A Tree-to-Tree Alignment-based Model for Statistical Machine Translation

Min ZHANG[1] **Hongfei JIANG**[12] **Ai Ti AW**[1] **Jun SUN**[13] **Sheng LI**[2] **Chew Lim TAN**[3]

[1]Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
{mzhang, aaiti, vishfj, visjs}@i2r.a-star.edu.sg

[2]School of Computer
Harbin Institute of Technology
Harbin 150001
{hfjiang, lishen}@mtlab.hit.edu.cn

[3]School of Computing
National University of Singapore
Singapore 117543
{sunjun, tancl}@comp.nus.edu.sg

Abstract

This paper presents a novel statistical machine translation (SMT) model that uses tree-to-tree alignment between a source parse tree and a target parse tree. The model is formally a probabilistic synchronous tree-substitution grammar (STSG) that is a collection of aligned elementary tree pairs with mapping probabilities (which are automatically learned from word-aligned bi-parsed parallel texts). Unlike previous syntax-based SMT models, this new model supports multi-level global structure distortion of the tree typology and can fully utilize the source and target parse tree structure features, which gives our system more expressive power and flexibility. The experimental results on the HIT bi-parsed text show that our method performs significantly better than Pharaoh, a state-of-the-art phrase-based SMT system, and other syntax-based methods, such as the synchronous CFG-based method on the small dataset.

Keywords: statistical machine translation, syntax-based statistical machine translation, tree-to-tree alignment, synchronous tree-substitution grammar, elementary tree

Motivation

Phrase-based SMT

Phrase-based approach (Marcu and Wong, 2002; Koehn et al, 2003; Och and Ney, 2004) to statistical machine translation (SMT) has recently achieved significant improvements in translation accuracy over the original IBM word-alignment-based model (Brown et al., 1993). In phrase-based models, a phrase can be any string of adjacent words without constraints imposed by any syntactic theory. These phrases allow a model to learn local reorderings, translations of multiword expressions, or insertions and deletions that are sensitive to local context. These make it a simple and powerful mechanism for machine translation. However, there exist many open issues to be resolved in phrase-based models. For examples, the handling of discontiguous phrases and modeling of global reordering, estimation of phrase translation probabilities and phrase partition probabilities are not yet effectively addressed in phrase-based models (Quirk and Menezes, 2006). Much research has been carried out to look into the above issues. One natural extension is to utilize syntax-based structure features for SMT.

Syntax-based SMT

Recent work in SMT has evolved from the word-based and phrase-based models to syntax-based models, that include hierarchical phrase models (Wu, 1997; Chiang, 2007), bilingual synchronous grammars (Eisner, 2003; Ding and Palmer, 2005; Quirk et al, 2005; Cowan et al., 2006;) and other syntax-based models (Yamada and Knight, 2001; Gildea, 2003; Och et al, 2004b; Liu et al., 2006). Wu (1997) and Chiang (2007)'s methods are formally syntax-based, i.e., their methods are not informed by any linguistically syntactic theory. Wu (1997) proposes Inversion Transduction Grammars (ITGs, an instance of synchronous CFGs), treating translation as a process of parallel parsing of the source and target languages via ITGs. Chiang (2007) uses a formal binary synchronous CFG to model hierarchical phrase structures. Yamada and Knight (2001) use noisy-channel model to transfer a target parse tree into a source sentence. Och et al (2004) explore using various morphologic and syntactic features to re-rank the translation outputs of a phrase-based system. Ding and Palmer (2005) propose a syntax-based translation model based on a probabilistic synchronous dependency insertion grammar, a version of synchronous grammar defined on dependency trees. Quirk et al. (2005) propose a dependency treelet-based translation model. They project the source dependency parse onto the target sentence, extract dependency treelet translation pairs and train a tree-based ordering model. Cowan et al. (2006) propose a feature-based discriminative model for prediction of the target language syntactic structures, given the source language parse trees. Riezler and Maxwell III (2006) present an approach to SMT that combines ideas from phrase-based SMT and traditional grammar-based SMT. They incorporate the concept of multi-word translation units into transfer of dependency structure snippets, and model and train statistical components according to phrase-based SMT system. Zhang et al. (2006) study the synchronous rule binarization for MT. They devise a linear-time algorithm for factoring syntactic re-orderings by binarizing synchronous rules when possible and show that the resulting rules set significantly improves the speed and accuracy of a state-of-the-art syntax-based machine translation system. Zollmann and Venugopal (2006) present a syntax-based machine translation method that generates translation results by a chart parsing decoder operating on phrase tables augmented and generalized with target language syntactic categories. The motivation behind all these advances is to exploit syntactic structure features to model translation process: lexical selection, reordering, structure transfer and generation.

Structural Divergences between Languages

One of the major challenges in applying syntax to SMT is structural divergences between languages (Dorr, 1994),

which are due to either systematic differences between two languages in expressing a concept syntactically or relatively free translations in the training corpora. As a result, syntax-based MT systems have to transduce between non-isomorphic tree structures that is beyond the power of Synchronous CFGs (only sibling nodes are allowed to reorder independently prior to translation). For example, the *S(VO)* structure in English cannot be translated into a *VSO* word order in Arabic by any SCFGs.

Many researchers have investigated and studied the above issues. Fox (2002) examines the issue of linguistic phrasal cohesion between English and French and discovers that while there is less cohesion than we might desire, there is still a large amount of regularity in constructions where breakdowns occur. The paper also examines the differences in cohesion phrase-structure-based parse tree, trees with flatten verb phrases and dependency structures, and concludes that the highest degree of cohesion is presented in dependency structures. Eisner (2003) studies how to learn non-isomorphic tree-to-tree or tree-to-string mappings for machine translation. The paper sketches an EM algorithm to learn the probabilities of elementary tree pairs by training on pairs of full trees, and a Viterbi decoder to find optimal translations. However, the above two papers do not verify their methods empirically on a real MT system. Gildea (2003) proposes a new subtree cloning operation to either tree-to-tree or tree-to-string alignment algorithms for MT. His method is evaluated on word alignment rather than machine translation. Galley et al. (2004) propose a theory that gives formal semantics to word-alignments defined over parallel corpora and use the theory to derive from word-aligned parallel corpora the minimal set of syntactically motivated transformation rules that explain human translation data. They find that local transformations (primarily child-node re-orderings) of one-level parent-children substructures are an inadequate model for parallel corpora, so they learn rules involving much larger tree fragments. Melamed (2004) studies how to infer the synchronous structures hidden in parallel texts for the syntax-aware SMT by generalizing ordinary parsing algorithms to synchronous ones. Huang et al. (2006) study a tree substitution grammar-based tree-to-string alignment model for SMT. Liu et al. (2006) propose a tree-to-string alignment template-based method for SMT. Wellington et al. (2006) study empirically the lower bounds on alignment failure rates with and without gaps for bilingual/monolingual bitexts under the constraints of word alignment alone or with one or both side parse trees. Their study finds surprisingly many examples of translational equivalence that could not be analyzed using binary-branching structures without discontinuities.

Previous research discussed above suggests using more powerful grammars whose rules can be applied to larger tree fragments to address the non-isomorphic issue. Shieber and Schabes (1990) introduce synchronous tree-adjoining grammar (STAG) preliminary for semantics and Eisner (2003) uses a synchronous tree-substitution grammar (STSG), which is a restricted version of STAG without adjunctions (Chiang, 2006), for machine translation. STAGs and STSGs use elementary tree structure, which is beyond the scope of two-level context-free rules, to generate more tree relations than SCFGs.

Synchronous TSG-based Tree-to-Tree Alignment

In this paper, we propose a synchronous TSG-based tree-to-tree alignment model for machine translation. Specifically, we use elementary tree-based structure alignments, which are automatically learned from word-aligned bi-parsed parallel texts, to model the translation process. We separate the source language analysis from the recursive transformation. Therefore, to translate a source sentence, we first employ a CFG-based Treebank parser to produce a source parse tree and then use the set of learned elementary tree pairs to transform the source parse tree to a target parse tree, which is then used to generate target sentence.

There are two major benefits of our STSG-based tree-to-tree alignment model. First, it is possible to explicitly model the syntax of the target language, thereby improve the grammaticality of target sentence. Second, our model has more expressive power and flexibility since it allows multi-level global structure distortion of the tree typology and fully utilizes source and target parse tree structure features. Therefore, it can solve the deficiencies in SCFG and phrase-based models such as non-isomorphic tree alignment, global reordering and discontiguous phrase.

To the best of our knowledge, no previous work explores empirically STSG-based tree-to-tree alignment over phrase-structure parse trees for machine translation. Eisner (2003) studies STSG-based alignment on dependency trees, but no empirical verification on machine translation is done. Compared to Eisner (2003), we use different training and decoding algorithms and modeling methods. Graehl and Knight (2004) define tree transducers that have multi-level trees only on the source-side. Yamada and Knight (2001) and Zollmann and Venugopal (2006) and Galley et al. (2004) only utilize target parse tree information. Ding and Palmer (2005) and Chris et al (2005) work on dependency grammars while Huang et al (2006) and Liu et al (2006) work on tree-to-string alignment models. Our method, in terms of modeling, training and decoding algorithms are different from theirs at one or more points.

In the rest of this paper, we elaborate our modeling, training and decoding methods and report our experimental results in detail.

Tree-to-Tree Alignment-based Model

In this section, we first introduce what STSG is and then based on which we define our tree-to-tree alignment-based SMT model. Finally, we present the modeling process based on log-linear framework.

Synchronous TSG (STSG) for SMT

Shieber (2004) gives a formal and general definition of STSG. Here we give a more concrete definition of STSG with respect to its application in SMT. A STSG is a septet $G = < \Sigma_s, \Sigma_t, N_s, N_t, S_s, S_t, P >$, where:

- Σ_s and Σ_t are source and target terminal alphabets (POSs or lexical words), respectively, and
- N_s and N_t are source and target non-terminal alphabets (linguistic phrase tag, i.e., NP/VP…), respectively, and

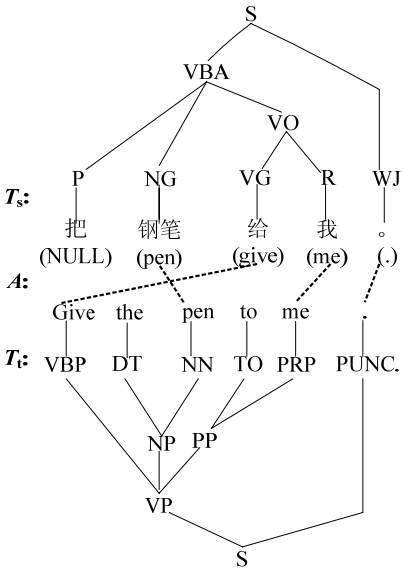


Figure 1: A word-aligned parse tree pairs of a Chinese sentence and its English translation

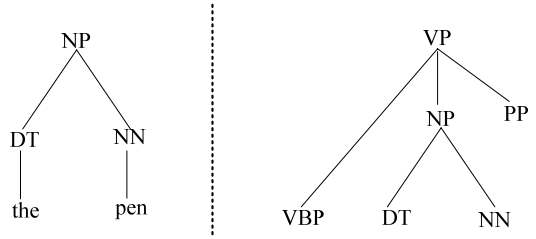


Figure 2: Examples of elementary trees

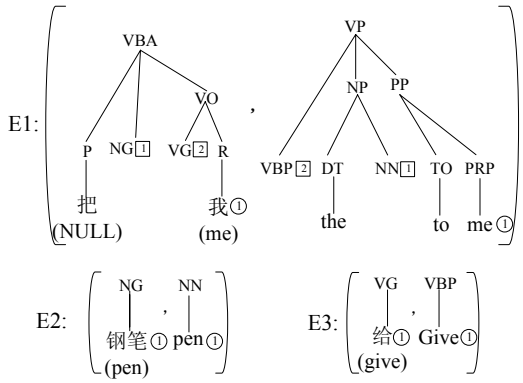


Figure 3: Three examples of *PET*

- $S_s \in N_s$ and $S_t \in N_t$ are the source and target start symbols (roots of source and target parse trees), and
- P is a production rule set, where a production rule is a pair of elementary tree ($\xi_s \leftrightarrow \xi_t$) with linking relation between leaf nodes in source elementary tree (ξ_s) and leaf nodes in target elementary tree (ξ_t).

In TSG and STSG, an elementary tree is a tree fragment whose leaf nodes can be either non-terminal symbols or terminal symbols. For example, Figure 2 illustrates two examples of elementary trees which belong to the English parse tree T_t shown in Figure 1. Obviously, a normal subtree (whose leaf nodes must be terminal symbols) is an elementary tree but not always true vice versa.

In STSG, a production or a rule is a pair of elementary tree with alignment information (hereafter, *PET*). We can define a *PET* as a triple $< \xi_s , \xi_t , \tilde{A} >$, where:

- ξ_s is a source elementary tree, and
- ξ_t is a target elementary tree, and
- $\tilde{A}$ is the alignments between leaf nodes of two elementary trees. It is defined as a subset of the Cartesian product of source and target leaf node positions:

$$\tilde{A} \subseteq \{(i, j) : i \text{ is the position of } i^{\text{th}} \text{ leaf node of } \xi_s \text{ ; } j \text{ is the position of } j^{\text{th}} \text{ leaf node of } \xi_t \}$$

Figure 3 shows three examples of *PET* extracted from the word-aligned parse tree pair in Figure 1. We use boxed and circled indices to indicate non-terminal and terminal alignments, respectively. Obviously, *PET* allows any tree node insertion, deletion and substitution between the two elementary trees. We believe this property of *PET* can well address the issues of non-isomorphic structures, global reordering and phrase gaps that we discussed in the previous section.

STSG-based Tree-to-Tree Alignment

We use a STSG to represent tree-to-tree alignment, i.e., a STSG-based tree-to-tree alignment template is a *PET* $< \xi_s , \xi_t , \tilde{A} >$. In the following, we formally describe how to develop *PET*s into probabilistic dependencies to model the translation process.

Given the source and target sentences f_1^J and e_1^I, we first introduce two hidden variable T_s and T_t that denote the source and target parse trees, respectively, then we have[1]:

$$\begin{aligned} Pr(e_1^I \mid f_1^J) &= \sum_{T_s, T_t} Pr(e_1^I, T_t, T_s \mid f_1^J) \\ &= \sum_{T_s, T_t} (Pr(T_s \mid f_1^J) \cdot Pr(T_t \mid T_s, f_1^J) \quad (1) \\ &\qquad \cdot Pr(e_1^I \mid T_t, T_s, f_1^J)) \end{aligned}$$

Next, we introduce another hidden variable D to detach T_s and T_t into a sequence of K *PETs* $< \xi_s^{1,K} , \xi_t^{1,K} , \tilde{A}^{1,K} >$. We assume that each source elementary tree ξ_s^i produces a target elementary tree ξ_t^i *independently* and they are aligned by $\tilde{A}^i$. Then, we have:

[1] The notational convention in our paper is as follow. We use the symbol $Pr(\cdot)$ to denote general probability distribution with no specific assumptions. In contrast, for model-based probability distributions, we use generic symbol $p(\cdot)$

$$
\begin{aligned}
Pr(T_t \mid D, T_s, f_1^J) &= Pr(\xi_t^{1,K} \mid \xi_s^{1,K}, D, f_1^J) \\
&= Pr(\xi_t^{1,K} \mid \xi_s^{1,K}) \\
&= \prod_{k=1}^{K} Pr(\xi_t^k \mid \xi_s^k)
\end{aligned} \tag{2}
$$

where we omit the explicit dependences on D and f_1^J to avoid notational overhead. Based on eq. (2), we have:

$$
\begin{aligned}
Pr(T_t \mid T_s, f_1^J) &= \sum_D Pr(T_t, D \mid T_s, f_1^J) \\
&= \sum_D (Pr(D \mid T_s, f_1^J) \cdot Pr(T_t \mid D, T_s, f_1^J)) \\
&= \sum_D (Pr(D \mid T_s, f_1^J) \cdot Pr(\xi_t^{1,K} \mid \xi_s^{1,K})) \\
&= \sum_D (Pr(D \mid T_s, f_1^J) \cdot \prod_{k=1}^{K} Pr(\xi_t^k \mid \xi_s^k))
\end{aligned} \tag{3}
$$

To further decompose $Pr(\xi_t^k \mid \xi_s^k)$, the elementary tree alignment $\tilde{A}^k$ is introduced as another hidden variable:

$$
\begin{aligned}
Pr(\xi_t^k \mid \xi_s^k) &= \sum_{\tilde{A}^k} Pr(\xi_t^k, \tilde{A}^k \mid \xi_s^k) \\
&= \sum_{\tilde{A}^k} (Pr(\tilde{A}^k \mid \xi_s^k) \cdot Pr(\xi_t^k \mid \tilde{A}^k, \xi_s^k))
\end{aligned} \tag{4}
$$

From eqs. (1) to (4), we can see that our STSG-based tree-to-tree translation model is comprised of four sub-models:

1) parse model: $Pr(T_s \mid f_1^J)$
2) detachment model: $Pr(D \mid T_s, f_1^J)$
3) translation model: $Pr(\xi_t^{1,K} \mid \xi_s^{1,K})$, including:
   3.1) tree alignment selection model: $Pr(\tilde{A}^k \mid \xi_s^k)$
   3.2) structure transfer model: $Pr(\xi_t^k \mid \tilde{A}^k, \xi_s^k)$
4) generation model: $Pr(e_1^I \mid T_t, T_s, f_1^J)$

Figures 1 and 3 show how our tree-to-tree translation model works. First, the source sentence is parsed into a source parse tree T_s (the upper tree in Figure 1). Next, the parse tree T_s is detached into three elementary trees (the left hand side of three *PET*s shown in Figure 3). Then the three *PET*s shown in Figure 3 are selected to map the three source elementary trees to three target elementary trees, which are then combined to generate a target parse tree T_t (the lower tree in Figure 1). Finally, a target translation is generated from the target parse tree.

Features

Our model is implemented under log-linear framework (Och and Ney, 2002). Hence, all knowledge sources, including source and target string and all hidden variables and any additional knowledge source, such as language model or additional dictionaries, are described as feature functions. In our implementation, we further simplify our model as follows:

1) The parse model $Pr(T_s \mid f_1^J) \equiv 1$ since we usually only use the best parse tree for structure mapping.
2) The detachment model $Pr(D \mid T_s, f_1^J) \equiv 1$ since we assume all detachments have the same probability.
3) The generation model $Pr(e_1^I \mid T_t, T_s, f_1^J) \equiv 1$ since we just output the leaf nodes of T_t to generate the target translation e_1^I regardless of T_s and f_1^J and any further morphological generation.

After model simplification, we have:

$$
Pr(e_1^I \mid f_1^J) = \sum_D Pr(\xi_t^{1,K} \mid \xi_s^{1,K}) \tag{5}
$$

$$
Pr(e_1^I, PET_1^K \mid f_1^J) = \frac{\exp[\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J, PET_1^K)]}{\sum_{e_1'^I, PET_1'^K} \exp[\sum_{m=1}^{M} \lambda_m h_m(e_1'^I, f_1^J, PET_1'^K)]} \tag{6}
$$

$$
\hat{e}_1^I = \arg\max_{e_1^I, PET_1^K} (\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J, PET_1^K)) \tag{7}
$$

Eq. (5) is the simplified model. Eq. (6) formalizes the modeling process based on log-linear framework. Eq. (7) formulizes the decoding, i.e., the translation process.

Finally, for our experiments we use the following seven feature functions that are analogous to the default feature set of Pharaoh (Koehn, 2004a).

1) Bidirectional elementary tree mapping probability:

$$
\phi(e \mid f) = \log \prod_{k=1}^{K} \frac{N(\xi_e^k, \xi_f^k)}{N(\xi_f^k)}
$$

$$
\phi(f \mid e) = \log \prod_{k=1}^{K} \frac{N(\xi_f^k, \xi_e^k)}{N(\xi_e^k)}
$$

2) Bidirectional elementary tree lexical translation probability: $lex(f \mid e)$ and $lex(e \mid f)$. Here, we only consider terminal translation probability and set the non-terminal translation probability to 1.
3) Language model (*lm*): $\log \prod_{i=1}^{I} p(e_i \mid e_{i-2}, e_{i-1})$.
4) Number of elementary tree pairs used (*pp*): *K*.
5) Number of target words (*wp*): *I*.

Rule Extraction

Rules or *PET*s are extracted from word-aligned, bi-parsed sentence pairs $< T(f_1^J), T(e_1^I), A >$, where $T(z)$ denotes a parse tree covering string z. For better understanding our rule extraction algorithm, we classify *PET*s into two categories:

- ***initial PET***, if all leaf nodes in both source and target elementary trees of a *PET* are terminals
- ***abstract PET***, otherwise

Hence, an *initial PET* $<T(f_{j1}^{j2}),T(e_{i1}^{i2}),\tilde{A}>$ would satisfy the following constraints:

- $\forall(i,j)\in A: i_1 \le i \le i_2 \leftrightarrow j_1 \le j \le j_2$
- $T(f_{j1}^{j2})$ is a subtree of $T(f_1^J)$
- $T(e_{i1}^{i2})$ is a subtree of $T(e_1^I)$

We introduce another concept here. Given an *initial PET* $<T(f_{j1}^{j2}),T(e_{i1}^{i2}),\tilde{A}>$, a triple $<T(f_{j3}^{j4}),T(e_{i3}^{i4}),\hat{A}>$ is its *sub initial PET* if and only if:

- $<T(f_{j3}^{j4}),T(e_{i3}^{i4}),\hat{A}>$ is an *initial PET*
- $\forall(i,j)\in \tilde{A}: i_3 \le i \le i_4 \leftrightarrow j_3 \le j \le j_4$
- $T(f_{j3}^{j4})$ is a subtree of $T(f_{j1}^{j2})$
- $T(e_{i3}^{i4})$ is a subtree of $T(e_{i1}^{i2})$

Our rule extraction algorithm includes two steps:

1) Extracting ***initial PETs*** from $<T(f_1^J),T(e_1^I),A>$:

It is straightforward to extract *initial PET*s. We just iterate all source and target subtree pairs $<T(f_{j1}^{j2}),T(e_{i1}^{i2})>$. If the condition "$\forall(i,j)\in A: i_1 \le i \le i_2 \leftrightarrow j_1 \le j \le j_2$" is satisfied, the triple $<T(f_{j1}^{j2}),T(e_{i1}^{i2}),\tilde{A}>$ is an *initial PET*, where $\tilde{A}$ are alignments between leaf nodes of $T(f_{j1}^{j2})$ and $T(e_{i1}^{i2})$.

2) Extracting ***abstract PETs*** from extracted ***initial PETs***:

We derive *abstract PETs* from an *initial PET* by removing one or more of its *sub initial PET*s. Following is the algorithm for extracting *abstract PETs*.

Input: *initial PET* set Output: *abstract PET* set
1) **foreach** $PET_i \in$ *initial PET* set, **do** 2) 2.1) put all *sub initial PET*s of PET_i into a set PET_{sub} 2.2) **foreach** subset $\xi \subset PET_{sub}$ **do** 2.2.1) remove the portion covered by ξ from PET_i 2.2.2) add it into *abstract PET* set 2.3) **end do** 3) **end do**

Finally, same as previous work (Liu et al, 2006; Chiang, 2007), we set two parameters to control the number of extracted *PET*s:

1) The height of an elementary tree is no greater than h.
2) The number of non-terminals as leaf nodes is no great than c.

Decoding

We separate source language analysis from structure recursive transformation. Hence, in brief, our decoder carries out two-pass (or two-step) search by the following two modules.

1) The 1st one is a CFG-based chart parser as a pre-processor for mapping an input sentence to a parse tree T_s (for details of chart parser, please refer to Charniak (1997)).
2) The 2nd one is a STSG-based (or *PET*-based) bottom-up beam search algorithm for mapping the source parse tree T_s generated in the 1st pass to a target parse tree T_t. In this 2nd pass, a list of *candidate translations*[2] are computed for the input subtree rooted at each node with a post-order traversal[3]. The root of T_s is the last visited node. Hence, the best candidate translation of T_s is finally outputted as the target parse tree T_t.

Following is the pseudo-code of our 2nd pass search:

Input: source parse tree T_s Output: target parse tree T_t
Data structures: 1) n_i : the i^{th} node of T_s in post-order traversal 2) T_{ni}: the subtree rooted at node n_i 3) R_i: to store all usable[4] *Rules* or *PET*s to T_{ni} 4) H_i: to store all *Hypothesises* or *candidate translations* of T_{ni}
Algorithm: the 2nd pass search for tree structure mapping 1) **foreach** node n_i (post-order), **do** 2) 2.1) extract all usable *PET*s and put them in R_i 2.2) R_i pruning 2.3) **foreach** *PET* ξ =< ξ_s , ξ_t , $\tilde{A}$ >$\in R_i$ **do** 2.3.1) if ξ is an *initial PET*, then put it into H_i as one *candidate translation* of T_{ni} 2.3.2) else a list of candidate translations of T_{ni} are derived from the *abstract PET* ξ by replacing the non-terminal leaf nodes of ξ_t with candidate translations (which are stored in H_0~H_{i-1}) of the corresponding source subtrees that are not covered by the current *abstract PET* ξ. 2.4) **end do** 2.5) H_i pruning 3) **end do** 4) output the best *candidate translation* of T_s as T_t

The above algorithm maps the source subtrees to target ones recursively in the post-order sequence of source

[2] A candidate translation is a target subtree with accumulated feature values and accumulated probability.

[3] The post-order traversal can guarantee that when translating the current subtree rooted at current node, all subtrees rooted at descendants of the current node have already been translated before.

[4] A *PET* < ξ_s , ξ_t , $\tilde{A}$ > or a structure mapping rule is usable to a parse tree T if and only if ξ_s is rooted at the root of T and exactly covers a certain upper part of T.

Data Set	# of Sentence Pair	# of Chinese Word	# of English Word
Training Set	9,000	75,026	78,223
Development Set	528	4,432	4,630
Test Set	1,000	8,334	8,614

Table 1: Statistics of our experimental data

System	Features							
	d	$\phi(e \mid f)$	*lex*(*e*|*f*)	*pp*	*wp*	*lm*	$\phi(f \mid e)$	*lex*(*f*|*e*)
Pharaoh	0.047	0.232	-0.025	0.126	-0.0995	0.167	0.130	0.172
SCFG	—	0.191	-0.03	0.054	0.318	0.180	0.20	0.012
STSG	—	0.209	-0.045	-0.207	0.152	0.227	0.148	0.010

Table 2: Feature weights obtained by MER training on the development set

system	# of extracted rules	# of rules used in testing
Pharaoh	499,423	64,491
SCFG	70,000	24,302
STSG	2,629,146	98,422

Table 3: Extracted rules

system	BLEU4
Pharaoh	0.1208 ± 0.0069
SCFG	0.0867 ± 0.0048
STSG	0.1394 ± 0.0073

Table 4: System performance with 95% confidence intervals

subtrees. When translating a subtree, if the source elementary tree is equal to the subtree (i.e., the current *PET* is an *initial PET*), then the target elementary tree is a candidate translation (line 2.3.1). Otherwise, we have to combine the current *abstract PET* with previously generated *candidate translations* to form the current new *candidate translations* (line 2.3.2). This combination operation is the most time-consuming in our decoder. Hence, to speed up the decoder, we use several thresholds to limit search beams. For *PET* pruning (line 2.2), we use a fixed threshold (*pTableLen*) that specifies the maximum number of *PET*s and a probability threshold (*pTablePro*) that specifies the minimal probability of a *PET*. For *candidate translations* pruning (line 2.5), we also use a fixed threshold (*hTableLen*) and a probability threshold (*hTablePro*) to remove unpromising hypothesises. These pruning techniques are widely used in SMT and speech recognition. In addition, we only keep the best one among the same translations generated from different paths in order to further speed up the decoder and produce better n-best list. With regard to language model features, we use the method of *cube pruning* (Chiang, 2007) to incorporate the language model score into the feature function.

Experiments

The aim of our experiments is to verify the effectiveness of our STSG-based tree-to-tree alignment model for SMT.

Experimental Settings

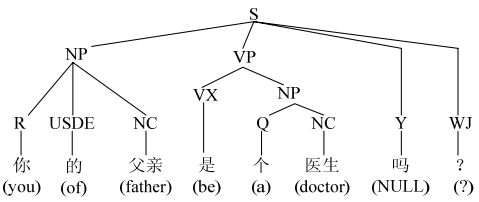
Dataset and Evaluation

Our experiments were on Chinese-to-English translation. We use part of HIT Chinese-English corpus (Yajuan Lv, 2003) as our experimental data. It is a balance corpus, gathered from various newspapers, newswires and broadcasts and covering many different topics. Table 1 shows the statistics on the corpus. We use an automatic parser to parse the bilingual corpus. To minimize the effect of parse errors on our system performance, we check the parse tree results manually before training and testing. This enables our evaluation on our tree-to-tree model more accurate. In addition, there is only one reference in our test set.

For language model, we used SRI Language Modeling Toolkit (Stolcke, 2002) to train a trigram language model with modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman 1998) on the 9k English sentences. Our evaluation metric is BLEU (Papineni et al., 2002), as calculated by the NIST script (version 11a) with its default settings, which performs case-insensitive matching of *n*-grams up to $n = 4$. Instead of GIZA++ (Och and Ney, 2004) which usually shows much lower performance on small data set, we use Lv's alignment toolkits (Yajuan Lv, 2003) to do m-n word alignment for each sentence pair. For MER training (Och, 2003), which tunes the feature weights to maximize the system's BLEU score on development set, we use Koehn's trainer (Koehn, 2004a) for the phrase-based model (Pharaoh), and further modify it for our tree-based system. For significance test, we use Zhang et al's implementation (Zhang et al, 2004), which uses bootstrapping resampling (Koehn, 2004b).

Implementation

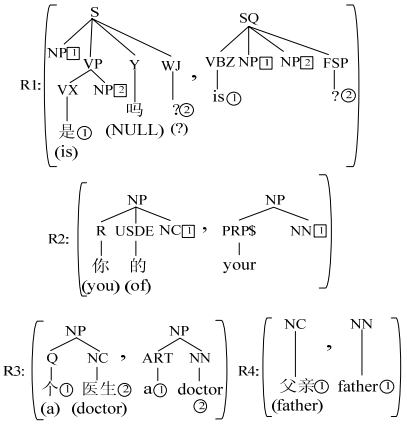
We implement our system using ANSI C++ in Linux, and set two baseline systems for comparison. One is Pharaoh (Koehn, 2003; Koehn, 2004a), a phrase-based translation



(a) Input Chinese parse tree

Input:	你/you 的/of 父亲/father 是/be 个/a 医生/doctor 吗/null ?/?
Pharaoh:	your father is a doctor?
SCFG:	do your father is a doctor?
STSG:	**Is your father a doctor?**

(b) The best translation results using the three methods



(c) STSG rules used

R1: S(NP[0],VP[1],Y(吗),WJ[2]) ↔ SQ(VB(do),NP[0],VP[1],FSP[2])		**P1:** <你(you) 的(of) , your>
R2: NP(R(你),USDE(的),NC(父亲)) ↔ NP(PRP$(your),NN(father))		**P2:** <父亲(father) 是(be) , father is>
R3: VP(VX[0],NP[1]) ↔ VP(VBZ[0],NP[1])		**P3:** <个(a) 医生(doctor) , a doctor>
R4: VX(是) ↔ VBZ(is)	**R6:** NC(医生) ↔ NN(doctor)	**P4:** **<? , ?>**
R5: NP(Q(个),NC[0]) ↔ NP(ART(a),NN[0])	**R7:** WJ(？) ↔ FSP(?)	

(d) Translation rules used in generating the best translation results: SCFG (left) and Phrase-based (right)

Figure 4: A translation example using the three methods

model; another one is a SCFG-based tree-to-tree translation model[5]. For Pharaoh, we use default settings for whole processing, including phrase table extraction, model training, MER training and decoding. For SCFG and our model, we use the same settings except the parameter *h* (*h*=5 for STSG and *h*=2 for SCFG). We optimize these parameters on development set and obtain the following settings: *c*=5; *pTableLen*=30; *pTablePro*=-100 (log probability); *hTableLen*=100 and *hTablePro*=-100 (log probability).

Experimental Results

For Pharaoh, eight default features are used: distortion model *d*, a trigram *lm*, phrase translation probability $\phi(e \mid f)$ and $\phi(f \mid e)$, lexical weightings *lex*(*e*|*f*) and *lex*(*f*|*e*), phrase penalty *pp* and word penalty *wp*. For the other two systems, the seven features described previously in this paper are used. The different feature weights obtained by MER training are showed in Table 2. Table 3 reports the number of extracted translation rules.

Table 4 compares the performance of the three systems with 95% confidence intervals. We show that:

1) Our STSG-based tree-to-tree model statistically significantly outperforms ($p < 0.01$) Pharaoh with an absolute improvement of 0.0186 (0.1394-0.1208) in BLEU score, representing a relative performance improvement of 15.3% (0.0186/0.1208). This suggests that a) the linguistically motivated structure features are very useful for SMT in modeling global reordering and structure transfer while phrase-based method is only effective for modeling local reordering and b) our STSG-based model is very effective in capturing such kinds of features since we allow any tree node operations: insertion, deletion and substitution among any different nodes.
2) Our STSG-based model statistically significantly outperforms ($p < 0.01$) the SCFG-based model with an absolute improvement of 0.0527 (0.1394-0.0867) in BLEU score, representing a relative performance improvement of 60.8% (0.0527/0.0867). This is largely because SCFG only allows sibling nodes reordering while STSG allows any node reordering within an elementary tree.
3) SCFG-based model also performs much worse than Pharaoh. This further verifies that linguistically motivated two-layer SCFG rule is inadequate in modeling language structure transfer.

Figure 4 exemplifies the advantage of STSG-based over SCFG and Phrase-based models for machine translation.

Conclusions

In this paper, we study how to utilize linguistic syntax structure features for SMT. The experimental results on the small dataset shows that our proposed STSG-based tree-to-tree alignment method is much more effective in modeling global reordering and structure transfer than phrase-based and SCFG-based methods. In the future, we will test our method on large data set using automatic

[5] The SCFG-based tree-to-tree model is easily implemented based on our STSG-based tree-to-tree model by setting the parameter *h* (the maximal height of an elementary tree) to 2.

parser. We will also study how to optimize the translation rule set.

Acknowledgements

We are grateful to Prof. YANG Muyun for kindly letting us use the HIT bi-parsed corpus and other supporting toolkits.

Bibliographical References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer. (1993). The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2):263-311

Eugene Charniak. (1997) Statistical parsing with a context-free grammar and word statistics. AAAI-97

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

David Chiang. (2006). An Introduction to Synchronous Grammars. Tutorial on ACL-06

David Chiang. (2007). Hierarchical phrase-based translation. Computational Linguistics, 33(2)

Brooke Cowan, Ivona Kucerova and Michael Collins. (2006). A discriminative model for tree-to-tree translation. EMNLP. 232-241

Bonnie J. Dorr (1994). Machine Translation Divergences: A formal description and proposed solution. Computational Linguistics, 20(4): 597-633

Yuan Ding and Martha Palmer. (2005). Machine translation using probabilistic synchronous dependency insertion grammars. ACL-05. 541-548

Jason Eisner. (2003). Learning non-isomorphic tree mappings for machine translation. ACL-03 (companion volume)

Heidi J. Fox. (2002). Phrasal Cohesion and Statistical Machine Translation. EMNLP-02

Michel Galley, Mark Hopkins, Kevin Knight and Daniel Marcu. (2004). What's in a translation rule? HLT-NAACL-2004

Daniel Gildea. (2003). Loosely Tree-Based Alignment for Machine Translation. ACL-03. 80-87

Jonathan Graehl and Kevin Knight. (2004). Training tree transducers. HLT-NAACL-2004. 105-112

Liang Huang, Kevin Knight and Aravind Joshi. (2006). Statistical Syntax-Directed Translation with Extended Domain of Locality. AMTA-06. (poster)

Reinhard Kenser and Hermann Ney. (1995). Improved backing-off for M-gram language modeling. ICASSP-95, 181-184

Philipp Koehn, Franz J. Och and Daniel Marcu. (2003). Statistical phrase-based translation. HLT-NAACL-03. 127-133

Philipp Koehn. (2004a). Pharaoh: a beam search decoder for phrase-based statistical machine translation models. AMTA-04, 115-124

Philipp Koehn. (2004b). Statistical significance tests for machine translation evaluation. EMNLP-04, 388-395

Yang Liu, Qun Liu and Shouxun Lin. (2006). Tree-to-String Alignment Template for Statistical Machine Translation. ACL-06, 609-616

Yajuan Lv. (2003) Research on bilingual corpus alignment and automatic translation knowledge acquisition. Ph.D. Thesis. Harbin Institute of Technology, 2003 (in Chinese)

Daniel Marcu and William Wong. (2002). A phrase-based, joint probability model for statistical machine translation. EMNLP-02, 133-139

Dan Melamed. (2004). Statistical machine translation by parsing. ACL-04. 653-660

Franz J. Och and Hermann Ney. (2002). Discriminative training and maximum entropy models for statistical machine translation. ACL-02, 295-302

Franz J. Och. (2003). Minimum error rate training in statistical machine translation. ACL-03, 160-167

Franz J. Och and Hermann Ney. (2004). The alignment template approach to statistical machine translation. Computational Linguistics, 30(4):417-449

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin and Dragomir Radev. (2004). A Smorgasbord of Features for Statistical Machine Translation. HLT-NAACL-04

Kishore Papineni, Salim Roukos, ToddWard and Wei-Jing Zhu. (2002). BLEU: a method for automatic evaluation of machine translation. ACL-02. 311-318

Chris Quirk and Arul Menezes. (2006). Do we need phrases? Challenging the conventional wisdom in Statistical Machine Translation. ACL-06. 9-16

Chris Quirk, Arul Menezes and Colin Cherry. (2005). Dependency treelet translation: Syntactically informed phrasal SMT. ACL-05. 271-279

Stefan Riezler and John T. Maxwell III. (2006). Grammatical Machine Translation. HLT-NAACL-06. 248-255

S. Shieber. (2004). Synchronous grammars as tree transducers. In Proceedings of the Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms

S. Shieber and Y. Schabes. (1990). Synchronous tree adjoining grammars. COLING-90

Andreas Stolcke. (2002). SRILM - an extensible language modeling toolkit. ICSLP-02. 901-904

Benjamin Wellington, Sonjia Waxmonsky and I. Dan Melamed. (2006). Empirical Lower Bounds on the Complexity of Translational Equivalence. ACL-06. 977-984

Dekai Wu. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, 23(3):377-403

Kenji Yamada and Kevin Knight. (2001). A syntax-based statistical translation model. ACL-01. 523-530

Hao Zhang, Liang Huang, Daniel Gildea and Kevin Knight. (2006). Synchronous Binarization for Machine Translation. HLT-NAACL-06. 256-263

Ying Zhang, Stephan Vogel and Alex Waibel. (2004). Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? LREC-04, 2051-2054

Andreas Zollmann and Ashish Venugopal. (2006). Syntax Augmented Machine Translation visa Chart Parsing. SMT workshop in HLT-NAACL-06. 138-141