Acta Technica *62* No. 1A/2017, 531–540

Hierarchical phrase machine translation decoding method based on tree-to-string model enhancement[1]

Ximeng Wen[2]

Abstract. Statistical machine translation model based on the syntax has gained unprecedented growth over the last ten years. In order to study the hierarchical phrase machine translation decoding method based on tree-to-string model enhancement, the hierarchical phrase model was used as the basic model in this paper. The tree-to-string model was used as a supplementary of hierarchical phrase model to increase the size of the translation inference space. And statistical machine translation decoding technology was mainly studied. Several decoding strategies including exact decoding strategy based tree, fuzzy decoding strategy based tree and decoding strategy based string were proposed. Finally, the experimental results on the NIST Chinese English translation task confirmed that the method studied in this paper could improve the translation performance of the baseline hierarchical phrase system effectively. For example, the data on newswire and web were raised by 1.3 and 1.2 BLEU points, respectively.

Key words. Hierarchical phrase model, tree-to-string model, machine translation decoding.

1. Introduction

The most successful statistical Machine translation model based syntax is a synchronous context free grammar based on hierarchical phrase model. In this model, no target language or source language syntax information is used to restrict translation in the process of extracting translation rules, so as to satisfy the learning of a large number of translation rules. However, the number of translation rules cannot be extended. Therefore, in order to control the number of translation rules, they are made within the acceptable range of the machine. The most common method is to set some restrictions in the extraction and use of hierarchical phrases, so as to achieve this effect. For example, the source language span referenced in decoding cannot exceed a threshold value. These restriction rules have already achieved good

[1]This work is supported by project from Education department of Hainan province, Automobile English (NO. Hyjc2011-11).

[2]Hainan College of Vocation & Technique, Haikou, Hainan, 570100, China

results and can even help the hierarchical phrase system to be practical. When the syntax needed to translate is complex or problems that need to be dealt with are dependent on long distance dependencies, these limitations also lead to a significant reduction in the processing power of hierarchical phrase systems. In order to better solve this problem and make the hierarchical phrase system have better translation and decoding performance, there are generally two ways to consider. The first is to add syntactic features of the target language to the system. The second is to combine hierarchical phrases with tree-to-string systems based on the idea of system integration. And the second method is the method which this article carries on the thorough discussion.

2. State of the art

Many scholars have begun to study and explore the language between machine translation since the beginning of the design and manufacture of the first computer. Rule-based methods were the mainstream method of machine translation research until 90s [1]. The statistical machine translation method was based on the content of Brown and other papers in 1993. The word based statistical machine translation method was first proposed innovatively [2]. This approach is a reversal of the process of translation and considers the target language sentence E as the input of the channel, it distorts through the noise channel, and the output side outputs the source language sentence, that is to say, it is needed to find the target language sentence to product F. At some level, the translation model and the language model reflect the fidelity and fluency of translation. At the time, the performance of this translation method exceeds that of rule-based SYSTRAN system, which attracts many researchers' interest [3]. Since then, more and more scholars have begun to do research on large-scale bilingual corpus analysis, probabilistic translation model exploration and model parameter research, which has opened up the era of machine translation and made many important breakthroughs. Phrase model performance growth has slowed down in recent years. This is mainly because only the size of the translation granularity is changed but there is no fundamental solution to the problem of remote reordering and lack of global information, which makes the model exhibit a plateau trend [4]. Yamad proposed the first statistical machine translation model based on syntax in 2001. In 2005, Chiang combined the model based phrase with the idea of tree structure and proposed an efficient decoding algorithm based on hierarchical phrase model and line graph analysis. The model is modeled based on simultaneous upper and lower independent grammars and does not use any of the displayed annotation information [5].

3. Methodology

Hierarchical phrase model relies on synchronous context free grammar (SCFG). A synchronous context free grammar can describe the generative process of bilingual strings containing hierarchical structures [6]. Formally, a synchronous context free

grammar is represented as a regular system called (N, W_s, W_t, R). Among them, N represents a set of non-terminating symbols, W_s and W_t represent termination symbols (or vocabulary) collections of source language and target language. Symbol R represents a production set [7]. Each production in R corresponds to a SCFG rule in the form of $X \to (\alpha, \beta, \sim)$. The resulting left-hand X represents a non-terminating character, α at the right hand represents a source language terminator and non-termination sequences, which is called the source language side, β represents the terminator and non-termination sequence of a target language, which is called the target language side. Finally, $\sim$ represents a one-to-one correspondence between non-terminating characters in α and β. Typically, $\sim$ can be represented as a non-subscript index [8].

Probabilistic synchronous context free grammars can be automatically extracted from word aligned data by using heuristic information [9]. For example, firstly, the initial set of translated phrases can be extracted, and then these initial phrases are used to obtain the translation rules of the hierarchical phrases (i.e., translation rules containing variables). After getting the SCFG rule, the SCFG rules can be used to decode the new sentence and complete the translation of the unknown sentence. An example of a SCFG rule that is extracted from a word - aligned example is given. Among them, rules of h_7, h_1 and h_3 correspond to a translation deduction, which can cover the whole bilingual sentence pair [10]. The decoding problem of hierarchical phrase model can also be treated as syntactic analysis. In other words,the source language side of SCFG is used to analyze the input sentences, and then to construct a SCFG derivation forest (or a hyper-graph structure). The translation model and the language model are used to derive the score, and the optimal derivation and output are obtained in the derivation forest [11].

In real systems, some constraints are introduced to enable the decoding process to be completed within an acceptable time usually. The details are as follows: When decoding, a hierarchical phrase rule can be applied to span size, which is called span limit. The usual limit is 10. The order of a rule (the number of variables allowed by the rule) is usually not more than two. Rule source language side variables can't appear continuously (except for glue rules). Rules must be Wie lexicalization rules (except for glue rules) and so on [12].

Tree-to-string translation model makes the translation process defined as the transformation from the source language to the target language string syntax tree. This translation process can be represented by a series of tree-to-string translation rules [13]. A tree-to-string rule r can be represented as $(s_r, t_r, \sim)$. Among them, s_r represents the source language fragment of the rule. The leaf node of s_r is either a terminator or variable (non-terminator). Symbol t_r represents the target language terminator and sequence of variables of a rule and $\sim$ represents a one-to-one correspondence between the leaf variables in s_r and the variables in t_r. The specific expression is shown in the following formula

$$\mathrm{VP}(\mathrm{VV}(\text{increase})x_1 : NN) \to \text{increases}(x_1)\,. \tag{1}$$

Formula (1) represents a tree-to-string translation rule. The "VP(VV(increase)x_1: NN)" represents the source language syntax tree fragment. "increases (x_1)" is the

target string. x_1 of two sections indicates that variables should correspond to each other.

The extraction of translation rules from tree-to-string is usually achieved by the GHKM method. The basic idea of GHKM method is to use the word alignment information to extract the minimum translation rules from the source language tree and target string [14].

The basic idea of integrating from the tree-to-string model integrated into the hierarchical phrase model is as follows: Hiero and GHKM are used to extract translation rules from bilingual data, at the same time, the extracted translation rules (tree-to-string) extracted by the GHKM method are added to the hierarchical phrase system to supplement the baseline SCFG. It should be noted that this method is different from the traditional system fusion and the mixed translation model, it does not simply equate different models (hierarchies phrases and trees to strings), and then fuse them together. Instead, the hierarchical phrase model is used as the underlying model, and then a small number of tree-to-string rules are used to enforce it [15]. In fact, the advantage of tree-to-string model is used to help the hierarchical phrase model improve its shortcomings, but it is not a symmetric system fusion method. Fig. 1 shows the basic framework of this method. The method uses both Hiero and GHKM methods to obtain rules and get a "larger" SCFG in the rule extraction phase. These SCFG rules of sentence words and syntactic information are used to decode the new sentence.

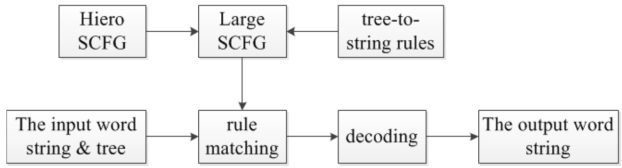

Fig. 1. Framework of tree-to-string model integration in hierarchical phrase system

As shown in Fig. 1, the method used in this paper requires simultaneous extraction of SCFG rules and tree-to-string rules extraction. These two kinds of rules can be obtained by using standard Hiero rule extraction method and GHKM rule extraction method respectively. However, tree-to-string translation rules and SCFG rules have different forms. Therefore, if you want to use tree-to-string translation rules in hierarchical phrase systems, you need to translate them into SCFG rules. Thus, it is possible to indirectly use the tree-to-string translation model information in a decoder based SCFG.

The conversion from tree-to-string rules to SCFG rules is very straightforward. For a given tree-to-string translation rule $(s_r, t_r, \sim)$, the sequence of leaf nodes corresponding to the source language side s_r is used as the source language side of the generated SCFG rule. And t_r and $\sim$ are kept unchanged in the SCFG rule. After that, all syntactic symbols in the rule are replaced by syntactic tags (such as X) used in hierarchical phrase systems, and the rules of SCFG are obtained. Thus, each tree-to-string translation rules will correspond to the only one SCFG rule after the above transformation. Therefore, the above results after translating of the original SCFG and the tree-to-string translation rules are merged to obtain a larger SCFG

rule.

Furthermore, the rules in the merged SCFG are divided into two types of rules. The type 1 rule is a rule that can be extracted by the Hiero. That is to say, all the rules in the baseline hierarchical phrase system are the first type rules. The second type rule is a rule that Hiero cannot extract, but it can be translated from tree-to-string translation rules.

In summary, the basic idea of the tree-to-string model is to obtain the phrase structure tree of the source language by a parser. And then, the tree-to-string alignment template is used to map the phrase structure tree of the source language into the string of the target language. In tree-to-string alignment templates (referred to as TAT), z is a three tuple $tildeT, \tilde{S}, \tilde{A}$). This three tuple describes the alignment relationship $\tilde{A}$ between the source language syntax tree $\tilde{T} = T(f_1^{\Gamma})$ and the target language string $\tilde{S} = S(e_1^{\Gamma})$. In this relation, T is used to represent a syntactic tree, and $T(z)$ is used to represent tree-to-string alignment of trees in template z. Similarly, $S(z)$ stands for tree-to-string alignment of strings in template z. The source language string f_1^{Γ} is the leaf node sequence of T. It may contain either a terminator or a non-terminating (part of speech mark or phrase structure class). The target language string e_1^{Γ} can also contain either a terminator or a non-terminator (placeholder).

The alignment relation $\tilde{A}$ is defined as a subset of the Descartes product of the source language and the target language symbol position, which is shown in the formula

$$\tilde{A} \subseteq \{(j,i) : j = 1 \cdots J', i = 1 \cdots I'\}. \tag{2}$$

Tree-to-string alignment templates can be divided into three categories according to the degree of lexicalization:

The first is the lexicalization alignment template. Leaf node and target language string of any source language syntax tree are all terminators.

The second is a partially lexicalization aligned template. Leaf node and the target language string of the source language syntax tree contain both the non-terminator and the terminator.

The third is a non-lexical alignment template. Leaf nodes and target language symbols of any source language syntax tree are non-terminating characters.

4. Result analysis and discussion

The experiment in this paper was carried out in the Chinese English translation task of NIST. The experiment used 2 million 700 thousand pairs of bilingual data. NiuTrans Hierarchy was chosen as the basic system of experiment, the decoder of the system was based on CKY algorithm, and the beam pruning and cubic pruning were used to speed up the decoder. The feature weights were automatically tuned on the development set by using minimum error rate training. All translation rules were obtained by standard Hiero extraction methods. The maximum span allowed in decoding and basic phrase rule extraction was 10.

The GHKM rules provided by NiuTrans were used to extract module for tree-to-

string rule extraction. Tree-to-string translation rules were extracted from a high-quality quantum set (500 thousand sentences) in training data. Each rule allowed up to 5 terminators and 5 variables at most. In addition, the tree-to-string rule was pruned by using translation probabilities. Pruning included the rule of discarding the forward translation probability which was less than 0.02 and the rule of non-lexicalization of discarding the forward translation probability which was less than 0.10.

Table 1 and Table 2 show the BLEU values for different experiments.

Table 1. BLEU value of Newswire translation system

	Tune	MT08	MT12	MT08.p	All test
	1181	691	400	688	1779
Standard hierarchical phrase base system	36.70	32.50	33.30	31.90	32.79
Exp01+ syntax soft constraint (feature)	36.84	32.44	33.30	31.99	32.83
Exp01+ removes span constraint	36.80	32.54	33.32	31.99	32.86
Exp03+ tree-to-string rule	37.19	33.06	33.79	32.27	33.20*
Exp04+ tree-to-string characterization	37.26	33.15	33.82	32.39	33.28**
Exp04+ fuzzy syntax mark	37.24	33.20	33.90	32.39	33.32**
Exp04+ fuzzy tree structure	37.45	33.39	33.97	32.66	33.49**
Exp04+ fuzzy tree structure & syntax mark	37.47	33.42	34.08	32.78	33.57**
Keywords Exp04+ based decoding	37.61	33.63	34.12	32.88	33.69**
Source language tree constraint	34.90	31.04	31.98	30.05	31.23**
Exp08 is done on span >10	37.12	33.20	33.63	32.20	33.17
Exp08+ left child optimization two fork	37.95	34.01	34.66	33.47	34.13**
Exp08+ right child optimization two fork	37.68	33.57	34.23	32.93	33.70**
Exp08+ forest based two forks	37.99	35.96	34.62	33.55	34.15**

Note: * or ** indicates a significant increase in baseline exp01 compared to the test set, P <0.05 or 0.01

Three baseline systems were selected for effective comparison. Exp01- standard hierarchical phrase system was NiuTrans Hierarchy. On the basis of exp01, if syntactic constraints were soft, exp02- used a better feature set, {NP+, NP=, VP+, VP=, PP+, PP=, XP+, XP=}. Exp03- was in exp01 decoding. When the source language fragment conformed to the syntactic structure, the span constraint was removed. This approach can be viewed as the simplest use of source language syntactic information in hierarchical phrase systems.

Table 2. BLEU value of Web translation system

	Tune	MT08	MT12	MT08.p	All test
	483	666	420	682	1768
Standard hierarchical phrase base system	31.80	23.90	21.90	25.00	24.21
Exp01+ syntax soft constraint (feature)	31.91	23.84	22.06	25.03	24.26
Exp01+ removes span constraint	31.85	23.95	21.86	25.00	24.22
Exp03+ tree-to-string rule	32.24	24.20	22.43	25.42	24.59
Exp04+ tree-to-string characterization	32.35	24.27	22.40	25.51	24.64*
Exp04+ fuzzy syntax mark	32.46	24.33	22.43	25.59	24.70**
Exp04+ fuzzy tree structure	32.60	24.46	22.48	25.65	24.81**
Exp04+ fuzzy tree structure & syntax mark	32.67	24.53	22.55	25.80	24.90**
Keywords Exp04+ based decoding	32.70	24.64	22.77	25.81	24.99**
Source language tree constraint	31.20	22.56	20.07	23.27	22.56
Exp08 is done on span >10	32.22	24.24	22.33	25.27	24.53
Exp08+ left child optimization two fork	33.04	24.99	23.04	26.24	25.44**
Exp08+ right child optimization two fork	32.77	24.60	22.87	25.86	25.07**
Exp08+ forest based two forks	33.02	24.94	23.07	26.30	25.48**

It can be seen from Table 1 and Table 2 that adding syntactic soft constraints (exp01) can lead to small performance gains over multiple test sets. On the one hand, this result confirms that source language syntax information is useful for machine translation. On the other hand, the results also show that simple syntactic features (without introducing new rules or increasing decoding space) cannot effectively improve the performance of hierarchical phrase system. In addition, removing the span constraint in exp03 will lead to certain BLEU improvements. The experimental results also verify that reducing span constraints is helpful for systems based syntactic constraint.

In addition, the running speeds of different decoding methods (tree-to-string rules and features were added for the baseline system, using string based decoding was used and the two forks method was added) were measured, as shown in Table 3.

Table 3 shows the average speed at which all the data is processed by the system. It can be seen that the translation speed of the system was only decreased by 10 % after introducing syntactic rules, which was consistent with the expected results. The introduction of less syntactic rules did not increase the system burden too much. On the other hand, when a string based decoding was introduced, the system run at

a half rate. The result is primarily for all span memory calculations due to string based decoding, and the system does not constrain the decoding family by decoding the syntax structure just as tree based decoding does. As a result, the system is burdened heavily.

Table 3. Operating speeds of different decoding methods

Number	system	Speed
Exp01	Standard hierarchical phrase base system	1.11 Sentence per second
Exp05	+Tree-to-string features and rules	1.01 Sentence per second
Exp09	+ decoding based String	0.47 Sentence per second
Exp12	+ left child priority two fork	0.42 Sentence per second

Table 3 shows the average speed at which all the data is processed by the system. It can be seen that the translation speed of the system was only decreased by 10 % after introducing syntactic rules, which was consistent with the expected results. The introduction of less syntactic rules did not increase the system burden too much. On the other hand, when a string based decoding was introduced, the system run at a half rate. The result is primarily for all span memory calculations due to string based decoding, and the system does not constrain the decoding family by decoding the syntax structure just as tree based decoding does. As a result, the system is burdened heavily.

In addition to examining the BLEU values of system output results, the use of different types of rules in optimal translation derivations were studied, as shown in Table 4.

Table 4. Percentage of different rule matching methods used

Rule matching method	Baseline (%)	+ Tree-to-string (%)	+ Binary tree (%)
String based	100	73	55
Tree based	0	27	45

5. Conclusion

The research focus of this thesis is the decoding of syntactic information of source language in hierarchical phrase system. The decoding strategy of tree-to-string rules in hierarchical phrase system was studied, and the corresponding decoding strategy was proposed. On the basis of the hierarchical phrase translation model, the number of string translation model was added as a supplementary model. Through the comparison of the experimental results of different decoding strategies, it can be found that the combination of the binary syntax tree and the decoding based on string can achieve maximum performance improvement. The conclusions can be obtained through this study that the biggest advantage of tree-to-string translation model is that the rules (and all variables) follow the syntax tree constraints. For example, all of the variables are required to cover legitimate and complete unit syn-

tactic sub-trees. Therefore, decoding of tree-to-string translation does not require forcing to join the constraint of the rule span. In addition, due to the use of source language syntax tree, the constraints such as the number of variables, the number of continuous variables in the source language, and the necessary lexicalization of rules in hierarchical phrases can be eliminated in tree-to-string translation models. Although good results have been achieved in this paper, there were still some problems that needed to be further studied in the future such as how to improve the accuracy of long sentence dependency analysis.

References

[1] W. Wang, J. May, K. Knight, D. Marcu: *Re-structuring, re-labeling, and re-aligning for syntax-based machine translation.* Journal Computational Linguistics *36* (2010), No. 2, 247–277.

[2] L. Adam: *Statistical machine translation.* ACM Computing Surveys (CSUR) *40* (2008), No. 3, article 8.

[3] D. Chiang: *Hierarchical phrase-based translation.* Journal Computational Linguistics *33* (2007), No. 2, 201–228.

[4] R. Q. Zhang, K. J. Yasuda, E. Sumita: *Chinese word segmentation and statistical machine translation.* Journal ACM Transactions on Speech and Language Processing *5* (2008), No. 2, Article No. 4.

[5] F. J. Och, H. Ney: *The alignment template approach to statistical machine translation.* Journal Computational Linguistics *30* (2004), No. 4, 417–449.

[6] F. J. Och, H. Ney: *A systematic comparison of various statistical alignment models.* Journal Computational Linguistics *29* (2003), No. 1, 19–51.

[7] C. Tillman, H. Nye: *Word reordering and a dynamic programming beam search algorithm for statistical machine translation.* Journal Computational Linguistics *29* (2003), No. 1, 97–133.

[8] H. Alshawi, S. Bangalore, S. Douglas: *Learning dependency translation models as collections of finite-state head transducers.* Computational Linguistics *26* (2000), No. 1, 45–60.

[9] P. F. Brown, J. Cocke, S. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, P. S. Roossin: *A statistical approach to machine translation.* Journal Computational Linguistics *16* (1990), No. 2, 79–85.

[10] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer: *The mathematics of statistical machine translation: Parameter estimation.* Journal Computational Linguistics–Special issue on using large corpora: II *19*, (1993), No. 2, 263–311.

[11] K. Knight: *Decoding complexity in word-replacement translation models.* Journal Computational Linguistics *25* (1999), No. 4, 607–615.

[12] K. Church, R. Patil: *Coping with syntactic ambiguity or how to put the block in the box on the table.* Journal Computational Linguistics *8* (1982), Nos. 3–4, 139–149.

[13] A. V. Aho, J. D. Ullman: *Syntax directed translations and the pushdown assembler.* Journal of Computer and System Sciences *3* (1969), No. 1 37–56.

[14] A. L. Berger, S. A. Della Pietra, V. J. Della Pietra: *A maximum entropy approach to natural language processing.* Journal Computational Linguistics *22*, (1996), No. 1, 39–71.

[15] D. Wu: *Stochastic inversion transduction grammars and bilingual parsing of parallel corpora.* Journal Computational Linguistics *23* (1997), No. 3, 377–403.

Received June 6, 2017