Utilizing Global Syntactic Tree Features for Phrase Reordering

Yeon-Soo LEE†, Hyoung-Gyu LEE†, Nonmembers, Hae-Chang RIM††, Member, and Young-Sook HWANG†††, Nonmember

SUMMARY  In phrase-based statistical machine translation, long-distance reordering problem is one of the most challenging issues when translating syntactically distant language pairs. In this paper, we propose a novel reordering model to solve this problem. In our model, reordering is affected by the overall structures of sentences such as listings, reduplications, and modifications as well as the relationships of adjacent phrases. To this end, we reflect global syntactic contexts including the parts that are not yet translated during the decoding process.

key words: phrase reordering model, global syntactic tree features, phrase-based statistical machine translation

1. Introduction

Since phrase-based statistical machine translation (PBSMT) is relatively simple, it can be easily applied to many language pairs. Also, its lexical translation coverage is very high because the translation unit can be any length of “phrase.” However, in recent years, PBSMT has been also applied to grammatically distant language pairs such as English-Korean, English-Japanese, or English-Turkish, thus making its reordering problem more prominent. Original PBSMT reordering model is based on the distribution of distance from a previously aligned phrase. However, the distance-based model has limits in terms of accurate reordering discrimination. Many studies have been conducted to improve the distance-based model; however, long sentences such as newspaper articles are still often not properly translated, mainly due to reordering failures. The phrases in long sentences should be reordered as complex types or long distance movements. If long distance reordering fails, translation performance barely changes even if several local reorderings succeed.

Long distance movement is caused by the differences between languages in the way they syntactically organize sentences such as subject-object-verb (SOV)/subject-verb-object (SVO), head-initial/head-final, and special grammatical conditions. Therefore, recent studies attempt to reflect syntax information in PBSMT reordering. First, there are preprocessing methods that change source words into a target sentence such as subject-object-verb (SOV) reordering model which uses syntactic information in PBSMT reordering. This model, reordering is affected by the overall structures of sentences such as listings, reduplications, and modifications as well as the relationships of adjacent phrases. To this end, we reflect global syntactic contexts including the parts that are not yet translated during the decoding process.

Second, post-processing methods are presented in [5]–[8]. It is difficult to integrate syntactic model into the PBSMT framework, since the syntactic boundary and phrase boundary of the PBSMT do not match. However, the methods have more potential to improve due to its ability to control the decoding process. The “Lexicalized reordering model” the most widely used method exploit lexical features between two phrases [5], [6]. In [7], they take advantage of syntactic cohesion characteristics. In recent years, reordering may be regarded as a classification problem and implemented as a discriminative model. In the [8]’s research, the source’s syntactic tree information is reflected in the word lattice in order to decide the visiting order of words. However, they still considered the reordering problem as a distance based movement of word and used the tree features insufficiently to resolve the long distance reordering problem. Nevertheless, since grammatical differences better explain the reordering phenomena, the performance was improved to a certain extent. Our method belongs to these soft-constraint approaches.

Using now, little attention has been given to questions such as “What range of syntactic context should be considered?” or “How is syntax and long distance reordering related?” If there is ambiguity in discriminating between local reordering and long distance reordering, then incorrect hypotheses remain in the hypothesis space even though syntactic information is used. If the incorrect hypothesis has a high language model score, finally translation errors are generated. It is worthwhile to examine the reordering problem more closely. In this paper, we focus on long distance reordering as well as local reordering. We refer to the ambiguity of using syntactic information, investigate global syntactic features, and propose a new reordering model to support these features.

Manuscript revised January 27, 2014.
†The authors are with the Dept. of Computer and Radio Communications Engineering, Korea University, Seoul, Korea.
††The author is with Platform R&D Office, SK Planet Co. Ltd., Seoul, Korea.
†††The author is with the Dept. of Computer and Radio Communications Engineering, Korea University.
a) E-mail: rim@nlp.korea.ac.kr (Corresponding author)
b) E-mail: youngsook.hwang@sk.com
DOI: 10.1587/transinf.E97.D.1694

Copyright © 2014 The Institute of Electronics, Information and Communication Engineers
2. Syntactically Featured Phrase Reordering Model

2.1 Long Distance Reordering & Syntax

First, we introduce an English-Korean translation example with complex reordering in Fig. 1. $S_1$ and $S_2$ are two of the several hypotheses in the decoding process. The decoder segments the source sentence $f$ into phrases $f_1, f_2, f_3, \ldots$, and reorders them. Hypothesis $S_1$ is correct but $S_2$ is not. $S_2$ seems to be correct in terms of lexical translation; however, the meaning is completely different from the original meaning because it is translated in the wrong order. It is translated as, “The government approves it after the UA officially sign the national assembly next month.” Most of the previous studies analyzed a junction point in the parse tree between a previous phrase and a current phrase. For example, when $S_1$ translates $f_5$ after $f_1$, the junction point on the parse tree is circle 1 the treelet $(S_1, NP_1, VP_1)$. Furthermore, at selection (hereafter referred to as “state”) 3 of $S_1$, $f_7$ meets $f_5$ at circle 3. NP3 precedes VP3 under $S_2$ at the target sentence. At state 3 of $S_2$, $f_5$ meets $f_2$ at circle 2. Under VP2, NP2 is translated before the phrase that includes VB. If the SOV characteristic of the target language is considered, all the above interpretations are correct. In other words, during the first three states, if we consider the local space (the junction points) on the parse tree, we cannot distinguish correct long distance movement from incorrect local reordering. Syntactic knowledge states, “The modification clause should be translated before the main verb and the subject is the first in that clause according to the target syntax.” If this is reflected, $S_2$ receives a very low score and disappears early from the hypothesis stack. However, this information is scattered on the parts that are not yet translated but related to the current phase and is based on the structural complexity of the sentence. To eliminate the $S_2$ in a limited search space, the information should be detected and reflected early.

2.2 Discriminative Phrase Precedence Model

Before we use the parse tree context, we first define a discriminative reordering model that estimates the precedence of phrases. The source sentence $f$ consists of $N$ phrases $\langle f_1, f_2, \ldots, f_N \rangle$ and the reordered sequence at target is $O = \langle o_1, o_2, \ldots, o_N \rangle$. Then $o_j$ represents the $j$th translated source phrase. The source phrases can be divided into three groups when $o_j$ is selected at the $j$th state. $P_j$ is a set of previously translated phrases, $C_j$ is a current phrase, and $L_j$ is a set of phrases that have not yet been translated. For example, in Fig. 1, when $f_7$ is translated after $f_1$ and $f_5$, $P_j$ is $\{f_1, f_5\}$, $C_j$ is $\{f_7\}$ and $L_j$ is $\{f_2, f_3, f_4, f_6\}$. Our model estimates the probability of the precedence relationship at state $j$, given the source tree $T_f$.

$$p(C_j, L_j | T_f, P_j),$$

$$P_j = \{o_{j_1}, o_{j_2}, \ldots, o_{j_m}\}, C_j = \{o_{j_0}\}, L_j = \{o_{j_n} | j < m \leq N\}$$

(1)

The sentence level reordering probability is represented by Eq. (2) and it can be rewritten as Eq. (3) in maximum entropy style.

$$\max_{j=1}^{N} \prod_{j=1}^{N} p(C_j, L_j | T_f, P_j)$$

$$= \max_{j=1}^{N} \frac{1}{Z(T_f, P_j)} \exp \left( \sum_k \lambda_k \phi_k(P_j, C_j, L_j, T_f) \right)$$

(3)
Table 1 Feature category.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>local</td>
<td>$T_{pc}$ - the relationship between the previously translated part and the current part</td>
</tr>
<tr>
<td>uncovered</td>
<td>$T_{cl}$ - the relationship between the current translated part and the part that will be translated in the future</td>
</tr>
</tbody>
</table>
| global   | $T_{long}$ - the longest path in the parse tree  
|          | $T_{wide}$ - The widest siblings  
|          | $T_{redup}$ - The nested sentence structure |

Here, $\phi_k$ is the binary valued feature function, $\lambda_k$ is the weight of the $\phi_k$, and $Z(T_f, P_j)$ represents the normalization factor.

2.3 Syntactic Mapping

In order to analyze the phrase precedence relationship in terms of syntax, the relationship is mapped to the source parse tree. We can attach one of the following three tags to each node in the tree: $p$, $c$, or $l$.

- $p$: all the words dominated by the node are translated
- $c$: one of the words dominated by the node belongs to $C_j$
- $l$: one of the words dominated by the node belongs to $L_j$

The tag indicates whether each node is associated with any of the three sets that were mentioned above: $P_j$, $C_j$, and $L_j$. Figure 1 shows an example of the state 3 of $S_1$. For example, tag $p$ is attached to NP1 by which all the dominated nodes are translated and all the ancestors of the current phrase “the month” have tag $c$. Also, tag $l$ is attached to all the ancestors of phrases that are not translated. We extract the sub-trees that represent the global context, and they are classified in Table 1. In Fig. 1, when $f_j$ is selected and translated, B, C and D represent the sub-tree features $T_{pc}$, $T_{cl}$, and $T_{pl}$, respectively. Also shown in Fig. 1, the global sub-tree features $T_{long}$, $T_{wide}$ and $T_{redup}$ are represented by E, F, and G respectively. $T_{long}$ features indicate which part of the longest path in the parse tree is currently translated. For the $T_{long}$ and $T_{wide}$ features, if a parent node has any child node attached to tag $p$ or $c$, the parent node is also attached to tag $p$ or $c$ respectively. To extract $T_{wide}$, we flatten the source parse tree; in other words, we integrate the parent and grand-parent that have the same head word. On the flattened tree, the parent-child treetlet with the most siblings is $T_{wide}$. The $T_{redup}$ features indicate the nested structure and the current part in the structure.

In order to use the sub-trees as the maximum entropy features, the trees are divided into parent-child treetlets. Also we assume independence between the treetlets. The above sub-tree features can be represented as a set of treetlet $t$:

$$T_{u,j} = \begin{cases} \{t | t = (u, pr, c_1, c_2, tag_{c_1}, tag_{c_2}, w)\} & \text{if } u \in \{pc, cl, pl\} \\ \{t | t = (u, n_1, n_2, \ldots , tag_{n_1}, tag_{n_2}, \ldots )\} & \text{if } u \in \{long, wide, redup\} \end{cases}$$

where $u$ is one of the sub-tree types. $pr$, $c_1$, and $c_2$ are parent, left child, and right child, respectively. The tag represents one of the following tags (described above): $p$, $c$, and $l$. The number of words dominated by the parent is represented by $w$. We differentiate the impact on the reordering according to the number of dominating words, even if the sub-tree shape is the same. $n$ is a node in the parse tree. In practice, to reduce the number of features and to avoid over-fitting, similar kinds of phrase tags are grouped and the weight is divided into three types: heavy, middle, and light.

As a result, Eq. (3) is rewritten as follows:

$$\max \prod_{j=1}^{n} \frac{1}{Z(T_f, P_j)} \exp \left( \sum_u \sum_{t \in T_u} \lambda_k \phi_k(P_j, C_j, L_j, T_f) \right)$$

Table 2 shows the sub-tree features in the $t$ format. The “correct” column shows the extracted features at the state 3 of $S_1$, and the “incorrect” column shows the extracted features at the state 3 of $S_2$. In the incorrect case, we know through the trees $T_{cl}$ and $T_{pl}$ that the modified clause “will officially sign the UA” is translated before the modifying clause “after . . .”; however, this order is wrong. Moreover, long distance reordering is needed because there are a large number of words in the modifying clause.

To train the above model, we must construct the correct positive and negative examples for the phrase reordering. Since this requires large costs, we construct a pseudo-answer set in the following way: First, we run the word alignment using GIZA++. From the word aligned parallel sentence, the phrases which are consecutively aligned words, can be $C_j$. With $C_j$ as the midpoint, $P_j$ is the part that is aligned forward than $C_j$ and $L_j$ is the part that is aligned backward. The negative example is generated by exchanging the $P_j$ and the $L_j$ for the same $C_j$. This method may not be accurate, but it has the advantage of generating a large amount of the training set. In this way, the positive set and the negative set each have an amount of 100,000 examples.

\(^1\)To measure the accuracy of the pseudo-answer set, we sampled 100 examples from the training set (which contained 100,000 examples). For each reordering example, the correctness was evaluated manually. As a result, the accuracy of the positive set is 82% and the accuracy of the negative set is 87%.
3. Experiments

We tested our method on three different language pairs: English-to-Korean(E2K), Chinese-to-Korean(C2K), and English-to-Chinese(E2C). The first two language pairs are SVO-SOV pairs which require a great deal of long distance movements. The last pair is an SVO-SVO pair with the same word order; however, it also requires long distance reorderings. The corpus\(^1\) for training and testing is shown in Table 3. For a language model, we used the SRI Language Modeling Toolkit to train a 5-gram language model on all the training (target) sentences. For parsing the input source sentences, we used the Stanford parser for English and Chinese. We used the SMT system, Moses[9], with default options for the baseline and implemented the proposed model by modifying the decoding module. BLEU [10] metric is used to measure the translation performance.

We measured the reordering accuracy using the following steps. First, we manually constructed word aligned data from 400 parallel sentences. Second, we automatically generated the positive and negative reordering hypothesis from the data in the same way we constructed the training set. Then we randomly selected 1,500 examples at three times from the total of 24,957 examples. We measured the accuracy of the three hypothesis sets by using the previously trained model and averaged the results. The reordering accuracy was evaluated only in the English-Korean set. Table 4 shows the results of distortion distance variation and the distance represents the minimum word level distance at target sentence between the previously aligned phrase and the current phrase. The baseline results “\(T_{pc}\) only” is produced by using only the precedence relationship with previously translated phrases. We can see that for long distance reordering, it is important to consider the uncovered part in advance. Although \(T_{long}\), \(T_{wide}\), and \(T_{redup}\) do not affect the short movement, they contribute to the accuracy of the long movements. Also, this result shows that considering only the relationship between the adjacent phrases (e.g., “\(T_{pc}\) only”), in the long distortion set, is insufficient.

Next, we evaluate the translation performance. The “Lexicalized” model uses lexical features. On the other hand, the “HPMT (Hierarchical reordering model)”, a basic syntax-based model, is known for its good performance on syntactically different language pairs. Table 5 shows the results of the comparative experiment on sentence length and features in English-Korean translation. These results show that the proposed features are effective, especially for the long sentence translation. Since the distribution of the NIST12 set is uneven, we do not conduct the length variation experiment; however, we can see that the overall performance is improved by using our proposed global features. Table 6 shows the performance of translations of language pairs. In the case of English-Korean and Chinese-Korean pairs, the proposed method performed better than HPMT. As for the English-Chinese pair, the performance of our method is comparable to that of HPMT.

4. Conclusion

We explore more discriminative features by analyzing long distance reordering in the context of global structures. Also, we propose a new reordering model and a learning method to reflect the global structure context. Through this work, we found that to solve long distance reordering problems in structurally different language pairs such as English-Korean, it is necessary to reflect the high level context of syntax. In our future work, we will use the information of target sentences and integrate it into our model.

Acknowledgements

This research was supported by Basic Science Research Program and Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Plannig (No.2011-0016878, No.2012M3C4A7033344).

References

[1] F. Xia and M. McCord, “Improving a statistical mt system with au-

---

\(^1\)They are provided by SK Planet Co., Ltd. (http://www.skplanet.com/) only for research purposes. They are crawled over the various Korean on-line news sites and refined semi-automatically.


