracted sequence	correct sequence	total accuracy	single-word accuracy	multi-word accuracy	token coverage	type coverage
10	3	4467	3989 / 4341	0.893 / 0.972	0.946 / 0.988	**0.712 / 0.918**	0.085	0.011
5	3	7654	6325 / 7271	0.826 / 0.950	0.937 / 0.986	**0.618 / 0.882**	**0.188**	**0.106**
10	10	4518	4002 / 4392	0.886 / 0.972	0.947 / 0.988	0.690 / 0.921	0.183	0.073
5	10	8007	6383 / 7387	0.797 / 0.922	0.938 / 0.986	**0.563 / 0.817**	**0.188**	**0.106**

Table 5: Result of Rigid and Gapped Sequences with minsup = 10. A default projectable constraint in Figure 3 is used.

minsup	maxpat	extracted sequence	correct sequence	total accuracy	single-word accuracy	multi-word accuracy	token coverage	type coverage
10	3	5792	4503 / 4979	0.777 / 0.860	0.950 / 0.989	**0.530 / 0.674**	0.085	0.012

Table 6: Comparison between Table 4 and Table 5 with minsup = 10, maxpat = 3

	single-word correct	single-word wrong	single-word all	multi-word correct	multi-word wrong	multi-word all
Both	3239	167	3406	554	181	735
Rigid only	25	18	43	**171**	**112**	283
Gapped only	0	2	2	**710**	**937**	1649

Table 7: Length Distribution of 171 correct Rigid multi-word Sequences Only (left) vs. Length Distribution of 112 wrong Rigid multi-word Sequences Only (right)

E \ J	1	2	3
1	n/a	16	0
2	15	**110**	6
3	5	7	12

E \ J	1	2	3
1	n/a	11	0
2	19	**29**	19
3	3	24	7

Table 8: Length Distribution of 710 correct Rigid and Gapped multi-word Sequences (left) vs. Length Distribution of 937 wrong Rigid and Gapped multi-word Sequences (right)

E \ J	1	2	3
1	n/a	17	0
2	45	**546**	15
3	9	43	35

E \ J	1	2	3
1	n/a	30	2
2	36	**229**	239
3	15	162	226

translation pairs and 7745 multi-word translation pairs. In this paper, we evaluate only the top 9000 pairs sorted by the similarity score.

For single-word translation, we get 93-99% accuracy at 19% token coverage and 11% type coverage. This implies that about 1/5 of content word tokens in the parallel corpora can find their correspondence with high accuracy. We cannot compare our word alignment result to (Moore, 2001), since the real rate of tokens that can be aligned by single-word translation pairs is not explicitly mentioned. Although our main focus is sequence-to-sequence correspondences, the critical question remains as to what level of accuracy can be obtained when extending coverage rate, for example to 36%, 46% and 90%. Our result appears much inferior to Moore (2001) and Melamed (2001) in this respect and may not reach 36% type coverage. A possible explanation for the poor performance is that our algorithm has no mechanism to check mutually exclusive constraints between translation candidates derived from the same paired parallel sentence.

For general multi-word translation, our method seems more comparable to Moore (2001). Our method performs 56-84% accuracy at 11% type coverage. It seems better than "compound accuracy" which is his proposal of hypothesizing multi-word occurrences, being 45-54% at 12% type coverage. However it is less favorable to "multiword accuracy" provided by Microsoft parsers, being 73-76% accuracy at 12% type coverage (Moore, 2001). The better performance could be attributed to our redundant generation of overlapped translation candidates in order to account for ambiguity. Although redundancy introduces noisier indirect associations than one-to-one mapping, our empirical result suggests that there is still a good chance of direct associations being selected.

Table 4 shows results of rigid sequences with a higher minimum support and a longer maximum length. Comparing with Table 3, setting a higher minimum support produces a slightly more cost-effective results. For example, $minsup = 10, maxpat = 3$, there are 4467 pairs extracted with 89.3-97.1% accuracy, while the top 4000 pairs in $minsup = 3, maxpat = 3$ are extracted with 89.1-97.1% accuracy. Table 4 reveals a drop in multi-word accuracy when extending $minpat$, indicating that care should be given to the length of a pattern as well as a cutoff threshold.

Our analysis suggests that an iterative method by controlling $minsup$ and $maxpat$ appropriately seems better than a single execution cycle of finding correspondences. It can take mutually exclusive constraints into account more easily which will improve the overall performance. Another interesting extension is to incorporate more linguistically motivated constraints in generation of sequences. Yamamoto et al. (2001) reports that N-gram translation candidates that do not go beyond the chunk boundary boosts performance. Had we performed a language dependent chunking in preparation of bilingual sequences, such a chunk boundary constraint could be simply represented in the projectable predicate. The issues are left for future research.

3.3.2 Gapped Sequences

One of advantages in our method is a uniform generation of both rigid and gapped sequences simultaneously. Gapped sequences are generated and extracted without recording offset and without distinguisting compositional compounds from non-compositional compounds. Although non-compositional compounds are rare and more difficult to extract, compositional compounds are still useful as collocational entires in bilingual dictionary.

There are positive and negarive effects in our gapped sequences using sequential pattern mining. Suppose we have English sequences of "My best friend wishes your father to visit $\cdots$" and "$\cdots$ best wishes for success". Then, we obtain a pattern "best wishes" that should be counted separately. However, if we have sequences of "staying at Hilton hotel" and "staying at Kyoto Miyako hotel", then we will obtain a kind of a phrasal template "staying at hotel" where the individual name of hotel, Hilton or Kyoto Miyako, is abstracted. Usefulness of such gapped sequences is still open, but we emperically evaluate the result of gapped sequences with $minsup = 10$ and $maxpat = 3$ shown in Table 5.

Comparing Table 4 and 5, we lose the multi-word accuracy substantially. Table 6 is a breakdown of rigid and gapped sequences with $minsup = 10$, $maxpat = 3$. The "Both" row lists the number of pairs found, under a category described in the column head, in both rigid and gapped sequences. The "Rigid only" row counts for those only found in rigid sequences, while the "Gapped only" row counts for those only found in gapped sequence. We learn that the decrease in multi-word accuracy is due to an increase in the portion of wrong pairs in sequences; 57% (937 / 1649) in gapped sequences whilst 40% (112 / 283) in rigid sequences.

However, gapped sequences have contributed to an increase in the absolute number of correct multi-word translation pairs (+539 correct pairs). In order to gain a better insight, we summarizes the length combination between English pattern and Japanese pattern as reported in Tables 7 and 8. It reveals that the word adjacency constraint in rigid sequences are too stringent. By relaxing the constraint, 436 (546 - 110) correct 2-2 translation pairs are encountered, though 200 (229 - 29) wrong 2-2 pairs are introduced at the same time. At this particular instance of $minsup = 10$ and $maxpat = 3$, considering gapped sequence of length 3 seems to introduce more noise.

Admittedly, we still require further analysis as to searching a break-even point of rigid/gapped sequences. Our preliminary finding supports the work on collocation by Smadja et al. (1996) in that gapped sequences are also an important class of multi-word translations.

4 Related Work

Moore (2001) presents insightful work which is closest to ours. His method first computes an initial association score, hypothesizes an occurrence of compounds, fuses it to a single token, recomputes association scores as if all translations are one-to-one mapping, and returns the highest association pairs. As for captoids, he also computes association of an inferred compound and its constituent words. He also uses language-specific features (e.g. capital letters, punctuation symbols) to identify likely compound candidates.

Our method is quite different in dealing with compounds. First, we outsource a step of hypothesizing compounds to language-dependent preprocessors. The reason is that an algorithm will become complicated if language-specific features are directly embedded. Instead, we provide an abstract interface, namely the projectable predicate in sequential pattern mining, to deal with language-specific constraints. Second, we allow items being redundantly counted and translation pair candidates being overlapped. This sharply contrasts with Moore's method of replacing an identified compound to a single token for each sentence pair. In his method, word segmentation ambiguity must be resolved before hypothesizing compounds. Our method reserves a possibility for word segmentation ambiguity and resolves only when frequently co-occured sequence-to-sequence pairs are identified.

Since we compute association scores independently, it is difficult to impose mutually exclusive constraints between translation candidates derived from a paired parallel sentence. Hence, our method tends to suffer from indirect association when the association score is low, as pointed out by Melamed (2001). Although our method relies on an empirical observation that "direct associations are usually stronger than indirect association", it seems effective enough for multi-word translation. balanced by a

As far as we know, our method is the first attempt to make an exhaustive enumeration of rigid and gapped translation candidates of both languages possible, yet avoiding combinatorial explosion. Previous approaches effectively narrow down its search space by some heuristics. Kupiec (1993) focuses on noun-phrase translations only, Smadja et al. (1996) limits to find French translation of English collocation identified by his Xtract system, and Kitamura and Matsumoto (1996) can exhaustively enumerate only rigid word sequences.

Many of works mentioned in the last paragraph as well as ours extract non-probabilistic translation lexicons. However, there are research works which go beyond word-level translations in statistical machine translation. One notable work is that of Marcu and Wong (2002), which is based on a joint probability model for statistical machine translation where word equivalents and phrase (rigid sequence) equivalents are automatically learned form bilingual corpora.

Our method does not iterate an extraction process as shown in Figure 1. This could be a cause of poor performance in single-word translation pairs, since there is no mechanism for imposing mutually exclusion constrains. An interesting question then is what kind of iteration should be performed to improve performance. Probabilistic translation lexicon acquisition often uses EM training on Viterbi alignments, e.g. (Marcu and Wong, 2002), while non-probabilistic ones employ a greedy algorithm that extracts translation pairs that give higher association scores than a predefined threshold where the threshold is monotonically decreasing as the algorithm proceeds, e.g. (Kitamura and Matsumoto, 1996). The issue is left for future work.

Last but not least, no previous works give an explicit mention to an efficient calculation of each cell in a contingency table. Our approach completes the process by a single run of sequential pattern mining. Since speed does not affect results of accuracy and coverage, its significance is often ignored. However, it will be important when we handle with corpora of large size.

5 Conclusions

We have proposed an effective method to find sequence-to-sequence correspondences from parallel corpora by sequential pattern mining. As far as multi-word translation is concerned, our method seems to work well, giving 56-84% accuracy at 19% token coverage and 11% type coverage.

In this work, we choose English-Japanese pair and empirically evaluate our method. However, we believe the method is applicable to any language pair with appropriate language-specific preprocessing tools. As by-product of our experiment, we obtain Japanese-English parallel corpora of 150,000 sentences where alignment of validated subsequence correspondences are back-annotated. This was accomplished by looking up to a Double Array dictionary of sequential patterns constructed in the extraction method. This shows that our method can be useful not only to development of semi-automatic lexicon for data-driven machine translation, but also to annotation of corresponding subsequences in translation memory system.

Acknowledgement

We would like to thank Masao Utiyama of CRL for creating a large-scale English-Japanese parallel corpora, Thorsten Brants and ChaSen development team for making NLP tools publicly available. In addition, we would like to thank anonymous reviewers for useful comments that have helped preparation of this paper.

References

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

J. Aoe. 1989. An Efficient Digital Search Algorithm by Using a Double-Array Structure. *IEEE Transactions on Software Engineering Vol. 15, 9*, pages 1066–1077.

T. Brants. 2000. TnT – A Statistical Part-of-Speech Tagger. *6th Applied Natural Language Processing Conference*, pages 224–231.

EDR Electronic Dictionary. 1995. http://www.iijnet.or.jp/edr.

T. Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics, Vol.19, No.1*, pages 61–74.

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

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

S. Kurohashi, T. Nakamura, Y. Matsumoto, and M. Nagao. 1994. Improvements of japanese morphological analyzer juman. *SNLR: Proceedings of the International Workshop on Sharable Natural Language Resources*, pages 22–28.

D. Marcu and W. Wong. 2002. A Phrase-Based, Joint Probability Model for Statistical Machine Translation. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 133–139.

Y. Matsumoto, A. Kitamuchi, T. Yamashita, H. Matsuda, K. Takaoka, and M. Asahara. 2000. Morphological analysis system chasen version 2.2.1 manual. *Nara Institute of Science and Technology*.

I.D. Melamed. 2001. *Empirical Methods for Exploiting Parallel Texts*. MIT Press.

R.C. Moore. 2001. Towards a Simple and Accurate Statistical Approach to Learning Translation Relationships among Words. *ACL Workshop on Data-Driven Machine Translation*, pages 79–86.

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

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

M. Utiyama and H. Isahara. 2002. Alingment of Japanese–English News Articles and Sentences (in Japanese). *IPSJ SIG-NL 151*, pages 15–21.

K. Yamamoto, Y. Matsumoto, and Kitamura M. 2001. A Comparative Study on Translation Units for Bilingual Lexicon Extraction. *ACL Workshop on Data-Driven Machine Translation*, pages 87–94.