Example-based Machine Translation Based on TSC and Statistical Generation

LIU Zhanyi, WANG Haifeng, WU Hua
Toshiba (China) Research and Development Center
5/F., Tower W2, Oriental Plaza
No.1, East Chang An Ave., Dong Cheng District
Beijing, 100738, China
{liuzhanyi, wanghaifeng, wuhua}@rdc.toshiba.com.cn

Abstract

This paper proposes a novel Example-Based Machine Translation (EBMT) method based on Tree String Correspondence (TSC) and statistical generation. In this method, the translation examples are represented as TSC, which consists of three parts: a parse tree in the source language, a string in the target language, and the correspondences between the leaf nodes of the source language tree and the substrings of the target language string. During the translation, the input sentence is first parsed into a tree. Then the TSC forest is searched out if it is best matched with the parse tree. The translation is generated by using a statistical generation model to combine the target language strings in the TSCs. The generation model consists of three parts: the semantic similarity between words, the word translation probability, and the target language model. Based on the above method, we build an English-to-Chinese Machine Translation (ECMT) system. Experimental results indicate that the performance of our system is comparable with that of the state-of-the-art commercial ECMT systems.

1 Introduction

This paper proposes a novel Example-Based Machine Translation (EBMT) method based on Tree String Correspondence (TSC) and statistical generation. In this method, the translation examples are represented as TSC. The translation consists of three steps. The input sentence is first parsed into a tree. Then the TSC forest is searched out if it is best matched with the input tree. Lastly, the translation is generated using a statistical generation model to combine the target language strings in the TSCs.

Many EBMT systems use annotated tree structures as translation examples (Watanabe, 1992; Poutsma, 2000; Al-Adhaileh et al., 2002; Way, 2003; Aramaki and Kurohashi, 2004). In these systems, it is necessary to parse both the source language sentences and the target language sentences in the examples. These structural translation examples have the advantage of high usability. However, it is difficult to build such a structural translation example database. Firstly, there is lack of the high-accuracy parsing tools in some languages. Secondly, if two languages in a language pair are quite different, it is difficult to solve the problem of the structural correspondences (Shieber, 1994; Al-Adhaileh et al., 2002). In order to deal with the problems, our system only parses the source language sentences in the examples, while the target language sentences are represented as word strings. In addition, the corresponding words in the bilingual examples are automatically aligned. Thus, the examples in our system are represented as TSC, which consists of three parts: a parse tree in the source language, a string in the target language, and the correspondences between the leaf nodes of the source language tree and the substrings of the target language string.

In the EBMT systems using the structural translation examples, the similarity between the translation example and the input sentence is calculated with the tree edit distance (Matsumoto et al., 1993; Watanabe, 1995; Al-Adhaileh and Tang, 1999). The calculation involves a rather complex tree-matching operation. It is certainly a considerable computation cost (Somers, 1999). In order to alleviate this problem, this paper proposes a tree-matching algorithm based on TSC to find the TSC forest that is best matched with the input tree.

For EBMT systems, there are two major approaches to select the appropriate translation fragments and generate the translation. The semantic-based approach (Aramaki et al., 2003; Armaki and Kurohashi, 2004) obtains an appropriate translation fragment for each part of the input sentence. The final translation is generated by combining the translation fragments in a pre-defined order. This approach does not take into account the fluency between the translation fragments. The statistical approach (Kaki et al., 1999; Callison-Burch and Flournoy, 2001; Akiba et al., 2002; Imamura et al., 2004) selects translation fragments with a statistical model. The statistical model can improve the fluency by using n-gram co-occurrence statistics. However, the statistical model does not take into account the semantic relation between the translation example and the input sentence. In this paper, we propose a new method to select the translation fragments and generate the translation, which combines the semantic-based approach and the statistical approach. The generation model consists of three parts: the semantic similarity between the tree in TSC and the input tree, the translation probability from source word to target word, and the standard n-gram language model for target language.

The remainder of the paper is organized as follows. The next section presents TSC. Section 3 describes the translation method based on TSC. Section 4 presents the evaluation results. Section 5 concludes this paper and presents the future work.

2 Tree String Correspondence

In our system, we use TSC to represent the translation example. TSC consists of three parts: a parse tree in the source language, a string in the target language and the correspondences between the leaf nodes of the source language tree and the substrings of the target language string. The corresponding words in the bilingual examples are automatically aligned using the GIZA++ toolkit[1]. The source language sentences in the examples are parsed using the Collins' parser[2] (Collins, 1999).

Figure 1 illustrates a translation example. The English sentence is "Mary borrowed a book from her friend." The Chinese translation is "玛丽 从 她 朋友 那里 借 了 一 本 书 。".

In this paper, the original parse tree rooted at r_0 is modified as below:

(1) If a punctuation node n is the most left/right leaf node, then set r_0 as the parent node of n. Otherwise, let n_r be the nearest right neighbor of n, then the nearest common ancestor of n and n_r is set as the parent node of n.

(2) In the tree, there are two kinds of nodes: the nonterminal node that consists of the category and the headword; and the leaf node that consists of the word and its Part-of-Speech. To simplify the presentation, these two kinds of nodes are uniformly represented as a pair of the category and the headword.

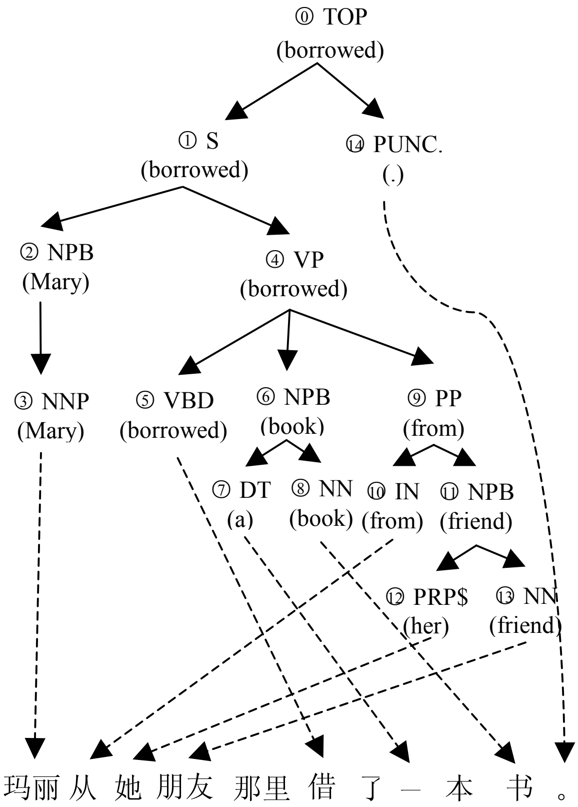


Figure 1. Translation Example

Given a parse tree T, if a set S satisfies the following conditions:

(1) There is more than one element in S.

(2) All the elements of S are the nodes of T.

(3) In S, there is only one node r whose parent node isn't in S. All the other nodes are descendant nodes of r.

(4) For any node n in S except r, the sibling node of n is also in S.

Then, all the nodes in S form a *sub-tree* (*ST*) of T. If all the descendant nodes of r are in S, then ST is a *maximal sub-tree* rooted at r. Otherwise ST is a *partial sub-tree* rooted at r. E.g., in Figure 1, the nodes ④, ⑤, ⑥, ⑦, ⑧, ⑨, ⑩, ⑪, ⑫, ⑬ form a maximal sub-tree (see Figure 2 (a)). The nodes ④, ⑤, ⑥, ⑨, ⑩, ⑪ form a partial sub-tree (see Figure 2 (b)). The nodes ④, ⑤, ⑥ don't form a sub-tree of T. A parse tree is a maximal sub-tree of itself.

If a leaf node n of a partial sub-tree is a nonterminal node in T, then n is called a *substitution node*. A substitution node only contains the category of the node. In Figure 2 (b), there are two substitution nodes ⑥, ⑪.

TSC can be represented as a triple defined as follow:

$$<t, s, c>$$

t is a sub-tree of the parse tree.

s is a string in the target language corresponding to t.

[1] http://www.clsp.jhu.edu/ws99/projects/mt/toolkit/

[2] http://www.ai.mit.edu/people/mcollins

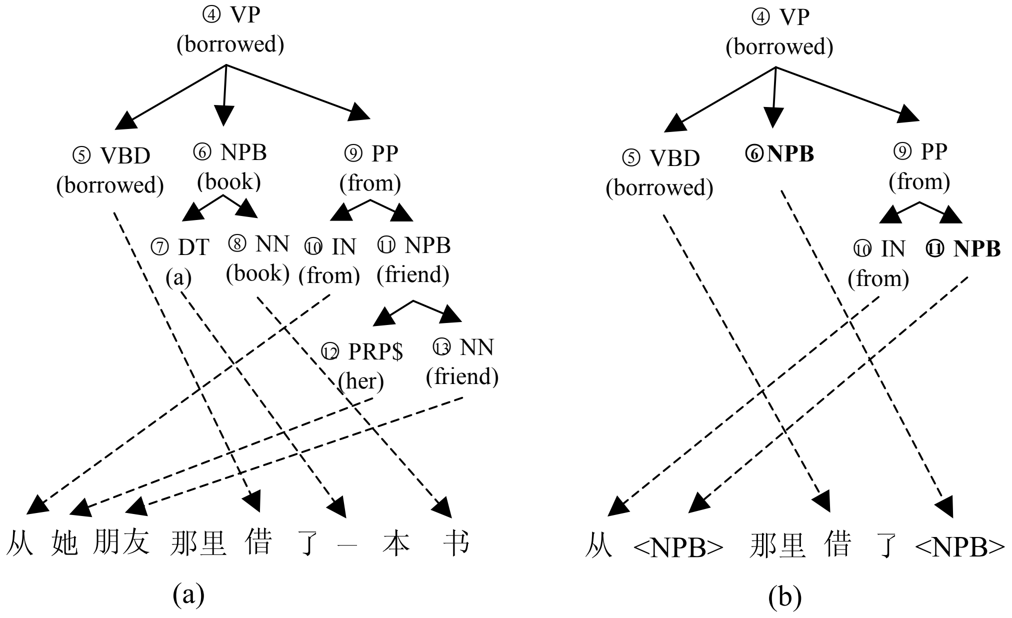


Figure 2. Example of TSC
(a) contains a maximal sub-tree, (b) contains a partial sub-tree

c consists of the correspondences between the leaf nodes of *t* and the words or the substitution symbols of *s*.

For a maximal sub-tree *t*, let S_t include all the words in the target language corresponding to the leaf nodes in *t*. *s* is the corresponding sub-string of *t* if

$$w_L = \arg\min_{w_i \in S_t}(i)$$

$$w_R = \arg\max_{w_i \in S_t}(i)$$

$$s = w_L \ldots w_R$$

For a partial sub-tree *t*, the corresponding *s* contains the words and the substitution symbols. A substitution symbol corresponds to a substitution node in *t*. In Figure 2 (b), which is derived from Figure 2 (a), the strings in the angle brackets indicate the substitution symbols. There are two substitution symbols in Figure 2 (b). The first one is to replace "她 朋友", which is corresponded to the maximal sub-tree rooted at the node ⑪. The second one is to replace "一 本 书", which is corresponded to the maximal sub-tree rooted at the node ⑥.

If a <*t*, *s*, *c*> is matched with a tree *T*, it should satisfy the following conditions.

(1) The nonterminal node of *t* is matched with the corresponding node of *T* by both the headword and the category of the node.

(2) The leaf node of *t* is matched with the corresponding node of *T* at least by the category of the node.

3 Translation Based on TSC

The translation consists of three steps.

(1) **Parsing**: The input sentence is parsed into a tree.

(2) **Tree-matching**: the TSC forest is searched out if it is best matched with the input tree.

(3) **Generation**: The final translation is generated using a statistical generation model to combine the target language strings in the TSCs.

We use the same parser described in section 2 to parse an input sentence in step (1). In the following two subsections, we will describe the tree-matching algorithm and the translation generation method.

3.1 Tree-Matching

3.1.1 Matching Score

The matching score of a TSC and an input sentence is defined as the similarity between the tree in the TSC and the parse tree of the input sentence. It is calculated as shown in Equation 1.

$$M(<t,s,c>,T) = \sum_i Sim(n_i, n_i') \qquad (1)$$

Where,

T is the input tree;

n_i is the i^{th} node in *t*;

n_i' is the corresponding node of n_i in T;

$Sim(n_i, n_i')$ is the similarity between n_i and n_i'.

If both the headword and the category of n_i are identical with that of n_i', $Sim(n_i, n_i')$ is set to 1.0. Otherwise, $Sim(n_i, n_i')$ is defined as the semantic similarity between the headwords of the two nodes.

$$Sim(n_i, n_i') = WordSim(w_i, w_i') \qquad (2)$$

Where,

w_i is the headword of n_i;

w_i' is the headword of n_i'.

In this paper, we take English to Chinese translation as a case study. The semantic similarity between English words is calculated by using WordNet. We employ the same method as described in (Lin, 1998) to calculate the semantic similarity between words in the semantic hierarchy.

$$WordSim(w_1, w_2) = \frac{2 \times \log p(C_0)}{\log p(C_1) + \log p(C_2)} \qquad (3)$$

Where,

C_1 is the concept that w_1 belongs to;

C_2 is the concept that w_2 belongs to;

C_0 is the nearest common ancestor in the semantic hierarchy that subsumes both C_1 and C_2;

$p(C_i)$ is the probability of encountering an instance of C_i.

3.1.2 Tree-Matching Algorithm

For a TSC forest and an input tree, if the source language trees of the forest can exactly compose the input tree, then this TSC forest is matched with the input tree. If the average matching score of the TSC forest is the highest in all the forests that can be matched with the input tree, then this forest is best matched with the input tree.

For an input tree, there may be many TSC forests matched with it. We need to find a TSC forest that is best matched with the input tree.

In this paper, we use a greedy algorithm to search for a TSC forest, which is good but not always best matched with the input tree. Figure 3 shows the detailed algorithm.

If *<t, s, c>*$_i$, *<t, s, c>*$_j$ $\in F$ (F is a TSC forest produced by the tree-matching algorithm) and the root of t_j is corresponded to a substitution node of t_i, then *<t, s, c>*$_j$ is called the *child TSC* of *<t, s, c>*$_i$, *<t, s, c>*$_i$ is called the *parent TSC* of *<t, s, c>*$_j$.

```
Input: Parse tree $T_{in}$
(1)     TSC forest $F$ is set to NULL.
(2)     If $SearchTSC(T_{in}, F)$ = TRUE
(3)        Return $F$.
(4)     Else
(5)        Return NULL.
Output: TSC forest $F$ or NULL

Function $SearchTSC$(Sub-tree $T$, TSC forest $F$)
(S1)    Set $n_0$ to the root of $T$.
(S2)    Search for all TSC matched with $T$ and
        rooted at $n_0$. Set $c$ to the number of
        searched TSCs.
(S3)    If $c$ = 0 return FALSE.
(S4)    Sort the searched TSCs in descending
        order of the matching score.
(S5)    Set $i$ = 1.
(S6)    For each substitution node $n_k$ of $TSC_i$
(S7)       Set $n_s$ to the corresponding node of $n_k$
           in $T$.
(S8)       Set $T_s$ is the maximal sub-tree of $T$
           and $T_s$ is rooted at $n_s$.
(S9)       If $SearchTSC(T_s, F)$ = FALSE
(S10)         Remove all the TSCs, which are
              matched with $T$, from $F$.
(S11)         if $i$ >= $c$ return FALSE.
(S12)         $i = i + 1$.
(S13)         Goto (S6).
(S14)   Add $TSC_i$ to $F$.
(S15)   Return TRUE.
```

Figure 3. Tree-Matching Algorithm

3.2 Statistical Generation

Based on the searched TSC forest, the final translation is generated by combining the target language strings in the TSCs in a bottom-up manner (Imamura et al., 2004). For each TSC in the forest, if the target language string contains the substitution symbols, then the substitution symbol is substituted by the translation that is obtained from the child TSC. The generated translation of the current TSC is reused to produce the translation of the parent TSC.

In subsection 3.1.2, the tree-matching algorithm only finds one TSC for each t in the forest. In the generation step, for each TSC in the forest, all

TSCs with the same t are retrieved from the translation example database as the candidates. The best TSC is then selected from the candidates according to the following three factors:

(1) The matching score of TSC;
(2) The word translation probability between s and t in TSC;
(3) The probability of the translation fragment occurring in the target language.

The matching score is calculated as described in subsection 3.1.1.

The word translation probability of TSC is calculated by using the word translation model based on the word alignment information. The word translation probability of TSC is defined in Equation 4.

$$P_{TM}(<t,s,c>) = \exp(\sum_i \ln(p(f_{a_i} \mid e_i))/N) \quad (4)$$

Where,

$\exp(x) = e^x$;

e_i is the headword of the leaf node in t;

f_{a_i} is the corresponding word of e_i in s;

$p(f_{a_i} \mid e_i)$ is the probability of e_i translating to f_{a_i};

N is the total number of leaf nodes in t.

A trigram language model is used to calculate the probability of the translation fragment occurring in the target language. The language model is described in Equation 5:

$$P_{LM}(st) = \prod_i p(w_{\mathrm{i}} \mid w_{\mathrm{i\text{-}2}}, w_{\mathrm{i\text{-}1}}) \quad (5)$$

Where,

st is the produced translation fragment by combining the target language strings of TSCs;

w_i is the i^{th} word in the st.

Based on the above three factors, the final score of the translation fragment can be calculated as shown in Equation 6:

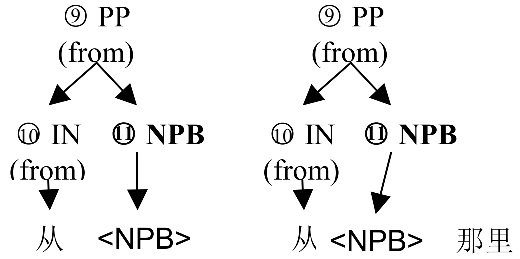
$$Q(st) = P_{LM}(st) \times \prod_i M(<t,s,c>_i, T) \times \prod_i P_{TM}(<t,s,c>_i) \quad (6)$$

Where,

$<t,\ s,\ c>_i$ is one of the TSCs that are used to generate st.

Thus, the best translation is obtained by maximizing Equation 6.

From the word alignments of the bilingual examples, it can be seen that the auxiliary words, quantifiers and other functional words in one language have no counterparts in the other language. For instance, in Figure 1, “那里” and “了” in the Chinese translation have no counterparts in the English sentence. However, these words are important to ensure the fluency of the generated translation. For instance, we construct two TSCs from the translation example of Figure 1, which are rooted the same node ⑨. One ignores the unaligned Chinese word “那里” while the other considers the unaligned Chinese word “那里”. The TSCs are shown in Figure 4.



(a) (b)

Figure 4. Example of TSC
(a) does not include the boundary word
(b) includes the boundary word

TSC (a) in Figure 4 is applied to translate (1a) and (2a) in Table 1. The final translations are (1b) and (2b), respectively. TSC (b) in Figure 4 is also applied to translated (1a) and (2a) in Table 1. The final translations are (1c) and (2c), respectively.

The LM column in Table 1 lists the language model probability for each translation. The value is calculated by using $-\log(P_{LM})$. The larger the log-probability is, the worse the translation is. From the results, it can be seen that the fluency of (1b) is better than (1c) and the fluency of (2c) is better than (2b). Thus, (1a) should select TSC (a) in Figure 4 to translate it. (2a) should select TSC (b) in Figure 4 to translate it.

ID	Source sentence & translation	LM
1a	He got down from the bus.	-
1b	他从公共汽车下来。	10.98
1c	他从公共汽车那里下来。	14.77
2a	He got the alms from the government.	-
2b	他从政府得到救助金。	18.36
2c	他从政府那里得到救助金。	16.39

Table 1. Examples of translations

In order to improve the fluency of the generated translation, the unaligned words adjacent to *s* are taken into account as optional words of the target language string of TSC when we construct TSC. In this paper, this kind of the unaligned word is called *boundary word*. During translation generation, the statistical model determines whether the boundary word is selected and added to the translation.

4 Evaluation

We evaluate the performance of our EBMT system in terms of translation quality. We first evaluate the component of our system, and then compare our system with several commercial machine translation systems. The NIST score (Doddington, 2002) is used for evaluation. The evaluation tool is the NIST MT Evaluation Toolkit[3].

4.1 Resource

Bilingual Corpus The bilingual corpus includes 262,560 English-Chinese bilingual sentence pairs collected from the general domain. The average length of the English sentences is 12.1 words while the average length of the Chinese sentences is 12.5 words. All of the sentence pairs are processed by the word alignment toolkit and the English sentences are parsed.

Testing Set We randomly select 500 sentence pairs from the bilingual corpus, the English sentences of which are used as the testing set. Each sentence in the testing set has two translation references.

Translation Dictionary We employ an English-Chinese translation dictionary to translate the words that can't be translated using the translation examples. The dictionary has about 90,000 entries. Each translation in the dictionary is assigned a translation probability. The translation probability is calculated based on the word-aligned bilingual corpus, which is described in Equation 7.

$$p(c \mid e) = \frac{p(e,c)}{p(e)} \qquad (7)$$

Where,

e and c represent the English word and the Chinese word, respectively;

$p(e)$ is the probability of occurrence of e;

$p(e, c)$ is the probability of co-occurrence of e and c.

Language Model The Chinese language model in our system is a standard trigram model. The language model is trained on a general Chinese corpus using the SRILM toolkit (Stolcke, 2002). The training data include 228 million Chinese words. The perplexity of the language model with respect to the Chinese sentences of the bilingual corpus is 31.85.

4.2 Component Evaluation

In this section, we design some comparable experiments to evaluate the translation generation component of our EBMT system.

Baseline: For the source language tree of each TSC in the forest, the target language string, which occurs most frequently in the candidates with the highest matching score, is selected. The final translation is generated by simply combining the target language strings of TSCs in a bottom-top manner.

LM + BW: During translation generation, the boundary words are taken into account. Only the language model is used to select the best target language string for the source language tree of each TSC in the forest.

LM + TM + BW: During translation generation, the boundary words are taken into account. The language model and the word translation model are used.

LM + TM + MS: During translation generation, the boundary words are not taken into account. The language model, the word translation model and the matching score are used.

The evaluation results are shown in Table 2. From the evaluation results, it can be seen that our system achieves the best translation result among all of the systems. This indicates that our translation generation method is very effective to improve translation quality.

Method	NIST
Baseline	4.7722
LM + BW	4.6611
LM + TM + BW	5.0429
LM + TM + MS	4.8174
Our system	5.2577

Table 2. Evaluation Results of Translation Generation Component

From the results of "LM + BW" and "LM + TM + BW", it can be seen that it is useful to improve the translation quality by introducing the word translation model to the generation model. In fact, the word translation model is helpful to improve

[3] http://www.nist.gov/speech/tests/mt/mt2001/index.htm

the target word selection. By comparing these two systems with our system, we can find that our system outperforms both of them. This is because we use the matching score, which is calculated using word semantic similarity, besides the language model and the word translation model. This result indicates that our translation generation method is useful to improve translation quality by combining the above three aspects.

The result of “LM + TM + MS” is much worse than the result of “Our system”. This is because of the effect of boundary words on the translation fluency. It also indicates that it is helpful to determine the boundary word using statistical generation model.

4.3 Compared with Commercial Translation Systems

Besides the above evaluation, we also compare our system with several commercial ECMT systems available in Chinese Market. We evaluate the commercial systems using the same testing set as described in section 4.1. Table 3 shows the evaluation results on our system and the top 3 commercial systems. Although the score of our system is slightly lower than the best system, our translation quality is comparable with the commercial systems. This indicates that TSC and the translation generation method used in our system are very effective for automatic machine translation.

System	NIST
Commercial System 1	5.0189
Commercial System 2	5.0677
Commercial System 3	5.2870
Our System	5.2577

Table 3. Evaluation Results of the Commercial Systems and Our System

5 Conclusion and Future Work

This paper describes an example-based machine translation system. The system effectively improves the quality of the translation by using TSC and a statistical generation model. In the system, we use a source language parser and a word alignment tool to construct the translation examples. For an input sentence to be translated, the system first parses it into a tree. Then the system uses a tree-matching algorithm to obtain the TSC forest, which is best matched with the input tree. The final translation is generated using a hybrid statistical generation model. This generation model combines the semantic similarity between words, the translation probability from the source word to the target word, and the target language model. In addition, the boundary words are introduced into the translation generation model. The appropriate use of these words improves the fluency of the generated translation. Experimental results indicate that the generation model is effective to select the translation for each sub-tree and to generate the final translation.

By comparing our system with the commercial machine translation systems, we also find that the translation quality of our system is comparable with the best commercial system.

In future work, we will investigate constructing TSC using the dependency tree to extend the applications of TSC. We will also explore using structure alignment to alleviate the problem of word alignment to improve translation quality.

References

Yasuhiro Akiba, Taro Watanabe and Eiichiro Sumita. 2002. Using Language and Translation Models to Select the Best among Outputs from Multiple MT systems. In *Proc. of the 19th International Conference on Computational Linguistics (COLING-2002)*, pages 8-14.

Mosleh H. Al-Adhaileh and Tang Enya Kong. 1999. Example-Based Machine Translation Based on the Synchronous SSTC Annotation Schema. In *Proc. of Machine Translation Summit VII*, pages 244-249.

Mosleh H. Al-Adhaileh, Tang Enya Kong and Zaharin Yusoff. 2002. A Synchronization Structure of SSTC and its Applications in Machine Translation. *The COLING-2002 Post-Conference Workshop on Machine Translation*.

Eiji Aramaki and Sadao Kurohashi. 2004. Example-Based Machine Translation Using Structural Translation Examples. *International Workshop on Spoken Language Translation (IWSLT)*, pages 91-94.

Eiji Aramaki, Sadao Kurohashi, Hideki Kashioka and Hideki Tanaka. 2003. Word Selection for EBMT based on Monolingual Similarity and Translation Confidence. In *Proc. of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts*, pages 57-64.

Chris Callison-Burch and Raymond S. Flournoy. 2001. A Program for Automatically Selecting the Best Output from Multiple Machine Translation Engines. In *Proc. of the Machine Translation Summit VIII*, pages 63-66.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *Proc. of the Second Conference on Human Language Technology (HLT-2002)*, pages 128-132.

Kenji Imamura, Hideo Okuma, Taro Watanabe and Eiichiro Sumita. 2004. Example-Based Machine Translation Based on Syntactic Transfer with Statistical Models. In *Proc. of the 20th International Conference on Computational Linguistics (COLING-2004)*, pages 99-105.

Satoshi Kaki, Setsuo Yamada and Eiichiro Sumita. 1999. Scoring Multiple Translations Using Character N-gram. In *Proc. of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-1999)*, pages 298-302.

Dekang Lin. 1998. An Information-Theoretic Definition of Similarity. In *Proc. of the 15th International Conference on Machine Learning*, pages 296-304.

Yuji Matsumoto, Hiroyuki Ishimoto and Takehito Utsuro. 1993. Structural Matching of Parallel Texts. In *Proc. of 31st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, pages 23-30.

Arjen Poutsma. 2000. Data-Oriented Translation. In *Proc. of the 18th International Conference on Computational Linguistics (COLING-2000)*, pages 635-541.

Stuart M. Shieber. 1994. Restricting the Weak Generative Capacity of Synchronous Tree Adjoining Grammar. *Computational Intelligence*, 10(4):371-385.

Harold Somers. 1999. Review Article: Example-based Machine Translation. *Machine Translation*, 14(2): 113-157.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proc. of the International Conference on Spoken Language Processing (ICSLP-2002)*, pages 901-904.

Hideo Watanabe. 1992. A Similarity-Driven Transfer System. In *Proc. of the 14th International Conference on Computational Linguistics (COLING-1992)*, pages 770-776.

Hideo Watanabe. 1995. A Model of a Bi-Directional Transfer Mechanism Using Rule Combinations. *Machine Translation*, 10(4): 269-291.

Andy Way. 2003. *Machine Translation Using LFG-DOP*. In R. Bod, R. Scha & K. Sima'an (eds.) Data-Oriented Parsing, CSLI Publications, Stanford, CA., pages 359-384.