Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Syntactic Skeleton-Based Translation

Tong Xiao[1,2], **Jingbo Zhu**[1,2], **Chunliang Zhang**[1,2]
[1]Northeastern University, Shenyang 110819, China
[2]YaTrans Co., Ltd., Shenyang 110004, China

Tongran Liu[3]
[3]Institute of Psychology (CAS)
Beijing 100101, China

Abstract

In this paper we propose an approach to modeling syntactically-motivated skeletal structure of source sentence for machine translation. This model allows for application of high-level syntactic transfer rules and low-level non-syntactic rules. It thus involves fully syntactic, non-syntactic, and partially syntactic derivations via a single grammar and decoding paradigm. On large-scale Chinese-English and English-Chinese translation tasks, we obtain an average improvement of +0.9 BLEU across the newswire and web genres.

Introduction

The use of source-language syntax and structural information has been a popular approach to replacing surface string translation with learning of tree-string mappings from parsed data. Unlike word or phrase-based models, beginning in the 1990s, source-language syntactic models rely on parse trees of input sentence. This enhances the ability of handling long-distance movement and complex reordering of multiple constituents (Liu, Liu, and Lin 2006; Huang, Kevin, and Joshi 2006). However, source tree-based models have robustness problems in that sparse and limited-coverage rules can lead to poor translation. A straightforward implementation has been found to underperform the (hierarchical) phrase-based counterpart (Liu et al. 2009).

In machine translation (MT), the power of source syntax lies in its good ability of modeling the skeletal structure of input sentence (Liu, Liu, and Lin 2006; Xiao et al. 2014). This property is very promising if we use MT systems as analogies to human translation - given a source sentence, we first have a high-level structure/pattern of the sentence in mind with our syntactic knowledge. Then, we determine the translation and order of the key elements in this structure/pattern, and then finish the remaining job of lexical selection and local reordering. As sentence structures can be well explained with the language syntax, a natural issue that arises is that: can we apply source syntax to where it can contribute most - i.e., translating the skeletal structure of the source sentence - and meanwhile benefit from (hierarchical) phrase-based models on non-syntactic segments?

To address this question, we propose an approach to learning to translate with a special syntactic structure (call it *syntactic skeleton* or *skeleton* for short) which models the high-level source syntax for MT. It combines two merits in one framework: 1) applying tree-to-string rules to translate the syntactic skeleton; 2) and applying hierarchical phrase-based rules to handle the lower-level lexical translation and reordering.

Our model is very flexible. It involves fully syntactic, non-syntactic, and partially syntactic derivations via a single grammar and decoding paradigm. Thus the hierarchical phrase-based and tree-to-string systems can be cast as two special cases of this approach.

In addition, our model fits in the general framework of synchronous context-free grammars (SCFGs). It is very easy to implement and speed-up if one already has an SCFG decoder. On large-scale Chinese-English and English-Chinese translation tasks, we obtain an average improvement of +0.9 BLEU across different genres.

Background

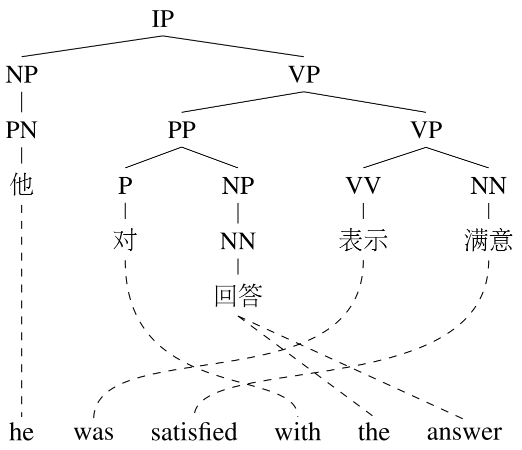
SCFG and Hiero-Style Translation

This work is based on synchronous context-free grammars which have been widely used in statistical machine translation (SMT). More formally, an SCFG rule can be represented as: LHS $\rightarrow \langle \alpha, \beta, \sim \rangle$, where the left hand side LHS is a non-terminal, α and β are sequences of terminals and non-terminals in the source and target languages, $\sim$ is a 1-to-1 alignment between the non-terminals in α and β.

Probabilistic SCFGs can be learned from unparsed, word-aligned parallel data using Hiero-style heuristics and constraints (Chiang 2007). Once the SCFG is obtained, new sentences can be decoded by finding the most likely derivation of SCFG rules. See Figure 1 for example Hiero-style rules extracted from a sentence pair with word alignments, where the non-terminals are labeled with X only. A sequence of such rules covering the source sentence is an SCFG derivation, e.g., rules h_5, h_1 and h_3 generate a derivation for the sentence pair.

Learning Translation from Source Syntax

Hiero-style grammars are formally syntactic, but rules are not constrained by source (or target) language syntax. A natural extension is to use the source-language parse tree to



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 对 X_2 表示 满意, X_1 was satisfied with X_2⟩
...	

Rules Transformed from GHKM Extraction

r_1	NP → ⟨ 他, he ⟩
r_2	NP → ⟨ 回答, the answer ⟩
r_3	VP → ⟨ 表示 满意, was satisfied ⟩
r_4	IP → ⟨ NP_1 VP_2, NP_1 VP_2 ⟩
r_5	VP → ⟨ 对 NP_1 VP_2, VP_2 with NP_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.

guide the rule extraction and translation. To do this, a popular way is to perform the GHKM rule extraction (Galley et al. 2006) on the bilingual sentences with both word alignment and source-language phrase structure tree annotations.

In the GHKM approach, translation equivalency relations are modeled from source-language syntactic trees to target language strings using derivations of GHKM rules (Liu, Liu, and Lin 2006). A GHKM rule is a tuple of a source tree-fragment s_r, a target string t_r, and the alignments between the frontier non-terminals of s_r and t_r, for example,

$$\text{VP(VV(提高) } x_1\text{:NN)} \rightarrow \text{increases } x_1$$

is a GHKM rule. We can transform this rule into the SCFG form by keeping the frontier non-terminal annotations and discarding the internal tree structure (Xiao et al. 2014), like this

$$\text{VP} \rightarrow \langle \text{ 提高 } \text{NN}_1, \text{ increases NN}_1 \ \rangle$$

In this work we refer to the SCFG rules transformed from the GHKM rules as tree-to-string rules. As the non-terminals are annotated with source syntactic labels, applying tree-to-string rules are constrained to the ”well-formed” constituents. See Figure 1 for tree-to-string rules extracted from a tree-string pair. Note that the tree-to-string rules used here ignore the multi-level tree structure of the original GHKM rule but keep the frontier nodes. It gives us a generation ability on new sentences (Zhu and Xiao 2011).

Decoding with tree-to-string rules can be seen as an instance of SCFG parsing. A popular method is string-parsing (or string-based decoding) which parses the input sentence with a chart decoder (e.g., the CYK decoder). Also, we can decode the parse tree using tree-parsing (or tree-based decoding) if source-language parse is available on the test data. In this way, source syntax is used to impose hard constraints that derivations must respect the constituents of the input parse tree.

The Model

It is well-known that Hiero-style systems and tree-to-string systems have different strengths and weaknesses (Cmejrek, Mi, and Zhou 2013; Xiao et al. 2014). For example, Hiero-style models are powerful in lexical selection and reordering that is inherent in lexicalized rules but have several constraints to complex constituent movement[1]. Tree-to-string models characterize the movement of hierarchical structures by linguistic notions of syntax and is promising in high-level syntactic reordering. But they suffer from the sparsity and limited coverage problems.

In an ideal case, we can apply the two types of models to where they can contribute most: 1) tree-to-string rules handle higher-level syntactic movement; 2) and Hiero-style rules handle lower-level lexical translation and reordering. To this end, we propose a method to ”combine” the two merits in one model. We reuse the Hiero-style and tree-to-string grammars in translation, and develop a new type of rules - partially syntactic rules - to link tree-to-string rules with Hiero-style rules.

A rule is partially syntactic if its left-hand side (LHS) is a source-language syntactic label and at least one of the non-terminals on the right-hand side (RHS) has the X symbol. See following for a partially syntactic rule

$$\text{VP} \rightarrow \langle \text{ 提高 } \text{X}_1, \text{ increases X}_1 \ \rangle$$

where the left-hand side represents a Verb Phrase (VP), and the right-hand side involves the X non-terminals as in standard Hiero-style rules. We can apply this rule on top of a partial Hiero-style derivation, and produce a derivation rooted at VP. Then, tree-to-string rules can be applied to substitute the VP derivation as usual in syntactic MT systems.

[1] The common constraints are 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.

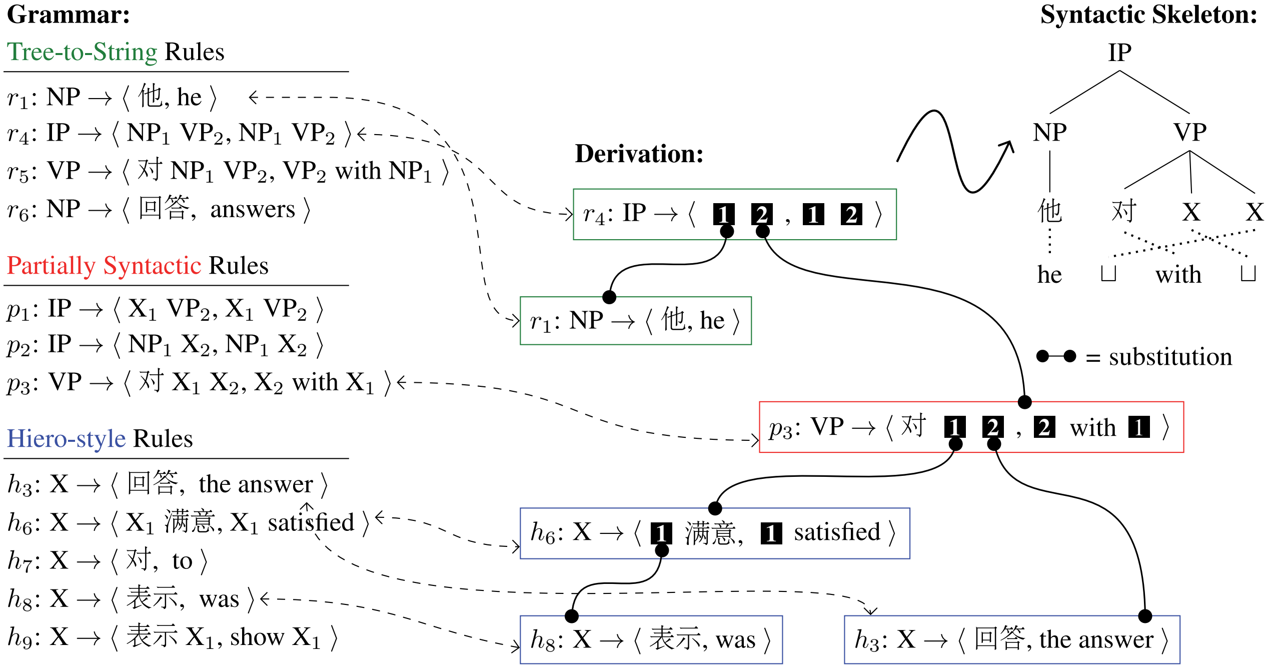

Grammar:
Tree-to-String Rules
r1: NP → ⟨ 他, he ⟩
r4: IP → ⟨ NP1 VP2, NP1 VP2 ⟩
r5: VP → ⟨ 对 NP1 VP2, VP2 with NP1 ⟩
r6: NP → ⟨ 回答, answers ⟩
Partially Syntactic Rules
p1: IP → ⟨ X1 VP2, X1 VP2 ⟩
p2: IP → ⟨ NP1 X2, NP1 X2 ⟩
p3: VP → ⟨ 对 X1 X2, X2 with X1 ⟩
Hiero-style Rules
h3: X → ⟨ 回答, the answer ⟩
h6: X → ⟨ X1 满意, X1 satisfied ⟩
h7: X → ⟨ 对, to ⟩
h8: X → ⟨ 表示, was ⟩
h9: X → ⟨ 表示 X1, show X1 ⟩
Derivation:
r4: IP → ⟨ 1 2 , 1 2 ⟩
r1: NP → ⟨ 他, he ⟩
p3: VP → ⟨ 对 1 2 , 2 with 1 ⟩
h6: X → ⟨ 1 满意, 1 satisfied ⟩
h8: X → ⟨ 表示, was ⟩
h3: X → ⟨ 回答, the answer ⟩
●–● = substitution
Syntactic Skeleton:
IP
NP
VP
他
对
X
X
he
⊔
with
⊔

Figure 2: Derivation and syntactic skeleton generated via a sample grammar.

As partially syntactic rules linkup Hiero-style rules with tree-to-string rules, we can use all these rules to form a partially syntactic derivation. See Figure 2 for a derivation built from a sample grammar of Hiero-style, tree-to-string, and partially syntactic rules. In this derivation, Hiero-style rules (h_3, h_6 and h_8) are applied to lower-level translation. A syntactic structure is created by applying syntactic rules (partially syntactic rule p_3, and tree-to-string rules r_1 and r_4) on top of the partial derivations of X.

The most interesting part of this derivation is the structure created via the source-language sides of the syntactic rules (see the top-right of Figure 2). We call it the syntactic skeleton. Basically it is a high-level syntactic tree-fragment with terminals and non-terminal at leaves. By using this skeleton structure, it is easy to capture the reordering of the constituents in ”对 NP VP”, and leave the translation of ”回答” and ”表示 满意” to Hiero-style rules.

For grammar extraction, learning Hiero-style and tree-to-string rules is trivial because we can resort to the baseline approaches as described in the Background section. To acquire partially syntactic rules, we employ a simple but straightforward method. For each tree-to-string rule, we transform it into partially syntactic rules by generalizing the symbols of one or two non-terminals on the RHS to Xs while keeping LHS unchanged. For example, for rule r_5 in Figure 1 (VP → ⟨对 NP$_1$ VP$_2$, VP$_2$ with NP$_1$⟩), there are three partially syntactic rules:

$$\mathrm{VP} \;\rightarrow\; \langle \text{对 } \mathrm{X}_1\ \mathrm{X}_2,\ \mathrm{X}_2 \text{ with } \mathrm{X}_1 \rangle$$
$$\mathrm{VP} \;\rightarrow\; \langle \text{对 } \mathrm{X}_1\ \mathrm{VP}_2,\ \mathrm{VP}_2 \text{ with } \mathrm{X}_1 \rangle$$
$$\mathrm{VP} \;\rightarrow\; \langle \text{对 } \mathrm{NP}_1\ \mathrm{X}_2,\ \mathrm{X}_2 \text{ with } \mathrm{NP}_1 \rangle$$

Once all rules (i.e., Hiero-style, tree-to-string and partially syntactic rules) are ready, we collect them to build a bigger SCFG and feed it to the decoder. We compute the rule weight in a weighted log-linear fashion. As in the standard SCFG-based model (Chiang 2007), we have the following features for rule LHS $\rightarrow \langle \alpha, \beta, \sim \rangle$:

- Translation probabilities P($\alpha|\beta$) and P($\beta|\alpha$) estimated by relative frequencies[2].
- Lexical weights P$_{lex}(\alpha|\beta)$ and P$_{lex}(\beta|\alpha)$ estimated by Koehn et al.’s (2003) method.
- Rule bonuses (exp(1)) for Hiero-style, tree-to-string and partially syntactic rules individually.
- Indicators (exp(1)) for glue rules, lexicalized rules and non-lexicalized rules, which allow the model to learn a preference for specified types of rules.
- Number of the X non-terminals (exp(#)) on the source-language side of the partially syntactic rule, which controls how often the model violates the syntactic compatibility.

We then define the weight (score) of derivation in our model. Let d be the derivation of the above grammar. To distinguish syntactic rules (i.e., tree-to-string and partially syntactic rules) and Hiero-style rules, we define d as a tuple $\langle d_s, d_h \rangle$, where d_s is the partial derivation for the skeleton, and d_h is the set of rules used to form the remaining

[2] For partially syntactic rules, the rule frequency is the sum of the frequencies of all corresponding tree-to-string rules.

parts of d. For example, in Figure 2, $d_s = \{r_4, r_1, p_3\}$ and $d_h = \{h_6, h_8, h_3\}$.

Let t be the target string encoded in d. Then the score of d is defined as the product of rule weights, with a multiplication of an n-gram language model $lm(t)$ and a word bonus exp($|t|$).

$$\begin{aligned} s(d) &= \prod_{r_i \in d_s} w(r_i) \times \prod_{r_j \in d_h} w(r_j) \times lm(t)^{\lambda_{lm}} \\ &\quad \times \exp(\lambda_{wb} \cdot |t|) \end{aligned}$$

where $w(r_*)$ is the weight of rule r_*, and λ_{lm} and λ_{wb} are the feature weights of the language model and word bonus.

Another note on our model. The framework presented here is very flexible, and includes tree-to-string and Hiero-style models as special cases. E.g., d is a Hiero-style derivation if it consists of Hiero-style rules only (i.e., $d_s = \phi$); and d is a fully syntactic derivation if it consists of tree-to-string rules only (i.e., $d_h = \phi$). What we are doing here is to introduce partially syntactic derivations into a space of Hiero-style and tree-to-string derivations[3]. The decoder can select the best derivation from the enlarged pool of candidates in terms of model score.

Decoding

Decoding with our grammar can be regarded as a string-parsing problem - we parse the source string with the source-language side of the grammar rules and generate the target string using the target-language side. The output is the target string yielded by the derivation with highest model score. In this work we use a CYK-style decoder with beam search and cube pruning (Chiang 2007). It operates on the binarized rules which are obtained via synchronous binarization (Zhang et al. 2006).

As a large number of partially syntactic rules are introduced, the decoding is slow. To speed-up the system, we further prune the search space in several ways. First, we discard lexicalized, partially syntactic rules whose scope is larger than 3 (Hopkins and Langmead 2010). We remove these rules because they are one of the major factors that slow down the system but are not very beneficial to translation. Also, we discard non-lexicalized, partially syntactic rules with X non-terminals on the RHS only. This type of rules does not make ”syntactic sense” in most cases. E.g., rule VP → ⟨ X_1 X_2, X_1 X_2 ⟩ is too general so that it is ”imprudent” to induce a VP constituent from two consecutive blocks with no lexical or syntactic sign.

In addition to pruning rules, we can control the depth of syntactic skeletons using a parameter ω_s. The system is forced to use a smaller syntactic skeleton (and fewer syntactic rules) if ω_s chooses a smaller value. As extreme cases, the system goes back to a hierarchical phrase-based model if $\omega_s = 0$; and it can consider syntactic skeletons with arbitrary depth if $\omega_s = +\infty$. Thus we can seek a balance by tuning it on a development set.

Another option for system speed-up is to apply the tree-parsing technique (Eisner 2003). We feed the source parse tree to the decoder in addition to the source sentence. The source sentence is first parsed using Hiero-style rules as is usual in hierarchical phrase-based systems (Chiang 2007), with the exception that we impose no span limit on rule applications for source spans corresponding to constituents in the syntactic tree. Then, we apply tree-to-string rules on the tree. If the source side of a tree-to-string rule matches an input tree fragment: 1) that rule is converted to partially syntactic rules; and 2) the tree-to-string and corresponding partially syntactic rules are added to the rule list linked with the CYK grid cell associated with the span of the source syntactic tree fragment. After that, the remaining decoding steps, such as building translation hypergraph and language model intersection, proceed as usual. This method can efficiently match (partially) syntactic rules for decoding and does not require rule binarization. As a trade-off, fewer syntactically-sensitive derivations are taken into account due to the hard constraints imposed by the source parse tree.

Experiments

We experimented with our approach on Chinese-English (zh-en) and English-Chinese (en-zh) translation tasks.

Experimental Setup

We used 2.74 million sentence Chinese-English bitext from NIST12 OpenMT. We ran GIZA++ on the bitext to produce bidirectional alignments and then the grow-diag-final-and method to obtain symmetric alignments. For syntactic parsing, we ran the Berkeley parser on both sides of the bitext. The parse trees were then binarized in a left-heavy fashion for better generation on test sentences. Syntax-based (tree-to-string) rules with up to five non-terminals were extracted on the entire set of the training data. For the hierarchical phrase-based system, hierarchial rules with up to two non-terminals were extracted from a 0.94 million sentence subset and phrasal rules were extracted from all the training data. All these rules were learned using the NiuTrans open-source toolkit (Xiao et al. 2012).

We trained two 5-gram language models: one on the Xinhua portion of the English Gigaword in addition to the English-side of the bitext, used by Chinese-English systems; one on the Xinhua portion of the Chinese Gigaword in addition to the Chinese-side of the bitext, used by English-Chinese systems. All language models were smoothed using the modified Kneser-Ney smoothing method.

For Chinese-English translation, we evaluated our systems on newswire and web data, respectively. Our tuning sets (newswire: 1,198 sentences, web: 1,308 sentences) were drawn from the NIST MT 04-06 evaluation data and the GALE data. The test sets (newswire: 1,779 sentences, web: 1,768 sentences) contained all newswire and web evaluation data of NIST MT 08, 12 and 08-progress. For English-Chinese translation, our tuning set (995 sentences)

[3] Note that if no partially syntactic derivations are introduced, our model is doing something similar to hypothesis selection which chooses the best derivation from either Hiero-style or tree-to-string derivation space.

Entry		zh-en (nw)		zh-en (wb)		en-zh		Average
		Tune	Test	Tune	Test	Tune	Test	improve.
		(1198)	(1779)	(1308)	(1768)	(995)	(1859)	on test
Hierarchical Phrase-based (Hier.)		35.70	31.76	27.29	22.61	33.12	30.59	0
Treebank	Tree-to-String	34.41	30.76	25.59	21.69	32.29	30.20	-0.77
	Hier. + source syn. features	36.02	31.99	27.35	22.76	33.34	30.90	+0.23
	Hier. + source syn. SAMT rules	36.09	32.09	27.47	22.86	33.46	30.99*	+0.32
	SYNSKEL	36.34*	32.43*	28.00*	23.15	33.70	31.50*	+0.71
	SYNSKEL (tree-parsing)	36.44*	32.35	27.95	23.11	33.49	31.32*	+0.61
Bi-trees	Tree-to-String	34.82	31.21	25.83	21.88	33.03	30.69	-0.39
	Hier. + source syn. features	36.09	31.98	27.42	22.87	33.40	30.97	+0.29
	Hier. + source syn. SAMT rules	36.14*	32.18*	27.46	22.90	33.40	30.92	+0.35
	SYNSKEL	36.70*	32.75*	28.09*	23.29*	33.82*	31.77*	+0.95
	SYNSKEL (tree-parsing)	36.67*	32.64*	27.92	23.39*	33.95*	31.66*	+0.91
	SYNSKEL (forest-parsing)	36.77*	32.56*	28.05*	23.45*	33.95*	31.79*	+0.94

Table 1: BLEU[%] scores of various systems. * means that a system is significantly better than the hierarchical phrase-based (Hier.) baseline at $p < 0.01$.

and test set (1,859 sentences) were the evaluation data sets of SSMT 07 and NIST MT 08 Chinese-English track, respectively. All source-side parse trees were produced in the same way as that used on the training data.

We implemented our CYK decoder as described in the Decoding section. By default, string-parsing was used and ω_s was set to $+\infty$. All feature weights were tuned using minimum error rate training (MERT). Since MERT is prone to local optimums, we ran each experiment on the tuning set five times with different initial setting of feature weights. In the evaluation, we report uncased BLEU4 on zh-en and uncased BLEU5 on en-zh, respectively.

MT Systems for Comparison

For comparison to other state-of-the-art methods, we report results of several systems in our empirical study. Two of them are the standard implementations of the hierarchical phrase-based model (Chiang 2007) and the tree-to-string model (Liu, Liu, and Lin 2006). Because hierarchical phrase-based translation is one of the most popular MT models in recent MT evaluations, we selected it as the primary baseline in the experiments.

Also, we introduced the source syntax-based features (or soft constraints) into the hierarchical phrase-based system to build another syntax-aware MT system. It is a simple and straightforward method to incorporate source-language syntax into existing non-syntactic systems. In our experiment we chose one of the best feature sets reported in Marton and Resnik's work (Marton and Resnik 2008) [4].

In addition, we experimented with adding a source-language syntax-augmented MT (SAMT) grammar into the hierarchical phrase-based system, as described in (Zollmann and Vogel 2010). In this system, the hierarchical phrase-based and syntax-based models are bridged by glue rules. It thus allows monotonic concatenation of hierarchical and syntactic partial derivations. As source-language SAMT grammars have finer grained labels induced from the source-language parse trees, using them in a Hiero-style system can be regarded as a straightforward way of making use of both hierarchical phrase-based and syntactic models.

Results

Table 1 shows the result, where our syntactic skeleton-based system is abbreviated as SYNSKEL. We see, first of all, that the SYNSKEL system obtains significant improvements on all three of the test sets. Using CTB style parse trees yields an average improvement of +0.6 BLEU, and using binary trees yields an average improvement of +0.9 BLEU. Also, tree-parsing is promising for applying (partially) syntactic rules on top of a standard use of Hiero-style rules. It obtains comparable BLEU scores as the string-parsing method. However, involving more trees in a binary forest does not help[5]. These interesting results indicate that it might be difficult to introduce ”new” derivations into an already very large derivation space by considering more binary parse alternatives. More interestingly, it is observed that SYNSKEL even outperforms the ”Hier.+ SAMT” system over 0.6 BLEU points (the improvements are statistically significant). We found that the systems with SAMT rules ran very slow due to the large number of CCG-style labels. It made the system hard to tune. In comparison, the SYNSKEL system relies on a relatively small number of syntactic labels and applies syntactic rules on the higher level of sentence structure only. It thus obtains bigger improvements and is easy to run.

In addition, we study the system behavior under the control of the maximum depth of skeleton (i.e., ω_s). Figure 3 shows that too large skeletons are not always helpful. Using skeletons with $\omega_s \leq 5$ can obtain satisfactory improvements, with a 27% reduction in decoding time in comparison of the full use of skeletons.

We then investigate how often the system chooses different types of derivations. Table 2 shows a preference to par-

[4] The features are NP+, NP=, VP+, VP=, PP+, PP=, XP+ and XP=.

[5] The forest was produced by binarizing a CTB-style parse tree into a binary forest.

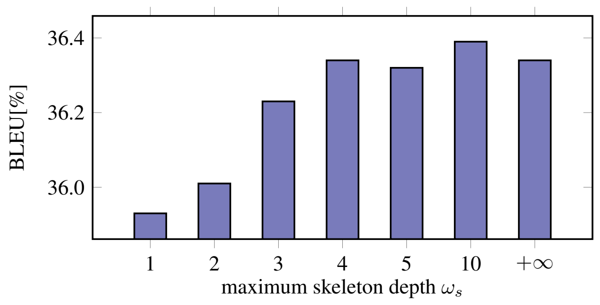


Figure 3: BLEU as a function of maximum depth of skeleton (tuning set of zh-en newswire).

tially syntactic and Hiero-style derivations on all three of the tasks. The en-zh task shows a heavy use of fully and partially syntactic derivations, followed by the zh-en newswire and zh-en web tasks. This result may reflect a fact of varying parse quality over different languages and genres.

Related Work

Recently MT models based on source syntax have achieved promising results due to their good abilities in modeling source-language syntax for lexicon selection and reordering (Eisner 2003; Liu, Liu, and Lin 2006; Huang, Kevin, and Joshi 2006). Improvements are continuing for a better use of the source-language syntactic information on top of non-syntactic models. A straightforward way is to introduce n-gram phrases into syntactic systems (Liu, Liu, and Lin 2006), or vice versa (Tinsley, Hearne, and Way 2007). Also, source-syntax constraints were developed to integrate tree-based constraints into hierarchical phrase-based systems (Marton and Resnik 2008).

More sophisticated models include system combination and joint decoding-like methods (Liu et al. 2009; Cmejrek, Mi, and Zhou 2013; Xiao et al. 2014). Unlike these methods, our focus is to study approaches to introducing partially syntactic (skeleton-based) derivations into MT, rather than developing new decoding paradigms. Beyond this, we develop a grammar to handle fully syntactic, partially syntactic, and Hiero-style derivations in one framework, which does not require modification of the SCFG decoder (or need little coding work if tree-parsing is adopted). By contrast, joint decoding-like methods need two or more individual models/grammars for input.

Another related line of work is to introduce source-language syntactic annotations to hierarchical phrase-based systems (Zhao and Al-Onaizan 2008; Hoang and Koehn 2010), or to simplify syntactic rules in syntax-based systems (Zhu and Xiao 2011; Zhao et al. 2011). E.g., the partially syntactic rules presented here are similar to the rules used in (Hoang and Koehn 2010; Zhao et al. 2011). Their rule extraction methods resort to either a single Hiero-like algorithm or a single GHKM-like algorithm, which may miss some useful rules. Moreover, the underlying structure of source sentence, such as sentence skeleton, is not well explained in previous work.

derivation type	zh-en (nw)	zh-en (wb)	en-zh
fully syn.	1.4%	0.6%	14.4%
non-syn.	19.7%	26.6%	11.6%
partially syn.	78.9%	72.8%	74.0%

Table 2: Derivation usage on the tuning sets.

Note that the approach presented here is doing something similar to fine-grained labeling of SCFGs (Chiang 2010; Mylonakis and Sima'an 2011). Previous work focuses more on enhancing Hiero-style grammars or ITGs with syntactic labels, while our approach "combines" two different models to model different levels of translation. The only exception is (Zollmann and Vogel 2010; 2011). They bridge Hiero-style models and SAMT models by glue rules in a single system. But they point out that the huge number of CCG-style non-terminals in SAMT grammars lead to a bad estimation of low-occurrence-count rules. Unlike their work, our approach uses standard labeling of non-terminals, and it uses the partially syntactic rules to bridge tree-to-string rules and Hiero-style rules. In this way, we apply the syntactic rules to where they can contribute most.

Though integrating sentence skeleton information into MT is promising, it is rare to see investigation on this issue. Perhaps the most related studies are (Mellebeek et al. 2006) and (Xiao, Zhu, and Zhang 2014). These methods rely on heuristics or expensive annotated data for skeleton acquisition. Our method instead views skeletons as latent structures and automatically induces them in MT decoding.

Conclusions

We have described an approach to introducing syntactic skeleton into MT for a better use of source syntax. Our model allows for applying tree-to-string rules to handle high-level syntactic movement, and meanwhile Hiero-style rules to handle low-level non-syntactic translation. Thus the system can search for best translation over a space of fully syntactic, non-syntactic and partially syntactic derivations. The hierarchical phrase-based model and the tree-to-string model can be regarded as two special cases of our framework. We experiment with our approach on large-scale MT tasks and obtain an average improvement of +0.9 BLEU across different languages and genres.

Acknowledgements

This work was supported in part by the National Science Foundation of China (61272376, 61300097 and 61432013). The authors would like to thank anonymous reviewers, Ji Ma and Adrià de Gispert for their valuable comments and discussions.

References

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

Chiang, D. 2010. Learning to Translate with Source and Target Syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1443–1452.

Cmejrek, M.; Mi, H.; and Zhou, B. 2013. Flexible and efficient hypergraph interactions for joint hierarchical and forest-to-string decoding. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 545–555.

Eisner, J. 2003. Learning Non-Isomorphic Tree Mappings for Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, 205–208.

Galley, M.; Graehl, J.; Knight, K.; Marcu, D.; DeNeefe, S.; Wang, W.; and Thayer, I. 2006. Scalable Inference and Training of Context-Rich Syntactic Translation Models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 961–968.

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

Hopkins, M., and Langmead, G. 2010. SCFG decoding without binarization. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 646–655.

Huang, L.; Kevin, K.; and Joshi, A. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of conference of the Association for Machine Translation in the Americas (AMTA)*, 66–73.

Koehn, P.; Och, F.; and Marcu, D. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT:NAACL)*, 48–54.

Liu, Y.; Mi, H.; Feng, Y.; and Liu, Q. 2009. Joint decoding with multiple translation models. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 576–584.

Liu, Y.; Liu, Q.; and Lin, S. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL)*, 609–616.

Marton, Y., and Resnik, P. 2008. Soft Syntactic Constraints for Hierarchical Phrased-Based Translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*, 1003–1011.

Mellebeek, B.; Owczarzak, K.; Groves, D.; Genabith, J. V.; and Way, A. 2006. A syntactic skeleton for statistical machine translation. In *Proceedings of the 11th Annual conference of the European Association for Machine Translation (EAMT)*, 195–202.

Mylonakis, M., and Sima'an, K. 2011. Learning hierarchical translation structure with linguistic annotations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*, 642–652.

Tinsley, J.; Hearne, M.; and Way, A. 2007. Exploiting parallel treebanks to improve phrase-based statistical machine translation. In *Proceedings of the 6th International Workshop on Treebanks and Linguistic Theories*.

Xiao, T.; Zhu, J.; Zhang, H.; and Li, Q. 2012. NiuTrans: An Open Source Toolkit for Phrase-based and Syntax-based Machine Translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistic (ACL): System Demonstrations*, 19–24.

Xiao, T.; de Gispert, A.; Zhu, J.; and Byrne, B. 2014. Effective incorporation of source syntax into hierarchical phrase-based translation. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, 2064–2074.

Xiao, T.; Zhu, J.; and Zhang, C. 2014. A hybrid approach to skeleton-based translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 563–568.

Zhang, H.; Huang, L.; Gildea, D.; and Knight, K. 2006. Synchronous binarization for machine translation. In *Proceedings of the Human Language Technology Conference: North American chapter of the Association for Computational Linguistics Annual Meeting (HLT:NAACL)*, 256–263.

Zhao, B., and Al-Onaizan, Y. 2008. Generalizing Local and Non-Local Word-Reordering Patterns for Syntax-Based Machine Translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 572–581.

Zhao, B.; Lee, Y.-S.; Luo, X.; and Li, L. 2011. Learning to transform and select elementary trees for improved syntax-based machine translations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*, 846–855.

Zhu, J., and Xiao, T. 2011. Improving decoding generalization for tree-to-string translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*, 418–423.

Zollmann, A., and Vogel, S. 2010. New parameterizations and features for pscfg-based machine translation. In *Proceedings of the 4th Workshop on Syntax and Structure in Statistical Translation*, 110–117.

Zollmann, A., and Vogel, S. 2011. A word-class approach to labeling pscfg rules for machine translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*, 1–11.