IJICTR International Journal of Information & Communication Technology Research

Volume 6- Number 4- Autumn 2014 (57-67)

Using Dependency Tree Grammar to Enhance the Reordering Model of Statistical Machine Translation Systems

Zahra Rahimi
Human Language Technology Lab
Amirkabir University of Technology
Tehran, Iran
zah-ra@aut.ac.ir

Shahram Khadivi
Human Language Technology Lab
Amirkabir University of Technology
Tehran, Iran
khadivi@aut.ac.ir

Heshaam Faili
Electrical and Computer Engineering Department
Tehran University
Tehran, Iran
hfaili@ut.ac.ir

Received: April 15, 2014-Accepted: October 16, 2014

***Abstract*—We propose three novel reordering models for statistical machine translation. These reordering models use dependency tree to improve the translation quality. All reordering models are utilized as features in a log linear framework and therefore guide the decoder to make better decisions about reordering. These reordering models are tested on two English-Persian parallel corpora with different statistics and domains. The BLEU score is improved by 2.5 on the first corpus and by 1.2 on the other.**

Keywords-statistical machine translation; reordering model; dependency tree; discriminative reordering model; discriminative decoder; long range reordering; maximum entropy; distortion model

I. INTRODUCTION

Generating a fluent translation is one of the main goals for statistical machine translation (SMT) systems. One of the main challenges in this regard is the differences between the word orders of two different languages. In addition, trying all possible word orders during the translation phase is an NP-complete problem [1].

In recent years several methods have been introduced to solve reordering problem. The basic reordering model that is used in many SMT systems is distance based distortion model [2]. This distortion model is simple and for languages with similar syntactic structures works well but it is context independent and for languages with significantly different syntactic structure is not appropriate.

Lexical reordering models [3,4,5,6] are context dependent reordering models which are extensively used in many translation tasks. These models have shown their superiority in practice. These models are appropriate for flat word surface structures, but they do not consider the syntactic structure in estimating the probability of reordering events.

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

Other approaches are discriminative reordering models [7, 5]. In these models many feature functions are defined. Weights for these features are found by maximum entropy principle and these weights are applied to improve machine translation quality.

Hierarchical model [8] assumes hierarchical structure for phrases. This model achieves good results compared to phrase based model. This method uses SCFG[1] grammars but is trained on bilingual corpora without any syntactic information. This method has a good performance for short and medium range reordering events but its performance for long range reordering events is not so good. Some researchers [9, 10, 11] proposed long reordering rules for this model.

Another approach uses syntax to improve translation quality. Syntax based reordering model can be viewed as a preprocessing method or as some search space constraints. The goal of preprocessing methods is to reorder the source sentences before the training step and to construct sentences that are similar to the target sentence structure. In these methods, rule extraction can be automatic [12, 13, 14] or can be derived by hand [15, 16]. Decoder constraints, limit the search space by using syntactic rules. Some researchers [17, 18] have proposed a cohesive constraint for a decoder by using non-syntactic phrases and translate sentence in an order that represents the dependency tree structure.

In another method [19] a novel reordering model have been proposed that employs source side dependency tree movements and constraints to generates a statistical distribution of sub-tree to sub-tree transition in training data and helps the decoder to make better decisions.

Another method [20] uses sequence labeling for solving the reordering problem. In this method nine tags have been defined. These tags are derived from word alignment structure. All sentences of training corpus are labeled with these tags. Then, a sequence labeling method is trained on this labeled data. Afterwards, test and development corpora are labeled by the trained sequence labeling model. In the decoding step, all words of current translation option are labeled, and these labels are compared with primary labels and according to differences between these two labels, decoder assigns a score to each translation option.

In this paper we introduce three novel reordering models. All of these methods use dependency tree to improve the translation quality. In the first method, discriminative approach is used and novel features are extracted from training data to improve the reordering decision by the decoder. In the second method, new orientation type with the help of source side dependency tree information is proposed. In the last method, in addition to extracting information from source dependency tree, target dependency tree information is also used. This paper is structured as follows:

In Section 2, discriminative decoder is explained. In Section 3, first reordering model (discriminative reordering model) is described. Second approach is explained in Section 4. Last approach is described in Section 5. Experimental results are described in Section 6.

II. Discriminative Decoder

In statistical machine translation, translation system gets a source sentence $f_1^J = f_1,...,f_J$ and generates a target sentence $e_1^I = e_1,...,e_I$ which is the most probable target sentence among all possible target sentences.

$$\hat{e}_1^I = \arg\max_{I,e_1^I}\{pr(e_1^I \mid f_1^J)\} \quad (1)$$

Posterior probability in Equation 1 is modeled using log linear framework:

$$pr(e_1^I \mid f_1^J) = \frac{\exp(\sum_{m=1}^{M}\lambda_m h_m(e_1^I, f_1^J))}{\sum_{I',e'^{I'}_1}\exp(\sum_{m=1}^{M}\lambda_m h_m(e'^{I'}_1, f_1^J))} \quad (2)$$

The denominator is independent of e_1^I so we can omit it therefore the decision rule with log linear frame work is as follows where h_m is mth feature function and λ_m is its weight:

$$\hat{e}_1^I = \arg\max_{I,e_1^I}\{\exp(\sum_{m=1}^{M}\lambda_m h_m(e_1^I, f_1^J))\} \quad (3)$$

One of these feature functions is the reordering model.

III. Discriminative Reordering model

A. Class Definition

This model should be able to predict the movement type of a phrase with respect to the previous phrases. In this method we use six classes of movements. These classes are combination of lexicalized reordering movement events and source dependency tree movement events.

Lexicalized reordering model [4] is learned based on the orientation type of current phrase with respect to previous phrase. This model checks the previous and next cells of current phrase in the alignment matrix and determines the orientation type of the current phrase with respect to previous phrase. These orientation types could be monotone (m), swap(s) or discontinuous (d). This model is parameterized as follows:

$$P(O|e, f) = \prod_{i=1}^{n} p(o_i|\bar{e}_i, \bar{f}_{a_i}, a_{i-1}, a_i) \quad (4)$$

[1] Synchronous Context-Free Grammar

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

Where e is the target sentence and f is the source sentence. $\bar{e}_i$ is the I_{th} target phrase and $a=(a_1,\ldots,a_n)$ is the set of phrase alignment. $\bar{f}_{a_i}$ is the source phrase that aligns with $\bar{e}_i$, O is the orientation sequence of o_i , $o_i \in \{m,s,d\}$ and defined in Equation 5.

$$o_i = \begin{cases} m & if\ a_i - a_{i-1} = 1 \\ s & if\ a_i - a_{i-1} = -1 \\ d & if\ |a_i - a_{i-1}| \neq -1 \end{cases} \quad (5)$$

Lexicalized reordering model is a good model for flat word surface structure [19] but this model does not consider the syntactic information. We use a source dependency tree the same as [19]. We extract two movement events from source side dependency tree.

Inside and outside movement events are obtained from sub-tree movements in dependency tree.

Assume T is a dependency tree and T(n) is a sub-tree rooted at node n. Each source phrase $\bar{f}$ which is used for constructing the current hypothesis state is a span. Each source span has a dependency structure s_h.

An open sub-tree is a sub-tree that its translation has been begun but not yet completed. A completed sub-tree is a sub-tree that has been translated. If a phrase $\bar{f}$ helps T(n) to be completed, $\bar{f}$ moves inside(I) of T(n) but, if we leave T(n) for translating $\bar{f}$ while its translation has not yet been completed, $\bar{f}$ moves outside of T(n). Inside and outside movements for source side of parallel sentence of TABLE I illustrated in Figure 1.

In Figure 1 translation of phrase [hamin/7,hala/8; 12/already] after [dar/9, janvie/10 ;10/in,11/January] is an outside movements because the sub-tree with root "Dar" is leaved to translate phrase [7,8;12] while the translation of this sub-tree have not yet been completed and phrase [7,8;12] is in another sub-tree(sub-tree with root "yeki").

Outside and inside movement classes combine with three lexical reordering classes, monotone, swap and discontinuous so, six movement classes are generated. These movement classes are as follows:

$$o-d = \{m-i, s-i, d-i, m-O, s-O, d-O\}$$

Where o represents the lexical reordering movements and d shows the source dependency tree movements. The reordering model is as Equation 6 where s_i, s_{i-1} are dependency structures of $\bar{f}_{a_i}$ and $\bar{f}_{a_{i-1}}$. D is a random variable that shows the sequence of $o-d$ events.

$$P(D|e,f) = \prod_{i=1}^{n} p((o-d)_i \mid \bar{e}_i, \bar{f}_{a_i}, a_{i-1}, a_i, s_{i-1}, s_i) \quad (6).$$

TABLE 1. A PERSIAN-ENGLISH SENTENCE

Source side (Persian)	"Nazarat darbareye bargozarie yeki az anha hamin hala dar janvie va tayeen digari baraye fevrie chist".
Target side (English)	How do you feel about holding one of them in January already and fixing the other one for February.

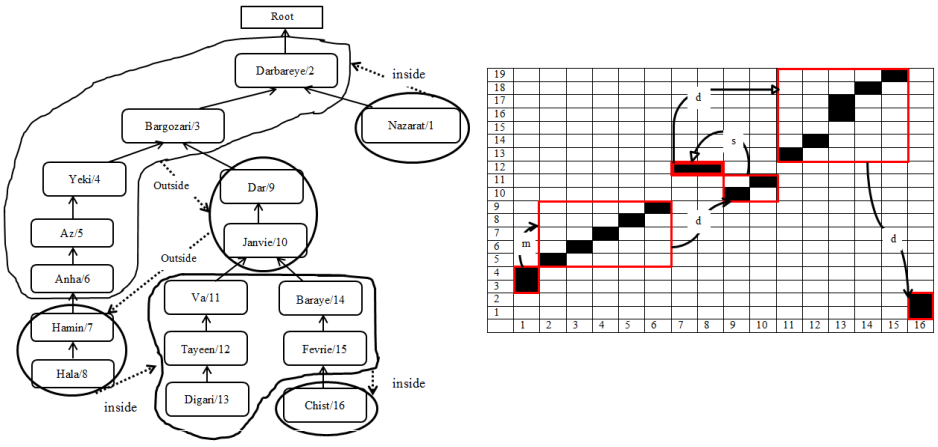

Fig. 1. The left figure is dependency tree for source side of bilingual sentence in Table I and thru right figure is alignment matrix for this bilingual sentence. Dotted lines in left figure show movement of each phrase with respect to previous phrase in dependency tree that can be inside or outside. Phrases specify by red rectangles in alignment matrix. "m", "s" and "d" show lexicalized reordering movements for each phrase with respect to previous phrase. in this figure the combination of lexicalized reordering movement events and source dependency tree movement events for each phrase are {i-m, O-d, O-s, i-d, i-d}.

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

B.Training

We have to extract dependency tree for each sentence in the source side of the corpus. The feature functions are extracted after phrase extraction. To find inside or outside movement event for each phrase with respect to previous phrases, the interruption check algorithm [17] is used. This algorithm investigate that sub-trees of previously extracted phrases are completed or opened, if this algorithm return true then d=i else d=O.

Monotone, swap and discontinuous events are found according to Equation 5. To train the model parameters λ_1^N (feature weights), the GIS [2] [21] algorithm is used. The optimization criterion is convex thus there is only one optimum and the convergence problem does not occur. To avoid overfitting in finding weights we use smoothing with Gaussian prior distribution. This method of smoothing tries to avoid assigning very large weights to some features. Feature weights are the outputs of this step. These weights are going to use in the decoding step.

We use maximum entropy principle for computing $p((o-d)_i \mid \bar{e}_i, \bar{f}_{a_i}, a_{i-1}, a_i, s_{i-1}, s_i)$. The formulation of this method is as follows:

$$p_{\lambda_1^N}(c_i|\bar{e}_i, \bar{f}_{a_i}, a_{i-1}, a_i, s_{i-1}, s_i) = \frac{\exp\left(\sum_{n=1}^{N} \lambda_n h_n(\bar{e}_i, \bar{f}_{a_i}, c)\right)}{\sum_{c'} \exp \sum_{n=1}^{N} \lambda_n h_n(\bar{e}_i, \bar{f}_{a_i}, c)} \quad (7)$$

Where λ_n is the weight of nth feature function (h_n) and c_i is the *ith* movement class.

C. Feature definition

Three types of Feature functions are used in this method:

- Word features
- Phrase number feature
- Orientation memory feature

D. Word Features

These features are like features proposed in [7]. These features depend on the alignment link (j,i) of a phrase where i is the first target position of current phrase and j is the position of source that aligns with i. j' is the source position that aligns with the last target position of previous phrase. These features are as follows:

1. Source words within a window l around position j:

$$h_{f,l,c}\left(f_1^J, e_1^I, i, j, j', s_k, s_{k-1}\right) = \delta\left(f_{j+l}, f\right)\delta\left(c, c_{j,j',s_k,s_{k-1}}\right) \quad (8)$$

2. Target words within a window l around position k:

$$h_{e,l,c}\left(f_1^J, e_1^I, i, j, j', s_k, s_{k-1}\right) = \delta\left(e_{i+l}, e\right)\delta\left(c, c_{j,j',s_k,s_{k-1}}\right) \quad (9)$$

Where $\delta(.,.)$ is the Kronecker-function. In the experiments, we will use $l \in \{-1,0,1\}$ whose values represent position before, same and after of current position respectively. s_k is the dependency structure of phrase k (current phrase) and s_{k-1} is the dependency structure of previous phrase.

E. Phrase Number feature

Phrase number is the rank of a phrase in a phrase sequence that is the complete translation. We extract phrase numbers from reordering graph [22]. In reordering graph each node is a bilingual phrase and each edge is the orientation type of current phrase with respect to the previous phrase. An example of such a graph is shown in Figure 2. Phrase number for node N is equal to the length of path from start node to node N. In such a graph many paths from start node to node N may exist therefore many phrase numbers for each phrase could be obtained. This feature is defined as follows:

$$h_{phr\#,\bar{f}_{a_k},\bar{e}_k,c}(f_1^J, e_1^I, \bar{e}_k, \bar{f}_{a_k}, a_k, s_k, s_{k-1}, i, j, j', k) = \delta(phr\#_{\bar{f}_{a_k},\bar{e}_k}, phr\#)\,\delta\left(c, c_{j,j',s_k,s_{k-1}}\right) \quad (10)$$

The aim of designing this feature is to help capturing long distance reordering events. This feature may be useful for some language pairs like Persian with SOV structure and English with SVO structure, in which long distance reordering events are usual. In Persian the verb of the sentence comes at the end but in English verb almost comes at first (after subject). When the verb is translated from Persian to English, a long distance reordering occurs. In this situation, the phrase number of a phrase that contains a verb is less than the phrase number of a phrase that contains subject of the sentence. Therefore phrase number could be a good feature that controls the order of phrase in a translation. This can force the decoder to have different biases for each phrase number. As shown in Figure 2. Phrase number for phrase [6, 6; 2, 2] that contains verb "raftam/go" is 2 but for phrase [3, 3; 6, 7] that is "dustam/ my friend" is bigger than 3.

F. Orientation memory feature

This feature's design is inspired by [20] work where they use sequence labeling concept in their reordering model. Here, this feature is designed to model dependency of the current phrase orientation to the orientation of the previous phrase. Indeed this feature is an orientation memory for current phrase. This feature is simply a reordering n-gram, i.e., the orientation of the current phrase is estimated according to the previous n-1 phrase orientations. Due to the sparsity problem, we set n to two, i.e., a bigram phrase orientation feature.

The orientation memory feature is defined as follows:

$$h_{O,\bar{f}_{a_i,c}}(\bar{e}_k, \bar{f}_{a_k}, \bar{e}_{k-1}, \bar{f}_{a_{k-1}}, s_k, s_{k-1}, i, j, j', k) = \delta(O_{\bar{f}_{a_k},\bar{f}_{a_{k-1}}}, O)\,\delta\left(c, c_{j,j',s_k,s_{k-1}}\right) \quad (11)$$

[2] Generalized Iterative Scaling Algorithm

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

The orientation with respect to previous phrases is left or right. If the corresponding source word position of the first target word in the current phrase is larger than the corresponding source word position of the last target word in the previous phrase, then the orientation is right; otherwise the orientation is left. The left and right orientation can be defined as follows:

$$o = \begin{cases} left & if\ j' < j \\ right & if\ j' > j \end{cases} \qquad (12)$$

Where j is source position that is aligned with first target position of current phrase and j' is source position that is aligned with last target position of previous phrase. To implement this feature, we use reordering graph. In this situation the labels of edges in this graph are left (L) or right (R).

n example of this graph with alignment matrix of Figure 2 is depicted in Figure 3. Phrase [2,2; 5,5] is in the left side of [5,5; 4,4] and phrase [5,5; 4,4] is in the right side of [4,4; 3,3].

G. Decoding

We add this new reordering model as a feature to Moses decoder, when the decoder finds the orientation of current phrase to be for example monotone then two hypotheses, one with reordering type "m-i" and one with reordering type "m-O" are expanded. Where "I" is inside movement event and O is outside movement event and M is monotone movement event.

In the decoding step, all features that have been mentioned in training section are extracted for each translation option. After that, we can compute the score of reordering by Equation 3.

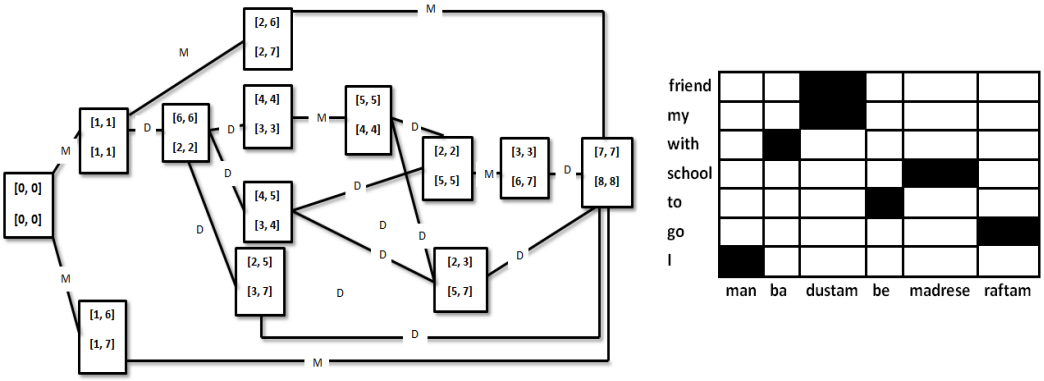

Fig. 2. each rectangle is a bilingual phrase and each edge is the orientation of each phrase with respect to previous phrase, phrase number for each phrase is the length of path from start node of the graph to that node for example for phrase [2,2;5,5]

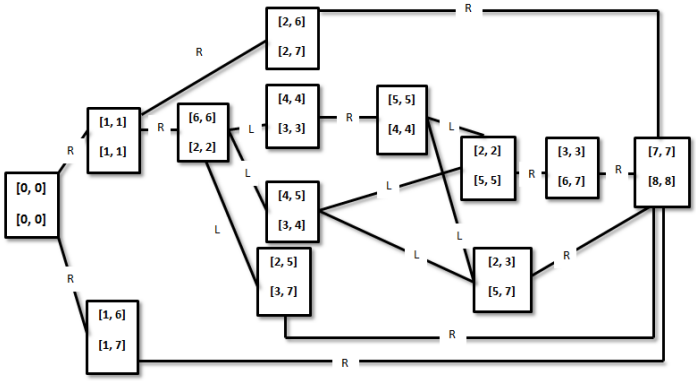

Fig. 3. Reordering Graph for Orientation feature. "L" is left and "R" is right.

IV. Using Parent Nodes

Persian is a SOV language but English is a SVO one. In Persian head of a syntactic phrase places after its modifiers, except noun and prepositional phrases, but in English, head places before its modifiers. In this section the effect of movement of a word with respect to its head in the dependency tree is investigated.

This movement is based on words but in this paper phrase based translation is used thus a heuristic is applied to adapt these movements with phrase based translation. In this heuristic, one word of a source phrase is selected as its indicator. Indicator of each phrase is a word that is closer to source dependency tree root. Indicator is named head word of phrase.

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

A. Movement Events

First the head word of current phrase is found. Next the direction of this phrase with respect to its parent in the dependency tree is found. This direction can be reversed (r) or Monotone (m) and is defined as follows:

$$\begin{cases} r & if\ (j_{child} - j_{parent})(i_{child} - i_{parent}) < 0 \\ m & otherwise \end{cases} \quad (13)$$

These directions are illustrated in Figure 4.

These two movement events are combined with lexical reordering movement events and six movement events are obtained:

$$o-d = \{m-m, s-m, d-m, m-r, s-r, d-r\}$$

Movement event of current phrase with respect to previous phrase is one of these 6 events.
This procedure is done for each phrase that is extracted.

In the next step of training, probability of belonging to any of these movement events for each phrase is computed by maximum likelihood principle in three following manners:

Do:

$$p(D|e,f) = \prod_{i=1}^{n} p((o-d)_i|\bar{e}_i, \bar{f}_{a_i}, a_i, a_{i-1}, s_i, s_{i-1}) = \frac{count(o_j - d_k) + \gamma}{\sum_k \sum_j (count(o_j - d_k) + \gamma)} \quad (14)$$

DOO:

$$p(D|e,f) = \prod_{i=1}^{n} p((o-d)_i|\bar{e}_i, \bar{f}_{a_i}, a_i, a_{i-1}, s_i, s_{i-1}) = \frac{count(o_j - d_k) + \gamma}{\sum_j (count(o_j - d_k) + \gamma)} \quad (15)$$

DOD:

$$p(D|e,f) = \prod_{i=1}^{n} p((o-d)_i|\bar{e}_i, \bar{f}_{a_i}, a_i, a_{i-1}, s_i, s_{i-1}) = \frac{count(o_j - d_k) + \gamma}{\sum_k (count(o_j - d_k) + \gamma)} \quad (16)$$

The pseudo code of this approach is shown in TABLE II.

B. Decodig

We add this new reordering model as a feature to Moses decoder. The decoder finds the orientation of lexical reordering. After that two hypotheses are expanded, for example if the orientation type of lexical reordering is monotone (m) then two hypotheses are expanded, one with orientation type m_m and one with reordering type m_r. r is reverse movement event and m is monotone movement event.

V. TARGET DEPENDENCY TREE

In this section we investigate the effect of using target dependency tree information to improving the reordering of machine translation system output. In order to use of target dependency tree information, two movement events are defined. These two movement events combine with two movement events that are extracted from source side dependency tree (Inside (i) and outside (o)) so four movement events are generated.
These movements are extracted from training data and the probability of each movement is computed by maximum likelihood criterion.

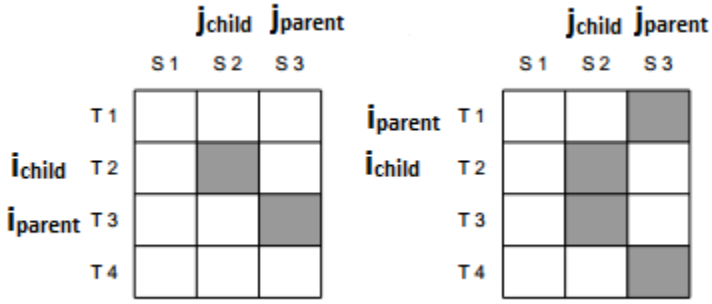

Fig. 4. The left figure shows a monotone movement event because $j_{child} < j_{parent}$ and $i_{chiled} < i_{parent}$. The right figure shows a reverse movement event because $j_{child} < j_{parent}$ and $i_{chiled} > i_{parent}$.

A.Movement Events

In this reordering model 4 movement events are used. These movements are as follows:

$$o-d = \{r-o, r-i, m-o, m-i\}$$

r and m are extracted of target side dependency tree and o and I are extract of source side dependency tree. These movements combine with each other and four movement events are generated as mentioned above.
The movements that are extracted from target side dependency tree:

The head of current target phrase and the head of its previous phrase are determined by target side dependency tree. The head of current target phrase is named $h_{current}$ and head of previous phrase is named h_{prev}. Next, parent of $h_{current}$ and h_{prev} are extracted from target side dependency tree and are named respectively parent ($h_{current}$) and parent (h_{prev}).

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

The orientation of current phrase with respect to previous phrase is defined as equation 17.

$$\begin{cases} r & parent(h_{current}) < parent(h_{prev}) \\ m & otherwise \end{cases} \quad (17)$$

B. Training:

After generating dependency tree for each sentence in the source and target sides of the corpus and extracting phrases, movement events for each phrase with respect to previous phrases are found. Next the probabilities of belonging to four movement events are estimated for each phrase. These probabilities are computed at three manners as follows:

Do:

$$p(D|e,f) = \prod_{i=1}^{n} p((o-d)|\bar{e}_i, \bar{f}_{a_i}, a_i, a_{i-1}, s_i, s_{i-1}) = \frac{count(o\,j - d\,k) + \gamma}{\sum_k \sum_j (count(o\,j - d\,k) + \gamma)} \quad (18)$$

DOO:

$$p(D|e,f) = \prod_{i=1}^{n} p((o-d)|\bar{e}_i, \bar{f}_{a_i}, a_i, a_{i-1}, s_i, s_{i-1}) = \frac{count(o\,j - d\,k) + \gamma}{\sum_j (count(o\,j - d\,k) + \gamma)} \quad (19)$$

DOD:

$$p(D|e,f) = \prod_{i=1}^{n} p((o-d)|\bar{e}_i, \bar{f}_{a_i}, a_i, a_{i-1}, s_i, s_{i-1}) = \frac{count(o\,j - d\,k) + \gamma}{\sum_k (count(o\,j - d\,k) + \gamma)} \quad (20)$$

The pseudo code of training step is illustrated in TABLE III.

TABLE 2. PSEUDO CODE OF USING PARENT NODE MODEL.

```
Training
Extract phrases
Generate dependency tree for each sentence in source side
For each phrase and its previous ones
        Find r or m movement from target dependency tree
        Find m, s or d movement as Equation5
        Combine these two movements and generate m-m, m-r, s-m, s-r, d-m or d-r
        Save to file
Find probability in three manners DO, DOD and DOO
```

TABLE 3. THE PSEUDO CODE OF TRAINING STEP FOR USING TARGET DEPENDENCY TREE MODEL.

```
Training
Extract phrases
Generate dependency tree for each sentence in source and target sides
For each phrase and its previous ones
        Find r or m movement from target dependency tree
        Find o or i movement from source dependency tree
        Combine these two movements and generate r-o, r-i, m-o or m-i
        Save to file
Find probability in three manners DO, DOD and DOO
```

C. Decoding

We add this new reordering model as a feature to Moses decoder. In hypothesis expansion four hypotheses are expanded. One with m-i, one with r-i, one with m-o and one with r-o orientation type.

VI. EXPERIMENTAL RESULTS

A. Corpus

For testing this method two English-Persian parallel corpora are used. Corpus1 is a small corpus. Sentences of this corpus are drawn from a simple domain (conversation about meeting scheduling). Corpus 2 is bigger than corpus 1 and sentences of this corpus are drawn from news and average sentence length of this corpus is longer than corpus 1. The direction of translation is Persian to English. The statistics summary of this corpus is given in TABLE IV.

TABLE 4. STATISTICS SUMMARY OF CORPUS 1.

	English	Persian
Train: Sentences	23145	23145
Running Words	249335	216577
Singleton	2501	2415
Tune Sentences	276	276
Test Sentences	250	250

The statistic summary of corpus 2 is given in TABLE V.

B. Baseline Systems

We use two baseline systems:

Baseline 1: Lexicalized reordering model (msd-bidirectional-fe)

Baseline 2: source side dependency reordering model [19].

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

TABLE 5. STATISTICS SUMMARY OF CORPUS 2.

	English	Persian
Train: Sentences	100000	100000
Running Words	2615626	2782206
Singleton	73687	73652
Tune Sentences	665	665
Test Sentences	1045	1045

C. Experimental Setup

For finding a dependency tree we use MST parser [23] which is trained with Persian dependency tree bank [3] [24]. The accuracy of the parser on the Dependency tree bank is 86%. Feature weights λ_1^N are generated by YASMET [25] toolkit.

We choose Moses [26] as the experimental decoder. GIZA++ [27] and the heuristic "grow-diag-final-and" are used for generating word alignments. A 3-gram language model is generated by SRILM toolkit [28]. Maximum length of bilingual phrases is set to 13 words. During decoding ttable-limit is set to 5 and distortion limit set to 6. MERT [29] is used for tuning various feature weights.

D. Experimental results for discriminative reordering model

Experimental results of discriminative reordering model on corpus 1 are shown in TABLE VI and on corpus 1 are given in TABLE VII. The BLEU metric for this method is 2.5 points better than baseline 1 and 1.3 points better than baseline 2.
The BLEU metric on corpus 2 is 1.25 points better than baseline 1 and 1 point better than baseline 2.

TABLE 6. EXPERIMENTAL RESULTS OF DISCRIMINATIVE REORDERING MODEL ON CORPUS 1.

Model	BLEU
Base line 1	22.31
Base line 2	23.51
Discriminative(word features)	22.36
Discriminative(word features+orient)	24.18
Discriminative(word features+orient+phrse number)	**24.82**

TABLE 7. EXPERIMENTAL RESULTS OF DISCRIMINATIVE REORDERING MODEL ON CORPUS 2.

model	BLEU
Base line 1	25
Baseline 2	25.21
Discriminative(word+orient+phrase num)	**26.25**

[3] http://www.dadegan.ir/perdt

E. Experimental result of using parent node

The experimental results for using parent node approach are shown in TABLE VIII and TABLE IX. The BLEU metric for this method is 0,8 point better than baseline1 on corpus 1 and 0.6 point better than baseline on corpus 2. This results show the improvement in translation system output. So considering the direction of the phrase with respect to parent node is almost beneficial.

F. Experimental Results for Target Dependency Tree

Experimental results for target dependency tree approach are shown in TABLE X and TABLE XI. This results show the improvement in the translation. This method in addition to use source side dependency tree information, use target side dependency tree information therefore this method have better estimation of reordering probabilities and gives the decoder more accurate information about reordering events.

G. Discussion and Analysis

In this paper three types of reordering models are introduced. Among them discriminative reordering model has better results on two corpora. This method extracts features from training data and use maximum entropy principle in calculating the probabilities. The merits of this method with respect to two other methods are:

- No sparsity problem is occurred. Two other methods use relative frequency so the sparsity problem can be occurred in calculating the probabilities.
- This method calculates the probabilities at decoding time while the other two methods pre-compute the probabilities in training step and use them in decoding so phrases that do not appear in training step cannot be reordered in decoding.

TABLE VI and TABLE VII show the results of this method. In TABLE VI, the effect of each feature is investigated. These results show that all feature functions improved the translation quality. The row 3 of this table shows the result of this method with just word features that is an implementation of [7] with 6 feature and using dependency tree.

The performance of this method is better than base line 2. The Base line 2 is an implementation of [19] that has used 6 movement events same as our method and has used relative frequency and has pre-computed the probabilities in training step.

Using target dependency perform better than using parent node on corpus 1 but using parent node model perform better than target dependency on corpus 2 but this difference in BLEU score on corpus 2 is trifle. These results show that the "using target dependency tree" model gives more information to the decoder than "using parent node" but "using parent node" model is faster than the other in training.

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

These reordering models work well on both corpora. On corpus 1 with shorter sentence length the most part of reorderings are short or medium range ones. The improvement of BLEU on this corpus shows that the proposed reordering models work well for short and medium range reordering. The most part of reorderings on corpus 2 are long range reordering and all models work well for long range reordering as well.

TABLE 8. EXPERIMENTAL RESULTS FOR USING OF PARENT NODE APPROACH ON CORPUS 1.

Reordering Model	**BLEU**	**NIST**
Baseline 1	22.31	4.51
DO	**23.10**	4.69
DOD	22.4	4.67
DOO	23	4.56

TABLE 9. EXPERIMENTAL RESULTS FOR USING OF PARENT NODE APPROACH ON CORPUS 2.

Reordering Model	**BLEU**	**NIST**
Baseline 1	22.3	4.51
DO	25.3	4.69
DOD	24.8	4.67
DOO	**25.6**	4.56

TABLE 10. EXPERIMENTAL RESULTS FOR USING TARGET DEPENDENCY TREE ON CORPUS 1

Reordering Model	**BLEU**	**NIST**
Baseline 1	22.31	4.51
DO	22.13	4.7
DOD	**23.4**	4.83
DOO	22.8	4.62

TABLE 11. TABLE 1. EXPERIMENTAL RESULTS FOR USING TARGET DEPENDENCY TREE ON CORPUS 2

Reordering Model	**BLEU**	**NIST**
Baseline 1	25	**7.2**
DO	**25.5**	**7.41**
DOD	25.03	**7.28**
DOO	25.01	**7.36**

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

VII. CONCLUSION

In this paper three novel reordering models for statistical machine translation have been presented. One of these methods is a novel discriminative Reordering Model that uses source side dependency tree movement for statistical machine translation. For training this model with maximum entropy principle several features have been used:

- Features based on source and target words
- Phrase number feature that considered the order of translation of a phrase
- Orientation memory feature that hold the orientation of current phrase with respect to previous phrases.

In experimental result effect of each feature on translation quality has been investigated and it has been found that each feature is effective and system with phrase number feature has the best performance. But system in all experiments performs better than baselines. In the future we can extend model by adding new features or changing the movement methods.

The other method in addition to extracting information from source side dependency tree uses information of target dependency tree. This method improves the translation quality with respect to baseline. The other method defines new movements by considering orientation of phrase with respect to head of phrase. This method improves the translation quality with respect to baseline.

We decide to change the target dependency tree model to compute the probabilities in decoding step and use cohesive constraint in "using parent node" model.

REFERENCES

[1] Kevin Knight. 1999. Squibs and Discussions: De-coding Complexity in Word-Replacement Translation Models. Computational Linguistics, 25(4).

[2] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In Proc. of HLT-NAACL 2003, pages 127-133

[3] Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In Proc. of HLT-NAACL 2004, Short Papers, pages 101-104.

[4] Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In Proceedings of ACL'05, pages 531–540.

[5] Xiong, Deyi, Qun Liu, and Shouxun Lin. "Maximum entropy based phrase reordering model for statistical machine translation." In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 521-528. Association for Computational Linguistics, 2006.

[6] Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In Proc. of EMNLP 2008, pages 848-856.

[7] Richard Zens and Hermann Ney. 2006. Discriminative Reordering Models for Statistical Machine Translation. In Proc. of Workshop on Statistical Machine Translation 2006, pages 521-528.

[8] David Chiang. 2005. A Hierarchical Phrase Based Model for Statistical Machine Translation, In Proceedings of ACL'05, pages 263–270.

[9] Matthias Huck, Stephan Peitz, Markus Freitag, Hermann Ney. *2010. Discriminative Reordering Extensions for Hierarchical Phrase Based Machine Translation*, Proceedings of the 16th EAMT Conference, pages 304-311.

[10] *Huck, Matthias, Joern Wuebker, Felix Rietig, and Hermann Ney. "A Phrase Orientation Model for Hierarchical Machine Translation." Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 452–463,Sofia, Bulgaria, August 8-9, 2013*

[11] Shu Cai, Yajuan Lu, Qun Liu.2009. Improved Reordering Rules for Hierarchical Phrase-Based Translation. In Proc of IALP '09 Proceedings of the 2009 International Conference on Asian Language Processing, pages 65-70.

[12] Huang, Fei, and Cezar Pendus. "Generalized Reordering Rules for Improved SMT." In ACL (2), pp. 387-392. 2013

[13] Karthik Viewswariah, Jiri Navratil, Jeffrey Sorensen. 2010. Syntax Based Reordering with Automatically Derived Rules for Improved Machine Translation, In proc of the 23rd International conference on Computational Linguistics(colling 2010), pages 1119-1127.

[14] Genzel, Dmitriy. "Automatically learning source-side reordering rules for large scale machine translation." In Proceedings of the 23rd international conference on computational linguistics, pp. 376-384. Association for Computational Linguistics, 2010.

[15] Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In Proceedings of EMNLP'07, pages 737–745.

[16] Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve smt for subject-object-verb languages. In Proceedings of NAACL-HLT'09, pages 245–253

[17] Colin Cherry. 2008. Cohesive phrase-based decoding for statistical machine translation. In Proceedings of ACL-08: HLT, pages 72–80.

[18] Nguyen Bach, Stephan Vogel, and Colin Cherry. 2009. Cohesive constraints in a beam search phrase-based decoder. In Proceedings of NAACL-HLT'09.

[19] Nguyen Bach, Qin Gao, Stephan Vogl, 2009, Source Side Dependency Tree Reordering Models with Subtree Movements and Constraints, In proceedings of MT-SummitXII.

[20] Feng, Minwei, Jan-Thorsten Peter, and Hermann Ney. "Advancements in Reordering Models for Statistical Machine Translation." In *ACL (1)*, pp. 322-332. 2013.

[21] J. N. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. Annals of Mathematical Statistics, 43:1470–1480.

[22] Jinsong Su, Yang Liu, Yajuan LÜ, HaitaoMi, Qun Liu. 2010. Learning lexicalized reordering models from reordering graphs. In Proc. of ACL 2010, Short Papers, pages 12-16.

[23] Ryan McDonald, Kevin Lerman, Fernando Pereira. 2006. Multilingual Dependency Parsing with a Two-Stage Discriminative Parser, In proc of Tenth Conference on Computational Natural Language Learning (CoNLL-X).

[24] Mohammad Sadegh Rasooli, Manouchehr Kouhestani, and Amirsaeid Moloodi. (2013). Development of a Persian Syntactic Dependency Treebank. In The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), Atlanta, USA.

[25] Franz Josef Och. 2001. YASMET: Toolkit for conditional maximum entropy models. http://www-i6.informatik.rwthaachen.de/web/Software/YASMET.html.

[26] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi,Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In Proc. of ACL 2007, Demonstration Session, pages 177-180.

[27] Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1):19-51.

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

[28] Andreas Stolcke. 2002. SRILM - *An extensible language modeling toolkit.* In Proc. of ICSLP, Vol. 2, pages 901-904.

[29] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In Proc. of ACL 2003, pages 160-167.

Zahra Rahimi received her B.Sc degree in software engineering from Shahrood University of Technology and M.Sc degree in Artificial Intelligence from Amirkabir university of technology in 2010 and 2013 respectively. Her research interests include statistical machine translation , natural language processing and machine learning.

Shahram Khadivi received the B.S. and M.S. degrees in computer engineering from Amirkabir University of Technology, Tehran, Iran, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from RWTH Aachen University, Aachen, Germany, in 2008. Since September 2008, he has been assistant professor at Computer Engineering department at Amirkabir University of Technology. His research interests include statistical machine translation, computational natural language processing, information retrieval, machine learning, and data analysis.

Hesham Faili has his B.Sc.. and M.Sc. in software engineering and Ph.D. in artificial intelligence from Sharif University of Technology. He is an assistant professor at Tehran University in the School of Electrical and Computer Engineering. His research interests include natural language processing, machine translation, data mining, and social networks.

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]

IJICTR

This Page intentionally left blank.

[Downloaded from ijict.itrc.ac.ir on 2024-04-26]