# Kyoto-U: Syntactical EBMT System for NTCIR-7 Patent Translation Task

Toshiaki Nakazawa Sadao Kurohashi
Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
nakazawa@nlp.kuee.kyoto-u.ac.jp kuro@i.kyoto-u.ac.jp

## Abstract

*This paper describes "Kyoto-U" MT system that attended the patent translation task at NTCIR-7. Example-based machine translation is applied in this system to integrate our study on both structural NLP and machine translation. In the alignment step, consistency criteria are applied to solve the alignment ambiguities and to discard incorrect alignment candidates. In the translation step, translation examples are combined using "bond" information, which can handle the word ordering without any statistics.*
**Keywords:** *EBMT, structural NLP, consistency.*

## 1 Introduction

Machine translation has been actively studied recently, and the major approach is Statistical Machine Translation (SMT), especially Phrase-based SMT. An alternative to SMT is Example-based machine translation (EBMT)[10]. The most important common feature between SMT and EBMT is to use a bilingual corpus, or translation examples, for the translation of new inputs. Both methods exploit translation knowledge implicitly embedded in translation examples, and make MT system maintenance and improvement much easier compared with Rule-Based Machine Translation.

On the other hand, EBMT is different from SMT in that SMT hesitates to exploit rich linguistic resources such as a bilingual lexicon and parsers; EBMT does not consider such a constraint. SMT basically combines words or phrases (relatively small pieces) with high probability [13]; EBMT tries to use larger translation examples. When EBMT tries to use larger examples, it can better handle examples which are discontinuous as a word-string, but continuous structurally. Accordingly, though it is not inevitable, EBMT can quite naturally handle syntactic information.

This paper describes our EBMT system, "Kyoto-U", challenged to NTCIR-7 patent translation task, and reports the evaluation results and discussion.

## 2 System Overview

Figure 1 shows the overview of our EBMT system on Japanese-English translation.

Using the example database, new input sentence is translated. The input sentence is parsed and transformed into dependency structure. For all the arbitrary sub-trees, available examples are searched. Translation examples are also parsed in both source and target sides. Of course there are many available examples for one sub-tree, so we give some scores to the examples and use the highest scored example. Also there are many types of sub-tree combinations. We search the best combination by beam-search based on the calculated scores.

In the example, four examples are used. They are combined and finally we can get the output dependency tree. We call the outside nodes of the actually used nodes as "bond" nodes. The bond nodes of one example are replaced by the other example, and thus two examples can be combined. Using the bond information, we don't need to consider word or phrase orders. Bond information naturally resolve the reordering problem.

## 3 Alignment of Parallel Sentences

Our system consists of two modules: an alignment module for parallel sentences and a translation module retrieving appropriate translation examples and combining them. First, we explain the alignment module.

The alignment of Japanese-English parallel sentences is achieved by the following steps, using a Japanese parser, an English parser, and a bilingual dictionary.

1. Dependency analysis of Japanese and English sentences.
2. Detection of Word/phrase correspondences.
3. Disambiguation of correspondences.
4. Handling of remaining words.

We explain these alignment steps in detail.

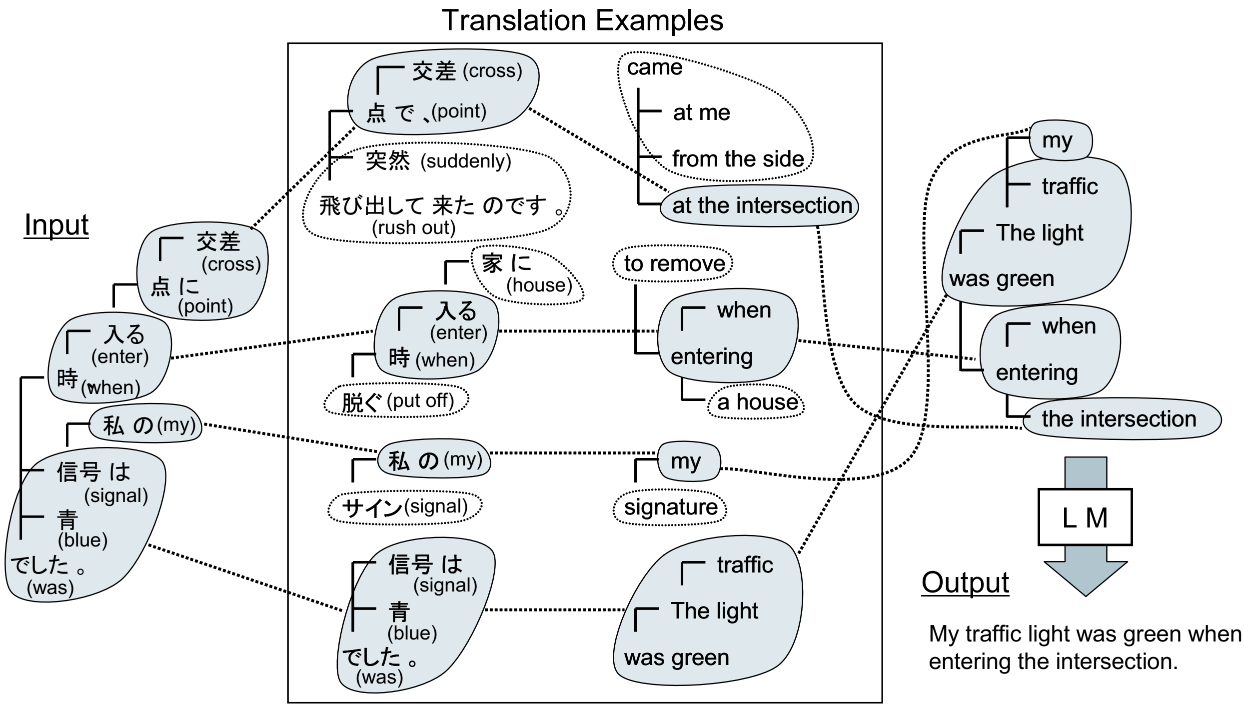


**Figure 1. An example of Japanese-English translation.**

### 3.1 Dependency Analysis of Japanese and English Sentences

Japanese sentences are converted into dependency structures using a morphological analyzer, JUMAN [6], and a dependency analyzer, KNP [5]. These tools can detect Japanese sentence structures with high accuracy: for the news article domain, 99% for segmentation and POS-tagging, and 90% for dependency analysis. They are robust enough to handle patent sentences and the accuracy is almost the same with news article sentences.

Japanese dependency structure consists of nodes which correspond with content words. Function words such as postpositions, affixes, and auxiliary verbs are included in content words' nodes.

For English sentences, Charniak's nlparser is used to convert them into phrase structures [2], and then they are transformed into dependency structures by rules defining head words for phrases. In the same way as Japanese, each content word composes a node of English dependency tree.

Figure 2 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.

### 3.2 Detection of Word/Phrase Correspondences

To detect the corresponding candidates between Japanese word/phrase and English word/phrase, we utilize multiple clues: bilingual dictionaries, transliteration, numeral expression matching, and sub-string alignment method.

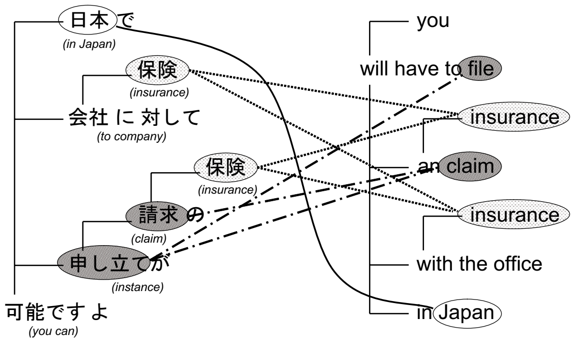


**Figure 2. Example of alignment.**

#### Bilingual Dictionaries

By looking up all of the pairs of Japanese words and English words in a dictionary, corresponding candidates are detected deterministically.

#### 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 automatically produced, and the similarities with words in the English sentence are calculated based on the edit distance. If there are similar word pairs whose edit distance exceeds a threshold, they are handled as a corresponding candidate.

For example, the following words can be considered as correspondences 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)

#### Numeral Expression Matching

By normalizing numeral expressions in each language, we get correspondence between them. For example, "二百六十万" in Japanese and "2.6 million" in English represent the same number "2600000".

#### Sub-String Alignment

Some parallel sentences include particular expressions, technical terms, and sometimes they are liberal translations. The clues listed above are not sufficiently able to handle such sentences. In other words, not able to find enough correspondences for aligning the parallel sentences with very high accuracy. Therefore, it is also necessary to use statistical alignment method.

As one of the statistical alignment methods, we make use of Cromieres's work [3]. This method can create arbitrary sub-string alignments by computing the co-occurrence counts of any pair of sub-string in parallel corpus. One big advantage of this method lies in unnecessity of morphological analyzer. This is effective for non-segmented languages such as Japanese or Chinese.

Let us consider a parallel sentence for example:

> Source: 参院選での社会党の大敗は必至と言われる。
> Target: It is said that the Social Democratic Party will suffer a major loss at the House of Councillors election.

a parallel dictionary finds only one correspondence "言われる ↔ said that". With the statistical method, more correspondences, "参院 ↔ the House of Councillors", "選 ↔ election", "の社会 ↔ the social", "党の ↔ Democratic Party" are found.

As a result of using the information above, for example, 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 (figure 2).

### 3.3 Selection of Correspondence Candidates

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

In figure 2, 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 globally consistent alignment, we define "consistency score" for a pair of correspondences [12]. We select the best set of correspondences by the following equation:

$$\underset{alignment}{\operatorname{argmax}} \frac{\sum_{i=1}^{n}\sum_{j=i+1}^{n} cons.score(a_i, a_j)}{n(n-1)/2} \tag{1}$$

where $a_i$ and $a_j$ are one of correspondences.

The detail is explained in the following sections.

#### 3.3.1 Consistency of Alignment

What is most important for the accuracy of alignment is the way of selecting correct correspondences among many candidates including ambiguous or unsuitable ones. Therefore, robust architecture which can align the parallel sentence as a whole is indispensable.

Using only a statistical method, it is difficult to realize high accuracy alignment for a linguistically different language pair such as Japanese-English. In contrast to this, our alignment system is based on dependency tree structure and using deeper NLP, it has the great advantage of absorbing the difference of linguistic structure with rich information.

Before introducing our proposed method, let us think of alignment consistency itself. In figure 3, the triangles represent the clauses of tree structure of each language, and the lines represent the correspondences. Among all the correspondences, one (on which a cross is placed) seems to be strange, which disturbs the consistency of whole alignment.

This inconsistency is visually apparent. To measure the inconsistency 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 in target language is near. Since the tree structure is constructed based on dependency information, such a case rarely happens. In other words, it is unlikely that two corresponding phrase pairs are structurally close in target language and they are far in source language simultaneously.

Therefore, suitably capturing the distance in each tree structure of all the pairs of lines leads to the overall

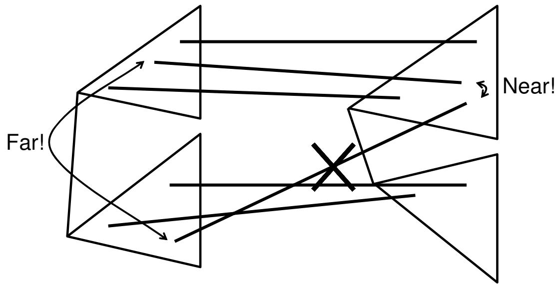


**Figure 3. Example of consistency.**

| Japanese | |
|---|---|
| predicate:level C | 6 |
| predicate:level B+/B | 5 |
| predicate:level B-/A | 4 |
| case *no* / rentai | 2 |
| inside clause predicate:level A- | 1 |
| others | 3 |

| English | |
|---|---|
| S/SBAR/SQ ... | 5 |
| VP/WHADVP WHADJP | 4 |
| ADVP/ADJP NP/PP/INTJ QP/PRT/PRN | 3 |
| others | 1 |

**Figure 4. Definition of dependency type distance.**

consistent alignment of the parallel sentence. For this purpose, we introduce “consistency score”, which is explained in the following sections.

#### 3.3.2 Consistency Score

To obtain consistent alignment within a sentence, we define a consistency score based on the dependency tree. Consistency scores are calculated for all the combinations of correspondences. If the relation of a pair is proper, the score is positive, otherwise, the score is negative.

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 the source language and $p_{Ti}$ represents the phrase of $a_i$ in the target language. $p_{Sj}$ and $p_{Tj}$ are defined in the same way.

Then, the dependency distance of the source language $d_S(a_i, a_j)$ is defined as the distance between $p_{Si}$ and $p_{Sj}$, and the dependency distance of the 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:

$$cons.score(a_i, a_j) = f(d_S, d_T) \qquad (2)$$

where $d_S = d_S(a_i, a_j)$ and $d_T = d_T(a_i, a_j)$. $f(d_S, d_T)$ is a function that maps a pair of distance to the score. The detail of the distances $d_S$, $d_T$ and the function $f$ are illustrated in the following sections.

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

Correct correspondences are supported by their neighbor correspondences (here, *neighbor* means the distance is small in *both* sides). Such relations produce good scores and contribute to the alignment consistency.

#### 3.3.3 Dependency Type Distance

In this section, we explain how we calculate the dependency distances $d_S$ and $d_T$. In a simple setting, all the distances of each branch are set to 1. However, it is natural to use deeper knowledge derived from NLP resources.

The Japanese dependency analyzer KNP outputs dependency type information for each phrase, and Charniak’s nlparser also outputs phrase tag information. According to such information, we set dependency type score for each branch to show the strength of separation between a phrase and its parent phrase. For example, if the branch represents the segmentation of compound noun, the strength of dependency is strong (dependency score is small), on the other hand, if it represents the segmentation of clauses, the strength is weak (dependency score is high).

Since there are at most around 30 pattens of dependency type, the scores are set by hand subjectively and figure 4 shows part of them. An example of applying dependency type score is shown in figure 5, where the label placed on each branch represents the dependency type and the number placed above the label represents dependency type score.

The distances $d_S$ and $d_T$ between two correspondences are defined as the sum of dependency type scores from one node to the other in source language and target language respectively. For example, in figure 5, the distance between the connected two correspondences in Pair 1 is $(d_S, d_T) = (1, 1)$. Also in Pair 2, the distance is $(d_S, d_T) = (1, 7)$.

Without tree structure, the difference between these two patterns could be regarded as small (in some cases, no difference) because the two “insurance” is both close to the “claim” from the view point of simple sequence of words. This illustrates the advantage of using tree structure.

#### 3.3.4 Distance-Score Function

The distance-score function $f(d_S, d_T)$ maps the distance $(d_S, d_T)$ to a score which reflects the soundness of the distance. To define this function, we observed what was the linguistic “behavior” in some real data. Using a gold standard alignment data [15], which includes about 40K sentence pairs, we learned the frequency distribution of distance pair. Figure 6 shows the result of automatic learning form gold stan-

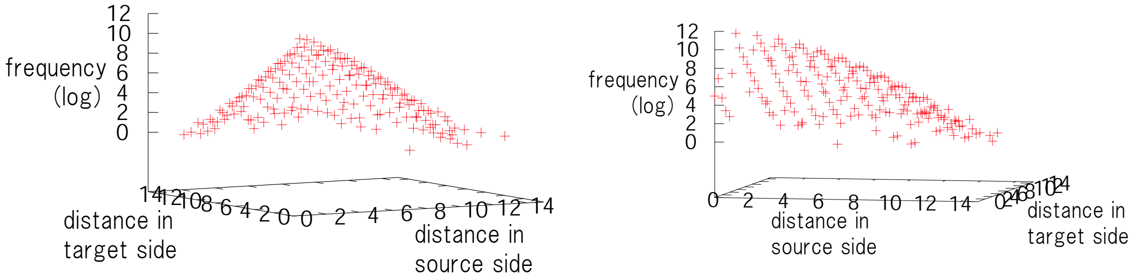


**Figure 6. Distance pair distribution learned from gold standard data.**

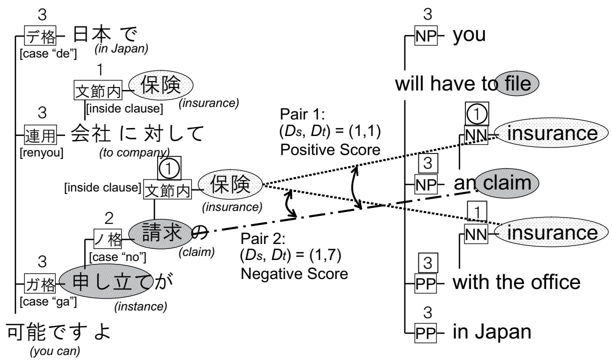


**Figure 5. Example of dependency type distance and distance score.**

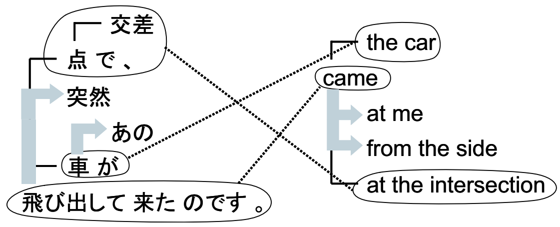


**Figure 7. Example of remaining words handling.**

dard data. The result shows that the frequency of the equidistant pairs are high and that of non-equivalent pairs are extremely low.

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

- Positive if both $d_S$ and $d_T$ are small, which means the relation between the two correspondences is appropriate.
- 0 if both $d_S$ and $d_T$ are large, for the relation is not so important if they are far from each other.
- Negative if the margin between $d_T$ and $d_S$ is large, which means the relation between the two correspondences is inappropriate.

For example, in figure 5, a positive score is given to $(d_S, d_T) = (1, 1)$, and a negative score to $(d_S, d_T) = (1, 7)$. To meet the requirement above, we defined the distance-score function on the basis of sigmoid function:

$$f(d_S, d_T) = \begin{cases} 0 & \text{if } d_S, d_T > b \\ 1 - \frac{a}{1+e^{-t(d_S, d_T)}} & \text{otherwise} \end{cases} \quad (3)$$

where the parameters for the function, $a$, $b$, $t(d_S, d_T)$, are set by hand. Automatic optimization of both dependency type distance and distance-score function parameters is a future work.

### 3.4 Handling of Remaining Words

The alignment procedure so far found all 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 all parallel sentences have an appropriate translation relation. Accordingly, if neither root nodes (of the Japanese dependency tree and the English dependency tree) are included in any correspondences, the new correspondence of the two root nodes is generated. If either root node is remaining, it is merged into the correspondence of the other root node.

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

In the case of figure 2, "あの (that)" is merged into the correspondence "車 (car) ↔ the car", since it is within an NP. Then, "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".

### 3.5 Translation Example Database

Once we detect basic correspondences in the parallel sentences, all basic correspondences and all combination of adjoining basic correspondences (both in Japanese and English dependency trees) are registered into the translation example database.

From the parallel sentences in figure 7, the three basic correspondences and their combinations such as "交差点で，突然飛び出して来たのです ↔ came at me from the side at the intersection" and "突然あの車が飛び出して来たのです ↔ the car came at me from the side" are registered.

## 4 Translation

In the translation process, first, a Japanese input sentence is converted into the dependency structure as in the parallel sentence alignment. Then, translation examples for each sub-trees are retrieved. Finally, the best translation examples are selected, and their English expressions are combined to generate the English translation (figure 1).

### 4.1 Retrieval of Translation Examples

At first, the root of the input sentence is set as the retrieval root, and each sub-tree whose root is the retrieval root is retrieved step by step. If there is no translation example for a sub-tree, the retrieval for the current retrieval root stops. Then, each child node of the current retrieval root is set to the new retrieval root and its sub-trees are retrieved.

In the case of figure 1, sub-trees from the root node "でした (was)" are retrieved: "でした (was)", "青 (blue) でした (was)", "信号 (signal) は でした (was)", "信号 (signal) は 青 (blue) でした (was)" and so on. Then, sub-trees from "青 (blue)" and sub-trees from " 信号 (signal) は" are retrieved step by step.

If no translation example is found for a Japanese node, the bilingual dictionary is looked up and its translation is used as an translation example. (If there is no entry in the dictionary we output nothing for the node.)

### 4.2 Selection of Translation Examples

Then, among the retrieved translation examples, the good ones are selected to generate the English translation.

The basic idea of example-based machine translation is preferring to use larger translation example, which considers larger context and could provide an appropriate translation. According to this idea, our system also selects larger examples.

The selection criterion is based on the size of translation examples (the number of matching nodes with the input), plus the similarities of the neighboring outside nodes, ranging from 0.0 to 1.0 depending on the similarity calculated by a thesaurus. The similar outside node is used as a bond to combine two translation examples, as explained in the next section.

For example, if the size of a translation example is two, and the outside parent node is similar to the outside parent node of the matching Japanese input sub-tree by 0.3 similarity, and one outside child node is also similar to the corresponding input by 0.4, the score of the translation example becomes 2.7.

The set of translation examples just enough for the input is searched in a greedy way. That is, the best translation example is selected among all the examples first, and then the next best example is selected for the remaining input nodes, and this process is repeated.

### 4.3 Combination of Translation Examples

It is easy to generate an English expression from a translation example, because it contains enough information about English dependency structure and word order. The problem is how to combine two or more translation examples.

However, in most cases, the bond node is available outside of the example, to which the adjoining example is attached. There are two types of bond nodes: a child bond and a parent bond.

If there is a child node, it is easy to attach the adjoining example on it. For example, in figure 1, the translation example "入る (enter) 時 (when)" has a child bond, "家 (house) に", corresponding to "a house" in the English side. The adjoining example "交差点 (で) ↔ (at) the intersection" is attached on "家に", which means "house" is replaced with "the intersection".

On the other hand, a parent bond tells that the translation example modifies its head from the front or from behind, but there is no information about the order with the other children. Currently, we handle it as the first child if it modifies from the front; as the last child if it modifies from behind. In figure 1, "私の ↔ my " has a parent bond, "サイン ↔ sign" and it tells that "my" should modify its head from the front. Then, "my" is put to the first child of "the light", before "traffic".

It is not often the case, but if there is no bond, we use translation patterns which are automatically acquired from the training corpus [11].

## 5 Results and Discussion

Our translation system utilized PSD-1 for training [4]. In addition, Japanese and English parsers and a bilingual dictionary were used as external resources.

Table 1 and 2 shows the result of formal trun of NTCIR-7 patent translation task.

**Table 1. Intrinsic JE Evaluation Result.**

| | BLEU | Improved | BLEU-m300 | BLEU-m600 | Adequacy | Fluency |
|---|---|---|---|---|---|---|
| Kyoto-U | 21.57 | 21.98 | 29.35 | 35.49 | **2.85** | 3.35 |
| Moses | **27.14** | — | 36.02 | 43.40 | 2.81 | 3.55 |

We examined the translation results and found out that it was not the case that there was a few major problems, but there were variety of problems, such as parsing errors of both languages, excess and deficiency of the bilingual dictionary, and the inaccurate and inflexible use of translation examples.

Now, let us discuss the biggest question: "is the current parsing technology useful and accurate enough for machine translation?" If the translation performance was significantly better than the other systems without parsing, we could answer "YES" to the question. However, unfortunately our performance is not best in both BLEU and human evaluation.

There is a very interesting point that our results are in the middle of the whole systems in view of BLEU score, they are higher-ranked in view of human-evaluation. Compared to the results of Moses (table 1, JE direction), although our result is more than 5 points behind in BLEU score, there is almost no difference in human-evaluation, and ours overcomes Moses in the measure of Adequacy. From this, we can say two things:

- As is often addressed, BLEU score does not always appropriately reflect the goodness of the translations, and it is unfavorable for comparing the translation qualities of the systems which are not using same techniques.
- Our EBMT system exploits the grobal sentence structrure information even during alignment (training) step. This might lead to the correct sentence structure of the translations and could achieve reasonably high Adequacy.

For the result of EJ direction, it was not good in both BLEU score and human-evaluation. This is because our system was developed with the source language supposed to be Japanese, so it works incorrectly in a case where the source language is not Japanese. In Japanese sentence, all the phrases depend on the latter phrase than itself, and the last phrase is always a head of the sentence. Of course this is not same to English. The handling of the case where a phrase depends on the former phrase was not appropriate.

The up-to-date BLEU scores with the modifications above and other trivial problems are shonw in "Improved" columns. There is small improvement in JE direction, however in the EJ direction, there is about 1 point improvement. While it seems the result was not changed drastically in view of BLEU score, we could see much improvement by manual check of translations. We can say the quality is very close to the Moses.

**Table 2. Intrinsic EJ Evaluation Result.**

| | BLEU | Improved | Adequacy | Fluency |
|---|---|---|---|---|
| Kyoto-U | 22.65 | 23.54 | 2.42 | 2.54 |
| Moses | 30.58 | — | 2.90 | 3.69 |

As we mentioned above, parsing errors are not a principal cause of translation errors, but these are not a few. One of the possible countermeasures is to reconsider the learning process of an English parser. The English parser used here is learned from the penn Treebank, and seems to be vulnerable to patent sentences.

There often occurs typical expressions, mathematical or chemical formulas and so on in patent corpus. Given such sentences, the parser will produce unusual parse trees. We did not adopt any pre-processes to the training corpus and input sentences. However, it must be necessary to handle such typical expressions appropriately.

Furthermore, it is quite possible to improve parsing accuracies of both languages complementarily by taking advantage of the difference of syntactic ambiguities between the two languages [8]. This approach may not substantially improve the parsing accuracy of the patent sentences, but is promising for translating longer general sentences.

Other main points are as follows:

- Automatic evaluation methods are a little advantageous to SMT [7],[1].
- The soundness of dictionaries heavily affects on the accuracy of alignment.
- The extension rules of remaining nodes should be revised.
- The constraint of selecting translation examples should be more robust. It is currently impossible to use 'almost equal' examples to the input sentence, such as those that differ perhaps only with respect to whether or not it contains a negation adverb such as 'not'.

## 6 Related Work

MSR's MT system [14] also applied EBMT integrating syntax structure of both source language and target language. While our method is different from that work in the follwing point: They use "Logical Form(LF)" [9] which abstracts away language-particular aspects of the sentence pair such as function words. On the other hand, we use full structure of the sentences, and this can handle not only shorter, easy sentences but longer, complicated sentences precisely.

## 7 Conclusion

As we stated in Introduction, we not only aim at the development of machine translation through some evaluation measure, but also tackle this task from the comprehensive viewpoint including the development of structural NLP. The examination of translation errors revealed several problems, such as parsing, soundness of dictionaries and selection of translation examples. Resolving such problems is considered to be an important issue not only for MT but also for other NLP applications. We pursue the study of machine translation from this standpoint continuously.

## References

[1] C. Callison-Burch, M. Osborne, and P. Koehn. Re-evaluating the role of bleu in machine translation research. In *Proceedings of 11 th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, 2006.

[2] E. Charniak and M. Johnson. 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, 2005.

[3] F. Cromieres. Sub-sentential alignment using substring co-occurrence counts. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, pages 13–18, 2006.

[4] A. Fujii, M. Utiyama, M. Yamamoto, and T. Utsuro. Overview of the patent trajnslation task at the ntcir-7 workshop. In *Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access*, 2008.

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

[6] S. Kurohashi, T. Nakamura, Y. Matsumoto, and M. Nagao. Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of The International Workshop on Sharable Natural Language*, pages 22–28, 1994.

[7] G. Leusch, N. Ueffing, and H. Ney. Cder: Efficient mt evaluation using block movements. In *Proceedings of 11 th Conference of the European Chapter of the Association for Computational Linguistics*, pages 241–248, 2006.

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

[9] A. Menezes and S. D. Richardson. 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, 2001.

[10] M. Nagao. A framework of a mechanical translation between Japanese and English by analogy principle. In *Proceedings of the international NATO symposium on Artificial and human intelligence*, pages 173–180, 1984.

[11] T. Nakazawa and S. Kurohashi. Example-based machine translation using automatically acquired translation patterns for function words. In *Proceedings of the 14th Annual Meeting of the Natural Language Processing*, 2008. in Japanese.

[12] T. Nakazawa, K. Yu, and S. Kurohashi. Structural phrase alignment based on consistency criteria. In *In Proceedings of Machine Translation Summit XI (MT-Summit XI)*, pages 337–344, 2007.

[13] F. J. Och, C. Tillmann, and H. Ney. Improved alignment models for statistical machine translation. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, 1999.

[14] S. D. Richardson, W. B. Dolan, A. Menezes, and M. Corston-Oliver. Overcoming the customization bottleneck using example-based mt. In *Proceedings of the workshop on Data-driven methods in machine translation - Volume 14*, pages 1–8, 2001.

[15] K. Uchimoto, Y. Zhang, K. Sudo, M. Murata, S. Sekine, and H. Isahara. 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, 2004.