Structural Phrase Alignment Based on Consistency Criteria

Toshiaki Nakazawa **Yu Kun** **Sadao Kurohashi**

Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku
Kyoto, 606-8501, Japan
{nakazawa, kunyu}@nlp.kuee.kyoto-u.ac.jp kuro@i.kyoto-u.ac.jp

Abstract

In this paper, we propose a new method for phrase alignment using a dependency type distance and a distance-score function. With this method, appropriate correspondences can be selected among correspondence candidates that often include ambiguous or incorrect ones. Furthermore, this method makes it possible to measure the overall alignment consistency. We conduct an alignment experiment using 500 parallel sentences on newspaper domain, and achieve an F-measure improvement of 35 points over the simple statistical method (GIZA++), and 3.0 points over a baseline system. We also conducted a translation experiment and achieved a BLEU score improvement of 0.4 points over a baseline system.

1 Introduction

In machine translation task, how to align the training parallel corpus with high accuracy is a big problem, and thus a number of studies have been done. The alignment methods can be categorized into two groups: one is probabilistic method and the other is heuristic method with structural information.

Probabilistic methods are mainly used in Statistical Machine Translation (SMT) systems (Och and Ney, 2003a). The main issue is how to decompose the alignment probabilities $Pr(\mathbf{A}|\mathbf{S}, \mathbf{T})$ reasonably to make good use of some approximations.

The simplest statistical method is based on word level alignment, in which the IBM Model (Brown et al., 1993) is mostly used as the baseline method. Recently, more sophisticated methods have been proposed by (Watanabe et al., 2002) and (Zhang and Vogel, 2005), which handle not only a word but a larger block which is usually a multiple word or a phrase. However, even if these methods are oriented to use larger block or structure, data sparseness is still a big problem on its way. For this reason, it is not easy to achieve high performance for the language pair whose linguistic structure is quite different from each other.

While, by using heuristic rules in alignment procedure, structural methods can easily use NLP resources, such as a morphological analyzer and a syntactic analyzer, to grasp characteristics of language pairs with large difference in linguistic structure.

(Menezes and Richardson, 2001) proposed a kind of tree structure called "Logical Form", which is a disordered graph representing the relations among the most meaningful elements of a sentence. With this structure, they proposed a "best-first" alignment method. This method starts from the nodes with the tightest lexical correspondence and then goes to close nodes from the first node. (Groves et al., 2004) used parsed tree structure of an original sentence, and then aligned the trees with some heuristic rules that constrain the order of alignment.

Although these structural methods utilize profound knowledge of NLP and achieve high accuracy, the manner of alignment is still heuristic, which is often not in general purpose. To resolve this issue, (Gildea, 2003) proposed a probabilistic tree-based alignment between Korean and English. They use some cloning operations to calculate the probability, so they make the structure more complicated. Moreover, it is not apparent that the same operations are effective and suitable for different language pairs.

In this paper, we propose an alignment method applying dependency type distance and distance score function into the structural alignment. Our motivation is to measure the alignment consistency based on distance, which is not only keeping the simple sentence structure but also language independent. Experimental result shows our proposed method can achieve about 3 points improvement on alignment accuracy.

In the following section, we briefly introduce the basic structural alignment module in our machine translation system. In Section 3, our proposed methods are introduced: there are three methods: baseline, model 1, and model 2. We performed some experiments to evaluate our proposal, and it is reported in Section 4. At last, we give a short conclusion and introduce our future work.

2 Procedure of Structural Phrase Alignment

Our machine translation system works mainly for Japanese-English, and the alignment is achieved by the following steps, using a Japanese parser, an English parser, and a bilingual dictionary.

2.1 Dependency Analysis of Sentences

Japanese sentences are converted into dependency structures using the morphological analyzer, JUMAN (Kurohashi et al., 1994), and the dependency analyzer, KNP (Kurohashi and Nagao, 1994). Japanese dependency structure consists of nodes which correspond to content words. Function words such as post-positions, affixes, and auxiliary verbs are included in the nodes.

For English sentences, Charniak's nlparser is used to convert them into phrase structures (Charniak and Johnson, 2005), and then they are transformed into dependency structures by rules defining head words for phrases. In the same way as Japanese, each node in this dependency tree consists of a content word and related function words.

Figure 1 shows an example of tree structure. The root of a tree is placed at the extreme left and phrases are placed from top to bottom.

2.2 Detection of Word/Phrase Correspondence Candidates

Correspondence candidates between Japanese word/phrase and English word/phrase are detected by a Japanese-English dictionary.

At this moment, the dictionary is not probabilistic. By looking up the whole pair of Japanese words and English words in the dictionary, correspondence candidates are detected deterministically.

In addition to the dictionary, we also handle transliteration. For possible person names and place names suggested by the morphological analyzer and Katakana words (Katakana is a Japanese alphabet usually used for loan words), their possible transliterations are produced and their similarity with words in the English sentence is calculated based on the edit distance. If there are similar word pairs whose edit distance exceeds a threshold, they are handled as a correspondence candidate.

For example, the following words can be considered as correspondence by the transliteration module, which cannot be handled by the existing bilingual dictionary entries:

新宿 → Shinjuku ↔ Shinjuku (similarity:1.0)
ローズワイン → rosuwain ↔ rose wine (similarity:0.78)

In Figure 1, the correspondence candidates “日本 (Japan) ↔ Japan”, “請求 (claim) ↔ claim”, “申し立て (allegation) ↔ file / claim”, and combination of “保険 (insurance) ↔ insurance” are found.

2.3 Selection of Correspondence Candidates

Results of previous procedure may contain ambiguous or incorrect correspondence candidates.

In Figure 1, for example, Japanese word “保険 (insurance)” and English word “insurance” occurs twice for each sentence, so there happens ambiguity. Moreover, “申し立て (allegation)” has two possible translations, “file” and “claim” in the English sentence. In addition, unambiguous but incorrect correspondence candidates might be sometimes detected. Thus, we need to select plausible correspondences among correspondence candidates.

In order to construct consistent over all alignment, we define consistency score for a pair of correspondences (introduced in Section 3). Then, we select the best set of correspondences in which a summation of consistency scores of all the combinations of correspondences is maximum.

$$\underset{alignment}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{j=i+1}^{n} consistencyscore(a_i, a_j) \qquad (1)$$

where a_i and a_j are one of correspondences.

2.4 Handling of Remaining Words

The alignment procedure so far finds some correspondences in parallel sentences. Then, we merge the remaining nodes into existing correspondences.

First, the root nodes of the dependency trees are handled as follows. In the given training data, we suppose that all parallel sentences have appropriate translation relation. Accordingly, if neither of the root nodes (of the Japanese dependency tree and the English dependency tree) is included in any correspondences, the new correspondence between the two root nodes is generated. If either root node is remaining, it is merged into the correspondence of the other root node.

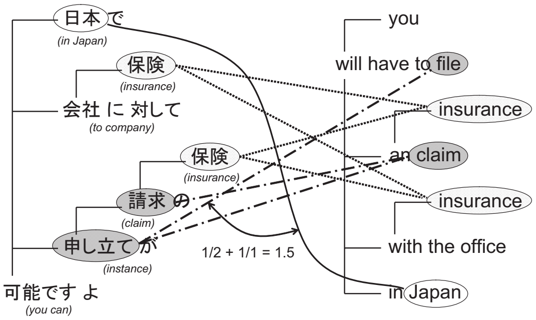

Figure 1: Example of ambiguities.

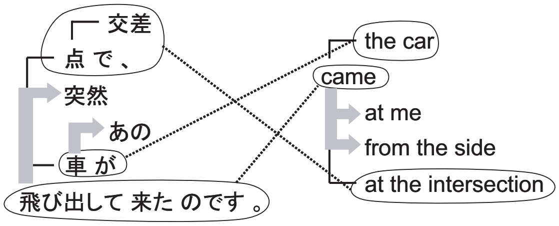

Figure 2: Example of extension.

Then, for both Japanese remaining node and English remaining node, if it is within a base NP and another node in the NP is in a correspondence, it is merged into the correspondence. At last, other remaining nodes are merged into correspondences of their parent (or ancestor) nodes.

In an example shown in Figure 2, “あの (that)” is merged into the correspondence “車 (car) ↔ the car”, since it is within an NP. “突然 (suddenly)”, “at me” and “from the side” are merged into their parent correspondence, “飛び出して来たのです (rush out) ↔ came”.

3 Structural Phrase Alignment Based on Consistency Criteria

3.1 Consistency of Alignment

Before introducing our proposing method, let us think of the word “consistency” itself. In Figure 3, the triangles represent the abstract of tree structure of each language, and the lines represent the correspondences. Among many lines, one line (on which a cross is placed) seems to be strange, which means disturbing the conformity of whole alignment.

This instability is apparent in visual. To measure the instability quantitatively, we focus on the “distance” in each language tree structure between two lines. In the example, although the distance between two lines in source language is far, the distance that in target language is near. Since the tree structure is constructed based on dependency information, such case rarely happens. In other words, it is impossi-

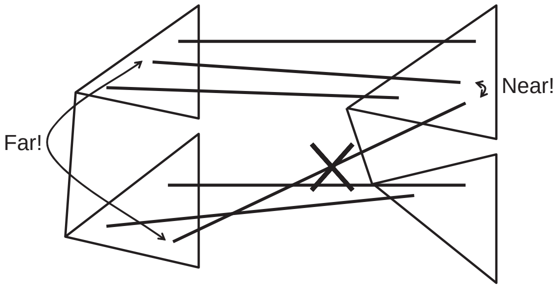

Figure 3: Example of Consistency.

ble that the two corresponding phrase pairs are semantically close in target language, but they are far in source language.

Therefore, suitably capturing the distance in each tree structure of all the pairs of lines leads to the overall consistent alignment of the parallel sentence. For this purpose, we introduce "consistency score", and this is explained in the following sections.

3.2 Consistency Score

To obtain consistent alignment within a sentence, we define a consistency score based on the dependency analysis tree.

First, we focus on an arbitrary pair of correspondences a_i as (p_{Si}, p_{Ti}) and a_j as (p_{Sj}, p_{Tj}), where p_{Si} represents the phrase of a_i in source language and p_{Ti} represents the phrase of a_i in target language. It is same for p_{Sj} and p_{Tj}.

Then, the dependency distance of source language $d_S(a_i, a_j)$ is defined as the distance between p_{Si} and p_{Sj}, and the dependency distance of target language $d_T(a_i, a_j)$ is defined as the distance between p_{Ti} and p_{Tj}. Then, the consistency score is defined as follows:

$$consistencyscore(a_i, a_j) = f(d_S(a_i, a_j), d_T(a_i, a_j))$$

It is referred as $f(d_S, d_T)$ for short. $f(d_S, d_T)$ is a function that maps a pair of distance to the score. Function d_S, d_T and f are defined differently in different model.

The consistency of whole alignment is defined as a summation of the consistency scores of all the pairs of correspondences (as shown in Equation 1).

Correct correspondences are supported by their neighbor correspondences, or the distance between them is small in both sides. Such relations produce good scores and contribute to the alignment consistency.

3.3 Baseline Method

In the baseline method, all the unambiguous correspondences are adopted without any constraint. For the ambiguous correspondences, we use distance and distance-score function.

Here, all the branches are treated as same: the distance is 1 for all. This means that we define the distance between correspondences as the number of traversing nodes in a dependency tree. Furthermore, the distance-score function is also simple. Suppose there is a correspondence a_{amb} with ambiguity, and there is an unambiguous correspondence a_{unamb} with the distance $d_S(a_{amb}, a_{unamb})$ in the Japanese dependency tree and the distance $d_T(a_{amb}, a_{unamb})$ in the English dependency tree, we give a score $f(d_S, d_T) = 1/d_S(a_{amb}, a_{unamb}) + 1/d_T(a_{amb}, a_{unamb})$ to the correspondence a_{amb} and a_{unamb}.

Then, we hold an assumption that the closer a_{unamb} is to a_{amb}, the more strongly a_{unamb} supports a_{amb}. Consequently, we accept the ambiguous correspondence with the highest score and reject the others conflicting with the accepted one. This calculation is repeated until all the ambiguous correspondences are resolved.

For example, in Figure 1, considering only one determined correspondences "日本 (Japan) ↔ Japan" as a clue, the scores are calculated, and the correspondence "申し立て (allegation) ↔ file" with the highest score is adopted. At the same time, the conflicting correspondence "申し立て (allegation) ↔ claim" is rejected. After that, the correspondence "請求 (claim) ↔ claim" is unambiguous, so it is adopted.

3.4 Proposed Model

Here we would like to refine the heuristic definition of distance and distance-score function. We proposed two models.

Proposed Model 1

First, we refine distance-score function in order to reject unambiguous and incorrect correspondences. Here, the definition of $d_S(a_i, a_j)$ and $d_T(a_i, a_j)$ is the same as the baseline model.

Using the gold standard alignment data from NICT(Uchimoto et al., 2004), which includes about 40,000 sentence pairs, we learned the frequency distribution of distance pair. Figure 4 shows the result of automatic learning form gold standard data.

Based on the observation of gold standard data, we design $f(d_S, d_T)$ as follows:

Criteria 1: $f(d_S, d_T)$ is positive if both d_S and d_T are small, which means the relation between the two correspondences is appropriate;

Criteria 2: $f(d_S, d_T)$ is 0 if both d_S and d_T are large, for the relation is not so important if they are far from each other;

Criteria 3: $f(d_S, d_T)$ is negative is d_S is large but d_T is small, or d_T is large but d_S is small, which means the relation between the two correspondences is inappropriate.

From these assumptions, the function $f(d_S, d_T)$ is modified like the graph in Figure 5. The modification is done satisfying the assumptions and covering the learning result.

Proposed Model 2

Japanese dependency analyzer KNP outputs dependency type information for each phrase, and Charniak's nlparser also outputs tag information. According to such information, we define dependency type score $ts_S(p_{Si})$ to show the strength of segmentation for source language between p_{Si} and the phrase p_{Si} depends on, and define $ts_T(p_{Ti})$ for target language. This definition is set by hand and Figure 6 shows part of them. An example of dependency type score is shown in Figure 7, where the label placed on each branch represents dependency type and the number placed over the label represents dependency type score.

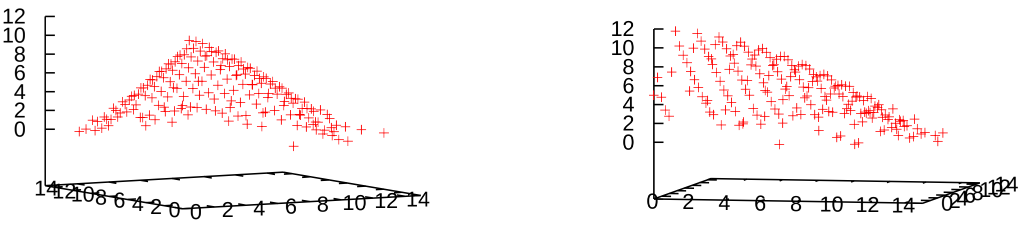

Figure 4: Learned Distance Pair Distribution for Proposed Model 1.

predicate:level C	6
predicate:level B+ / B	5
predicate:レベル B- / A *to* case	4
wo case / *ni* case / *de* case	3
ga case / *no* case / adnominal	2
inside *bunsetu* predicate:level A+	1

S/SBAR/SA/:	5
VP/ADVP/ADJP WHADVP/WHADJP	4
NP/PP/INTJ QP/PRT/PRN	3
others	2

Figure 6: An Example of Dependency Type Distance.

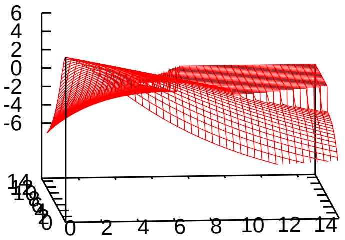

Figure 5: Smoothed Distance-Score Function for Proposed Model 1.

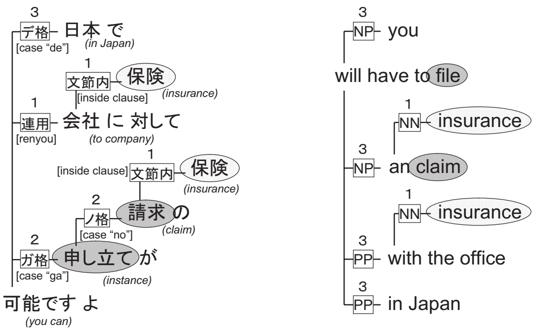

Figure 7: Example of Dependency Type Distance.

Then the dependency distance $d_S(a_i, a_j)$ and $d_T(a_i, a_j)$ are defined as:

$$d_S(a_i, a_j) = d_{Stype}(a_i, a_j)$$
$$d_T(a_i, a_j) = d_{Ttype}(a_i, a_j)$$

where $d_{Stype}(a_i, a_j)$ is defined as the dependency type distance between correspondence a_i and correspondence a_j in source language and $d_{Ttype}(a_i, a_j)$ is defined as the dependency type distance in target language.

The detailed definition of $d_{Stype}(a_i, a_j)$ and $d_{Ttype}(a_i, a_j)$ are as follows, where $a_i = (p_{Si}, p_{Ti})$ and $a_j = (p_{Sj}, p_{Tj})$:

If phrases p_{Si} and p_{Sj} are located in the same path of the dependency tree, then $d_{Stype}(a_i, a_j)$ is defined as the assumption of $ts_S(p_{Si})$ where p_{Si} is the node existing in the path between p_{Si} and p_{Sj}. $d_{Ttype}(a_i, a_j)$ is defined in the same way.

If phrases p_{Si} and p_{Sj} belong to different subtrees, then $d_{Stype}(a_i, a_j)$ is defined as the assumption of $ts_S(p_{Sx})$ where p_{Sx} is the node existing in the path between p_{Si} and the root node plus the assumption of $ts_S(p_{Sx})$ where p_{Sx} is the node existing in the path between p_{Sj} and the root node. $d_{Ttype}(a_i, a_j)$ is defined in the same way.

For example, in Figure 8, the distance between the connected two correspondences is $(d_S, d_T) = (1, 1)$, and they are very close both in Japanese and English, so the score is plus. On the other hand, in Figure 9, the distance is $(d_S, d_T) = (1, 7)$. They are close in Japanese, but far in English, so the score is minus. This is because their relation may be inappropriate.

Figure 10 shows the function $f(d_S, d_T)$. The dots are the result of automatic learning form gold standard data,

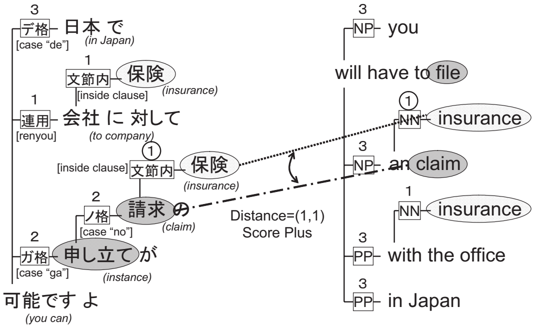

Figure 8: Example of Good Relation.

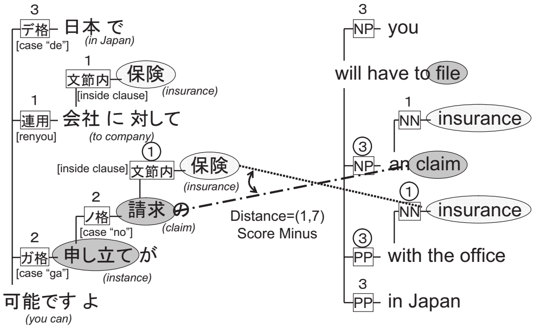

Figure 9: Example of Bad Relation.

and the meshed graph is manually modified function. The modification is done in the same manner as model 1.

3.5 Alignment Consistency Score

The consistency of alignment as a whole is denoted as a summation of $f(d_S, d_T)$ of all the combinations of correspondences. The best alignment is acquired by selecting correspondences to make the summation the greatest. However, it is not reasonable to expand all the cases and check them all since it easily falls into the combinatorial explosion. Therefore, the best alignment is searched greedily.

Focusing on an arbitrary correspondence, calculate the distance and the score between focusing correspondence and all the other correspondences. The score of the focusing correspondence is defined as a summation of the calculated scores. For all the other correspondences, the score of the correspondence is calculated in the same manner. The highest scored correspondence is regarded as a correct one and adopted. At the same time, negatively scored correspondences which exceed the negative threshold are rejected.

These steps are iterated until all the correspondences are adopted or rejected, and the approximately best alignment is acquired.

4 Experimental Results and Discussions

4.1 Alignment Experiment

We selected randomly 500 sentences of the newspaper domain corpus from NICT and there are also gold standard alignment data(Uchimoto et al., 2004). The unit of English sentence is a word, but the unit for Japanese sentence is

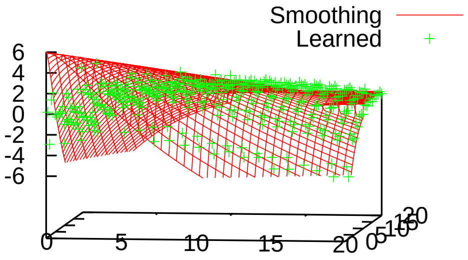

Figure 10: Distance-Score Function for Proposed Model 2.

different between our output and gold standard (although both ours and gold standard are phrase-base, the criterion of phrase is different). Therefore, the evaluation was done on character base for Japanese.

Character base evaluation should have a problem caused by the length of words. In the case of that, long words are misaligned, the accuracy get worse heavily. However, correctly aligned long words give good effect on the accuracy vice versa. Thus, it doesn't seem to be a big problem.

We used two bilingual dictionaries. One is KENKYUSYA's Japanese-English dictionary, consisting of 36K entries, and we extracted 214K one-to-one translations. The other is KENKYUSYA's English-Japanese dictionary, consisting of 50K entries, and we extracted 303K one-to-one translations.

The evaluation criterion is basic precision, recall, and F-measure. The gold standard data is annotated with only sure (S) alignments (no possible (P) alignments (Och and Ney, 2003a)). Figure 11 shows an example of evaluation. The black colored cells represent the gold standard alignment, and white boxes represent our output. Precision is an accuracy of the output, in the example, the precision is 9 / 12 = 75%. Recall is a coverage of the gold standard, in the example, the precision is 9 / 11 = 82%. The F-measure is a harmonic mean of Precision and Recall, and it is defined as:

$$F-measure = \frac{1}{\frac{1}{2*Precision} + \frac{1}{2*Recall}}$$

In the examle, F-measure is 78 points.

The evaluation results are shown in Table 1. "Baseline" is the heuristic method introduced in Section 3.3. "Model 1" is one of the proposed methods in which the distance-score function is refined. "Model 2" is a method using dependency type distance.

For comparison, we segmented the data using the morphological analyzer JUMAN (Kurohashi et al., 1994) and created alignments using the simple and standard statistical alignment tool GIZA++ (Och and Ney, 2003b). 10 iterations of each of the IBM model 1, 2, 3 and 4 were used for statistical alignment.

In Table 1, we can see about 2.0 points improvement be-

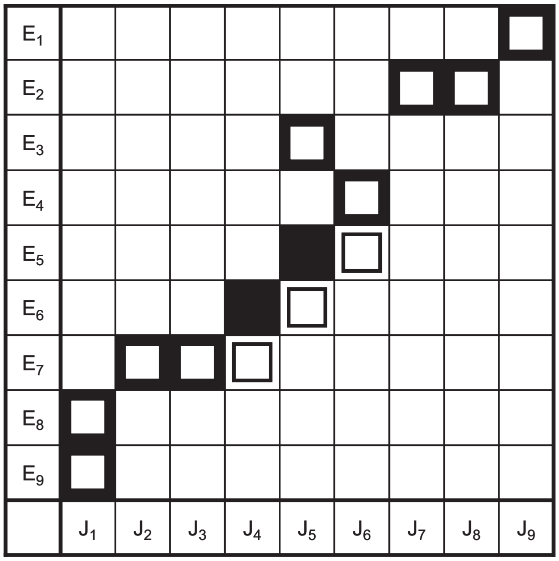

Figure 11: Alignment Evaluation Example.

Table 1: Alignment Evaluation Results.

	Precision	Recall	F-measure
Baseline	65.6	65.1	58.8
Model 1	67.4	68.0	60.8
Model 2	**68.6**	**68.5**	**61.9**
GIZA++	59.9	17.0	26.4

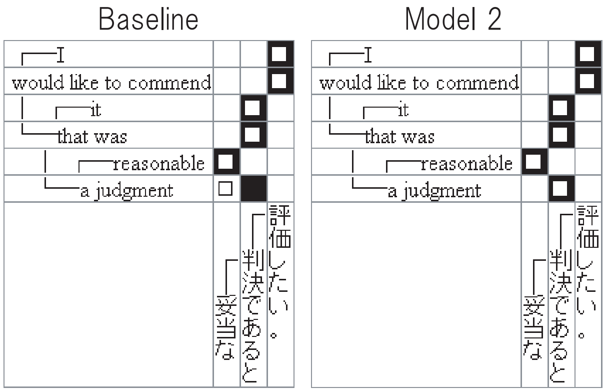

Figure 12: Sample Alignment 1.

tween Baseline and Model 1. This shows that the refining of distance-score function has great effect on the alignment accuracy. Moreover, we can say dependency type distance also works well from the improvement between Model 1 and Model 2.

At present, the distance-score function is defined by hand with the clue of automatically learned data, but we have not done any tuning to the function. In addition, the threshold of rejecting the bad-scored correspondence candidate also needs to be tuned. It is also important to reconsider the procedure of calculating consistency score. Through these modifications, the accuracy is expected to be more improved.

The dependency type score is also defined manually now. Since the number of types are not so large, it is not so hard to define by hand. However, there is no confidence in the degree of the score. There may seem to be several ways to learn the strength of the segmentation: learn from the monolingual corpus, or use parallel corpus, and so on. We need to take some of them and try to learn automatically.

The F-measure result of GIZA++ is extremely bad. Precision is not so different between GIZA++ and ours, but the Recall is very low. This is because, as it is often said, the statistical methods work well for language pairs that are not so different in the point of language structure (e.g., English and Spanish). Japanese and English have significantly different structure. Most famous difference is that Japanese sentences consist of SOV word order, but English word order is SVO. For such language pair as Japanese and English, deeper sentence analysis using NLP resources is necessary, like our method.

Sample alignments are shown in Figure 12 and Figure 13. Wrongly aligned parts in the baseline method are modified in the model 2.

4.2 Translation Experiment

We also conducted translation experiment. For this purpose, we utilized around 218K parallel sentences for training, and 500 sentences for testing. All the sentences are on newspaper domain(Utiyama and Isahara, 2003). The translation results are summarized in Table 2 and Table 3. Results were evaluated by n-gram precision based metrics, BLEU and NIST, with only one reference. We show 3, 4, 5-gram evaluation results in the tables.

From the result, it is able to be said that the translation quality is improved by our proposed method. The improvement of alignment accuracy leads to the improvement of the quality of translation examples used in translation step.

Sample translations are shown in Table 4. The numerals following the method name represent the 4-gram BLEU score of the output.

5 Conclusion

We have proposed a new alignment method using distance-score function $f(d_S, d_T)$ and dependency type distance $d_{type}(a_i, a_j)$ to improve structural phrase alignment. We defined the overall alignment consistency as a summation of $f(d_S, d_T)$ of all the pairs of correspondences. Our new method can evaluate alignment as a whole, and eliminate inappropriate correspondences, which baseline method was not able to eliminate. With the new method, we succeed to improve structural phrase alignment and achieved 35 points higher alignment accuracy than simple statistical method, and 3 points than baseline.

What we need to do in the future is to find the dependency type distance by machine learning, which is set by hand at present. There are several ways to learn: using parallel corpus or huge size single language corpus.

Our proposed method utilize the advantage of tree structure and highly rely on the information. This is very useful as shown in experiments, but it is also easy to cause alignment errors derived from the parsing errors. For Japanese, even thought the parsing accuracy is basically high, it

Table 4: Sample Translations.

Reference	under the revised foreign exchange and foreign trade law the government can restrict or suspend remittances to and trade with north korea
Baseline (35.2)	first of all it will serve restrictions and halting remittances to and to trade with north korea the foreign exchange and foreign trade law
Model 2 (48.6)	one is that restrictions and halting remittances to and trade with north korea the foreign exchange and foreign trade law
Reference	the dependence on the united states for security has allowed the country to devote all its energy to economic development
Baseline (0.0)	great relying on the us arsenal as the united states security security has allowed economic in to devote himself to japan
Model 2 (28.2)	great its dependence on the united states security security has allowed economic in to devote himself to japan

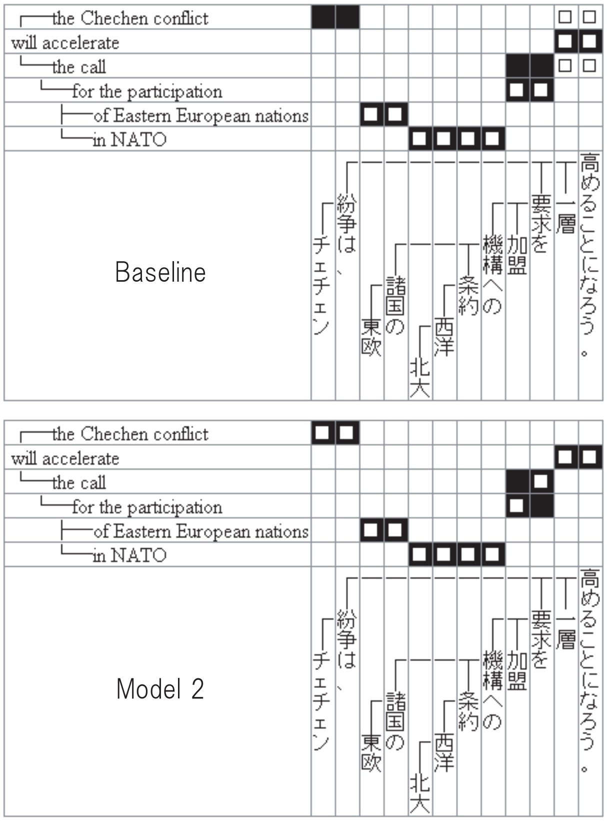

Figure 13: Sample Alignment 2.

Table 2: Translation Evaluation Results (BLEU).

	3-gram	4-gram	5-gram
Baseline	8.19	5.05	3.38
Model 2	**8.64**	**5.40**	**3.59**

Table 3: Translation Evaluation Results (NIST).

	3-gram	4-gram	5-gram
Baseline	2.6993	2.7003	2.7006
Model 2	**2.7063**	**2.7073**	**2.7077**

sometimes fails to parse the sentence and outputs wrong tree structure. For English, generally said to be difficult to correctly parse, the condition is rather tragic.

Therefore, we need some strategies which can modify the tree structure itself. This is able to be done by using our consistency criteria.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Association for Computational Linguistics*, 19(2):263–312.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan, June.

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 80–87.

Declan Groves, Mary Hearne, and Andy Way. 2004. Robust sub-sentential alignment of phrase-structure trees. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 1072–1078.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4):507–534.

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of The International Workshop on Sharable Natural Language*, pages 22–28.

Arul Menezes and Stephen D. Richardson. 2001. A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL) Workshop on Data-Driven Machine Translation*, pages 39–46.

Franz Josef Och and Hermann Ney. 2003a. A systematic comparison of various statistical alignment models. *Association for Computational Linguistics*, 29(1):19–51.

Franz Josef Och and Hermann Ney. 2003b. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kiyotaka Uchimoto, Yujie Zhang, Kiyoshi Sudo, Masaki Murata, Satoshi Sekine, and Hitoshi Isahara. 2004. Multilingual Aligned Parallel Treebank Corpus Reflecting Contextual Information and Its Applications. In *Proceedings of the MLR2004: PostCOLING Workshop on Multilingual Linguistic Resources*, pages 63–70.

Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 72–79.

Taro Watanabe, Kenji Imamura, and Eiichiro Sumita. 2002. Statistical machine translation based on hierarchical phrase alignment. In *9th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 188–197.

Ying Zhang and Stephan Vogel. 2005. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and long corpora. In *European Association for Machine Translation 2005 Conference Proceedings*, pages 294–301.