Log-linear Generation Models for Example-based Machine Translation

Zhanyi Liu, Haifeng Wang, Hua Wu

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 describes log-linear generation models for Example-based Machine Translation (EBMT). In the generation model, various knowledge sources are described as the feature functions and are incorporated into the log-linear models. Six features are used in this paper: matching score and context similarity, to estimate the similarity between the input sentence and the translation example; word translation probability and target language string selection probability, to estimate the reliability of the translation example; language model probability and length selection probability, to estimate the quality of the generated translation. In order to evaluate the performance of the log-linear generation models, we build an English-to-Chinese EBMT system with the proposed generation model. Experimental results show that our EBMT system significantly outperforms both a baseline EBMT system and a phrase-based SMT system.

Introduction

In example-based Machine Translation (EBMT), translation generation plays a crucial role (Somers, 1999; Hutchins, 2005). For EBMT systems, there are two major approaches to selecting the translation fragments and to generating the final translation. Semantic-based approaches obtain an appropriate target language fragment for each part of the input sentence by means of a thesaurus. The final translation is generated by recombining the target language fragments in a predefined order (Aramaki et al., 2003; Aramaki & Kurohashi, 2004). This approach does not take into account the transition between fragments. Therefore, the fluency of the translation is weak. Statistical approaches select translation fragments with a statistical model (Knight & Hatzivassiloglou, 1995; Kaki et al., 1999; Callison-Burch & Flournoy, 2001; Akiba et al., 2002; Hearne & Way, 2003&2006; Imamura et al., 2004; Badia et al., 2005; Carl et al., 2005). The statistical model can solve the transition problem by using *n*-gram co-occurrence statistics. However, this approach does not take into account the semantic relations between the translation example and the input sentence. As a result, the accuracy of translation is poor. Liu et al. (2006) presented a hybrid generation model which combines these two approaches.

In this paper, we propose log-linear generation models for EBMT. Unlike the hybrid model presented in (Liu et al., 2006), our generation model uses various knowledge sources that are described as feature functions. The feature functions are incorporated into the log-linear models (Och & Ney, 2002&2004). In this paper, we use six feature functions. Matching score and context similarity are used to estimate the similarity between the input sentence and the source part of the translation example. Word translation probability and target language string selection probability are used to estimate the reliability of the translation example. Language model probability and length selection probability are used to estimate the quality of the generated translation. Experimental results show that the performance of the EBMT system is significantly improved by using the log-linear generation models. Such an EBMT system also achieves a significant improvement of 0.0378 BLEU score (17.2% relative) as compared with a phrase-based SMT system.

The remainder of the paper is organized as follows. The next section briefly introduces the Tree String Correspondence based EBMT method. And then we describe the log-linear generation models and the feature functions. After that, the search algorithm is described. Finally, we present the experimental results and conclude this paper.

Tree String Correspondence Based EBMT

In this paper, we improve the Tree String Correspondence (TSC) based EBMT method (Liu et al., 2006) with the log-linear generation models.

Definition of TSC

Given a phrase-structure tree T and a subtree T_s of T, T_s is a matching-tree of T if T_s satisfies the following conditions:

1. There is more than one node in T_s.
2. In T_s, there is only one node r (the root node of T_s) whose parent node is not in T_s. All the other nodes in T_s are descendant nodes of r.
3. For any node n in T_s except r, the sibling node of n is also in T_s.

Here, each node of the parse tree is labeled with its headword and category.

TSC is defined as a triple <*t*, *s*, *c*>, where *t* is a matching-tree of the source language parse tree; *s* is a target language string corresponding to *t*; *c* denotes the word correspondence, which consists of the links between the leaf nodes of *t* and the substrings of *s*.

If the leaf node of the matching-tree in TSC is a non-terminal node of the parse tree, then this kind of leaf node is also called a substitution node. The correspondence in the target language string of the substitution node is called a substitution symbol. The substitution symbol can represent a single word, or phrase that can be expanded by other matching-tree. During translation, for each

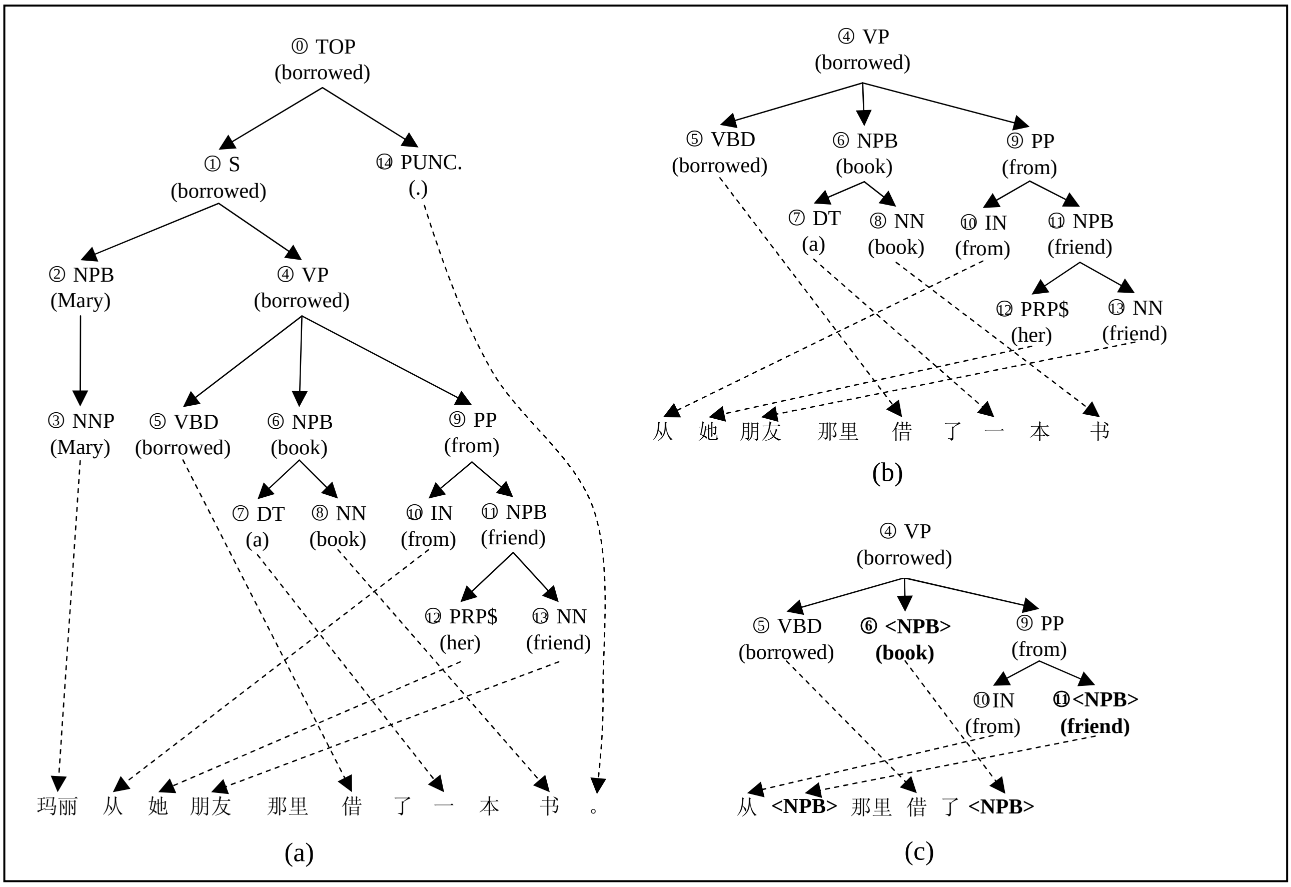

Figure 1. Examples of TSC

substitution node, its corresponding substitution symbol will be replaced by the translation candidate of the TSC whose root node corresponds to this substitution node.

A TSC is used to represent either of static translation examples or dynamic translation example fragments. In the TSC-based EBMT system, a preprocessed translation example is statically stored as a TSC in the example database. During the translation, a translation example fragment, which is identified to match the input, is represented as a TSC.

In this paper, we use English-to-Chinese MT as a case study. Figure 1 shows three examples of English-to-Chinese TSC. TSC (a) indicates the following translation example:

Mary borrowed a book from her friend.
玛丽 从 她 朋友 那里 借 了 一 本 书 。
(Mary from her friend there borrow a book .)

In this TSC, the matching-tree of the source language and the target language string are composed of the source part and the target part of the translation example, respectively. TSC (b) and (c) are derived from TSC (a). The matching-tree of TSC (b) and (c) matches the subtrees of the TSC (a) that are rooted at Node 4. The matching-tree of TSC (b) matches all descendant nodes of Node 4 and no substitution nodes are included in matching-tree. The target language string corresponding to the subtree is considered the translation of the matching-tree. Different from TSC (b), the leaf nodes 6 and 11 in the matching-tree of TSC (c) are the non-terminal nodes of the matching-tree of TSC (a). The two nodes in this TSC are the substitution nodes. Their corresponded parts in the target language string are the substitution symbols "<NPB>". Thus, the target language string of TSC (c) consists of the target language words and the substitution symbols.

Two TSCs are homologous if their source language matching-trees are the same, which means that the same source language matching-tree can be translated into the different target language strings.

A TSC forest matching a parse tree means that the source language matching-trees of the TSC forest can exactly compose the parse tree. For TSC T_1 and T_2 in the TSC forest, if the root node of T_1 corresponds to a substitution node of TSC T_2, then T_1 is the child TSC of T_2 and T_2 is the parent TSC of T_1.

EBMT Based on TSC

In the EBMT system based on TSC, the translation example is presented as the TSC. For an input sentence to be translated, it is first parsed into a tree. Then the TSC forest which best matches the input tree is searched out. Finally, the translation is generated by combining the target language strings of TSCs.

For the parse tree of the input sentence, there are many TSC forests that match the parse tree. In Liu et al. (2006), a TSC can better match a parse tree if the TSC has more nodes or the TSC has higher matching score with the parse tree. Thus, a TSC forest best matches a parse tree if the TSC forest has the highest matching score in the TSC forest candidates. A greedy tree-matching algorithm was used to search for the TSC forest that best matches the parse tree of the input sentence. The algorithm first searches for the best matching TSC whose root node corresponds to that of the parse tree. Then, for each substitution node of the TSC, the algorithm continues to search for the TSC that best matches the subtree of the parse tree rooted at the substitution node. The procedure is iterated until all substitution nodes are expanded.

During searching for the final translation based on the TSC forest, the TSC forest is first extended by adding the homologous TSCs in order to include more possible translation candidates. Then the hybrid generation model, including matching score of TSC, word translation probability of source words and target words, and target language model probability, is used to generate the final translation. The final translation is produced by combining the target language strings in the TSC forest in a bottom-up manner. If the target language string contains the substitution symbol, then the substitution symbol is replaced with the translation of the corresponding substitution node.

Log-linear Generation Models

We incorporate various features into our log-linear models. Given the input (source language sentence) $\mathbf{f}=f_1^J=f_1,...,f_j,...,f_J$, the translation (target language sentence) $\mathbf{e}=e_1^I=e_1,...,e_i,...,e_I$ with the highest probability is chosen from the possible target language sentences according to Eq. 1.

$$\mathbf{e} = \arg\max_{\mathbf{e}'}\{p(\mathbf{e}' \mid \mathbf{f})\} \qquad (1)$$

Based on the maximum entropy framework, we directly model the posterior probability $p(\mathbf{e} \mid \mathbf{f})$ using the same method as described in (Berger et al., 1996). In this framework, there are M feature functions $h_m(\mathbf{e},\mathbf{f})$, m=1,...,M. For each feature function, there exists a model parameter λ_m. We can get the translation probability as described in Eq. 2.

$$p(\mathbf{e} \mid \mathbf{f}) = \frac{\exp[\sum_{m=1}^{M}\lambda_m h_m(\mathbf{e},\mathbf{f})]}{\sum_{\mathbf{e}'}\exp[\sum_{m=1}^{M}\lambda_m h_m(\mathbf{e}',\mathbf{f})]} \qquad (2)$$

Thus, we obtain the decision rule:

$$\begin{aligned}\mathbf{e} &= \arg\max_{\mathbf{e}'}\{p(\mathbf{e}' \mid \mathbf{f})\} \\ &= \arg\max_{\mathbf{e}'}\{\sum_{m=1}^{M}\lambda_m h_m(\mathbf{e}',\mathbf{f})\}\end{aligned} \qquad (3)$$

Typically, $p(\mathbf{e}|\mathbf{f})$ can be decomposed by adding hidden variables. To include the dependence on the hidden variables, we extend the feature functions by including the following hidden variables: the parse tree $\mathbf{F}$ of the input sentence and the TSC forest $\mathbf{Z}$ with K TSCs $z_1^K=z_1,...z_k,...z^K$. $\mathbf{Z}$ is used to generate the final translation. Thus, we obtain M feature functions of the form $h_m(\mathbf{e},\mathbf{f},\mathbf{F},\mathbf{Z})$, and the following rule:

$$\mathbf{e} = \arg\max_{\mathbf{e}',\mathbf{F},\mathbf{Z}}\{\sum_{m=1}^{M}\lambda_m h_m(\mathbf{e}',\mathbf{f},\mathbf{F},\mathbf{Z})\} \qquad (4)$$

In our system, only 1-best parse tree is considered, so we get Eq. 5:

$$\mathbf{e} = \arg\max_{\mathbf{e}',\mathbf{Z}}\{\sum_{m=1}^{M}\lambda_m h_m(\mathbf{e}',\mathbf{f},\mathbf{F},\mathbf{Z})\} \qquad (5)$$

Feature Functions

We use six feature functions in the log-linear generation models.

Matching score and context similarity are used to estimate the similarity between the translation example and the input sentence. The former describes the semantic similarity of the matched fragments. The latter describes the context similarity of the sentences.

Word translation probability and target language string selection probability are used to estimate the reliability of the translation example. The former is based on the word alignment probability. The latter makes use of the probability of the target language string given the source language matching-tree.

Language model probability and length selection probability are used to estimate the quality of the generated translation. The former scores the fluency of the generated translation. The latter adjusts the estimation based on the target sentence length.

Matching Score

The matching score between TSC and **F** is defined as the sum of the semantic similarity between the nodes in TSC and the matched nodes in **F**. It is calculated as shown in Eq. 6:

$$M(<t,s,c>,\mathbf{F}) = \sum_{n_i \in t} Sim(n_i, n_i') \qquad (6)$$

Here n_i is the ith node in t; n_i' is the corresponding node of n_i in **F**; $Sim(n_i,n_i')$ is the semantic similarity between the headwords of n_i and n_i'.

The semantic similarity between English words is calculated based on WordNet (Fellbaum, 1998). We employ the same method as described in (Lin, 1998).

$$\begin{aligned}Sim(n_1,n_2) &= Sim(f_1,f_2) \\ &= \frac{2\times\log p(C_0)}{\log p(C_1)+\log p(C_2)}\end{aligned} \qquad (7)$$

Here C_1 and C_2 are the headwords of n_1 and n_2, respectively; C_1 and C_2 are the concepts subsuming the words f_1 and f_2, respectively; 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 in the corpus.

Hence, we get the feature function h_{MS}:

$$h_{\text{MS}}(\mathbf{e},\mathbf{f},\mathbf{F},\mathbf{Z}) = \log\prod_{k=1}^{K} M(z_k,\mathbf{F}) \qquad (8)$$

Context Similarity

Context similarity is used in various NLP tasks (Karov & Edelman, 1998; Schafer & Yarowsky, 2002). We use it to select the approximate translation example and to further improve translation selection. The context similarity is defined as the word-based cosine distance between sentences.

$$CS(\mathbf{f},z)=\frac{\mathbf{V}\cdot\mathbf{V}'}{\sqrt{\sum_i (v_i)^2}\times\sqrt{\sum_j (v'_j)^2}} \qquad (9)$$

Here **V** denotes the sequence of the word count of **f**; **V′** denotes the sequence of the word count of the source language sentence of the example from which *z* is derived. Thus, the context similarity h_{CS} is defined as shown in Eq. 10:

$$h_{\text{CS}}(\mathbf{e},\mathbf{f},\mathbf{F},\mathbf{Z})=\log\prod_{k=1}^{K} CS(\mathbf{f},z_k) \qquad (10)$$

Word Translation Probability

The quality of word alignment in the translation example affects the quality of the translation. To estimate the quality of the word alignment in TSC, we use the single-word translation probability between the source language matching-tree and the target language string. The word translation probability of TSC is defined in Eq. 11:

$$W(<t,s,c>)=(\prod_{(j,i)\in c} p(e_i \mid f_j))^{1/|c|} \qquad (11)$$

Here e_i is the correspondence word of f_j; $p(e_i|f_j)$ is the word translation probability, which is derived from the word-aligned translation examples using Eq. 12.

$$p(e \mid f)=\frac{C(e,f)}{\sum_{e'} C(e',f)} \qquad (12)$$

Here $C(e,f)$ is the count of aligned word pair (e,f) in the word-aligned corpus.

The word translation probability h_{WTP} is described in Eq. 13:

$$h_{\text{WTP}}(\mathbf{e},\mathbf{f},\mathbf{F},\mathbf{Z})=\log\prod_{k=1}^{K} W(z_k) \qquad (13)$$

Target Language String Selection Probability

The target language string selection probability is described in Eq. 14:

$$\begin{aligned} T(<t,s,c>) &= p(s \mid t) \\ &= \frac{\mathrm{C}(t,s)}{\sum_{\forall(t,s',c)} \mathrm{C}(t,s')} \end{aligned} \qquad (14)$$

Here $p(s \mid t)$ is the probability of *s* given *t*.

Higher target language string selection probability may result in more reliable target language string. Thus, we define the target language string selection probability h_{SSP} as in Eq. 15:

$$h_{\text{SSP}}(\mathbf{e},\mathbf{f},\mathbf{F},\mathbf{Z})=\log\prod_{k=1}^{K} T(z_k) \qquad (15)$$

Language Model Probability

A trigram language model is used to calculate the probability of the translation fragment occurring in the target language. This feature function h_{LMP} is defined in Eq. 16:

$$h_{\text{LMP}}(\mathbf{e},\mathbf{f},\mathbf{F},\mathbf{Z})=\log(\prod_{i=1}^{I} p(e_i \mid e_{i-2},e_{i-1}))^{1/I} \qquad (16)$$

Here the geometric mean of the probability is used to prevent preference for a short translation.

Length Selection Probability

Generally, the length of the target language sentence depends on that of the source language sentence. To ensure that the translation does not become too long or too short, we use a sentence length selection model to calculate the probability of the length of the target language sentence given the source language sentence.

$$p(I \mid J)=\frac{C(I,J)}{\sum_{I'} C(I',J)} \qquad (17)$$

Here the word number of the sentence is defined as the length of the sentence. The model is trained on the bilingual example database.

In the translation process, we need to score the length of the translation fragment, instead of the length of the sentence. The length of the fragment is more flexible. It is difficult to directly model the fragment length selection. The target language length tends to follow a normal distribution on the fixed source language length in the parallel corpus (Brown et al., 1991). Based on the normal distribution, we approximately model the fragment length selection using the ratio of the target language fragment length to the source language fragment length.

$$p_N(r;\mu,\sigma^2)=\frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(r-\mu)^2}{2\sigma^2}) \qquad (18)$$

Here *r=I/J*. We approximately estimate μ and σ^2 using the length of the sentences in the bilingual example database. μ and σ^2 are 1.03 and 5.56, respectively.

Thus, the length selection probability h_{LSP} is calculated as in Eq. 19.

$$\begin{aligned} &h_{\text{LSP}}(\mathbf{e},\mathbf{f},\mathbf{F},\mathbf{Z}) \\ &= \begin{cases} \log p(I \mid J) & \text{If } \mathbf{e} \text{ is a sentence} \\ \log p_N(\frac{I}{J};\mu,\sigma^2) & \text{Else} \end{cases} \end{aligned} \qquad (19)$$

Search

To include more possible translation candidates, we extend the TSC forest by adding the homologous TSCs to the TSC forest.

We use a beam search method to find the final translation in the extended TSC forest in a bottom-up manner. The search algorithm is shown in Figure 2. For each TSC in the extended TSC forest, if the target language string

```
INPUT: The extended TSC forest $\overline{\mathbf{Z}}$ including $K$ homological TSC sets $\mathbf{H}_1^K$, where $\mathbf{H}_k$ $(1 \le k \le K)$ consists of
       $G_k$ TSCs $\bar{z}_1^{G_k}$.

Let $\mathbf{T}_k$ be the translation candidate set of $\mathbf{H}_k$ $(1 \le k \le K)$.
for each $\mathbf{H}_k \in \overline{\mathbf{Z}}$ do
    for each $\bar{z}_g \in \mathbf{H}_k$ do
        if $\bar{z}_g$ includes $B$ ($B$>0) substitution symbols $u_1^B$ then
            for each $u_b$ $(1 \le b \le B)$ do
                Find the translation candidate set $\overline{\mathbf{T}}_b$ corresponding to $u_b$.
            end for
            for each of Cartesian Product of $\tau_1^B$ of $\overline{\mathbf{T}}_1^B$ do
                Replace each $u_b$ $(1 \le b \le B)$ with $\tau_b$ to generate the translation.
                Add the translation to $\mathbf{T}_k$.
            end for
        else
            Add the target language string of $\bar{z}_g$ to $\mathbf{T}_k$.
        end if
    end for
    Rank $\mathbf{T}_k$ using the log-linear generation models and keep the $n$-best translation candidates.
end for

OUTPUT: The top-1 translation candidate of $\mathbf{H}_i$, which matches the root node of the input tree.
```

Figure 2. Search Algorithm of the Log-linear Generation Models

contains the substitution symbols, then each substitution symbol would be substituted with the translation candidates of the corresponding child TSCs. The generated translation candidates are ranked by the log-linear generation models. The *n*-best translation candidates are chosen and reused to produce the translations of the parent TSCs.

The best translation candidate of the homologous TSC set, which matches the root node of the input tree, is considered the final translation of the input sentence.

Figure 3 shows an example of search. The input sentence is "The older employees are the backbone of the industry." The TSC forest includes four TSCs:

TSC (a): matching Nodes 0~6;
TSC (b): matching Nodes 3, 7~9;
TSC (c): matching Nodes 6, 10~15;
TSC (d): matching Nodes 15~17.

TSC (b) and (d) do not include any substitution symbol. The target language strings of TSC (b) and (d) are considered the translations of the TSCs. TSC (c) contains one substitution symbol and the translation is generated by replacing the substitution symbol <NPB> with the translation candidates of TSC (d). In the same way, for TSC (a), the translation is obtained by replacing the substitution symbol <NPB> with the translation candidates of TSC (b) and replacing the substitution symbol <NP-A> with the translation candidates of TSC (c). During replacing the substitution node, the number of the translation candidates of the TSC exponentially increase with the number of the substitution nodes. Therefore, only the *n*-best translation candidates are chosen. Finally, the translation system outputs the best translation, "那些 老 雇员 是 行业 的 骨干 。".

Experiments

In order to evaluate the performance of the log-linear generation models, we develop an English-to-Chinese EBMT system based on TSC. Our system has two major differences from the EBMT system described in (Liu et al., 2006): first, we use the log-linear generation models to incorporate the knowledge sources; and second, more feature functions are introduced into our generation model. In our system, the weights of the feature functions are tuned on a development set using Powell's algorithm with a grid-based line optimization method (Press et al., 2002). We start from different initial parameter values to avoid finding a poor local optimum. The number of the translation candidates is set to 100 during the searching for the final translation.

The BLEU score (Papineni et al., 2002) is used to evaluate the translation quality. We calculate the 95% confidence intervals using the same method as described in (Zhang et al., 2004) for all experimental results.

Resources

Translation Examples: The translation examples include 262,060 English-Chinese bilingual sentence pairs collected from general language texts. The average length of the English sentences is 12.1 words and that of the Chinese sentences is 12.5 words (21.8 characters). The

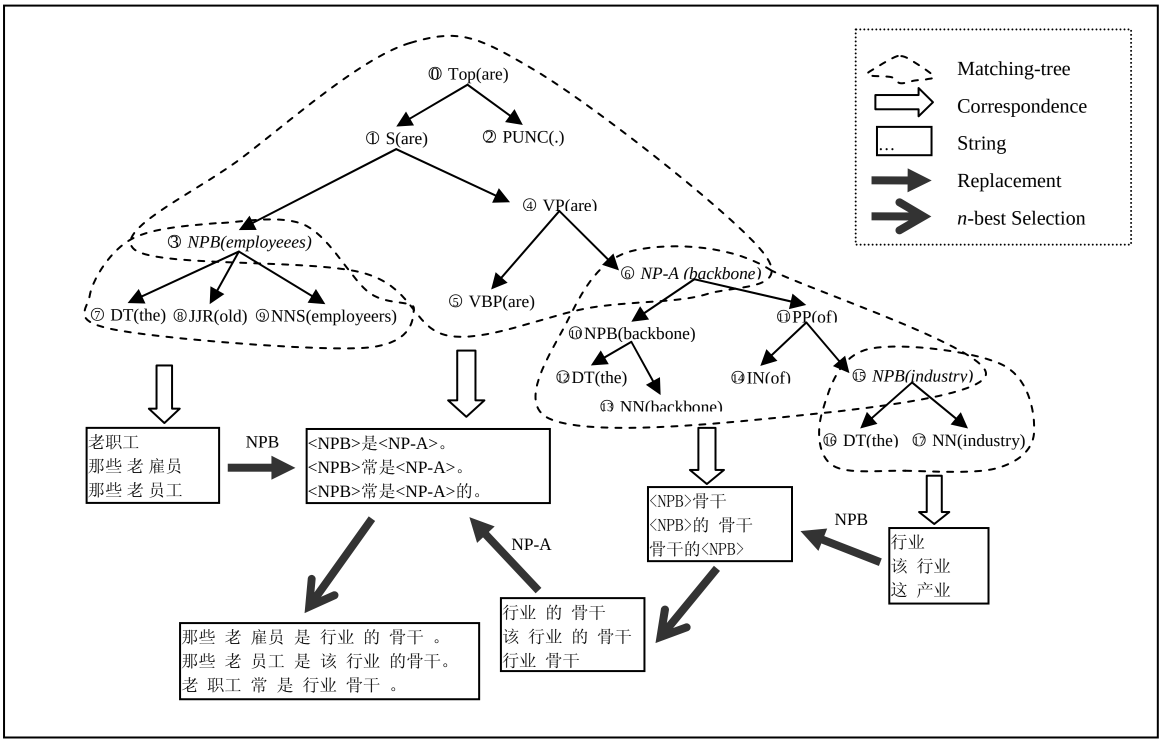

Figure 3. Example of Search

English words and Chinese words in the sentence pairs are automatically aligned using GIZA++ (Och & Ney, 2000).

Test Set & Development Set: The test set contains 400 English sentences and each sentence has two reference (human) translations. The development set contains 100 and each sentence has two references. Both of them are not included in the translation examples.

Translation Dictionary: A general English-Chinese translation dictionary is used to translate words that cannot be covered by the translation examples.

Language Model: The Chinese language model in our experiments is a word-based trigram model, which is trained using the SRILM toolkit (Stolcke, 2002) on a general Chinese corpus with 228 million words.

English Parser: The English sentence is parsed by the parser of Collins (1999). We use its model 3 and default settings. In the original result of the parser, the punctuation node always occurs as a right sibling of the previous leaf node. If so, the punctuation node cannot always act as a coordinating conjunction (Bikel, 2004). Thus, we change the position of the punctuation node in the tree. For the punctuation node *n*, if it is the most left/right leaf node, then we set the root node of the tree 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*.

Evaluation of Log-linear Generation Models

We conduct four experiments to investigate the performance of the log-linear generation models.

- **E1 (MS+WTP+LMP)** In the experiment, the log-linear generation models use three feature functions: matching score (MS), word translation probability (WTP), and language model probability (LMP), which are similar to the generation model in (Liu et al., 2006). It is used as the baseline.
- **E2 (E1+LSP)** Besides the feature functions used in E1, the length selection probability (LSP) is added to the log-linear generation models.
- **E3 (E2+SSP)** Besides the feature functions used in E2, we incorporate the target language string selection probability (SSP) into the log-linear generation models.
- **E4 (E3+CS)** Besides the feature functions used in E3, the context similarity (CS) is incorporated into the log-linear generation models.

The experimental results are shown in Table 1. The baseline using the three feature functions (MS, WTP, and LMP) achieves a BLEU score of 0.2219. By analyzing the result of E1, we find that the translations contain some redundant words such as the auxiliary words, which is partially caused by the geometric mean of the language model probability. This feature function prevents preference for a short translation. However, this feature function tends to select longer sentences.

In order to alleviate this problem, we add the length selection probability (LSP) to the log-linear generation models. The results show that the average length of the translations decreases by 16.2%. The EBMT system gets an increase of 0.0139 in BLEU score. The results indicate that the length selection probability is effective in improving the quality of the translation.

In E3, we add the target language string selection probability to the log-linear generation models. From the results, it can be seen that E3 outperforms E2, which is

Experiments	λ_{MS}	λ_{WTP}	λ_{LMP}	λ_{LSP}	λ_{SSP}	λ_{CS}	BLEU Score
E1 (MS+WTP+LMP)	0.556	0.166	0.278	-	-	-	0.2219 ± 0.0022
E2 (E1+LSP)	0.582	0.083	0.250	0.085	-	-	0.2358 ± 0.0024
E3 (E2+SSP)	0.629	0.037	0.222	0.074	0.038	-	0.2410 ± 0.0026
E4 (E3+CS)	0.115	0.086	0.384	0.192	0.096	0.127	0.2571 ± 0.0026

Table 1. Results of Log-linear Generation Models

due to the fact that the target language string probability effectively scores the reliability of the target language string of the translation example.

It can be seen from the result of E4 that the translation quality is improved with a BLEU score upgrade from 0.2410 to 0.2571, which indicates that the context similarity makes a contribution to the improvement of the translation selection.

From the analysis, it can be seen that the translation quality is relatively improved by 15.9% in BLEU score as compared with the baseline. Experimental results indicate that the log-linear generation models using various knowledge sources for EBMT are effective in improving the performance of translation generation.

Comparison with Phrase-based SMT

In (Groves & Way, 2005&2006; Way & Gough, 2005), performances of EBMT system and SMT system using the same corpora were discussed. We make a comparison between our EBMT system with six feature functions and the phrase-based SMT system, Pharaoh (Koehn, 2004). We use the default features of Pharaoh, including language model, reordering model, phrase translation table, and word penalty. We run the trainer with its default settings and then use Koehn's implementation of minimum-error-rate training (Och, 2003) to tune the feature weights of Pharaoh on our development set. The training data is the same as used in our system.

Table 2 shows the comparison results. Our system achieves a relative improvement of 17.2% over Pharaoh.

Systems	BLEU Score
Pharaoh	0.2193 ± 0.0031
Our System	0.2571 ± 0.0026

Table 2. Comparison Results between Phrase-based SMT System and Our System

Conclusions

This paper presented the log-linear generation models for EBMT. In the generation model, various knowledge sources are described as the feature functions and incorporated into the log-linear models. In this paper, we used six feature functions: matching score and context similarity, to estimate the similarity between the input sentence and the translation example; word translation probability and target language string selection probability, to estimate the reliability of the translation example; and language model probability and length selection probability, to estimate the quality of the translation.

We built an English-to-Chinese EBMT system with the proposed log-linear generation models. Experimental results show that our system achieves an absolute improvement of 0.0352 in BLEU score (15.9% relative) as compared with the baseline EBMT system.

We also compared our method with a phrase-based SMT system. Experimental results indicate that the EBMT system significantly outperforms Pharaoh, achieving an absolute improvement of 0.0378 in BLEU score (17.2% relative).

References

Akiba, Y., Watanabe, T., and Sumita, E. (2002). Using Language and Translation Models to Select the Best among Outputs from Multiple MT Systems. In Proceedings of the 19th International Conference on Computational Linguistics (pp. 8--14). Taipei, Taiwan.

Aramaki, E. & Kurohashi, S. (2004). Example-based Machine Translation Using Structural Translation Examples. In Proceedings of International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation (pp. 91--94). Kyoto, Japan.

Aramaki, E., Kurohashi, S., Kashioka, H., and Tanaka, H. (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 (pp. 57--64). Edmonton, Canada.

Badia, T., Boleda, G., Melero, M., and Oliver, A. (2005). An N-gram Approach to Exploiting a Monolingual Corpus for Machine Translation. In Proceedings of MT Summit X Workshop on Example-based Machine Translation (pp.1--7). Phuket, Thailand.

Berger, A.L., Della Pietra, S.A. and Della Pietra, V.J. (1996). A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics, 22(1):39--72.

Bikel, D.M. (2004). Intricacies in Collins' Parsing Model. Computational Linguistics, 30(4):479--511.

Brown, P.F., Lai, J.C., and Mercer, R.L. (1991). Aligning Sentences in Parallel Corpora. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (pp. 169--176). Berkley, CA.

Callison-Burch, C. & Flournoy, R.S. (2001). A Program for Automatically Selecting the Best Output from Multiple Machine Translation Engines. In Proceedings of MT Summit VIII (pp. 63--66). Santiago de Compostela, Spain.

Collins, M. (1999). Head-driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania.

Carl, M., Schmidt, P., and Schutz, J. (2005). Reversible Template-based Shake & Bake Generation. In Proceedings of MT Summit X Workshop on Example-based Machine Translation (pp. 17--25). Phuket, Thailand.

Fellbaum, C. (1998). Wordnet: an Electronic Lexical Database. MIT Press, Cambridge, MA.

Groves, D. & Way, A. (2005). Hybrid Example-based SMT: the Best of Both Worlds? In Proceedings of ACL-2005 Workshop on Building and Using Parallel Text (pp. 183--190). Michigan.

Groves, D. & Way, A. (2006). Hybridity in MT: Experiments on the Europarl Corpus. In Proceedings of the 11th Conference of the European Association for Machine Translation (pp. 115--124). Oslo, Norway.

Hearne, M. & Way, A. (2003). Seeing the Wood for the Trees: Data-Oriented Translation. In Proceedings of MT Summit IX (pp. 165--172). New Orleans, Louisiana.

Hearne, M. & Way, A. (2006). Disambiguation Strategies for Data-Oriented Translation. In Proceedings of the 11th Conference of the European Association for Machine Translation (pp. 59--68). Oslo, Norway.

Hutchins, J. (2005). Towards a Definition of Example-based Machine Translation. In Proceedings of MT Summit X Workshop on Example-based Machine Translation (pp. 63--70). Phuket, Thailand.

Imamura, K., Okuma, H., Watanabe, T., and Sumita, E. (2004). Example-based Machine Translation Based on Syntactic Transfer with Statistical Models. In Proceedings of the 20th International Conference on Computational Linguistics (pp. 99--105). Geneva, Switzerland.

Kaki, S., Yamada, S., and Sumita, E. (1999). Scoring Multiple Translations Using Character N-gram. In Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (pp. 298--302). Beijing, China.

Karov, Y. & Edelman, S. (1998). Similarity-based Word Sense Disambiguation. Computational Linguistics, 24(1):41--59.

Knight, K. & Hatzivassiloglou, V. (1995). Two-level, Many-paths Generation. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (pp. 252--260). Cambridge, Mass.

Koehn, P. (2004). Pharaoh: A Beam Search Decoder for Phrase-based Statistical Machine Translation Models. In Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (pp. 115--124). Washington, DC.

Lin, D. (1998). An Information-theoretic Definition of Similarity. In Proceedings of the 15th International Conference on Machine Learning (pp. 296--304). San Francisco, CA.

Liu, Z., Wang, H., and Wu H. (2006). Example-based Machine Translation Based on Tree-string Correspondence and Statistical Generation. Machine Translation, 20(1): 25--41.

Och, F.J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (pp. 160--167). Sapporo, Japan.

Och, F.J. & Ney, H. (2000). Improved Statistical Alignment Models. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (pp. 440--447). Hong Kong, China.

Och, F.J. & Ney, H. (2002). Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (pp. 295--302). Philadelphia, PA.

Och, F.J. & Ney, H. (2004). The Alignment Template Approach to Statistical Machine Translation. Computational Linguistics, 30(4):417--449.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (pp. 311--318). Philadelphia, PA.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. (2002). Numerical Recipes in C++. Cambridge University Press, Cambridge, UK.

Schafer, C. & Yarowsky, D. (2002). Inducing Translation Lexicons via Diverse Similarity Measures and Bridge Languages. In Proceedings of the 6th Conference on Natural Language Learning (pp. 1--7). Taipei, Taiwan.

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

Stolcke, A. (2002). SRILM - An Extensible Language Modeling Toolkit. In Proceedings of the International Conference on Spoken Language Processing (pp. 901--904). Denver, Colorado.

Way, A. & Gough, N. (2005). Comparing Example-based and Statistical Machine Translation. Natural Language Engineering, 11(3):295--309.

Zhang, Y., Vogel, S., and Waibel, A. (2004). Interpreting BLEU/NIST Scores: How Much Improvement Do We Need to Have a Better System? In Proceedings of the 4th International Conference on Language Resources and Evaluation (pp. 2051--2054). Lisbon, Portugal.