Example-based Machine Translation based on Deeper NLP

Toshiaki Nakazawa, Kun Yu

Graduate School of
Information Science and Technology
University of Tokyo
nakazawa@kc.t.u-tokyo.ac.jp

Daisuke Kawahara

National Institute of
Information and Communications
Technology
dk@nict.go.jp

Sadao Kurohashi

Graduate School of
Informatics
University of Kyoto
kuro@i.kyoto-u.ac.jp

Abstract

This paper describes our Kyoto-U system that attended the IWSLT06 Japanese-English machine translation task. Example-based machine translation is applied in this system to integrate our study on both structural NLP and machine translation.

1. Introduction

Machine translation has been actively studied recently, and the major approach is Statistical Machine Translation (SMT). An alternative to SMT is Example-based machine translation (EBMT)[1]. 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 [2]; 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.

Besides that, the difference in the problem setting between EBMT and SMT is also important. SMT is a natural approach when linguistic resources such as parsers and a bilingual lexicon are not available. On the other hand, in case that such linguistic resources are available, it is also natural to see how accurate MT can be achieved using all the available resources.

We chose the latter problem setting to conduct EBMT research, and here we would like to mention two reasons we chose this setting.

First, we are aiming at the improvement of structural NLP. We have been conducting research on Japanese morphological analyzers, parsers, and anaphora/omission analyses. MT is considered as an application of these fundamental technologies. Amelioration of fundamental NLP technologies naturally improves applications, and applications give some feedback to fundamental NLP, pointing out the shortcomings. Needless to say, MT is not the only NLP application, and monolingual NLP applications such as man-machine interfaces and information retrieval can benefit from the improvement of fundamental NLP.

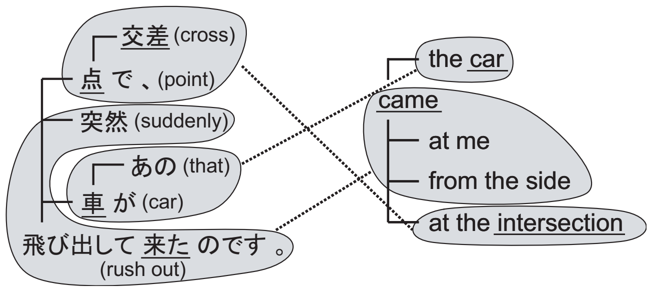

Figure 1: An example of parallel sentence alignment. (The root of a tree is placed at the extreme left and phrases are placed from top to bottom. Correspondences of underlined words were detected by a bilingual dictionary.)

The second point is that, in practice, we often encounter some cases in which the EBMT problem setting is suitable. That is, if there is no huge bilingual corpus which enables SMT, but some very similar translation examples are available, it would be nice if automatic translation or translation assistance can be provided by exploiting the examples. For example, translation of manuals when translations of the old version manuals are available, and patent translation when translations of the related patents are available. Or, in the translation of an article, the translations to a certain point can be used effectively as translation memory step by step, because the same or similar expressions/sentences are often used in an article. In such cases, an EBMT approach which tries to find larger translation examples is suitable.

This paper describes our Japanese-English EBMT system, Kyoto-U, challenged to IWSLT06, and reports the evaluation results and discussion.

2. 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 (see Figure 1).

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

Among IWSLT06 39,953-sentence training data, some pairs consist of two or more sentences. We utilized the pairs with the same number of Japanese sentences and English sentences, and separated them into one-to-one Japanese–English sentence pairs. As a result, we utilized 43,318 sentence pairs.

We explain these alignment steps in detail.

2.1. Dependency Analysis of Japanese and English Sentences

Japanese sentences are converted into dependency structures using a morphological analyzer, JUMAN, and a dependency analyzer, KNP [3]. 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 travel domain conversations 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 [4], 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.

The "Self-trained parser" introduced in ACL-COLING06 [5] is used in our system. Since Charniak's nlparser was trained on the penn Treebank, it is not necessarily suitable for travel domain conversations. In some cases, basic English sentences were wrongly parsed by the parser.

2.2. Pronoun Estimation

In Japanese-English translation, omission of pronouns often causes problems. In conversational utterances, Japanese pronouns such as "私 (I)", "あなた (you)", "これ (this)" are often omitted, and this could cause erroneous translations.

There are two cases when pronoun omission causes erroneous translations. One is that a pronoun is omitted in a translation example and not omitted in an input sentence. In such cases, there is no correspondence for the English pronoun, and it is merged into the other (usually predicate's) correspondence. If this merged pronoun is used in the translation, it overlaps with the pronoun from the input. For example, if the translation example "胃 (stomach) が 痛いのです (ache) ↔ I 've a stomachache" is used to translate "私 (I) は胃 (stomach) が 痛いです (ache)", the translation becomes "I I 've a stomachache" naively.

The opposite case also causes erroneous translations. That is, when a pronoun is in a translation example and is omitted in an input, the ungrammatical English sentence without pronoun is generated. For example, when "これ (this) を 日本 (Japan) へ 送って下さい (mail) ↔ will you mail this to Japan" is used to translate "日本 (Japan) へ 送って下さい (mail)", the translation becomes "will you mail to Japan" by eliminating "これを ↔ this".

To solve these problems, omitted pronouns are estimated by the clues of modality and subject case. Then, an extra node, which represents pronoun omission, is inserted.

2.3. Detection of Word/Phrase Correspondences

Japanese word/phrase to English word/phrase correspondences are detected by two methods.

One is to use a Japanese-English dictionary. We utilized KENKYUSYA Japanese-English and English-Japanese dictionaries. From these two dictionaries, we got about 300K entries.

The other method handles transliteration. For possible person names and geo 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 words exceeding the threshold, they are handled as a correspondence.

For example, the following words can be looked as correspondence by the transliteration module, which can not 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)

The units of correspondences are nodes, and function words in nodes are included in the correspondences of content words. If the bilingual dictionary and transliteration module detect a correspondence with two or more content words, the correspondence of two or more nodes is generated accordingly. In Figure 1, for example, the two Japanese nodes "交差 (cross)" and "点 (point) で" corresponds to the one English node "at the intersection".

After all the correspondences are found, the inserted nodes explained in section 2.2 are aligned if there are no aligned pronouns in English side.

2.4. Disambiguation of Correspondences

The method described in the previous section sometimes detects ambiguous correspondences, that is, one-to-many or many-to-many correspondences. Such ambiguity is resolved based on harmonious criteria.

Suppose there is a correspondence X with ambiguity, and there is an unambiguous correspondence Y with the distance n in the Japanese dependency tree and the distance m in the English dependency tree, we give the score $1/n + 1/m$ to the correspondence X, since we can consider that the nearer Y is to X, the more strongly Y supports X. Here we define the distance of correspondences as the number of traversing nodes in a dependency tree. For example, in Figure 1, the distance between “the car” and “came” is 1, and that between “the car” and “at the intersection” is 2.

Then, we accept the correspondence with the highest sum of its neighbouring correspondences’ score, and reject the others conflicting with the accepted one. This calculation is repeated until all the ambiguous correspondences are resolved.

2.5. 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 1, “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”.

We call the correspondences constructed so far as *basic correspondences*.

2.6. 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 1, 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.

3. 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 2).

3.1. 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 2, 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.)

3.2. 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

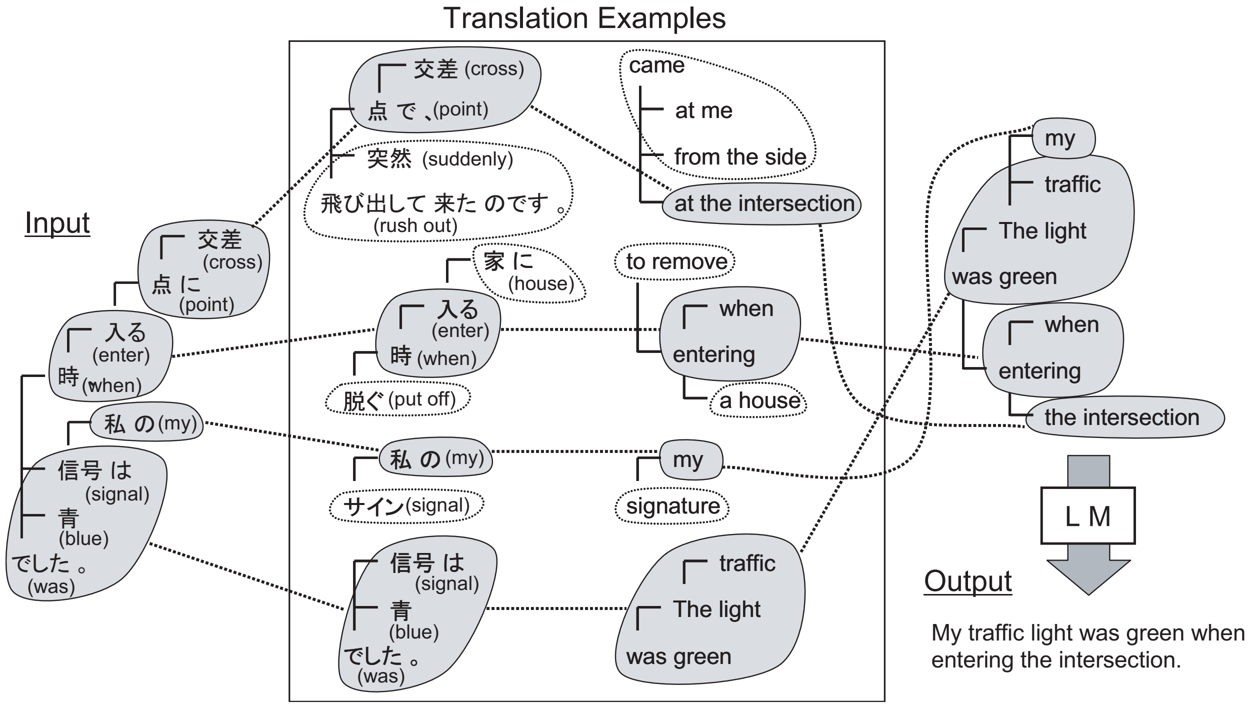

Figure 2: An example of Japanese-English translation.

becomes 2.7. [1]

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.

3.3. 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 2, 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 2, "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, the order of combining two translation examples is controlled by heuristic rules.

3.4. Handling of Numerals

Numerals in Japanese are translated into English in several ways.

- cardinal : 124 → one hundred twenty four
- ordinal (e.g., day) : 2 日 → second
- two-figure (e.g., room number, year) : 124 → one twenty four
- one-figure (e.g., flight number, phone number) : 124 便 → one two four
- non-numeral (e.g., month) : 8 月 → August

At the time of parallel sentence alignment, it is checked in which type Japanese numerals are translated.

[1]We proposed a method of selecting translation examples based on translation probability [6]. Though we used size- and similarity-based criteria for IWSLT06 because of time constraints, we are planning to use probability-based criteria from now on.

Translation examples of non-numeral type are used only if the numerals match exactly (“8 月 → August” cannot be used to translate “7 月”). However, translation examples of the other types can be used by generalizing numerals, and the input numeral is transformed according to the type. For example, “2 日 → second” can be used to translate “13 日”, transforming to the ordinal, “thirteenth”.

3.5. Language Model

Even though we are handling omitted pronouns, not aligned pronouns are still sometimes left in English side. These pronouns are merged into the head node of the sentence, and it causes the same problem shown in section 2.2.

To solve this problem, the merged pronoun is marked at the alignment, and two translations with it and without it are generated and ranked using a language model of English. In addition, a bond node, which is not used for translation in a normal case, is also used as a translation candidate when the bond node is a pronoun, and the best translation is selected using a language model of English.

In the IWSLT06, we used English sentences in 39,953 training data and Cam_Toolkit by CMU for a English language model [7].

4. Japanese Flexible Matching

In natural language, many expressions have almost the same meaning, which brings great difficulty to many natural language processing tasks, including machine translation.

For example, suppose an input sentence “ホテル に (to) 一番 (best) 近い (near) 駅 (station) はどこ ですか (where is)” is given to an example-based machine translation system. Even if a very similar translation example “旅館 (Japanese-style hotel) の (to) 最寄り (nearest) の (of) 駅 (station) はどこ ですか (where is) ↔ Where’s the nearest station to the hotel?” exists in the translation memory, a simple exact matching method cannot utilize this example for the translation.

To solve these problems, we use flexible matching method, which can assimilate the expressive divergence. It has the following two features:

1. Synonym relations and hypernym-hyponym relations are automatically extracted from an ordinary dictionary.

2. Extracted synonymous expressions are effectively handled by SYNGRAPH data structure, which can pack expressive divergence.

A thesaurus is a knowledge source to provide synonym and hypernym-hyponym relations. However, existing thesauri are not appropriate for flexible matching. One reason is that the number of words assigned to one unit is often too large, and it is difficult to distinguish synonyms in a narrow sense from similar words. Such distinction is important for machine translation in which translation examples with almost the same meaning should be retrieved. Information retrieval also can utilize such distinction as a degree of matching score. Another problem of existing thesauri is that they rarely contain relations between a word and a phrase.

To overcome such problems in existing thesauri, we use an ordinary dictionary to extract wide-coverage and precise synonym and hypernym-hyponym expressions. The extracted expressions include not only hypernym-hyponym relations such as 夕食 (dinner) and 食事 (meal), and basic synonym relations such as 旅館 (Japanese-style hotel) and ホテル, but also adverbial synonym such as 一番 (best) and 最も (most) and synonym relations between a word and a phrase such as 最寄り (nearest) and 一番 近い (best near).

Another point of our method is introducing SYNGRAPH data structure to this flexible matching. So far, the effectiveness of handling expressive divergence has been only shown for information retrieval using a thesaurus. However, it is based on a bag-of-words approach and does not pay attention to sentence-level synonymy with syntactic structure. However, machine translation requires such precise handling of synonymy, and advanced information retrieval and question answering also need it. To handle sentence-level synonymy precisely, we have to consider the combination of expressive divergence, which may cause combinatorial explosion. To overcome this problem, we assign an ID to each synonymous expression and then introduce SYNGRAPH data structure to pack expressive divergence.

For a sentence, by applying synonymous expressions extracted in the previous section, many paraphrases can be generated in a combinatorial way. The following are some example paraphrases generated from a fairly short seed sentence.

	ホテル	に	一番	近い	駅	は	どこ	ですか
	(hotel	to	best	near	station	TOP	where	is)
=	ホテル	に	最も	近い	駅	は	どこ	ですか
	(hotel	to	most	near	station	TOP	where	is)
=	ホテル	の	最寄り	の	駅	は	どこ	ですか
	(hotel	to	nearest	of	station	TOP	where	is)
=	旅館	に	*ichiban*	近い	駅	は	どこ	ですか
	(hotel	to	best	near	station	TOP	where	is)
	ホテル	に		近い	駅	は	どこ	ですか
	(hotel	to		near	station	TOP	where	is)
	...							

To do a flexible matching, we need to recognize the synonymous relations among this expressive divergence. However, the combination of synonymous expressions will cause combinatorial explosion, which makes both pre-unfolding and dynamic search are infeasible.

To handle this problem, we introduce SYNGRAPH data structure which packs all synonymous expressions and can generate all the possible paraphrase sentences. Figure 3 shows an example SYNGRAPH which can generate the above paraphrases.

The basis of SYNGRAPH is the dependency structure of the original sentence (so, in this paper we always employ a robust parser [3]). In the dependency structure, each

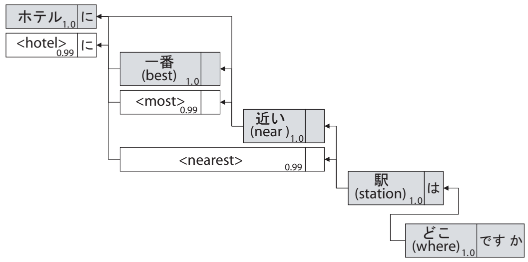

Figure 3: Example of SYNGRAPH.

Table 1: JP to EN Evaluation results.

	BLEU	NIST
Dev 1	0.5087	9.6803
Dev 2	0.4881	9.4918
Dev 3	0.4468	9.1883
Dev 4	0.1921	5.7880
Dev 4 ASR	0.1590	5.0107
Test	0.1655	5.4325
Test ASR	0.1418	4.8804

node consists of one content word and zero or more function words, which is called as a *basic node* hereafter. If the content word of a basic node belongs to a synonymous group, a new node with the SYNID is attached to it, and it is called a *SYN node* hereafter. For example, in Figure 3, the shaded nodes are basic nodes and the other nodes are SYN nodes [2].

Then, if the expression conjoining two or more nodes corresponds to one synonymous group, a SYN node will be added there. For example, in Figure 3, ⟨ nearest ⟩ is such a SYN node. Furthermore, if one SYN node has a hyper synonymous group in the synonymy database, the SYN node with the hyper SYNID is also added.

In this SYNGRAPH data structure, each node has a score, NS (Node Score), which reflects how much the expression of the node is shifted from the original expression.

5. Results and Discussion

Our Japanese-English translation system tried two tasks: manual manuscript translation and ASR output translation (for ASR output we just translated the best path, though). Our system utilized Japanese and English parsers and a bilingual dictionary, and it was categorized to "Open Data Track".

Table 1 shows evaluation scores for development set 1 to 4 and the test set.

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.

[2] The reason why we distinguish basic nodes from SYN nodes is to give priority to exact matching over synonymous matching.

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 average and we cannot claim that. Currently, we can at least dispel the suspicion that parsing might cause side-effects and lower translation performance.

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 conversational sentences in travel domain.

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 travel domain sentences, because of their short length, but is promising for translating longer general sentences.

Other main points are as follows:

- Punctuation marks are removed and automatic insertion sometimes failed. This causes parsing errors and leads to translation errors.
- Automatic evaluation methods are a little advantageous to SMT [9],[10].
- 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' exanples 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 [11] also applied EBMT integrating syntax structure of both source language and target language. While our method is different from that work in the follwing two points:

- They use "Logical Form(LF)" [12] 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.

- We use Japanese flexible matching method. This can greately help translations of Japanese because Japanese has quite a variation in expressions. With Japanese flexible matching, we can find more correspondences in alignment part, and can use more appropriate examples in translation part.

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.

8. References

[1] 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*, 1984, pp. 173–180.

[2] F. J. Och, C. Tillmann, and H. Ney, "Improved alignment models for statistical machine translation," in *Proceedings of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999, pp. 20–28.

[3] S. Kurohashi and M. Nagao, "A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures," *Computational Linguistics*, vol. 20, no. 4, pp. 507–534, 1994.

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

[5] D. McClosky, E. Charniak, and M. Johnson, "Reranking and self-training for parser adaptation," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 337–344. [Online]. Available: http://www.aclweb.org/anthology/P/P06/P06-1043

[6] E. Aramaki, S. Kurohashi, H. Kashioka, and H. Tanaka, "Probabilistic model for example-based machine translation," in *Proceedings of MT Summit X*, 2005, pp. 219–226.

[7] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the CMU-Cambridge toolkit," in *Proceedings of the European Conference on Speech Communication and Technology*, 1997, pp. 2707–2710.

[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*, 1993, pp. 23–30.

[9] 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*, 2006, pp. 241–248.

[10] 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*, 2006, pp. 249–256.

[11] 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*, 2001, pp. 1–8.

[12] 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*, 2001, pp. 39–46.