Probabilistic Model for Example-based Machine Translation

Eiji Aramaki[1], Sadao Kurohashi[2], Hideki Kashioka[3] and Naoto Kato[4]

[1] Planning, Information and Management Dept. University of Tokyo Hospital
[2] Graduate School of Information Science and Tech. University of Tokyo
[3] ATR Spoken Language Translation Research Laboratories
[4] Science and Technical Research Laboratories of NHK
[1]aramaki@hcc.h.u-tokyo.ac.jp, [2]kuro@kc.t.u-tokyo.ac.jp
[3]hideki.kashioka@atr.jp, [4]naoto.kato@atr.jp

Abstract

Example-based machine translation (EBMT) systems, so far, rely on heuristic measures in retrieving translation examples. Such a heuristic measure costs time to adjust, and might make its algorithm unclear. This paper presents a probabilistic model for EBMT. Under the proposed model, the system searches the translation example combination which has the highest probability. The proposed model clearly formalizes EBMT process. In addition, the model can naturally incorporate the context similarity of translation examples. The experimental results demonstrate that the proposed model has a slightly better translation quality than state-of-the-art EBMT systems.

1 Introduction

Nowadays, much attention has been given to data-driven (or corpus-based) machine translation, such as example-based machine translation or EBMT(Nagao, 1984) and statistical machine translation or SMT (Brown et al., 1993). This paper focuses on EBMT approach.

The idea of EBMT is that translation examples similar to a part of an input sentence are retrieved and combined to produce a translation. EBMT basically prefers larger translation examples, because the larger the translation is, the wider context is taken into account. So, most EBMT systems retrieve large examples as possible as they can, and the retrieving is based on some heuristic criterion/measures which prefer larger examples.

On the other hand, SMT approach basically breaks down translation examples into small word/phrases in order to calculate translation probability reliably. Of course, recent SMT studies incorporate larger phrase unit, for example, Och(Och et al., 1999) used alignment template to handle phrase chunks. However, SMT translation unit is smaller than EBMT, which has no limitation in its unit size.

Simply speaking, EBMT and SMT have two differences:

1. EBMT pays more attention to the size; SMT to the frequency.
2. EBMT relies on heuristic criterion/measures; SMT is statistically formalized.

For the formalization of EBMT, this paper proposes a probabilistic translation model, which deals not only with the example size but with the context similarity.

In the experiments, the proposed model has a slightly better translation quality than state-of-the-art EBMT systems. The results demonstrated the validity of the proposed model.

This paper is organized as follows. The next section presents the basic idea of our approach. Section 3 describes the algorithm. Then, Section 4 reports experimental results, Section 5 reports related works, and Section 6 presents our conclusions.

Though the proposed method does not depend on language pairs and translation directions, this paper describes Japanese-English translation.

2 Basic Idea

The basic principle of EBMT is to generate a translation using the larger example. To do so, a translation consisting the larger examples should have the higher probability. This section describes the basic idea of the proposed model.

First of all, let us consider that an input sentence can be decomposed in N ways,

$$D = \{d_1, ..., d_N\}. \qquad (1)$$

where d_i stands for a decomposed pattern of an input sentence, D is a set of each d_i.

Next, suppose that d_i decomposes an input tree into M_i sub-trees as follows:

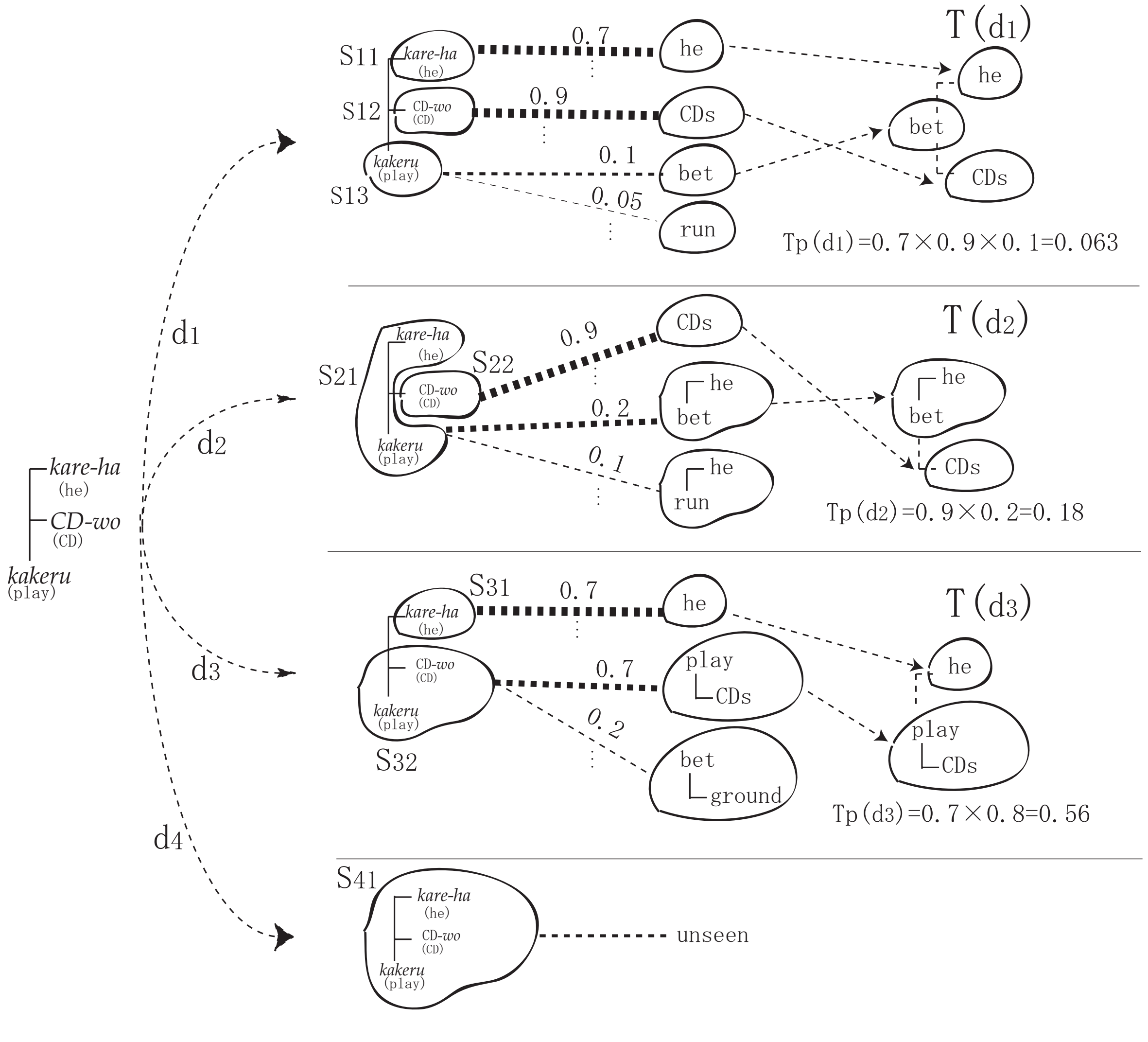

Figure 1: Example of Translation Flow.

$$d_i = \{s_{i1}, s_{i2}, ..., s_{iM_i}\}, \tag{2}$$

where s_{ij} is a sub-tree of an input sentence.

For example, an input sentence in the left of Figure 1 could be decomposed in four ways as shown in $d_1, ..., d_4$, and d_1 decomposes an input into three sub-trees s_{11}, s_{12} and s_{13}; d_2 decomposes an input into two sub-trees s_{21} and s_{22} and so on.

Then, for each sub-tree s_{ij}, its target expression t_{ij} is selected based on translation probability $P(t_{ij} \mid s_{ij})$ (whose definition is described in the following subsection), and we calculate a target sentence probability $T_p(d_i)$, which is the product of the sub-tree probabilities as follows:

$$T_p(d_i) = \prod_{s_{ij} \in d_i} P(t_{ij} \mid s_{ij}). \tag{3}$$

We regard $t_{i1}, ..., t_{iMi}$ as a translation of d_i, and notate it as $T(d_i)$.

Finally, the decomposition d_m which has the highest translation probability $T_p(d_m)$ is searched as follows:

$$d_m = \arg\max_{d_i \in D} T_p(d_i). \tag{4}$$

We regard $T(d_m)$ as the final translation. In short, Formula 4 searches the plausible translation unit, and Formula 3 searches the plausible target expression for each unit.

More importantly, the proposed framework naturally prefers a translation consisting of large sub-trees (examples), because the larger examples tend to have little ambiguities in target translations, leading to the high example probability. So, T_p which is product of them naturally gets the high translation probability. This preference has compatibility with the original EBMT idea.

For example, as shown in Figure 1, a Japanese word "*kakeru*" is very ambiguous, and its target translation could be various words, i.e., "bet", "run", "play" and so on.

In d_1, an input is decomposed into small parts. In this case, "*kakeru*" alone (s_{13}) has a small example probability, leading to a small translation probability $T_p(d_1)$.

In d_3, the example probability are estimated for a large unit s_{32} ("*kakeru*"+"*CD-wo*"). In this case, s_{32} has stable target translation, leading to both the high example probability and the high translation probability $T_p(d_3)$.

Note that retrieving a very large sub-tree as shown in d_4 might fail, because such a large example is unseen in training corpora.

2.1 Parameter estimation

This subsection describes the method to estimate parameters.

First, consider an example consisting an English sub-tree (t) and a Japanese sub-tree (s). The estimation of the example's probability or $P(t \mid s)$ is basically based on the direct counting of each sub-tree pair's appearance on the aligned corpus (the method to build the aligned corpus is described in Section 3) as follows:

$$P(t \mid s) = \frac{count(t, s)}{count(*, s)}, \qquad (5)$$

where $count(t, s)$ is the occurrence of sub-tree pairs (t, s) in the aligned corpus, $count(*, s)$ is the occurrence of Japanese sub-tree (s) in the aligned corpus.

Note that in counting the occurrences, we use context information as mentioned in the following subsection.

2.1.1 Example filtering based on context similarity

One of the most important clues to select translation examples is the size, and it is already realized in the previous section. However, in addition to the size, the similarity of context is also an important clue. The proposed model can naturally incorporate the context similarity by the following method.

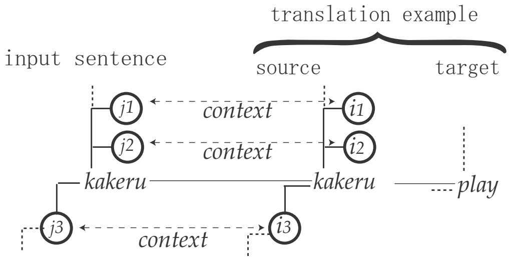

Figure 2: Definition of Context.

translation examples		context_sim
target parts	source parts (context)	
play	*kekeru* (CDs)	0.8
play	*kakeru* (music tapes)	0.8
put	*kekeru* (Mini Discs)	0.8
⋮	⋮	⋮
set	*kakeru* (alarm)	0.7
bet	*kakeru* (the races)	0.6
⋮	⋮	⋮
bet	*kakeru* (money)	0.3

* In fact, examples are stored in the tree forms, but for simplicity, this figure shows them without their structures.

Figure 3: Translation Examples including "*kakeru*" and their Context (shown in blankets).

Before explaining the methods, first, we define the context. In this paper, we regard the context as both surrounding phrases which directly connect to translation examples and their corresponding input phrases.

For example, as illustrated in Figure 2, a source part of translation example has its surrounding phrases, i_1, i_2 and i_3, which correspond to input phrases j_1, j_2 and j_3. In this example, we regard them, $i_{1..3}$ and $j_{1..3}$, as the context.

Then, we define the context similarity as follows:

$$context\_sim = \sum_{i \in N} sim(i, j), \qquad (6)$$

where i is a surrounding phrase of translation example, j is i's corresponding input phrase, N is a set of i. $sim(i, j)$ is the similarity, which is calculated using a source language thesaurus[1]

[1] In the experiments in Section 4, we used NTT the-

as follows:

$$sim(i,j) = \frac{2d_c}{d_i + d_j}, \quad (7)$$

where d_i and d_j are the depths of i and j in the thesaurus, and d_c is the depth of their lowest (most specific) common node. If i or j is compound (multi) words, its head word is consulted.

The point of the methods is to filter out low-context-similarity examples, which might lead to improper translations. To do so, when the system calculates an example's translation probability in Formula 5, the system counts only examples which have same or higher *context_sim* than itself. We call this operation a filtering based on *context_sim*. By using this filtering, the probability of an example with the high *context_sim* is calculated from only examples which has also high *context_sim*. This operation basically reduces the ambiguity of target expressions.

For example, consider a translation of an unseen input phrase "*record-wo kakeru*" (which means "play records"). Here, the entire phrase does not appear, but words in it appear, and are stored as translation examples as shown in Figure 3. When the *context_sim* between an input phrase"records" and an example phrase "CDs" is 0.8, the translation probability is calculated by using translation examples whose *context_sim* are equal or over 0.8 (shown in the dotted box in Figure 3). In this case, the number of translation example becomes only three, but its target translation becomes more stable, i.e., $P(play \mid kakeru) = \frac{2}{3}$, $P(set \mid kakeru) = \frac{1}{3}$.

Thus, a translation example with the similar context can naturally get a higher translation probability.

3 Algorithm

The Algorithm of proposed method consists of the following two modules: (1) an alignment module, which builds translation example from corpus, and (2) a translation module, which generates a translation.

Alignment module

Step 1: Conversion into phrasal dependency structures
First, sentence pairs are parsed by the Japanese parser KNP (Kurohashi and Nagao, 1994) and the English nl-parser (Charniak, 2000). The Japanese parser outputs a dependency structure, and we use it as is. The English parser outputs a phrase structure. Then, it is converted into a dependency structure by rules which decide on a head word in a phrase. A Japanese phrase unit consists of sequential content words and their following function words. An English phrase unit is a base-NP or a base-VP.

Step 2: Alignment based on translation dictionary
Then, correspondences are estimated by using translation dictionaries (Aramaki et al., 2001). We used four dictionaries: EDR, EDICT, ENAMDICT, and EIJIRO. These dictionaries have about two million entries in total.

Step 3: Building Translation Example Database
The system generate possible combinations of correspondences from aligned sentence pairs as shown in Figure 4. We regard these combinations of correspondences as translation examples, and store them in the database. In this operation, the system stores also their surrounding phrases, which are used for calculating *context_sim*.

Translation module

Step 1: Input sentence analysis
First, an input sentence is analyzed by the Japanese parser KNP(Kurohashi and Nagao, 1994), and the system gets its dependency structure.

Step 2: Select translation examples
The system decomposes the input tree into possible sub-tree combination as shown in Figure 1 left. Then, the system counts the occurrences of translation examples, and calculates their translation probability as mentioned in the previous section. We regard translation probability as the product of each translation example's probability (Formula 3), and the examples which have the highest translation probability are adopted.

Note that in the case that there are no translation examples, the system consults a

saurus(Ikehara et al., 1997).

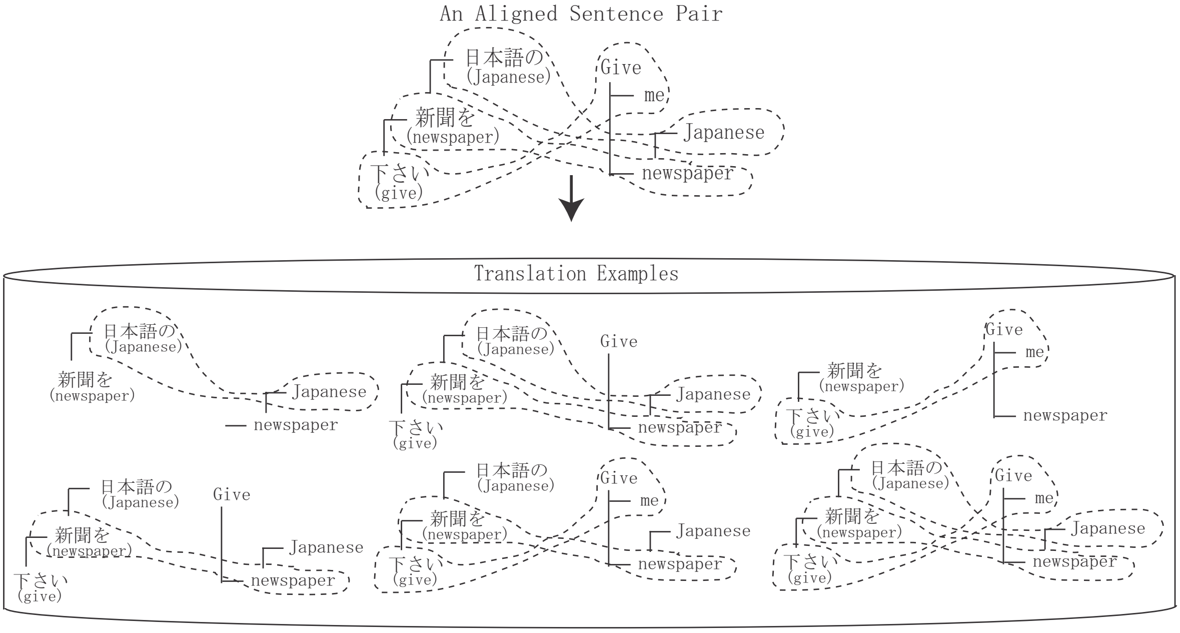

Figure 4: An Aligned Sentence Pair and Translation Examples build from it.

translation dictionary, and gets target expressions.

Step 3: Output sentence generation
Selected translation examples are combined into the output dependency structure. In this operation, the dependency relations are decided by the following two rules.

1. The relation in a translation example is preserved. For example, suppose two translation examples, TE1 and TE2, as shown in Figure 5. The target part of TE1 consists of two phrases, t_1 and t_3, and their dependency relation (shown in a bold line) is preserved in the output structure.

2. The relation between translation examples is equal to the relation between their corresponding input phrases. For example, TE2 has a corresponding input phrase i_2, and it has a relation to i_3 in the input structure. In this case, TE2 has a relation to an example phrase $t_($3), which corresponds to i_3(as shown in a dotted line).

Finally, the output word-order is decided based on the n-gram language model ($n = 3$).

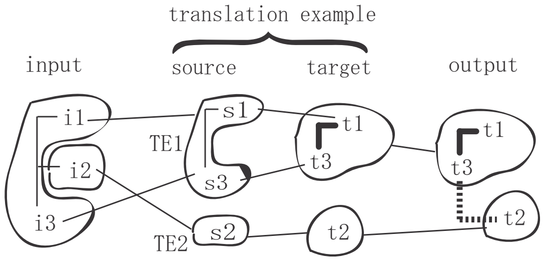

Figure 5: Output Sentence Generation.

4 Experiments

4.1 Experimental setting

For evaluation, we used corpora (training-set and test-set) which are provided in the IWSLT04(Akiba et al., 2004). The training-set consists of 20K English-Japanese sentence pairs in a travel conversation domain.

We built translation examples from the training-set by using the proposed alignment method mentioned in Section 3.

The test-set consists of 500 Japanese sentences and their English references (500 × 16). The experiments are conducted using the following five systems:

PROPOSED: The system which selects translation examples based on the proposed method .

BASIC: The system (Aramaki and Kurohashi, 2004) which selects translation examples based on the heuristic criterion. This system submitted translation evaluation workshop on IWSLT04(Akiba et al., 2004), and showed its basic feasibility. Note that BASIC uses the same alignment result as PROPOSED.

BASELINE: EBMT baseline, which searches the most similar translation examples by using a character-based DP matching method, and outputs its target parts as is.

C1, C2: Commercial machine translation systems under rule base approach.

4.2 Evaluation

Evaluation is conducted based on the following conditions and by using the five evaluation metrics in Table 1.

(1) case insensitive (lower case only)
(2) no punctuation marks (.,?!")
(3) no hyphen
(4) spelling out numerals

4.3 Results

The result is shown in Table 2. Because PROPOSED accuracy is slightly higher than BASIC, the result demonstrates validity of the proposed translation model.

4.4 Contribution of context similarity

We investigated the contribution of context similarity to the translation performance. This is conducted by performance comparison between the PROPOSED system and WITHOUT_SIM, which does not use the thesaurus (Table 2).

As shown in Table 2, PROPOSED improves the NIST score, and deteriorates the BLEU. Because the NIST is more sensible of word selection, we can say that the context similarity improves the word selection performance.

On the other hand, WITHOUT_SIM method uses the most frequent expression, and it might have some advantages in the BLEU.

4.5 Error analysis

For more concrete analysis, we randomly selected 100 PROPOSED translations, and checked them by hand. The hand check determined that 49 outputs are correct and the other 51 outputs are incorrect. Their errors are classified in Table 3.

Table 1: Evaluation Metrics.

BLEU	The geometric mean of n-gram precision by the system output with respect to the reference translations(Papineni et al., 2002).
NIST	A variant of BLEU using the arithmetic mean of weighted n-gram precision values(Doddington, 2002).
WER	word error rate; The edit distance between the system output and the closest reference translation(Niessen et al., 2000).
PER	Position-independent WER; A variant of mWER which disregards word ordering(Och et al., 2001).
GTM	general text matcher; Harmonic mean of precision and recall measures for maximum matchings of aligned words in a bitext grid.(Turian et al., 2003)

* Large scores are better in BLEU, NIST and GTM. Small scores are better in WER and PER.

DATA-SPARSENESS: DATA-SPARSENESS is the error caused by lack of translation examples. In such a case, the proposed method sometimes generates wrong translations by using a translation dictionary.

ZERO-PRONOUN: ZERO-PRONOUN is the error caused by Japanese zero pronouns. This case is re-classified in two types: (1) an input sentence includes a zero pronoun, and (2) a source (Japanese) part in translation example includes zero pronouns. In both cases, pronouns might drop out in translations.

ALIGNMENT-ERR: ALIGNMENT-ERR is the error caused by incorrect alignment results.

WORD-ORDER: WORD-ORDER refers to the case where the word order is ungrammatical.

SELECTION-ERR: SELECTION-ERR is the error caused by unsuitable translation examples.

Table 2: Experimental Results.

	bleu	nist	wer	per	gtm
PROPOSED	0.41	8.04	0.52	0.44	0.67
BASIC	0.39	7.92	0.52	0.44	0.67
BASELINE	0.31	6.65	0.62	0.54	0.59
C1	0.13	5.47	0.75	0.60	0.56
C2	0.27	7.31	0.54	0.47	0.65
WITHOUT_SIM	0.42	7.67	0.49	0.42	0.68

* WITHOUT_SIM is mentioned in Section 4.4.

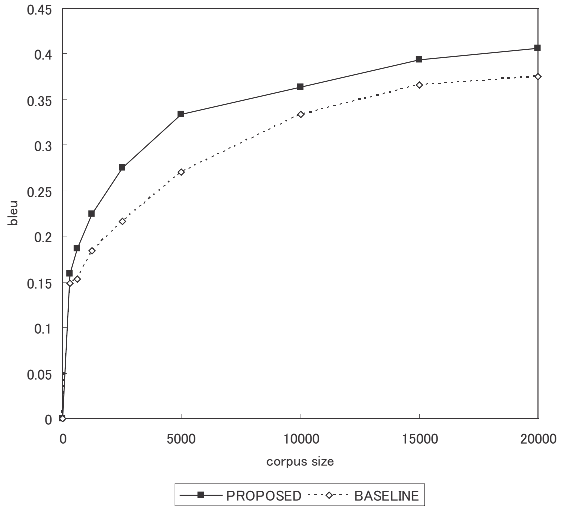

Figure 6: Corpus Size and Performance (BLEU).

- **OTHERS**: OTHERS is a case that multiple errors occur ,and we could not classify it into the above error types.

As shown in Table 3, DATA-SPARSENESS is the most outstanding problem. Therefore, we can believe that the system will achieve a higher performance if we obtain more corpora.

4.6 Corpus size and accuracy

Finally, we investigated the relation between the corpus size (the number of training sentence pairs) and its performance (BLEU) using two systems (PROPOSED and BASELINE) (Figure 6).

As shown in the figure, the difference between PROPOSED and BASELINE is larger in the small corpus size condition ($x \simeq 5,000$). This means that PROPOSED is more robust with respect to lack of translation examples than BASELINE.

More importantly, the score is not saturated at the max point ($x = 20,000$). It leads to the fact that, as mentioned before, the system will achieve a higher performance with larger corpora.

Table 3: Error Analysis.

21	DATA-SPARSENESS
6	ZERO-PRONOUN
4	ALIGNMENT-ERR
3	WORD-ORDER
3	SELECTION-ERR
12	OTHERS

5 Related Work

To our knowledge, there has been no work realizing EBMT based on the translation probability, and previous EBMT systems handle their translation examples using heuristic measures/criterion.

For instance, MSR-MT (Richardson et al., 2001) retrieves translation examples by using only the example size.

HPAT(Imamura, 2002) and TDMT(Furuse and Iida, 1994) are EBMT systems based on size and context similarity. UTOKYO-MT(Aramaki et al., 2003) used alignment confidence in addition to these metrics. Such a combination of multiple metrics leads to a problem of how to estimate the weight of each metric.

6 Conclusion

In order to formalize EBMT, this paper proposed the probabilistic translation model, in which the system searches the translation exmaple combination which the highest translation probability. Because the proposed model prefers larger and high context similarity translation exmaple, it has compatibility with the original EBMT idea.

In the experiments, the proposed model has a slightly better translation quality than state-of-the-art EBMT systems. The results demonstrated the validity of the proposed model.

However, we believe the main contribution of this paper is to provide a clear formalization to EBMT, which enables more precise comparison with SMT and possibly leads further improvement of EBMT quality.

References

Yasuhiro Akiba, Marcello Federico, Noriko Kando, Hiromi Nakaiwa, Michael Paul, and

Jun'ichi Tsujii. 2004. Overview of the IWSLT04 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 1–12.

Eiji Aramaki and Sadao Kurohashi. 2004. Example-based machine translation using structual translation examples. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 91–94.

Eiji Aramaki, Sadao Kurohashi, Satoshi Sato, and Hideo Watanabe. 2001. Finding translation correspondences from parallel parsed corpus for example-based translation. In *Proceedings of MT Summit VIII*, pages 27–32.

Eiji Aramaki, Sadao Kurohashi, Hideki Kashioka, and Hideki Tanaka. 2003. Word selection for ebmt based on monolingual similarity and translation confidence. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 57–64.

Peter F. Brown, Stephen A. Della Pietra, Vi cent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *In Proceedings of NAACL 2000*, pages 132–139.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of HLT*, pages 257–258.

Osamu Furuse and Hitoshi Iida. 1994. Constituent boundary parsing for example-based machine translation. In *Proceedings of the 15th COLING*, pages 105–111.

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentarou Ogura, and Yoshifumi Oyama Yoshihiko Hayashi, editors. 1997. *Japanese Lexicon.* Iwanami Publishing.

Kenji Imamura. 2002. Application of translation knowledgeacquired by hierarchical phrase alignment for pattern-based mt. In *Proceedings of TMI-2002*, pages 74–84.

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

Makoto Nagao. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In *Elithorn, A. and Banerji, R. (eds.): Artificial and Human Intelligence*, pages 173–180.

S. Niessen, G. Leusch F. J. Och, and H. Ney. 2000. An evaluation tool for machine translation: Fast evaluation for machine translation research. In *Proceedings of the Second Int. Conf. on Language Resources and Evaluation (LREC)*, pages 39–45.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proceedings of the Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP)*, pages 20–28.

Franz Josef Och, Nicola Uerng, and Hermann Ney. 2001. An effcient a* search algorithm for statistical machine translation. In *Proceedings of ACL 2001 Workshop on Data-Driven Machine Translation*, pages 55–62.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, pages 311–318.

Stephen D. Richardson, William B. Dolan, Arul Menezes, and Monica Corston-Oliver. 2001. Overcoming the customization bottleneck using example-based mt. In *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*, pages 9–16.

J. P. Turian, L. Shen, and I. D. Melamed. 2003. Evaluation of machine translation and its evaluation. In *Proceedings of MT Summmit IX*, pages 386–393.