Effective Incorporation of Source Syntax into Hierarchical Phrase-based Translation

Tong Xiao†‡, Adrià de Gispert§, Jingbo Zhu†‡, Bill Byrne§
† Northeastern University, Shenyang 110819, China
‡ Hangzhou YaTuo Company, Hangzhou 310012, China
§ University of Cambridge, CB2 1PZ Cambridge, U.K.
{xiaotong,zhujingbo}@mail.neu.edu.cn
{ad465,wjb31}@eng.cam.ac.uk

Abstract

In this paper we explicitly consider source language syntactic information in both rule extraction and decoding for hierarchical phrase-based translation. We obtain tree-to-string rules by the GHKM method and use them to complement Hiero-style rules. All these rules are then employed to decode new sentences with source language parse trees. We experiment with our approach in a state-of-the-art Chinese-English system and demonstrate +1.2 and +0.8 BLEU improvements on the NIST newswire and web evaluation data of MT08 and MT12.

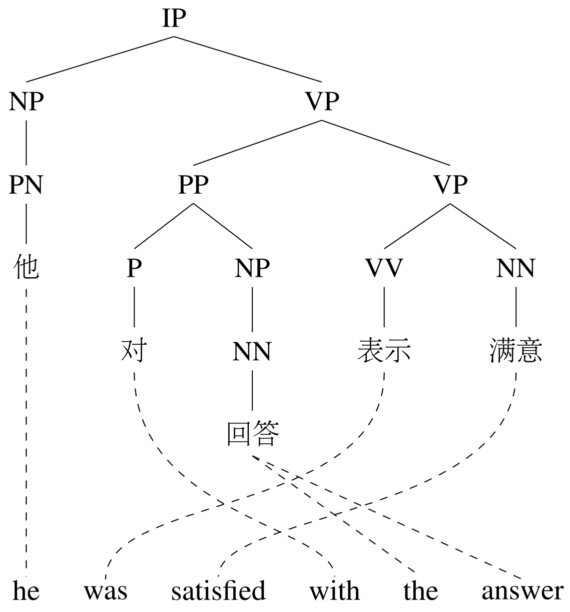
1 Introduction

Synchronous context free grammars (SCFGs) are widely used in statistical machine translation (SMT), with hierarchical phrase-based translation (Chiang, 2005) as the dominant approach. Hiero grammars are easily extracted from word-aligned parallel corpora and can capture complex nested translation relationships. Hiero grammars are formally syntactic, but rules are not constrained by source or target language syntax. This lack of constraint can lead to intractable decoding and bad performance due to the over-generation of derivations in translation. To avoid these problems, the extraction and application of SCFG rules is typically constrained by a source language span limit; (non-glue) rules are lexicalised; and rules are limited to two non-terminals which are not allowed to be adjacent in the source language. These constraints can yield good performing translation systems, although at a sacrifice in the ability to model long-distance movement and complex reordering of multiple constituents.

By contrast, the GHKM approach to translation (Galley et al., 2006) relies on a syntactic parse on either the source or target language side to guide SCFG extraction and translation. The parse tree provides linguistically-motivated constraints both in grammar extraction and in translation. This allows for looser span constraints; rules need not be lexicalised; and rules can have more than two non-terminals to model complex reordering multiple constituents. There are also modelling benefits as more meaningful features can be used to encourage derivations with ”well-formed” syntactic tree structures. However, GHKM can have robustness problems in that translation relies on the quality of the parse tree and the diversity of rule types can lead to sparsity and limited coverage.

In this paper we describe a simple but effective approach to introducing source language syntax into hierarchical phrase-based translation to get the benefits of both approaches. Unlike previous work, we do not resort to soft/hard syntactic constraints (Marton and Resnik, 2008; Li et al., 2013) or Hiero-style rule extraction algorithms for incorporating syntactic annotation into SCFGs (Zollmann and Venugopal, 2006; Zhao and Al-Onaizan, 2008; Chiang, 2010). We instead use GHKM syntactic rules to augment the baseline Hiero grammar and decoder. Our approach uses GHKM rules if possible and Hiero rules if not. We report performance on a state-of-the-art Chinese-English system. In a large-scale NIST evaluation task, we find significant improvements of over 1.2 and 0.8 BLEU relative to a strong Hiero baseline on the newswire and web evaluation data of MT08 and MT12. We also investigate variations in the GHKM formalism and find, for example, that our approach works well with binarized trees.

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2064–2074, Dublin, Ireland, August 23-29 2014.



Hiero-style SCFG Rules	
h_1	X → ⟨他, he⟩
h_2	X → ⟨对, with⟩
h_3	X → ⟨回答, the answer⟩
h_4	X → ⟨表示 满意, was satisfied⟩
h_5	X → ⟨X_1 表示 满意, was satisfied X_1⟩
h_6	X → ⟨X_1 表示 X_2, was X_2 X_1⟩
h_7	X → ⟨X_1 对 X_2 表示 满意, X_1 was satisfied with X_2⟩

Tree-to-String Rules	
r_1	NP(PN(他)) → he
r_2	P(对) → with
r_3	NP(NN(回答)) → the answer
r_4	VP(VV(表示) NN(满意)) → was satisfied
r_5	PP(x_1:P x_2:NP) → x_1 x_2
r_6	VP(x_1:PP x_2:VP) → x_2 x_1
r_7	IP(x_1:NP x_2:VP) → x_1 x_2
r_8	VP(PP(P(对) x_1:NP) x_2:VP) → x_2 with x_1

Figure 1: Hiero-syle and tree-to-string rules extracted from a pair of word-aligned Chinese-English sentences with a source language (Chinese) parse tree.

2 Background

2.1 Hierarchical Phrase-based Translation

In the hierarchical phrase-based approach, translation is modelled using SCFGs. In general, probabilistic SCFGs can be learned from word-aligned parallel data using heuristic methods (Chiang, 2007). We can first extract initial phrase pairs and then obtain hierarchical phrase rules (i.e., rules with non-terminals on the right hand side). Once the SCFG is obtained, new sentences can be decoded by finding the most likely derivation of SCFG rules. See Figure 1 for example rules extracted from a sentence pair with word alignments. A sequence of such rules covering the words of the source sentence is a SCFG derivation, e.g., rules h_7, h_1 and h_3 generate a derivation for the sentence pair.

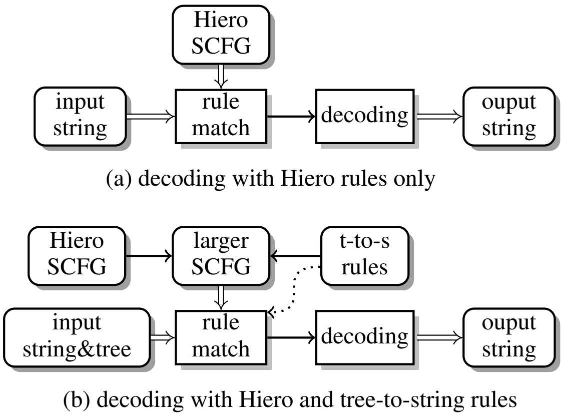
The Hiero SCFG allows vast numbers of derivations which can make unconstrained decoding intractable. In practice, several constraints are applied to control the model size and reduce ambiguity. Typically these are: (a) a rule span limit to be applied in decoding and sometimes also in rule extraction, set to 10; (b) a limit on the rank of the grammar (number of non-terminals that can appear on a rule), set to 2; and (c) a prohibition of consecutive non-terminals on the source language side of a rule (except the glue rules).

2.2 Tree-to-String Translation

Instead of modelling the problem based on surface strings, tree-to-string systems model the translation equivalency relations from source language syntactic trees to target language strings using derivations of tree-to-string rules (Liu et al., 2006; Mi et al., 2008; Huang and Mi, 2010; Feng et al., 2012). A tree-to-string rule is a tuple $\langle s_r, t_r, \sim \rangle$, where s_r is a source language tree-fragment with terminals and non-terminals at leaves; t_r is a string of target-language terminals and non-terminals; and $\sim$ is a 1-to-1 alignment between the non-terminals of s_r and t_r, for example, VP(VV(提高) x_1:NN) → increases x_1 is a tree-to-string rule, where the non-terminals labeled with the same index x_1 indicate the alignment.

To obtain tree-to-string rules, a popular way is to perform the GHKM rule extraction (Galley et al., 2006) on the bilingual sentences with both word alignment and source (or target) language phrase-structure tree annotations. In GHKM extraction, we first compute the set of the minimally-sized translation rules that can explain the mappings between source language tree and target-language string while respecting the alignment and reordering between the two languages. More complex rules are then learned by composing two or more minimal rules. See Figure 1 for rules extracted using GHKM.

One of the advantages of the above model is that non-terminals in tree-to-string rules are linguistically



(a) decoding with Hiero rules only

(b) decoding with Hiero and tree-to-string rules

Figure 2: Overview of the Hiero baseline (a) and our approach (b). ⇒ means input or output of the decoder. t-to-s is a short for tree-to-string.

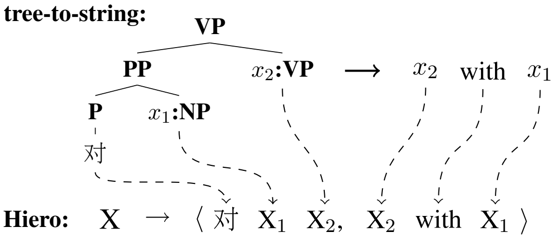


Figure 3: Converting the tree-to-string rule r_8 from Figure 1 to a Hiero-style rule.

motivated and can span word sequences with arbitrary length. Also, one can use rules with consecutive (or more than two) source language non-terminals when the source language parse tree is available. For example, r_8 in Figure 1 has a good Chinese syntactic structure indicating the reordered translations of NP and VP. However, such a rule would not normally be included in a Hiero grammar, as it would require consecutive source language non-terminals (see Figure 3).

3 The Proposed Approach

Both the tree-to-string model and the hierarchical phrase-based model have their own strengths and weaknesses. For example, tree-to-string systems are good at modelling long distance reordering, while hierarchical phrase-based systems are relatively more powerful in handling ill-formed sentences[1] and free translations (Zhao and Al-Onaizan, 2008; Vilar et al., 2010). Here we present a method to enhance hierarchical phrase-based systems with tree-to-string rules and benefit from both models. The idea is simple: we obtain both the tree-to-string grammar and the Hiero-style SCFG from the training data, and then use tree-to-string rules as additional rules in decoding with the SCFG.

Figure 2 shows an overview of our approach and the usual hierarchical phrase-based approach. Our approach requires source language parse trees to be input in both rule extraction and decoding. In rule extraction, we acquire tree-to-string rules using the GHKM method and Hiero-style rules using the Hiero-style rule extraction method to form a larger SCFG. Then, we make use of both the input string and parse tree to decode with the SCFG rules. We now describe our approach.

3.1 Transforming Tree-to-String Rules into SCFG Rules

As described in Section 2, tree-to-string rules have a different form from that of SCFG rules. We will use tree-to-string rules in our hierarchical phrase-based systems by converting each tree-to-string rule into an SCFG rule. The purpose of doing this is to make tree-to-string rules directly accessible to the Hiero-style decoder which performs decoding with SCFG rules.

The rule mapping is straightforward: given a tree-to-string rule $\langle s_r, t_r, \sim \rangle$, we take the frontier nodes of s_r as the source language part of the right hand side of the resulting SCFG rule, and keep t_r and $\sim$ unchanged. Then we replace the non-terminal label with that used in the hierarchical phrase-based system (e.g., X). See Figure 3 for rule mapping of rule r_8 of Figure 1.

In this way, every tree-to-string rule is associated with exactly one SCFG rule. Therefore we can obtain a larger SCFG by combining the rules from the original Hiero-style SCFG and the transformed tree-to-string rules. As explained next, to prevent computational problems we will apply these new rules

[1] For example, the parser fails for 4% of the sentences in our training corpus, and 3% and 6% of the newswire and web development/test sentences, indicating that the data is sometimes ill-formed.

only on the spans that are consistent with the input parse trees. The main goal is to use the tree and the adapted tree-to-string rules to provide the decoder with new linguistically-sensible translation hypotheses that may be prevented by the usual Hiero constraints, and to do so without incurring a computational explosion.

We categorize SCFG rules into two categories based on their availability in Hiero and GHKM extraction. If an SCFG rule is obtained from Hiero extraction, it is a *type 1* rule; If not (i.e., this rule is only available in GHKM extraction), it is a *type 2* rule. E.g., the SCFG rule in Figure 3 is a type 2 rule because it is not available in the original Hiero-style SCFG but can be generated from the tree-to-string rule.

Next we describe how each of these rule types are applied in decoding. We also describe which features are used and how they are computed for each rule type.

3.2 Decoding

Both types of SCFG rules can be employed by usual Hiero decoders with a slight modification. Here we follow the description of Hiero decoding by Iglesias et al. (2011). The source sentence is parsed under the Hiero grammar using the CYK algorithm. Each cell in the CYK grid has associated with it a list of rules that apply to its span; these rules are used to construct a recursive transition network (RTN) which represents all translations of the source sentence under the grammar. The RTN is expanded to a weighted finite state automaton for composition with n-gram language models (de Gispert et al., 2010). Translations are produced via shortest path computation.

This procedure accommodates type 1 rules directly. For tree-to-string rules associated with type 2, we attempt to match rules to the source syntactic tree. If a match is found: the source span of the matching tree fragment is noted and the CYK cell for that span is selected; the tree-to-string rule is converted to a Hiero-style rule; and that rule is added to the list of rules in the selected CYK cell. Once this process is finished, RTN construction, expansion, and language model composition proceeds as usual. Similar modifications could be made to incorporate these rules into cube pruning (Chiang, 2007), cube growing (Huang and Chiang, 2007), and PDT intersection and expansion (Iglesias et al., 2011). We now elaborate on the rule matching strategy.

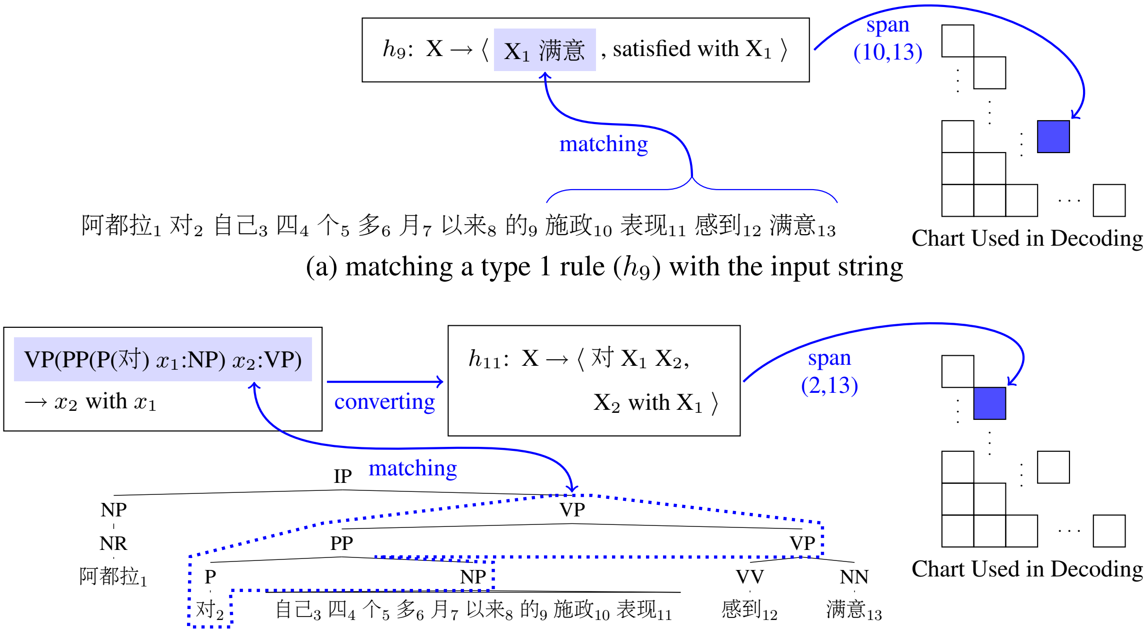
Type 1 Rules The source sentence is parsed as is usual in Hiero-style translation, with the exception that we impose no span limit on rule applications for source spans corresponding to constituents in the Chinese syntactic tree. Rule matching, the procedure that determines if a rule applies to a source span, is based on string matching (see Figure 4(a)). For example, the type 1 rule h_9 in Figure 4(c) can be applied to spans (1,13) and (2,13) since both of them agree with tree constituents (see Figure 4(b)). But h_9 is not applied to span (3,13) because that span is longer than 10 words and agrees with no syntactic tree constituent.

Type 2 Rules If the source side of a tree-to-string rule matches an input tree fragment: 1) that rule is converted to a Hiero-style SCFG rule (Section 3.1); and 2) the Hiero-style rule is added to the rules linked with the CYK grid cell associated with the span of the source syntactic tree fragment. Here, rules are applied via tree matching. For example, rule h_{11} in Figure 4(b) matches the tree fragment spanning positions (2,13).

It is worth noting that some type 1 rules may be found via both Hiero-style and tree-to-string grammar extraction. In this case we monitor whether a rule can be applied as a tree-to-string rule using tree-matching so that features (Section 3.3) and weights can be set appropriately. As an example, rule h_{10} in Figure 4 is available in both extraction methods. For span (2,11), this rule can be matched via both string matching and tree matching. We then note that we can apply h_{10} as a tree-to-string rule for span (2, 11) and activate the corresponding features defined in Section 3.3. For other spans (e.g., spans (2,3)-(2,10)), no tree fragments can be matched and the baseline features are used for h_{10}.

3.3 Features

The baseline feature set used in this work consists of 12 features (Pino et al., 2013), including a 4-gram language model, a strong 5-gram language model, bidirectional translation probabilities, bidirectional lexical weights, a word count, a phrase count, a glue rule count, a frequency-1 rule count, a frequency-2


h_9: X → ⟨ X$_1$ 满意 , satisfied with X$_1$ ⟩
span
(10,13)
matching
阿都拉$_1$ 对$_2$ 自己$_3$ 四$_4$ 个$_5$ 多$_6$ 月$_7$ 以来$_8$ 的$_9$ 施政$_{10}$ 表现$_{11}$ 感到$_{12}$ 满意$_{13}$
Chart Used in Decoding
VP(PP(P(对) x_1:NP) x_2:VP)
→ x_2 with x_1
converting
h_{11}: X → ⟨ 对 X$_1$ X$_2$,
X$_2$ with X$_1$ ⟩
span
(2,13)
matching
IP
NP
NR
阿都拉$_1$
VP
PP
VP
P
对$_2$
NP
自己$_3$ 四$_4$ 个$_5$ 多$_6$ 月$_7$ 以来$_8$ 的$_9$ 施政$_{10}$ 表现$_{11}$
VV
感到$_{12}$
NN
满意$_{13}$
Chart Used in Decoding

(a) matching a type 1 rule (h_9) with the input string

(b) matching a type 2 rule (h_{11}) with the input parse tree

ID	Type	Hiero-style Rule	Tree-to-string Rule	Applicable Spans
h_8	type 1	X → ⟨ 感到 满意, is satisfied ⟩	N/A	(12,13)
h_9	type 1	X → ⟨ X$_1$ 满意, satisfied with X$_1$ ⟩	N/A	(i,13), $i = 1, 2$ or $4 \leq i \leq 12$
h_{10}	type 1	X → ⟨ 对 X$_1$, with X$_1$ ⟩	PP(P(对) x_1NP) → with NPx_1	(2,j), $3 \leq j \leq 11$ or $j = 13$
h_{11}	type 2	X → ⟨ 对 X$_1$ X$_2$, X$_2$ with X$_1$ ⟩	VP(PP(P(对) x_1:NP) x_2:VP) → x_2 with x_1	(2,13)

(c) example rules used in decoding

Figure 4: Decoding with both Hiero-style and tree-to-string grammars (span limit = 10). A span (i,j) means spanning from position i to position j.

rule count, and a larger-than-frequency-2 rule count [2]. In addition, we introduce several features for applying tree-to-string rules.

- **Rule type indicators**. We consider four indicator features, indicating tree-to-string rules, lexicalized tree-to-string rules, rules with consecutive non-terminals, and non-lexicalized rules. Note that the tree-to-string rule indicator feature is in principle a generalization of the soft syntactic features (Marton and Resnik, 2008), in that a bonus (or penalty) is applied when a rule application is consistent with a source tree constituent. The difference lies in that the tree-to-string rule indicator feature does not distinguish between different syntactic labels, whereas soft syntactic features do.

- **Features in syntactic MT**. In general tree-to-string rules have their own features which are different from those used in Hiero-style systems. For example, the features in syntactic MT systems can be defined as the generation probabilities conditioned on the root symbol of the tree-fragment. Here we choose five popular features used in syntactic MT systems, including the bi-directional phrase-based conditional translation probabilities (Marcu et al., 2006) and three syntax-based conditional probabilities (Mi and Huang, 2008). All these probabilities can be computed by relative-frequency estimates. For example, the phrase-based features are the probabilities of translating between the frontier nodes of s_r and t_r. The syntax-based features are the probabilities of generating r conditioned on its root,

[2] We experimented with soft syntactic features (Marton and Resnik, 2008) but found no improvement over our baseline system.

source and target language sides, respectively. More formally, we use the following estimates for these probabilities:

$$\begin{aligned}
\mathrm{P}_{phr}(t_r \mid s_r) &= \frac{\sum_{r'':\varphi(s_{r''})=\varphi(s_r)\wedge t_{r''}=t_r} c(r'')}{\sum_{r':\varphi(s_{r'})=\varphi(s_r)} c(r')} \\
\mathrm{P}_{phr}(s_r \mid t_r) &= \frac{\sum_{r'':\varphi(s_{r''})=\varphi(s_r)\wedge t_{r''}=t_r} c(r'')}{\sum_{r':t_{r'}=t_r} c(r')} \\
\mathrm{P}(r \mid root(r)) &= \frac{c(r)}{\sum_{r':root(r')=root(r)} c(r')} \\
\mathrm{P}(r \mid s_r) &= \frac{c(r)}{\sum_{r':s_{r'}=s_r} c(r')} \\
\mathrm{P}(r \mid t_r) &= \frac{c(r)}{\sum_{r':t_{r'}=t_r} c(r')}
\end{aligned}$$

where $c(r)$ is the count of r, and $root(\cdot)$ and $\varphi(\cdot)$ are functions that return the source root symbol for a tree-to-string rule and the sequence of leaf nodes for a tree-fragment respectively.

4 Evaluation

4.1 Experimental Setup

We report results in the NIST MT12 Chinese-English task, where our baseline system was among the top academic systems. The parallel training corpus consists of 9.2 million sentence pairs which are provided within the NIST Chinese-English MT12 track. Word alignments are obtained using MTTK (Deng and Byrne, 2008) in both Chinese-to-English and English-to-Chinese directions, and then unioning the links. The data from newswire and web genres was used for tuning and test. The development sets contain 1,755 sentences and 2160 sentences for the two genres respectively. The test sets (newswire: 1,779 sentences, web: 1768 sentences) contain all newswire and web evaluation data of MT08 (mt08), MT12 (mt12), and MT08 progress test (mt08.p). All Chinese sentences in the training, development and test sets were parsed using the Berkeley parser (Petrov and Klein, 2007). A Kneser-Ney 4-gram language model was trained on the AFP and Xinhua portions of the English Gigaword in addition to the English side of the parallel corpus. A stronger 5-gram language model was trained on all English data of NIST MT12 and the Google counts corpus using the ”stupid” backoff method (Brants et al., 2007).

For decoding we use HiFST, which is implemented with weighted finite state transducers (de Gispert et al., 2010). A two-pass decoding strategy is adopted; first, only the 4-gram language model and the translation model are activated; and then, the 5-gram language model is applied for second-pass rescoring of the translation lattices generated by the first-pass decoding stage. We extracted SCFG rules from the parallel corpus using the standard heuristics (Chiang, 2007) and filtering strategies (Iglesias et al., 2009). The span limit was set to 10 in extracting basic phrases and decoding. All features weights were optimized using lattice-based minimum error rate training (Macherey et al., 2008).

For tree-to-string extraction, we used a reimplementation of the GHKM method (Xiao et al., 2012) and extracted rules from a 600K-sentence portion of the parallel data. To prune the tree-to-string rule set, we restricted the extraction to rules with at most 5 frontier non-terminals and 5 terminals. Also, we discarded lexicalized rules with a Chinese-to-English translation probability of $<$ 0.02 and non-lexicalized rules with a Chinese-to-English translation probability of $<$ 0.10.

4.2 Results

We report MT performance in Table 1 by case-insensitive BLEU (Papineni et al., 2002). The experiments are organized as follows:

- Baseline and Span Limits (exp01 and exp02)
  First we study the effect of removing the span limit for tree constituents, that is, SCFG rules can be

Entry	System	Newswire					Web				
		tune	mt08	mt12	mt08.p	all test	tune	mt08	mt12	mt08.p	all test
		(1755)	(691)	(400)	(688)	(1779)	(2160)	(666)	(420)	(682)	(1768)
exp01	baseline	35.84	35.85	35.47	35.50	35.63	29.98	25.15	23.07	27.19	25.33
exp02	+= no span limit	36.05	36.08	35.70	35.54	35.79	30.11	25.28	23.08	27.17	25.37
exp03	+= t-to-s rules	36.63	36.51	36.08	36.09	36.25*	30.80	26.00	23.08	27.80	25.83
exp04	+= t-to-s features	36.82	36.49	36.53	36.16	36.38*	30.91	26.03	23.27	27.85	25.98*
exp05	t-to-s baseline	34.63	34.44	34.87	33.66	34.25*	28.30	23.40	21.38	25.30	23.56*
exp06	exp04 on spans $>$ 10	36.17	36.11	35.71	35.86	35.92	30.18	25.30	23.12	27.36	25.45
exp07	exp04 with null trans.	36.10	36.03	35.35	34.86	35.42	29.96	25.32	22.58	23.33	24.12*
exp08	exp04 + left binariz.	37.11	37.46	37.03	36.30	36.91*	31.18	26.15	23.54	27.98	26.13*
exp09	exp04 + right binariz.	36.58	36.56	36.41	35.70	36.20*	31.06	25.94	23.47	27.48	25.88*
exp10	exp04 + forest binariz.	37.03	37.27	37.09	36.62	36.98*	31.20	25.99	23.59	28.09	26.15*

Table 1: Case-insensitive BLEU[%] scores of various systems. += means incrementally adding methods/features to the previous system. * means that a system is significantly different than the exp01 baseline at $p < 0.01$.

applied to any spans when they respect the tree constituents of the input tree. It can be regarded as the simplest way of using source syntax in Hiero-style systems. Seen from Table 1, removing the span limit shows modest BLEU improvements. It agrees with the previous result that loosening the constraints on spans is helpful to systems based on the hard syntactic constraints (Li et al., 2013).

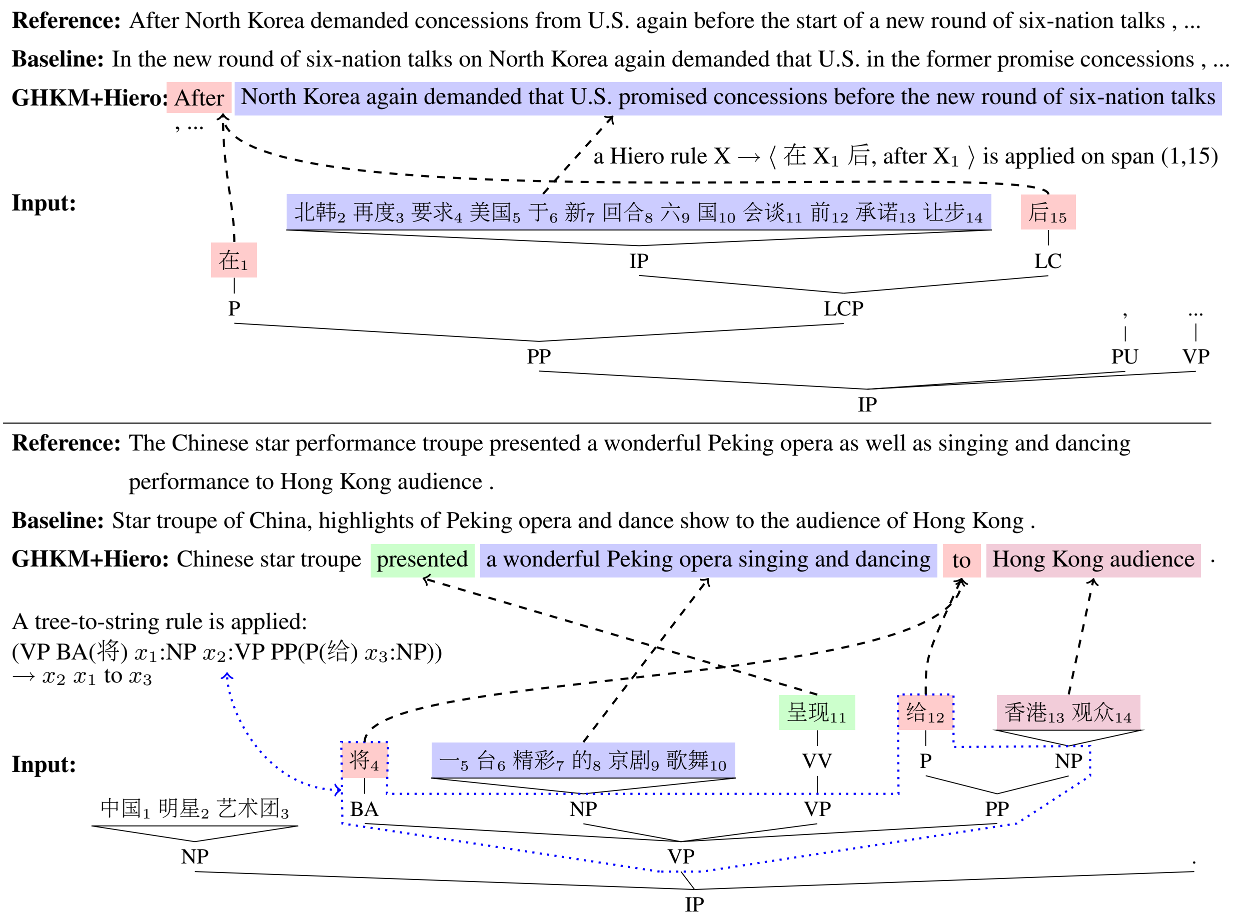
- GHKM+Hiero (exp03 and exp04)
  The results of our proposed approach (w/o new features) are reported in exp03 and exp04. We see that incorporating tree-to-string rules yields +0.6 and +0.5 improvements on the collected newswire and web test sets (exp03 vs exp01). The new features (Section 3.3) give a further improvement (exp04 vs exp03). This result confirms that the system can learn a preference for certain types of rules using the new features.

- Impact of Search Space (exp05)
  We also study the impact of search space on system performance. To do this, we force the improved system (exp04) to respect source tree constituents and to discard any hypotheses which violate the tree constituent constraints. Seen from exp05, this system has a lower BLEU score than both the Hiero baseline (exp01) and GHKM+Hiero system (exp04), strongly suggesting that restricting MT systems to a smaller space of hypotheses is harmful.

- GHKM+Hiero, Spans $>$ 10 Only (exp06)
  Another interesting question is whether tree-to-string rules and features are more helpful to larger spans. We restricted our approach to spans $>$ 10 only and conducted another experiment. As is shown in exp06, applying tree-to-string rules and features for large spans is beneficial (exp06 vs. exp01). But it underperforms the system with the full use of tree-to-string rules (exp06 vs. exp04). This interesting observation implies that applying tree-to-string rules on smaller spans introduces good hypotheses that can be selected with our additional features.

- Impact of Failed Parses (exp07)
  As noted in Section 3, the parser fails to parse some of the sentences in our experiments. In this case our approach generates the baseline result using the Hiero model (i.e., type 1 rules only). To investigate the effect of failed parse trees on system performance, we also report the BLEU score including null translations for which the parser fails. As shown in exp07, there are significantly lower BLEU scores when null translations are included. It indicates that our approach is more robust than standard tree-to-string systems which would generate an empty translation if the source language parser fails.

- Results on Binarization (exp08-10)
  Tree binarization is a widely used method to improve syntactic MT systems (Wang et al., 2010). exp08-10 show the results of our improved system with left-heavy, right-heavy and forest-based bina-



Reference: After North Korea demanded concessions from U.S. again before the start of a new round of six-nation talks , ...

Baseline: In the new round of six-nation talks on North Korea again demanded that U.S. in the former promise concessions , ...

GHKM+Hiero: After North Korea again demanded that U.S. promised concessions before the new round of six-nation talks , ...

a Hiero rule X → ⟨ 在 X₁ 后, after X₁ ⟩ is applied on span (1,15)

Reference: The Chinese star performance troupe presented a wonderful Peking opera as well as singing and dancing performance to Hong Kong audience .

Baseline: Star troupe of China, highlights of Peking opera and dance show to the audience of Hong Kong .

GHKM+Hiero: Chinese star troupe presented a wonderful Peking opera singing and dancing to Hong Kong audience .

A tree-to-string rule is applied:
(VP BA(将) x_1:NP x_2:VP PP(P(给) x_3:NP))
→ x_2 x_1 to x_3

Figure 5: Comparison of translations generated by the baseline and improved systems.

rization[3]. We see that left-heavy binarization is very helpful and exp08 achieves overall improvements of 1.2 and 0.8 BLEU points on the newsire and web data. In contrast, right-heavy binarization does not yield promising performance. This agrees with the previous report (Wang et al., 2010) that MT systems prefer to use certain ways of binarization in most cases. exp10 shows that the additional trees introduced in our forest-based scheme are not sufficient to make a big impact on BLEU scores. Possibly larger gains can be obtained if taking a forest of parse trees from the source parser, but this is outside the scope of this paper.

4.3 Analysis

We then analyse rule usage in the 1-best derivations for our improved system on the tuning set. We find that type 2 rules represent 13.97% of the rules used in the 1-best derivations. Also, 44.45% of the applied rules are available from the tree-to-string model (i.e., rules that use the features described in Section 3.3). These numbers indicate that the tree-to-string rules are beneficial and our decoder likes to use them.

Finally, we discuss two real translation examples from our tuning set. See Figure 5 for translations generated by different systems. In the first example, the Chinese input sentence contains 在 ... 后 which is usually translated into *after* ... (i.e., a Hiero rule X → ⟨ 在 X₁ 后, after X₁ ⟩). However, because the " 在 ... 后" pattern spans 15 words and that is beyond the span limit, our baseline is unable to apply this desired rule and chooses a wrong translation *in* for the Chinese word 在. When the source parse tree

[3] We found that the CTB-style parse trees usually have a very flat top-level IP (i.e., single clause) tree structure. As the IP structure in Chinese is very complicated, the system might prefer a flexible binarization scheme. Thus we considered both left and right-heavy binarization to form a binarization forest for IPs in Chinese parse trees, and binarized other tree constituents in a left-heavy fashion.

is available, our approach removes the span limit for spans that agree with the tree constituents. In this case, the MT system successfully applies the rule on span (1, 15) and generates a much better translation.

In the second example, the translation of the input sentence requires complex reordering of adjacent constituents. The baseline system cannot handle this case and generates a monotonic translation using the glue rules. This results in a wrong order for the translation of Chinese verb 呈现 (*show*). By contrast, the improved system chooses a tree-to-string rule with three non-terminals (some of which are adjacent in the source language) and perfectly performs a syntactic movement of the required tree constituents.

5 Related Work

Recently linguistically-motivated models have been intensively investigated in MT. In particular, source tree-based models (Liu et al., 2006; Huang et al., 2006; Eisner, 2003; Zhang et al., 2008; Liu et al., 2009a; Xie et al., 2011) have received growing interest due to their good abilities in modelling source language syntax for better lexicon selection and reordering. Alternatively, the hierarchical phrase-based approach (Chiang, 2005) considers the underlying hierarchical structures of sentences but does not require linguistically syntactic trees on either language side.

There are several lines of work for augmenting hierarchical phrase-based systems with the use of source language phrase-structure trees. Liu et al. (2009b) describe novel approaches to translation under multiple translation grammars. Their approach is very much motivated by system combination, and they develop procedures for joint decoding and optimisation within a single system that give the benefit of combining hypotheses from multiple systems. They demonstrate their approach by combining full tree-to-string and Hiero systems. Our approach is much simpler and emphasises changes to the grammar rather than the decoder or its parameter optimisation (MERT). Our aim is to augment the search space of Hiero with linguistically-motivated hypotheses, and not to develop a new decoder that is capable of translation under multiple grammars. Moreover, we consider Hiero as the backbone model and only introduce tree-to-string rules where they can contribute; we show that extracting tree-to-string rules from just 10% of the data suffices to get good gains. This results in a small number of tree-to-string rules and does not slow down the decoder.

Another related line of work is to introduce syntactic constraints or annotations to hierarchical phrase-based systems. Marton and Resnik (2008) and Li et al. (2013) proposed several soft or hard constraints to model syntactic compatibility of Hiero derivations and input source language parse trees. We note that, despite significant development effort, we were not able to improve our baseline through the use of these soft syntactic constraints; it was this experience that led us to develop the hybrid approach described in this paper.

Several research groups used syntactic labels as non-terminal symbols in their SCFG rules and develop new features (Zollmann and Venugopal, 2006; Zhao and Al-Onaizan, 2008; Chiang, 2010; Hoang and Koehn, 2010). However, all these methods still resort to rule extraction procedures similar to that of the standard phrase/hierarchical rule extraction method. In contrast, we use the GHKM method which is a mature technique to extract rules from tree-string pairs but does not impose those Hiero-style constraints on rule extraction. More importantly, we consider the hierarchical syntactic tree structure to make use of well-formed rules in decoding, while such information is not used in standard SCFG-based systems. We also keep to the simpler non-terminals of Hiero, and do not ‘decorate’ any non-terminals with syntactic or other information.

6 Conclusion

We have presented an approach to improving Hiero-style systems by augmenting the SCFG with tree-to-string rules and syntax-based features. The input parse trees are used to introduce new linguistically-sensible hypotheses into the translation search space while maintaining the Hiero robustness qualities and avoiding computational explosion. We obtain significant improvements over a strong Hiero baseline in Chinese-to-English. Further improvements are achieved when applying tree binarization.

Acknowledgements

This work was done while the first author was visiting the speech group at University of Cambridge, and was supported in part by the National Science Foundation of China (Grants 61272376 and 61300097), and the China Postdoctoral Science Foundation (Grant 2013M530131). We would like to thank the anonymous reviewers for their pertinent and insightful comments. We also would like to thank Juan Pino, Rory Waite, Federico Flego and Gonzalo Iglesias for building parts of the baseline system.

References

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large Language Models in Machine Translation. In *Proceedings of EMNLP-CoNLL*, pages 858–867, Prague, Czech Republic.

David Chiang. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of ACL*, pages 263–270, Ann Arbor, Michigan, USA.

David Chiang. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33:45–60.

David Chiang. 2010. Learning to Translate with Source and Target Syntax. In *Proceedings of ACL*, pages 1443–1452, Uppsala, Sweden.

Adrià de Gispert, Gonzalo Iglesias, Graeme Blackwood, Eduardo R. Banga, and William Byrne. 2010. Hierarchical Phrase-Based Translation with Weighted Finite-State Transducers and Shallow-n Grammars. *Computational Linguistics*, 36(3):505–533.

Yonggang Deng and William Byrne. 2008. HMM Word and Phrase Alignment for Statistical Machine Translation. *IEEE Transactions on Audio, Speech & Language Processing*, 16(3):494–507.

Jason Eisner. 2003. Learning Non-Isomorphic Tree Mappings for Machine Translation. In *Proceedings of ACL*, pages 205–208, Sapporo, Japan.

Yang Feng, Yang Liu, Qun Liu, and Trevor Cohn. 2012. Left-to-Right Tree-to-String Decoding with Prediction. In *Proceedings of EMNLP-CoNLL*, pages 1191–1200, Jeju Island, Korea.

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*, pages 961–968, Sydney, Australia.

Hieu Hoang and Philipp Koehn. 2010. Improved translation with source syntax labels. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 409–417, Uppsala, Sweden.

Liang Huang and David Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proceedings of ACL*, pages 144–151, Prague, Czech Republic.

Liang Huang and Haitao Mi. 2010. Efficient Incremental Decoding for Tree-to-String Translation. In *Proceedings of EMNLP*, pages 273–283, Cambridge, MA, USA.

Liang Huang, Knight Kevin, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of AMTA*, pages 66–73, Cambridge, MA, USA.

Gonzalo Iglesias, Adrià de Gispert, Eduardo R. Banga, and William Byrne. 2009. Rule Filtering by Pattern for Efficient Hierarchical Translation. In *Proceedings of EACL*, pages 380–388, Athens, Greece.

Gonzalo Iglesias, Cyril Allauzen, William Byrne, Adrià de Gispert, and Michael Riley. 2011. Hierarchical Phrase-based Translation Representations. In *Proceedings of EMNLP*, pages 1373–1383, Edinburgh, Scotland, UK.

Junhui Li, Philip Resnik, and Hal Daumé III. 2013. Modeling Syntactic and Semantic Structures in Hierarchical Phrase-based Translation. In *Proceedings of NAACL-HLT*, pages 540–549, Atlanta, Georgia.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In *Proceedings of COLING-ACL*, pages 609–616, Sydney, Australia.

Yang Liu, Yajuan Lü, and Qun Liu. 2009a. Improving Tree-to-Tree Translation with Packed Forests. In *Proceedings of ACL-IJCNLP*, pages 558–566, Suntec, Singapore.

Yang Liu, Haitao Mi, Yang Feng, and Qun Liu. 2009b. Joint decoding with multiple translation models. In *Proceedings of ACL-IJCNLP*, pages 576–584, Suntec, Singapore.

Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based Minimum Error Rate Training for Statistical Machine Translation. In *Proceedings of EMNLP*, pages 725–734, Honolulu, Hawaii.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical Machine Translation with Syntactified Target Language Phrases. In *Proceedings of EMNLP*, pages 44–52, Sydney, Australia.

Yuval Marton and Philip Resnik. 2008. Soft Syntactic Constraints for Hierarchical Phrased-Based Translation. In *Proceedings of ACL-HLT*, pages 1003–1011, Columbus, Ohio.

Haitao Mi and Liang Huang. 2008. Forest-based Translation Rule Extraction. In *Proceedings of EMNLP*, pages 206–214, Honolulu, Hawaii, USA.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-Based Translation. In *Proceedings of ACL-HLT*, pages 192–199, Columbus, Ohio.

Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL*, pages 311–318, Philadelphia, PA, USA.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of HLT-NAACL*, pages 404–411, Rochester, New York, USA.

Juan Pino, Aurelien Waite, Tong Xiao, Adrià de Gispert, Federico Flego, and William Byrne. 2013. The University of Cambridge Russian-English system at WMT13. In *Proceedings of WMT*, pages 200–205, Sofia, Bulgaria.

David Vilar, Daniel Stein, Stephan Peitz, and Hermann Ney. 2010. If i only had a parser: poor man's syntax for hierarchical machine translation. In *Proceedings of IWSLT*, pages 345–352.

Wei Wang, Jonathan May, Kevin Knight, and Daniel Marcu. 2010. Re-structuring, Re-labeling, and Re-aligning for Syntax-Based Machine Translation. *Computational Linguistics*, 36(2):247–277.

Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. NiuTrans: An Open Source Toolkit for Phrase-based and Syntax-based Machine Translation. In *Proceedings of ACL: System Demonstrations*, pages 19–24, Jeju Island, Korea.

Jun Xie, Haitao Mi, and Qun Liu. 2011. A novel dependency-to-string model for statistical machine translation. In *Proceedings of EMNLP*, pages 216–226, Edinburgh, Scotland.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A Tree Sequence Alignment-based Tree-to-Tree Translation Model. In *Proceedings of ACL-HLT*, pages 559–567, Columbus, Ohio, USA.

Bing Zhao and Yaser Al-Onaizan. 2008. Generalizing Local and Non-Local Word-Reordering Patterns for Syntax-Based Machine Translation. In *Proceedings of EMNLP*, pages 572–581, Honolulu, Hawaii.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax Augmented Machine Translation via Chart Parsing. In *Proceedings of WMT*, pages 138–141, New York City.