Hybrid Spoken Language Translation Using Sentence Splitting Based on Syntax Structure

Satoshi Kamatani, Tetsuro Chino and Kazuo Sumita
Corporate Research & Development Center, Toshiba Corporation
1 Komukai-Toshiba-cho Saiwai-ku Kawasaki, JAPAN, 212-8582
satoshi.kamatani@toshiba.co.jp

Abstract

In this paper, we propose a hybrid spoken language translation method utilizing sentence segmentation. By portioning the sentence using the result of syntax analysis, we can utilize rule-based control of the integration of subtranslations translated by a suitable method for each segment.

We also report a preliminary experiment on translation quality of our prototype Japanese-to-English translation system. We confirmed that our method achieved a 13.4% advantage in NIST score for the individual RBMT method, and a 6.0% advantage for the individual EBMT method.

1 Introduction

There is a great deal of research on the machine translation, and each of them has achieved surely advantage. There are three typical approaches: Rule-based Machine Translation (RBMT), Example-based MT (EBMT) and Statistical MT (SMT).

RBMT uses many translation rules(Amano et al., 1987): parsing, transfer, generation rules, etc. One part of these rules is described abstractly to overcome various linguistic phenomena, and another part is elaborated concretely to acquire skillful translation. Abstract rules give robustness to a system, but sometimes become a cause of lack of fluency.

EBMT is an analogical method based on human-translated examples(Nagao, 1984). Those examples are directly used as a result or are partially replaced to be matched to an input sentence. So translation tends to be more natural than in the case of RBMT. However, since the domain covered strongly depends on the example database, robustness is often inferior to that of RBMT.

SMT generates translation on the basis of statistical models derived from the analysis of bilingual corpora. It can cut development cost dramatically compared with RBMT and generate a natural translation result for a suitable domain. But, in some cases, well-developed RBMT outputs a more suitable translation and covers a larger domain.

These strengths and weaknesses of each translation method are not only inherent properties but also complementary properties. We propose a new hybrid translation method based on this complementarity. A characteristic of our proposal is that it divides the input sentence into optimum units based on its syntactic structure generated by RBMT and selects the best translation method for each segment.

It is especially effective in translating spoken language that is often breaking off fragmental speech. We think the most suitable approach for spoken language translation (SLT) is to pack such speech fragments into significant groups and translate them by switching the translator.

In the following section, using Japanese-to-English translation as a motif, we describe a detailed method. Next, we report on our evaluation experiments. Then, we present a comparison of other relevant studies and conclude the paper with a discussion of future work.

2 Hybrid Translation Method

EBMT is a powerful tool when an input sentence is long or idiomatic. But the use of an example match-

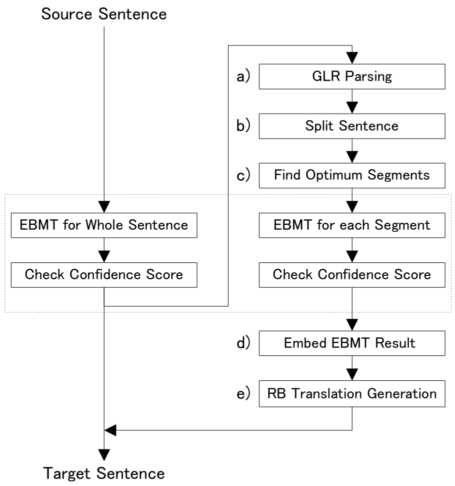

Figure 1: Process Flow of our Hybrid MT

ing such an input is less frequent. If such a long input can be translated by combination of examples, those shorter examples will be used efficiently. Furthermore, dividing an input into short units can contribute to computational efficiency of SMT.

Building on this concept, we design the hybrid SLT method. Figure 1 shows a process flow of our hybrid SLT system. The portion wrapped with a dotted line is the basic EBMT method, and the other portion is the extended RBMT method.

1. Try EBMT for a whole sentence
2. Evaluate Confidence score
   a) Parse an input sentence
   b) Split the sentence based on the syntax
   c) Find an optimum segments' combination
      1') Try EBMT for each segment
      2') Evaluate confidence scores
   d) Embed partial EBMT results
   e) Generate translation of the whole sentence

In the remainder of this section, we give a detailed explanation of each splitting step.

2.1 Parsing

We regard the sentence partitioning problem as the finding segment under the following conditions. 1) Each segment can be independently and correctly interpreted. 2) Each segment can be removed without changing a meaning of a remaining part. 3) Translation for a whole input sentence can be generated fluently, even if it is necessary to combine partial translations of each segment.

<Rewrite rules>

(1) $S \rightarrow C$
(2) $C \rightarrow SC\ C$
(3) $SC \rightarrow SC\ SC$
(4) $C \rightarrow VP$
(5) $SC \rightarrow VP\ CP$
(6) $NP \rightarrow N\ CM$
(7) $VP \rightarrow NP\ V$

<Nonterminal symbols>

S ··· *Sentence*
C ··· *Clause*
SC ··· *Subordinate Clause*
NP ··· *Noun Phrase*
CM ··· *Case Maker*
CP ··· *Conjunctive Particle*
VP ··· *Verb Phrase*
N ··· *Noun*
V ··· *Verb*

Figure 2: Part of our grammar

For Japanese, we use a clause as such a segment. A Japanese clause is a small and significant unit consisting of at most one subject and one predicate. To estimate such a clause structure, we utilize the method proposed by (Kamatani et al., 2006). They proposed an analysis method that estimates clause structures by treating input utterances as a sequence of fragmental phrases, and evaluates validities combined with dependency preferences. It allows evaluation of all candidates efficiently and choosing of the totally optimum one.

According to their analysis, even spoken language can be analyzed by using a grammar developed on the basis of the following two assumptions. 1) One utterance often consists of fragmental phrases. 2) When some fragments are unified as a clause, its internal structure is quite grammatical. By using their method, we can evaluate all combinations of segments cyclopaedically in real time.

We develop an original grammar centered on a clause structure. A part of our grammar is shown in Figure 2. Figure 3 shows a parsing example by GLR parser working with our grammar. For purposes of illustration, the packed shared forest structure [1](Tomita, 1991)is somewhat simplified and a node has an identifier with its syntactic category. For instance, the node marked (a) has a syntax category "NP" denoting "Noun Phrase". In the figure, the node (e) is shared by other nodes (f) and (g), and node (h) packs local ambiguity <h1> and <h2>.

This grammar includes some special treatments to classify a relation between segments. It is used to sensitively translate a relation between each segment. For instance, as shown in Figure 2, we handle a parenthetic expression as a dependency rela-

[1]We simply call this structure a syntax forest in this paper.

tion between (subordinate) clauses.

2.2 Sentence Splitting based on the Syntax

First, we introduce the following notations and functions to formulate our sentence splitting method.

- The parser derives a syntax forest f with a set of nodes N_f for one input sentence.
- One syntax forest can be divided into each individual syntax tree $t \in f$ and its nodes set $N_t \subseteq N_f$.
- Each node has one syntax category $c \in \mathcal{C}$.
- $Cat(n)$ gives a syntactic category of a node $n \in N_f$.
- $Prt(n)$ gives a set of nodes in a partial forest structure dominated by a node $n \in N_f$.
- $Trees(n)$ gives a set of nodes in trees including a node $n \in N_f$.

Our hybrid method enumerates two types of splitting candidates. They are "basic segment" and "pairing segment" which are defined as follows.

Basic segment candidates :

$$S_b \quad = \quad \{seg | seg = Prt(n) \;\; s.t. \;\; Cat(n) \in \mathcal{C}_s\}$$

Where $\mathcal{C}_s \subseteq \mathcal{C}$ is a set of syntactic categories predefined to elect the splitting candidate. For Japanese analysis, we use syntactic category "C" and "SC" shown in a Figure2.

Here we call this type of segment a "basic segment", and the root node of the segment a "dominator node". In the following explanation, we express a basic segment by using a notation "(n)". That means a basic segment dominated by a dominator node $n \in N_f$.

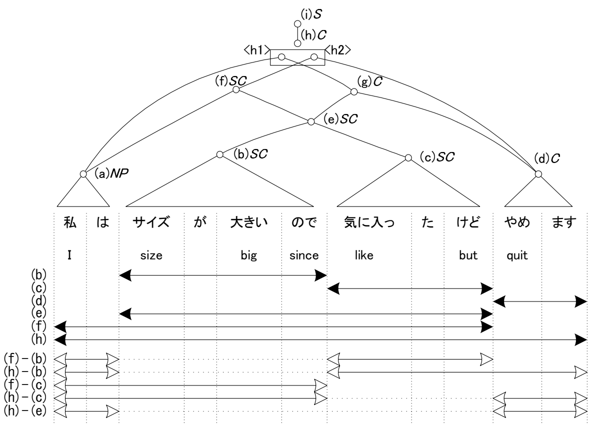

Figure 3: Splitting an input sentence

Pairing segment candidates :

$$\begin{aligned} S_p \quad = \quad & \{seg | seg = \\ & \quad Trees(n_j) \cap \{Prt(n_i) - Prt(n_j)\} \\ & \quad s.t. \;\; Prt(n_i), Prt(n_j) \in S_b \\ & \quad\quad \wedge Prt(n_j) \subset Prt(n_i)\} \end{aligned}$$

When any given two nodes are dominator nodes of a basic segment and one node has a structure dominated by the other node as its substructure, the remainder structure of the two syntactic structures will be chosen as a segment. This structural subtraction stands on a supposition that even if the meaningfully complete segment is removed from another segment, the remainder is understood correctly.

Here we call a segment of this type a "pairing segment", the root node of the remaining segment a "dominator node", and the root node of a deleted substructure an "exclusive node". In the following explanation, we express a basic segment by using a notation "(n_1, n_2)". That means a paring segment derived from a dominator node $n_1 \in N_f$ and an exclusive node $n_2 \in N_f$.
□

Figure 3 shows an example of sentence splitting. Here, we assume that the nodes marked b, c, d, e, f and h satisfy the condition as a dominator node of a basic segment. For example, the node(b) dominates a syntax structure for the partial input morphemes "サイズが大きいので*/Because the size is large.*" that can be regarded as a basic segment (b). In the figure, spans of each basic segment are indicated by black arrows.

All pairs of two dominator nodes of enumerated basic segments are checked to ascertain whether each of them satisfies a condition of a paring segment. In the example presented in Figure 3, we can consider 5 pairs that satisfy the condition for paring segments: $(f, b), (h, b), (f, c), (h, c)$ and (h, e). For example, the segmented morphemes "私はやめます*/I quit buying it.*" are found for a pair of the node h and e. In the figure, the spans of morphemes for each basic segment are indicated as white arrows. Clearly, even a discontinuous sequence of input morphemes is detected as a segment.

2.3 Choose Optimum Split

We introduce two additional functions to describe a way to find optimum splitting.

- $Mrp(seg)$ gives a set of morphemes expressed by one segment(seg).
  - If $seg \in S_b$, $Mrp(seg)$ gives a sequence of morphemes dominated by n.
  - If $seg \in S_p$, $Mrp(n_i, n_j)$ gives a relative complement of morphemes dominated by n_i and n_j.
- $Root(seg)$ gives a dominator node of a basic segment and a paring segment.

First, we classify the syntactic categories $c \in \mathcal{C}_s$. $\mathcal{C}_{sc} \subseteq \mathcal{C}_s$ includes categories given to nodes whose substructures can be translated independently. The other categories $c \in \mathcal{C}_s$ are classified into $\mathcal{C}_c$. That means $\mathcal{C}_c = \mathcal{C}_s - \mathcal{C}_{sc}$. For Japanese analysis, we use a classification $C_c = \{\text{“}C\text{”}\}, C_{sc} = \{\text{“}SC\text{”}\}$. Therefore, each segment in S_b and S_p can be classified into two types.

$$\begin{aligned} S_c &= \{seg | seg \in S_b \cup S_p \\ &\qquad s.t.\ \ Cat(Root(seg)) \in \mathcal{C}_c\} \\ S_c &= \{seg | seg \in S_b \cup S_p \\ &\qquad s.t.\ \ Cat(Root(seg)) \in \mathcal{C}_{sc}\} \end{aligned}$$

We calculate a combination of segments as the optimum split with the following two strategies. The first strategy chooses as many optimum segments dominated by a category $c \in C_{sc}$ node as possible. It can increase the chance of applying EBMT.

$$\begin{aligned} Split_{sc} &= \{A_p | A_p \subseteq S_{sc}\ \ s.t. \\ &\qquad \forall a_i \in A_p((\bigcup_{a_i, a_j \in A_p, i \neq j} a_i \cap a_j) = \phi) \\ &\qquad \wedge\ \exists N_t(\bigcup_{a_i \in A_p} Root(a_i) \subseteq N_t)\} \end{aligned}$$

A set $Split_{sc}$ represents possible segment combinations for a whole input sentence.

$$Opt_{sc} = \operatorname*{argmax}_{A_p \in Split_{sc}} \sum_{seg_i \in A_p} |seg_i|$$

The second strategy chooses optimum segments to maintain the interpretation for the whole utterance and translatability by RBMT.

$$\begin{aligned} Split_c = &\quad \{A_p | A_p \subseteq S_c\ \ s.t. \\ &\quad \forall a_i \in A_p((\bigcup_{a_i, a_j \in A_p, i \neq j} a_i \cap a_j) = \phi) \\ &\quad \wedge[\{(\bigcup_{a_k \in Opt_{sc}} a_k) \cap (\bigcup_{a_i \in A_p} a_i)\} = \phi] \\ &\quad \wedge[\exists N_t((\bigcup_{a_k \in Opt_{sc}} Root(a_k)) \\ &\qquad \cup(\bigcup_{a_i \in A_p} Root(a_i)) \subseteq N_t)] \end{aligned}$$

A set $Split_c$ represents possible segment combinations for a partial input sentence that is not covered by Opt_{sc}.

$$Opt_c \quad = \operatorname*{argmax}_{A_p \in Split_c} \sum_{seg_i \in A_p} |Mrp(seg_i)|$$

These two strategies extract just one combination of segments without evaluating a confidence score of each partial EBMT result and calculating the total score of the translation for a whole utterance. Accordingly, it does not assure that the split sequence can generate the best translation result.

Another choosing method is to consider all the pairs of EBMT results and calculate the total confidence scores. But there is trade-off between calculation cost and translation precision, and such a constitutively produced confidence score does not always assure quality. Because we can consider these segments to be briefly evaluated by syntax, it leads to the local maximum at least. For these reasons, we only use this strategy.

In the example described in Figure 3, basic segments (b) and (c), and a paring segment (h, e) are elected as an optimum combination that gives the best division of the utterance (Figure 4).

2.4 Embedding partial EBMT results

The segments composing the optimum splitting $Opt_{sc} \cup Opt_c$ are individually translated by EBMT. Then, the EBMT result with sufficient confidence score is used as a partial translation.

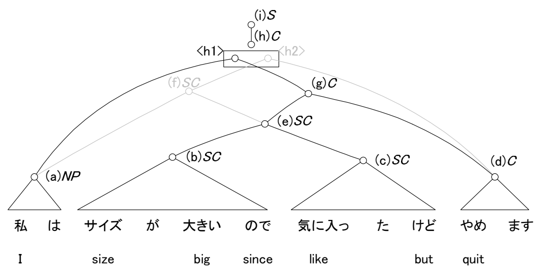

Figure 4: Optimum Splitting

We utilize an EBMT method proposed by (Wu et al., 2005). They improved quality and example coverage of a translation memory system by taking advantage of sentence-level matching, subsentential matching and pattern-based MT methods.

Their proposed method also includes input sentence splitting as a subsentential matching. But segments are estimated statistically and the whole sentence is translated by a single EBMT method. We use a translation result generated by the sentence-level matching method to embed, because we want to acquire a partial translation of as high quality as possible and evaluate individual performances.

We also used the confidence score of EBMT by using sentence similarity defined by them and tri-gram language model($F(T)$) of a target language.

$$F(T) \quad = \quad \left(\prod_{i=1 \ldots |T|} p(t_i | t_{i-2}, t_{i-1}) \right)^{\frac{1}{|T|}} \qquad (1)$$

where the T is a target sentence and the t_i is a morpheme in it.

$$Score \quad = \quad \beta_1 \cdot Sim(X,Y) + \beta_2 \cdot F(T) \qquad (2)$$

where Sim(X,Y) is similarity between an input utterance X and an source sentence Y of an example pair, β_1 and β_2 are weights which are experimentally given, and $0 \leq Score \leq 1.0$. The detailed definition of Sim(X,Y) is given in their paper.

EBMT results with sufficient confidence scores will be elected and embedded as a partial translation of the whole utterance. Two embedding styles are defined and switched by the segment kind.

Basic segment :

1. Delete the syntactic structure that has the dominator node as a root node and depends on only the morphemes in this segment.
2. Add a special terminal node that denotes the EBMT translation result. The new terminal node will be unified to a parent node of the dominator node. If the dominator node already has other nodes, the order of these nodes follows the order of input morphemes.

Pairing segment :

1. Move the exclusive node to the dominator node as a new parent node. If the dominator node already has other nodes, the order of these nodes follows the order of input morphemes.

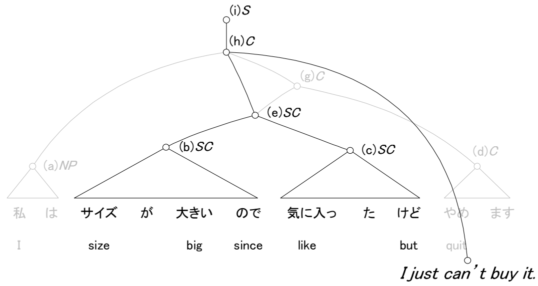

Figure 5: Embedding EBMT Result to the Tree(1)

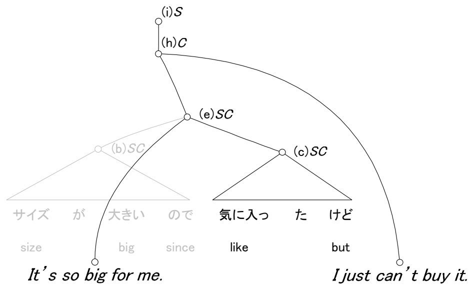

Figure 6: Embedding EBMT Result to the Tree(2)

2. Delete the syntactic structure that has the dominator node as a root node and depends on only the morphemes in this segment.
3. Add a special terminal node that denotes the EBMT result. The new terminal node will be unified to a parent node of the dominator node. If the dominator node already has other nodes, the order of these nodes follows the order of morphemes in the input sentence.

We define these processes so as to regard them as one of the transfers in the RBMT system. So, even after embedding, the syntax tree keeps translatability by RBMT.

Now, we assume that a paring segment (h, e) gets the EBMT result "*I just can't by it*" with a sufficient confidence score, and a basic segment (b) also acquires the EBMT result "*It's so big for me*". Figure 5 and Figure 6 show examples of the embedding process. The order in which segments are embedded is unrestricted.

2.5 Integration and Generation

Even after embedding an example to a syntax structure, it keeps its characteristics as a syntax tree. So, it is only necessary to develop new rules that handle such a partially translated structure. It is quite a natural task for RBMT.

3 Experiment

3.1 Experimental Settings

We evaluate three systems: individual EBMT(Wu et al., 2005) and RBMT, and hybrid SLT system. The performance of EBMT and RBMT can be a baseline for our proposed hybrid method. Here, we use only the sentence-level matched EBMT result with confidence $Score \geq 0.6$ in equation (2) for the hybrid SLT system. The result of EBMT is just to choose a translation with the highest confidence score, regardless of its absolute value. On the other hand, we allow subsentential matched translation result as a result of individual EBMT. Introducing a subsentential matching as a result makes it possible to compare the splitting method between the statistical method and ours.

Because we don't have enough bilingual travel domain corpora that we can use freely, we developed 123,819 Japanese-English translation pairs as an example base of EBMT. We prepare two types of test set aside from examples. One is a set of balanced travel domain sentences, and the other is a set of relatively long sentences. As mentioned in section 2, since our hybrid method divides an input sentence into several parts, it works only for the sentences that have a certain level of length.

For evaluation, we use NIST score(Doddington, 2002) and BLEU score(Papineni et al., 2002). Each sentence in the test set has one translation reference.

3.2 Evaluation Result

The evaluation result for the balanced test set is shown in Table 2. Since the DB for the same travel domain is used, both NIST and BLEU score higher for EBMT than for RBMT. Moreover, the

Table 1: Test set specification

	Number of Sentences	Japanese Source	English Reference
Balanced	1000	4.9372	0.2403
Long Sentence	200	4.4644	0.1885

Table 2: Evaluation for Japanese-to-English Translation of the Balanced Sentences Set

System	NIST	BLEU
EBMT	4.9372	0.2403
RBMT	4.4644	0.1885
Hybrid MT	5.0474	0.2511

Table 3: Evaluation for Japanese-to-English Translation of the Relatively Long Sentences Set

System	NIST	BLEU
EBMT	3.8798	0.1351
RBMT	3.8191	0.1252
Hybrid MT	4.1127	0.1597

Hybrid SLT method scored higher than each individual translation method. This result proves the effect of our method. The result of hybrid MT consists of 622 sentences by individual EBMT, 363 sentences by individual RBMT, and 15 sentences by composing partial EBMT and RBMT results.

The evaluation result for long sentences is shown in Table 3. The hybrid method outputs the 44 translations by EBMT, the 125 translations by RBMT, and the 31 translations composed by both EBMT and RBMT.

As an input sentence gets longer, it becomes harder to find an example translation matching it, and more complex to parse it. Both EBMT and RBMT get lower scores than the balanced test set. The score of the hybrid SLT is also reduced, but it is still higher than that of the individual MT method. This result highlights the advantage of our method.

Table 4 shows some translation results for the second test set, which are translated by composing the partial results of EBMT and RBMT. Three translation results are shown in the table. The row labeled "Ref." means the translation reference that was translated manually and used for the evaluation. And there are three translation results generated by each method for each source sentence (Src.).

4 Related Work

In the spoken language domain, research is often focused on determining the end of an utterance and subsentence punctuation predication, such as (Matusov et al., 2007). Such approaches are useful for

Table 4: Sample Japanese-to-English translations

1)Src.		ご迷惑でなければ、座席を少し倒していいですか？
	Ref.	If it's not much trouble, can I put my seat back a little?
	EBMT	Annoying, may I lower my seat a little?
	RBMT	As long as it is not troublesome, may I push down a little seat?
	Hybrid	If it's not too much trouble. I may push down a little seat.
2)Src.		隣の部屋の人が騒がしいので、部屋を替えて下さい。
	Ref.	Please change my room because the people next door are noisy.
	EBMT	Many people next door room noisy, to give room.
	RBMT	The person of the next room is noisy. Please change the room.
	Hybrid	People room next door is so noisy. Would you me a different room?
3)Src.		食事の時間になったら起こしてもらえますか？
	Ref.	Would you wake me up at meal time, please?
	EBMT	Trains wake be dinner at the same time?
	RBMT	Would you start, if the time of a meal comes?
	Hybrid	If the time of a meal comes Excuse me, let me wake.

cutting out a segment to parse, but they are deterministic and do not supply preference of a relation in each segment.

From the viewpoint of hybrid machine translation, (Akiba et al., 2006) and (Nakamura , 2006) proposed the multi-engine translation method that evaluates target sentences individually generated by each engine, and chooses the best one. However its evaluation target is a whole sentence. We think it can be used at our estimating step of optimum split.

(Bond et al., 2003) introduced a hybrid rule and example-based method for MT. Their system translates an input sentence using the most typical translation example that is similar to the input. Here, an example pair is chosen that both matches the input sentence and has a translation similar to other examples. However, the selection method still uses a whole-sentence translation as a unit.

(Doi et al., 2004) proposed a sentence splitting method that generates splitting candidates based on an N-gram model and selects the best one by calculating sentence similarity between the part and an example in the database for EBMT. The splitting model is given as a probability of insertion of segment start and end.

(Lavie et al., 1996) and (Langley et al., 2002) defined semantic dialog units that roughly correspond to a speech act and can be translated independently. The dialog units are estimated by acoustic cues and a pre-learned statistical model. Consequently, our method keeps totality between each segment based on a syntax given by the RBMT method. It allows finding discontinuous segmentation and translates a relation between segments appropriately.

(Furuse et al., 1998) also proposed input-splitting method for translating spoken language. It can exclude ill-formed expressions from a raw input. It aimed to find the best splitting to be translated efficiently by single translation method.

(Mellebeek et al., 2006) and (Rosti et al., 2007) combine translation results from multi-engine MT and find an optimum combination as a final translation result. But each chunk of translation is given for a continuous sequence in an input sentence. So, a dependency between non-continuous morphemes is sometimes missed in a final translation result.

5 Future Work

Our hybrid SLT method utilizes a Japanese clause as a unit to switch translation methods. A Japanese clause is small enough to understand a meaning, but it seems a rather big structure to increase a chance of applying EBMT. In particular, some short utterances do not fully benefit from our method, because a simple sentence usually consists of just one clause.

For the next step, we are studying use of a phrase. But it is more difficult to embed a phrase translation than a clause translation, since a phrase exhibits diverse behavior and other dependent are usually needed in order to determine translation. Among such segments that are somewhat awkward as units, a noun phrase is comparatively easy to handle.

While we are expanding coverage of the hybrid method, we are also examining a method of calculating the confidence score for each EBMT result and the final translation. For the first step, we have to evaluate using a monolingual language model translation for the whole utterance and check whether it gets a corresponding bless in calculation cost.

6 Conclusion

In this paper, we propose a hybrid spoken language translation (SLT) method which divides input sentence into some parts and translates them by switching RBMT and EBMT for each part. A characteristic of our method is that it splits an utterance based on its syntactic structure.

We also report fundamental experimental results. In the evaluation for balanced test set, our method achieves a 13.0% advantage in NIST score for the individual RBMT method and a 2.2% advantage for the baseline EBMT method. For long sentences, our hybrid method achieves a 6.0% advantage for the conventional EBMT and RBMT systems.

References

Yasuhiro Akiba, Kenji Imamura, Eiichiro Sumita, Hiromi Nakaiwa, Seiichi Yamamoto, Hiroshi G. Okuno. 2006. Using Multiple Edit Distances to Automatically Grade Outputs from Machine Translation Systems. *The IEEE Transactions on Speech and Audio Processing*. Vol.14, Issue.2, pages 393–402.

Shin-ya Amano, Hideki Hirakawa and Yoshinao Tsutsumi. 1987. TAURAS: the Toshiba machine translation system. In *Proc. of MT Summit*, pages 15–23.

Francis Bond and Satoshi Shirai. 2003. A Hybrid Rule and Example based Method for Machine Translation. In Michael Carl and Andy Way, eds., *Recent Advances in Example-based Machine Translation*,Kluwer Academic Publishers. Chapter 7, pages 211–224.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer and Paul S. Roossin, 1990. A statistical approach to machine translation. *Computational Linguistics*, Volume 16, Issue 2, pages 79–85.

Takao Doi and Eiichiro Sumita. 2004. Splitting Input Sentence for Machine Translation Using Language Model with Sentence Similarity. In *Proc. of International Committee on Computational Linguistics*. pages 113–120.

Osamu Furuse, Setsuo Yamada and Kazuhide Yamamoto. 1998. Splitting Long or Ill-formed Input for Robust Spoken-language Translation. In *Proc. of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*. pages 421–427.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *Proc. of the Second International Conference on Human Language Technology*. pages 138–145.

Satoshi Kamatani, Kentaro Furihata and Tetsuro Chino. 2006. Forest Driven Dependency Analysis Enhanced by Japanese Clause Structure Estimation. In *Proc. of The 20th Pacific Asia Conference on Language, Information and Computation*. pages 265–273.

Chad Langley, Alon Lavie, Lori Levin, Dorcas Wallace, Donna Gates and Key Peterson. 2002. Spoken Language Parsing Using Phrase-Level Grammars and Trainable Classifiers. In *Proc. of the Workshop on Speech-to-speech Translation: Algorithms and Systems*. pages 15–22.

Alon Lavie, Donna Gates, Noah Coccaro and Lori S. Levin. 1996. Input Segmentation of Spontaneous Speech in JANUS: A Speech-to-speech Translation System. In *Proc. of the Workshop on Dialogue Processing in Spoken Language Systems*. pages 86–99.

Evgeny Matusov, Dustin Hillard, Mathew Magimai-Doss, Dilek Hakkani-Tur, Mari Ostendorf and Hermann Ney. 2007. Improving Speech Translation with Automatic Boundary Prediction. In *Proc. of Interspeech*. pages 2449–2452.

Bart Mellebeek, Karolina Owczarzak, Josef Van Genabith and Andy Way. 2006. Multi-Engine Machine Translation by Recursive Sentence Decomposition. In *Proc. of the 7th Conference of the AMTA*. pages 110–118.

Nagao Makoto. 1984. A freamwork of mechanical translation between Japanese and English by analogy principle. In Elithorn & Banerji, eds., *Artificial and Human Inteligence*, Elsevier. pages 179–180.

Satoshi Nakamura, Konstantin Markov, Hiromi Nakaiwa, Genichiro Kikui, Hisashi Kawai, Takatoshi Jitsuhiro, Jin-Song Zhang, Hirofumi Yamamoto, Eiichiro Sumita, and Seiichi Yamamoto. 2006. The ATR Multilingual Speech-to-Speech Translation System. In *IEEE Trans. on Audio, Speech and Language Processing*. Vol. 14, No. 2, pp. 365–376.

Kishore Papineni, Salim Roukos, Todd Ward and Weiing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics* pages 311–318.

Antti-Veikko I. Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz and Bonnie J. Dorr. 2007. Combining Outputs from Multiple Machine Translation Systems. In *Proc. of NAACL HLT 2007* pages 228–235.

Masaru Tomita. 1991. Generalized LR Parsing. Kluwer Academic Publishers, Norwel, Massachusetts.

Wu Hua, Wang Haifeng, Liu Zhangyi and Tang Kai. 2005. Improving Translation Memory with Word Alignment Information. In *Proc. of MT Summit X*, pages 364–371.