Maximum Entropy based Rule Selection Model for Syntax-based Statistical Machine Translation

Qun Liu[1] and **Zhongjun He**[1,2] and **Yang Liu**[1] and **Shouxun Lin**[1]
[1]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, 100190, China
[2]Graduate University of Chinese Academy of Sciences
Beijing, 100049, China
{liuqun,zjhe,yliu,sxlin}@ict.ac.cn

Abstract

This paper proposes a novel maximum entropy based rule selection (MERS) model for syntax-based statistical machine translation (SMT). The MERS model combines local contextual information around rules and information of sub-trees covered by variables in rules. Therefore, our model allows the decoder to perform context-dependent rule selection during decoding. We incorporate the MERS model into a state-of-the-art linguistically syntax-based SMT model, the tree-to-string alignment template model. Experiments show that our approach achieves significant improvements over the baseline system.

1 Introduction

Syntax-based statistical machine translation (SMT) models (Liu et al., 2006; Galley et al., 2006; Huang et al., 2006) capture long distance reorderings by using rules with structural and linguistical information as translation knowledge. Typically, a translation rule consists of a source-side and a target-side. However, the source-side of a rule usually corresponds to multiple target-sides in multiple rules. Therefore, during decoding, the decoder should select a correct target-side for a source-side. We call this *rule selection*.

Rule selection is of great importance to syntax-based SMT systems. Comparing with word selection in word-based SMT and phrase selection in phrase-based SMT, rule selection is more generic and important. This is because that a rule not only contains terminals (words or phrases), but also contains nonterminals and structural information. Terminals indicate lexical translations, while nonterminals and structural information can capture short or long distance reorderings. See rules in Figure 1 for illustration. These two rules share the same syntactic tree on the source side. However, on the target side, either the translations for terminals or the phrase reorderings for nonterminals are quite different. During decoding, when a rule is selected and applied to a source text, both lexical translations (for terminals) and reorderings (for nonterminals) are determined. Therefore, rule selection affects both lexical translation and phrase reordering.

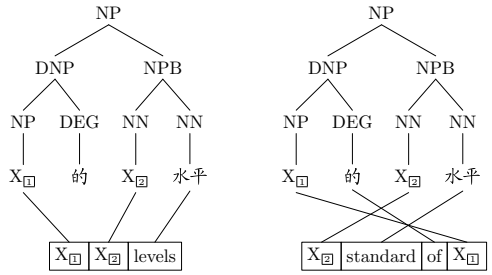


Figure 1: Example of translation rules

However, most of the current syntax-based systems ignore contextual information when they selecting rules during decoding, especially the information of sub-trees covered by nonterminals. For example, the information of $X_{[1]}$ and $X_{[2]}$ is not recorded when the rules in Figure 1 extracted from the training examples in Figure 2. This makes the decoder hardly distinguish the two rules. Intuitively, information of sub-trees covered by nonterminals as well as contextual information of rules are believed

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 89–97,
Honolulu, October 2008.

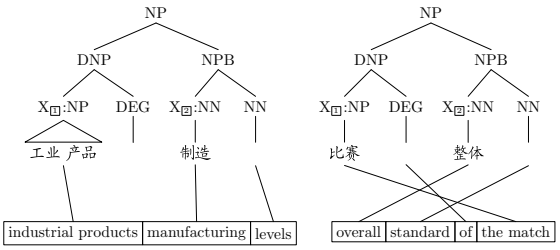


Figure 2: Training examples for rules in Figure 1

to be helpful for rule selection.

Recent research showed that contextual information can help perform word or phrase selection. Carpuat and Wu (2007b) and Chan et al. (2007) showed improvents by integrating word-sense-disambiguation (WSD) system into a phrase-based (Koehn, 2004) and a hierarchical phrase-based (Chiang, 2005) SMT system, respectively. Similar to WSD, Carpuat and Wu (2007a) used contextual information to solve the ambiguity problem for phrases. They integrated a phrase-sense-disambiguation (PSD) model into a phrase-based SMT system and achieved improvements.

In this paper, we propose a novel solution for rule selection for syntax-based SMT. We use the maximum entropy approach to combine rich contextual information around a rule and the information of sub-trees covered by nonterminals in a rule. For each ambiguous source-side of translation rules, a maximum entropy based rule selection (MERS) model is built. Thus the MERS models can help the decoder to perform a context-dependent rule selection.

Comparing with WSD (or PSD), there are some advantages of our approach:

- Our approach resolves ambiguity for rules with multi-level syntactic structure, while WSD resolves ambiguity for strings that have no structures;
- Our approach can help the decoder perform both lexical selection and phrase reorderings, while WSD can help the decoder only perform lexical selection;
- Our method takes WSD as a special case, since a rule may only consists of terminals.

In our previous work (He et al., 2008), we reported improvements by integrating a MERS model into a formally syntax-based SMT model, the hierarchical phrase-based model (Chiang, 2005). In this paper, we incorporate the MERS model into a state-of-the-art linguistically syntax-based SMT model, the tree-to-string alignment template (TAT) model (Liu et al., 2006). The basic differences are:

- The MERS model here combines rich information of source syntactic tree as features since the translation model is linguistically syntax-based. He et al. (2008) did not use this information.
- In this paper, we build MERS models for all ambiguous source-sides, including lexicalized (source-side which only contains terminals), partially lexicalized (source-side which contains both terminals and nonterminals), and unlexicalized (source-side which only contains nonterminals). He et al. (2008) only built MERS models for partially lexicalized source-sides.

In the TAT model, a TAT can be considered as a translation rule which describes correspondence between source syntactic tree and target string. TAT can capture linguistically motivated reorderings at short or long distance. Experiments show that by incorporating MERS model, the baseline system achieves statistically significant improvement.

This paper is organized as follows: Section 2 reviews the TAT model; Section 3 introduces the MERS model and describes feature definitions; Section 4 demonstrates a method to incorporate the MERS model into the translation model; Section 5 reports and analyzes experimental results; Section 6 gives conclusions.

2 Baseline System

Our baseline system is Lynx (Liu et al., 2006), which is a linguistically syntax-based SMT system. For translating a source sentence $f_1^J = f_1 ... f_j ... f_J$, Lynx firstly employs a parser to produce a source syntactic tree $T(f_1^J)$, and then uses the source syntactic tree as the input to search translations:

(1) $\widetilde{e}_1^I = \text{argmax}_{e'^I_1} Pr(e_1^I|f_1^J)$
$= \text{argmax}_{e'^I_1} Pr(T(f_1^J)|f_1^J) Pr(e_1^I|T(f_1^J))$

In doing this, Lynx uses tree-to-string alignment template to build relationship between source syntactic tree and target string. A TAT is actually a translation rule: the source-side is a parser tree with leaves consisting of words and nonterminals, the target-side is a target string consisting of words and nonterminals.

TAT can be learned from word-aligned, source-parsed parallel corpus. Figure 4 shows three types of TATs extracted from the training example in Figure 3: lexicalized (the left), partially lexicalized (the middle), unlexicalized (the right). Lexicalized TAT contains only terminals, which is similar to phrase-to-phrase translation in phrase-based model except that it is constrained by a syntactic tree on the source-side. Partially lexicalized TAT contains both terminals and non-terminals, which can be used for both lexical translation and phrase reordering. Unlexicalized TAT contains only nonterminals and can only be used for phrase reordering.

Lynx builds translation model in a log-linear framework (Och and Ney, 2002):

(2) $$P(e_1^I|T(f_1^J)) = \frac{\exp[\sum_m \lambda_m h_m(e_1^I, T(f_1^J))]}{\sum_{e'} \exp[\sum_m \lambda_m h_m(e_1^I, T(f_1^J))]}$$

Following features are used:

- Translation probabilities: $P(\widetilde{e}|\widetilde{T})$ and $P(\widetilde{T}|\widetilde{e})$;
- Lexical weights: $P_w(\widetilde{e}|\widetilde{T})$ and $P_w(\widetilde{T}|\widetilde{e})$;
- TAT penalty: $exp(1)$, which is analogous to phrase penalty in phrase-based model;
- Language model $P_{lm}(e_1^I)$;
- Word penalty I.

In Lynx, rule selection mainly depends on translation probabilities and lexical weights. These four scores describe how well a source tree links to a target string, which are estimated on the training corpus according to occurrence times of $\widetilde{e}$ and $\widetilde{T}$. There

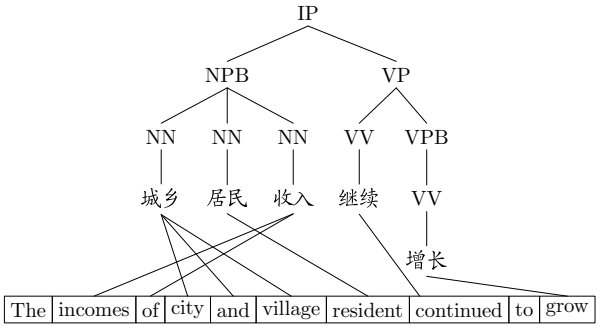


Figure 3: Word-aligned, source-parsed training example.

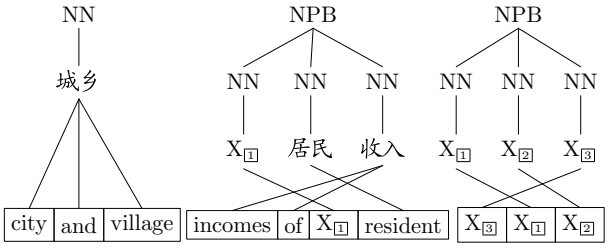


Figure 4: TATs learned from the training example in Figure 3.

are no features in Lynx that can capture contextual information during decoding, except for the n-gram language model which considers the left and right neighboring n-1 target words. But this information it very limited.

3 The Maximum Entropy based Rule Selection Model

3.1 The model

In this paper, we focus on using contextual information to help the TAT model perform context-dependent rule selection. We consider the rule selection task as a multi-class classification task: for a source syntactic tree $\widetilde{T}$, each corresponding target string $\widetilde{e}$ is a label. Thus during decoding, when a TAT $\langle \widetilde{T}, \widetilde{e'} \rangle$ is selected, $\widetilde{T}$ is classified into label $\widetilde{e'}$, actually.

A good way to solve the classification problem is the maximum entropy approach:

(3) $$P_{rs}(\widetilde{e}|\widetilde{T}, T(X_k)) = \frac{\exp[\sum_i \lambda_i h_i(\widetilde{e}, C(\widetilde{T}), T(X_k))]}{\sum_{\widetilde{e'}} \exp[\sum_i \lambda_i h_i(\widetilde{e'}, C(\widetilde{T}), T(X_k))]}$$

where $\widetilde{T}$ and $\widetilde{e}$ are the source tree and target string of a TAT, respectively. h_i is a binary feature functions and λ_i is the feature weight of h_i. $C(\widetilde{T})$ defines local contextual information of $\widetilde{T}$. X_k is a nonterminal in the source tree $\widetilde{T}$, where k is an index. $T(X_k)$ is the source sub-tree covered by X_k.

The advantage of the MERS model is that it uses rich contextual information to compute posterior probability for $\widetilde{e}$ given $\widetilde{T}$. However, the translation probabilities and lexical weights in Lynx ignore these information.

Note that for each ambiguous source tree, we build a MERS model. That means, if there are N source trees extracted from the training corpus are ambiguous (the source tree which corresponds to multiple translations), thus for each ambiguous source tree T_i ($i = 1, ..., N$), a MERS model M_i ($i = 1, ..., N$) is built. Since a source tree may correspond to several hundreds of target translations at most, the feature space of a MERS model is not prohibitively large. Thus the complexity for training a MERS model is low.

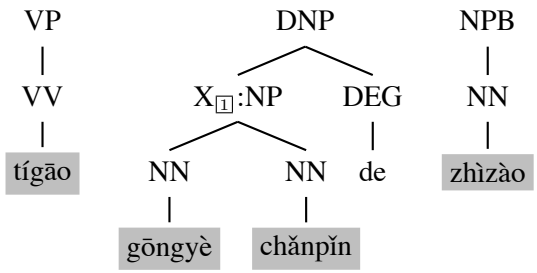
3.2 Feature Definition

Let $\langle \widetilde{T}, \widetilde{e} \rangle$ be a translation rule in the TAT model. We use $f(\widetilde{T})$ to represent the source phrase covered by $\widetilde{T}$. To build a MERS model for the source tree $\widetilde{T}$, we explore various features listed below.

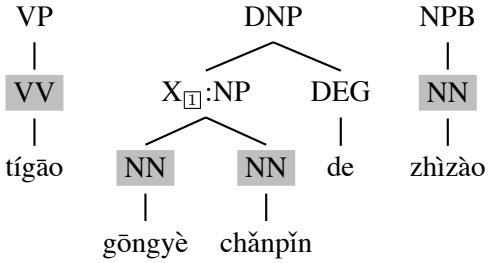
1. Lexical Features (LF)
   These features are defined on source words. Specifically, there are two kinds of lexical features: **external** features f_{-1} and f_{+1}, which are the source words immediately to the left and right of $f(\widetilde{T})$, respectively; **internal** features $f_L(T(X_k))$ and $f_R(T(X_k))$, which are the left most and right most boundary words of the source phrase covered by $T(X_k)$, respectively.

   See Figure 5 (a) for illustration. In this example, f_{-1}=tígāo, f_{+1}=zhìzào, $f_L(T(X_1))$=gōngyè, $f_R(T(X_1))$=chǎnpǐn.

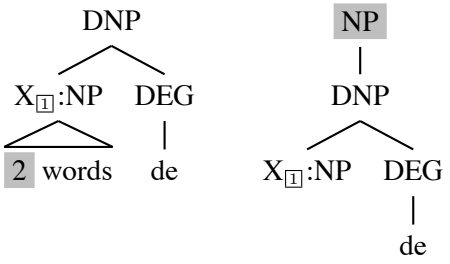
2. Parts-of-speech (POS) Features (POSF)
   These features are the POS tags of the source words defined in the lexical features: P_{-1}, P_{+1}, $P_L(T(X_k))$, $P_R(T(X_k))$ are the POS tags of f_{-1}, f_{+1}, $f_L(T(X_k))$, $f_R(T(X_k))$, re-



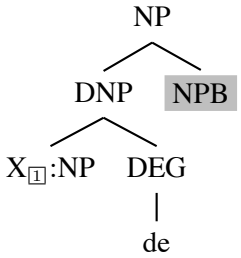
(a) Lexical Features



(b) POS Features



(c) Span Feature (d) Parent Feature



(e) Sibling Feature

Figure 5: Illustration of features of the MERS model. The source tree of the TAT is ⟨ DNP(NP $X_{[1]}$) (DEG de)⟩. Gray nodes denote information included in the feature.

spectively. POS tags can generalize over all training examples.

Figure 5 (b) shows POS features. P_{-1}=VV, P_{+1}=NN, $P_L(T(X_1))$=NN, $P_R(T(X_1))$=NN.

3. Span Features (SPF)
These features are the length of the source phrase $f(T(X_k))$ covered by $T(X_k)$. In Liu's TAT model, the knowledge learned from a short span can be used for a larger span. This is not reliable. Thus we use span features to allow the MERS model to learn a preference for short or large span.

In Figure 5 (c), the span of $X_{\boxed{1}}$ is 2.

4. Parent Feature (PF)
The parent node of $\widetilde{T}$ in the parser tree of the source sentence. The same source sub-tree may have different parent nodes in different training examples. Therefore, this feature may provide information for distinguishing source sub-trees.

Figure 5 (d) shows that the parent is a *NP* node.

5. Sibling Features (SBF)
The siblings of the root of $\widetilde{T}$. This feature considers neighboring nodes which share the same parent node.

In Figure 5 (e), the source tree has one sibling node *NPB*.

Those features make use of rich information around a rule, including the contextual information of a rule and the information of sub-trees covered by nonterminals. They are never used in Liu's TAT model.

Figure 5 shows features for a partially lexicalized source tree. Furthermore, we also build MERS models for lexicalized and unlexicalized source trees. Note that for lexicalized tree, features do not include the information of sub-trees since there is no nonterminals.

The features can be easily obtained by modifying the TAT extraction algorithm described in (Liu et al., 2006). When a TAT is extracted from a word-aligned, source-parsed parallel sentence, we just record the contextual features and the features of the sub-trees. Then we use the toolkit implemented by Zhang (2004) to train MERS models for the ambiguous source syntactic trees separately. We set the iteration number to 100 and Gaussian prior to 1.

4 Integrating the MERS Models into the Translation Model

We integrate the MERS models into the TAT model during the translation of each source sentence. Thus the MERS models can help the decoder perform context-dependent rule selection during decoding.

For integration, we add two new features into the log-linear translation model:

- $P_{rs}(\widetilde{e}|\widetilde{T}, T(X_k))$. This feature is computed by the MERS model according to equation (3), which gives a probability that the model selecting a target-side $\widetilde{e}$ given an ambiguous source-side $\widetilde{T}$, considering rich contextual information.

- $P_{ap} = exp(1)$. During decoding, if a source tree has multiple translations, this feature is set to $exp(1)$, otherwise it is set to $exp(0)$. Since the MERS models are only built for ambiguous source trees, the first feature $P_{rs}(\widetilde{e}|\widetilde{T}, T(X_k))$ for non-ambiguous source tree will be set to 1.0. Therefore, the decoder will prefer to use non-ambiguous TATs. However, non-ambiguous TATs usually occur only once in the training corpus, which are not reliable. Thus we use this feature to reward ambiguous TATs.

The advantage of our integration is that we need not change the main decoding algorithm of Lynx. Furthermore, the weights of the new features can be trained together with other features of the translation model.

5 Experiments

5.1 Corpus

We carry out experiments on Chinese-to-English translation. The training corpus is the FBIS corpus, which contains 239k sentence pairs with 6.9M Chinese words and 8.9M English words. For the language model, we use SRI Language Modeling Toolkit (Stolcke, 2002) with modified Kneser-Ney smoothing (Chen and Goodman, 1998) to train two tri-gram language models on the English portion of

Type	No. of TATs	No. of source trees	No. of ambiguous source trees	% ambiguous
Lexicalized	333,077	16,367	14,380	87.86
Partially Lexicalized	342,767	38,497	28,397	73.76
Unlexicalized	83,024	7,384	5,991	81.13
Total	758,868	62,248	48,768	78.34

Table 1: Statistical information of TATs filtered by test sets of NIST MT 2003 and 2005.

System	Features									
	$P(\widetilde{e}\|\widetilde{T})$	$P(\widetilde{T}\|\widetilde{e})$	$P_w(\widetilde{e}\|\widetilde{T})$	$P_w(\widetilde{T}\|\widetilde{e})$	lm_1	lm_2	TP	WP	P_{rs}	AP
Lynx	0.210	0.016	0.081	0.051	0.171	0.013	-0.055	0.403	-	-
+MERS	0.031	0.008	0.020	0.080	0.152	0.014	0.027	0.270	0.194	0.207

Table 2: Feature weights obtained by minimum error rate training on the development set. The first 8 features are used by Lynx. TP=TAT penalty, WP=word penalty, AP=ambiguous TAT penalty. Note that in fact, the positive weight for WP and AP indicate a reward.

the training corpus and the Xinhua portion of the Gigaword corpus, respectively. NIST MT 2002 test set is used as the development set. NIST MT 2003 and NIST MT 2005 test sets are used as the test sets. The translation quality is evaluated by BLEU metric (Papineni et al., 2002), as calculated by mteval-v11b.pl with case-insensitive matching of n-grams, where $n = 4$.

5.2 Training

To train the translation model, we first run GIZA++ (Och and Ney, 2000) to obtain word alignment in both translation directions. Then the word alignment is refined by performing "grow-diag-final" method (Koehn et al., 2003). We use a Chinese parser developed by Deyi Xiong (Xiong et al., 2005) to parse the Chinese sentences of the training corpus.

Our TAT extraction algorithm is similar to Liu et al. (2006), except that we make some tiny modifications to extract contextual features for MERS models. To extract TAT, we set the maximum height of the source sub-tree to $h = 3$, the maximum number of direct descendants of a node of sub-tree to $c = 5$. See (Liu et al., 2006) for specific definitions of these parameters.

Table 1 shows statistical information of TATs which are filtered by the two test sets. For each type (lexicalized, partially lexicalized, unlexicalized) of TATs, a great portion of the source trees are ambiguous. The number of ambiguous source trees accounts for 78.34% of the total source trees. This indicates that the TAT model faces serious rule selection problem during decoding.

5.3 Results

We use Lynx as the baseline system. Then the MERS models are incorporated into Lynx, and the system is called Lynx+MERS. To run the decoder, Lynx and Lynx+MERS share the same settings: tatTable-limit=30, tatTable-threshold=0, stack-limit=100, stack-threshold=0.00001. The meanings of the pruning parameters are the same to Liu et al. (2006).

We perform minimum error rate training (Och, 2003) to tune the feature weights for the log-linear model to maximize the systems's BLEU score on the development set. The weights are shown in Table 2.

These weights are then used to run Lynx and Lynx+MERS on the test sets. Table 3 shows the results. Lynx obtains BLEU scores of 26.15 on NIST03 and 26.09 on NIST05. Using all features described in Section 3.2, Lynx+MERS finally obtains BLEU scores of 27.05 on NIST03 and 27.28 on NIST05. The absolute improvements is 0.90 and 1.19, respectively. Using the sign-test described by Collins et al. (2005), both improvements are statistically significant at $p < 0.01$. Moreover, Lynx+MERS also achieves higher n-gram precisions than Lynx.

Test Set	System	BLEU-4	Individual n-gram precisions			
			1	2	3	4
NIST03	Lynx	26.15	71.62	35.64	18.64	9.82
	+MERS	**27.05**	72.00	36.72	19.51	10.37
NIST05	Lynx	26.09	70.39	35.12	18.53	10.11
	+MERS	**27.28**	71.16	36.19	19.62	10.95

Table 3: BLEU-4 scores (case-insensitive) on the test sets.

5.4 Analysis

The baseline system only uses four features for rule selection: the translation probabilities $P(\widetilde{e}|\widetilde{T})$ and $P(\widetilde{T}|\widetilde{e})$; and the lexical weights $P_w(\widetilde{e}|\widetilde{T})$ and $P_w(\widetilde{T}|\widetilde{e})$. These features are estimated on the training corpus by the maximum likelihood approach, which does not allow the decoder to perform a context dependent rule selection. Although Lynx uses language model as feature, the n-gram language model only considers the left and right n-1 neighboring target words.

The MERS models combines rich contextual information as features to help the decoder perform rule selection. Table 4 shows the effect of different feature sets. We test two classes of feature sets: the single feature (the top four rows of Table 4) and the combination of features (the bottom five rows of Table 4). For the single feature set, the POS tags are the most useful and stable features. Using this feature, Lynx+MERS achieves improvements on both the test sets. The reason is that POS tags can be generalized over all training examples, which can alleviate the data sparseness problem.

Although we find that some single features may hurt the BLEU score, they are useful in combination of features. This is because one of the strengths of the maximum entropy model is that it can incorporate various features to perform classification. Therefore, using all features defined in Section 3.2, we obtain statistically significant improvements (the last row of Table 4). In order to know how the MERS models improve translation quality, we inspect the 1-best outputs of Lynx and Lynx+MERS. We find that the first way that the MERS models help the decoder is that they can perform better selection for words or phrases, similar to the effect of WSD or PSD. This is because that lexicalized and partially lexicalized TAT contains terminals. Considering the following examples:

Feature Sets	NIST03	NIST05
LF	26.12	26.32
POSF	26.36	26.21
PF	26.17	25.90
SBF	26.47	26.08
LF+POSF	26.61	26.59
LF+POSF+SPF	26.70	26.44
LF+POSF+PF	26.81	26.56
LF+POSF+SBF	26.68	26.89
LF+POSF+SPF+PF+SBF	27.05	27.28

Table 4: BLEU-4 scores on different feature sets.

- Source: 马耳他 位于 欧洲 南部
- Reference: Malta is located in southern Europe
- Lynx: Malta in southern Europe
- Lynx+MERS: Malta is located in southern Europe

Here the Chinese word "位 于" is incorrectly translated into "in" by the baseline system. Lynx+MERS produces the correct translation "is located in". That is because, the MERS model considers more contextual information for rule selection. In the MERS model, $P_{rs}(\text{in}|\text{位于}) = 0.09$, which is smaller than $P_{rs}(\text{is located in}|\text{位于}) = 0.14$. Therefore, the MERS model prefers the translation "is located in". Note that here the source tree (VV 位于) is lexicalized, and the role of the MERS model is actually the same as WSD.

The second way that the MERS models help the decoder is that they can perform better phrase reorderings. Considering the following examples:

- Source: 按照 [在 中国 市场]$_1$ 的 [发展 战略]$_2$ ...
- Reference: According to its [development strategy]$_2$ [in the Chinese market]$_1$ ...
- Lynx: Accordance with [the Chinese market]$_1$ [development strategy]$_2$ ...
- Lynx+MERS: According to the [development strategy]$_2$ [in the Chinese market]$_1$

The syntactic tree of the Chinese phrase "在 中国 市场 的 发展 战略" is shown in Figure 6. However, there are two TATs which can be applied to the source tree, as shown in Figure 7. The baseline system selects the left TAT and produces a monotone translation of the subtrees "X[1]:PP" and "X[2]:NPB". However, Lynx+MERS uses the right TAT and performs correct phrase reordering by swapping the two source phrases. Here the source tree is partially lexicalized, and both the contextual information and the information of sub-trees covered by nonterminals are considered by the MERS model.

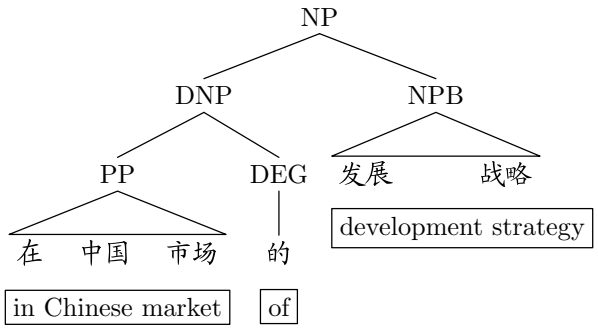


Figure 6: Syntactic tree of the source phrase "在 中国 市场 的 发展 战略".

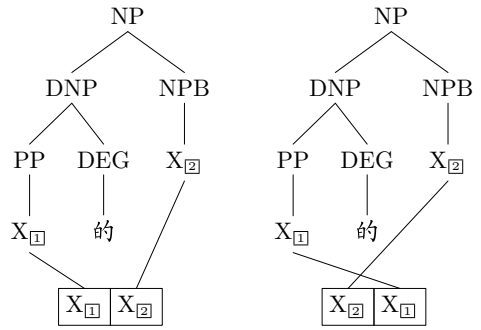


Figure 7: TATs which can be used for the source phrase "在 中国 市场 的 发展 战略".

6 Conclusion

In this paper, we propose a maximum entropy based rule selection model for syntax-based SMT. We use two kinds information as features: the local-contextual information of a rule, the information of sub-trees matched by nonterminals in a rule. During decoding, these features allow the decoder to perform a context-dependent rule selection. However, this information is never used in most of the current syntax-based SMT models.

The advantage of the MERS model is that it can help the decoder not only perform lexical selection, but also phrase reorderings. We demonstrate one way to incorporate the MERS models into a state-of-the-art linguistically syntax-based SMT model, the tree-to-string alignment model. Experiments show that by incorporating the MERS models, the baseline system achieves statistically significant improvements.

We find that rich contextual information can improve translation quality for a syntax-based SMT system. In future, we will explore more sophisticated features for the MERS model. Moreover, we will test the performance of the MERS model on large scale corpus.

Acknowledgements

We would like to thank Yajuan Lv for her valuable suggestions. This work was supported by the National Natural Science Foundation of China (NO. 60573188 and 60736014), and the High Technology Research and Development Program of China (NO. 2006AA010108).

References

Marine Carpuat and Dekai Wu. 2007a. How phrase sense disambiguation outperforms word sense disambiguation for statistical machine translation. In *11th Conference on Theoretical and Methodological Issues in Machine Translation*, pages 43–52.

Marine Carpuat and Dekai Wu. 2007b. Improving statistical machine translation using word sense disambiguation. In *Proceedings of EMNLP-CoNLL 2007*, pages 61–72.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual*

Meeting of the Association for Computational Linguistics, pages 33–40.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University Center for Research in Computing Technology.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 263–270.

M. Collins, P. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proc. of ACL05*, pages 531–540.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of COLING-ACL 2006*, pages 961–968.

Zhongjun He, Qun Liu, and Shouxun Lin. 2008. Improving statistical machine translation using lexicalized rule selection. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 321–328.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas*.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pages 127–133.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*, pages 115–124.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Andreas Stolcke. 2002. Srilm – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken language Processing*, volume 2, pages 901–904.

Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, and Yueliang Qian. 2005. Parsing the penn chinese treebank with semantic knowledge. In *Proceedings of IJCNLP 2005*, pages 70–81.

Le Zhang. 2004. Maximum entropy modeling toolkit for python and c++. available at http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.