Preference Grammars: Softening Syntactic Constraints to Improve Statistical Machine Translation

Ashish Venugopal Andreas Zollmann Noah A. Smith Stephan Vogel
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{ashishv,zollmann,nasmith,vogel}@cs.cmu.edu

Abstract

We propose a novel probabilistic synchoronous context-free grammar formalism for statistical machine translation, in which syntactic nonterminal labels are represented as "soft" preferences rather than as "hard" matching constraints. This formalism allows us to efficiently score *unlabeled* synchronous derivations without forgoing traditional syntactic constraints. Using this score as a feature in a log-linear model, we are able to approximate the selection of the most likely unlabeled derivation. This helps reduce fragmentation of probability across differently labeled derivations of the same translation. It also allows the importance of syntactic preferences to be learned alongside other features (e.g., the language model) and for particular labeling procedures. We show improvements in translation quality on small and medium sized Chinese-to-English translation tasks.

1 Introduction

Probabilistic synchronous context-free grammars (PSCFGs) define weighted production rules that are automatically learned from parallel training data. As in classical CFGs, these rules make use of nonterminal symbols to generalize beyond lexical modeling of sentences. In MT, this permits translation and reordering to be conditioned on more abstract notions of context. For example,

$$\text{VP} \rightarrow \text{ne VB}_1 \text{ pas \# do not VB}_1$$

represents the discontiguous translation of the French words "ne" and "pas" to "do not", in the context of the *labeled* nonterminal symbol "VB" (representing syntactic category "verb"). Translation with PSCFGs is typically expressed as the problem of finding the maximum-weighted derivation consistent with the source sentence, where the scores are defined (at least in part) by $\mathbb{R}$-valued weights associated with the rules. A PSCFG derivation is a synchronous parse tree.

Defining the translation function as finding the best derivation has the unfortunate side effect of forcing differently-derived versions of the same target sentence to compete with each other. In other words, the true score of each translation is "fragmented" across many derivations, so that each translation's most probable derivation is the only one that matters. The more Bayesian approach of finding the most probable *translation* (integrating out the derivations) instantiates an NP-hard inference problem even for simple word-based models (Knight, 1999); for grammar-based translation it is known as the consensus problem (Casacuberta and de la Higuera, 2000; Sima'an, 2002).

With weights interpreted as probabilities, the maximum-weighted derivation is the maximum *a posteriori* (MAP) derivation:

$$\hat{e} \leftarrow \operatorname*{argmax}_e \max_d p(e, d \mid f)$$

where f is the source sentence, e ranges over target sentences, and d ranges over PSCFG derivations (synchronous trees). This is often described as an approximation to the most probable translation, $\operatorname{argmax}_e \sum_d p(e, d \mid f)$. In this paper, we will describe a technique that aims to find the most probable *equivalence class* of *unlabeled* derivations, rather than a single labeled derivation, reducing the fragmentation problem. Solving this problem exactly is still an NP-hard consensus problem, but we provide approximations that build on well-known PSCFG decoding methods. Our model falls somewhere between PSCFGs that extract nonterminal symbols from parse trees and treat them as part of

the derivation (Zollmann and Venugopal, 2006) and unlabeled hierarchical structures (Chiang, 2005); we treat nonterminal labels as random variables chosen at each node, with each (unlabeled) rule expressing "preferences" for particular nonterminal labels, learned from data.

The paper is organized as follows. In Section 2, we summarize the use of PSCFG grammars for translation. We describe our model (Section 3). Section 4 explains the preference-related calculations, and Section 5 addresses decoding. Experimental results using preference grammars in a log-linear translation model are presented for two standard Chinese-to-English tasks in Section 6. We review related work (Section 7) and conclude.

2 PSCFGs for Machine Translation

Probabilistic synchronous context-free grammars (PSCFGs) are defined by a source terminal set (source vocabulary) $\mathcal{T}_S$, a target terminal set (target vocabulary) $\mathcal{T}_T$, a shared nonterminal set $\mathcal{N}$ and a set $\mathcal{R}$ of rules of the form: $X \rightarrow \langle \gamma, \alpha, w \rangle$ where

- $X \in \mathcal{N}$ is a labeled nonterminal referred to as the left-hand-side of the rule.
- $\gamma \in (\mathcal{N} \cup \mathcal{T}_S)^*$ is the source side of the rule.
- $\alpha \in (\mathcal{N} \cup \mathcal{T}_T)^*$ is the target side of the rule.
- $w \in [0, \infty)$ is a nonnegative real-valued weight assigned to the rule.

For visual clarity, we will use the # character to separate the source side of the rule γ from the target side α. PSCFG rules also have an implied one-to-one mapping between nonterminal symbols in γ and nonterminals symbols in α. Chiang (2005), Zollmann and Venugopal (2006) and Galley et al. (2006) all use parameterizations of this PSCFG formalism[1].

Given a source sentence f and a PSCFG G, the translation task can be expressed similarly to monolingual parsing with a PCFG. We aim to find the most likely derivation d of the input source sentence and read off the English translation, identified by composing α from each rule used in the derivation. This search for the most likely translation under the MAP approximation can be defined as:

$$\hat{e} = \mathrm{tgt}\left(\underset{d \in \mathcal{D}(G):\mathrm{src}(d)=f}{\mathrm{argmax}} p(d)\right) \quad (1)$$

where $\mathrm{tgt}(d)$ is the target-side yield of a derivation d, and $\mathcal{D}(G)$ is the set of G's derivations. Using an n-gram language model to score derivations and rule labels to constraint the rules that form derivations, we define $p(d)$ as log-linear model in terms of the rules $r \in \mathcal{R}$ used in d as:

$$\begin{aligned} p(d) &= p_{\mathrm{LM}}(\mathrm{tgt}(d))^{\lambda_0} \times \left(\prod_{i=1}^{m} p_i(d)^{\lambda_i}\right) \\ &\quad \times p_{\mathrm{syn}}(d)^{\lambda_{m+1}} / Z(\vec{\lambda}) \\ p_i(d) &= \prod_{r \in \mathcal{R}} h_i(r)^{\mathrm{freq}(r;d)} \quad (2) \\ p_{\mathrm{syn}}(d) &= \begin{cases} 1 & \text{if } d \text{ respects label constraints} \\ 0 & \text{otherwise} \end{cases} \quad (3) \end{aligned}$$

where $\vec{\lambda} = \lambda_0 \cdots \lambda_{m+1}$ are weights that reflect the relative importance of features in the model. The features include the n-gram language model (LM) score of the target yield sequence, a collection of m rule feature functions $h_i : \mathcal{R} \rightarrow \mathbb{R}_{\geq 0}$, and a "syntax" feature that (redundantly) requires every nonterminal token to be expanded by a rule with that nonterminal on its left-hand side. $\mathrm{freq}(r; d)$ denotes the frequency of the rule r in the derivation d. Note that λ_{m+1} can be effectively ignored when p_{syn} is defined as in Equation 3. $Z(\vec{\lambda})$ is a normalization constant that does not need to be computed during search under the argmax search criterion in Equation 1. Feature weights $\vec{\lambda}$ are trained discriminatively in concert with the language model weight to maximize the BLEU (Papineni et al., 2002) automatic evaluation metric via Minimum Error Rate Training (MERT) (Och, 2003).

We use the open-source PSCFG rule extraction framework and decoder from Zollmann et al. (2008) as the framework for our experiments. The asymptotic runtime of this decoder is:

$$\mathcal{O}\left(|f|^3 \left[|\mathcal{N}||\mathcal{T}_T|^{2(n-1)}\right]^K\right) \quad (4)$$

where K is the maximum number of nonterminal symbols per rule, $|f|$ the source sentence length, and

[1] Galley et al. (2006) rules are formally defined as tree transducers but have equivalent PSCFG forms.

n is the order of the n-gram LM that is used to compute p_{LM}. This constant factor in Equation 4 arises from the dynamic programming item structure used to perform search under this model. Using notation from Chiang (2007), the corresponding item structure is:

$$[X, i, j, q(\alpha)] : w \tag{5}$$

where X is the nonterminal label of a derivation, i, j define a span in the source sentence, and $q(\alpha)$ maintains state required to compute $p_{\text{LM}}(\alpha)$. Under the MAP criterion we can discard derivations of lower weight that share this item structure, but in practice we often require additional lossy pruning to limit the number of items produced. The Syntax-Augmented MT model of Zollmann and Venugopal (2006), for instance, produces a very large nonterminal set using "slash" (NP/NN → *the great*) and "plus" labels (NP+VB → *she went*) to assign syntactically motivated labels for rules whose target words do not correspond to constituents in phrase structure parse trees. These labels lead to fragmentation of probability across many derivations for the same target sentence, worsening the impact of the MAP approximation. In this work we address the increased fragmentation resulting from rules with labeled nonterminals compared to unlabeled rules (Chiang, 2005).

3 Preference Grammars

We extend the PSCFG formalism to include soft "label preferences" for unlabeled rules that correspond to alternative labelings that have been encountered in training data for the unlabeled rule form. These preferences, estimated via relative frequency counts from rule occurrence data, are used to estimate the feature $p_{\text{syn}}(d)$, the probability that an unlabeled derivation can be generated under traditional syntactic constraints. In classic PSCFG, $p_{\text{syn}}(d)$ enforces a hard syntactic constraint (Equation 3). In our approach, label preferences influence the value of $p_{\text{syn}}(d)$.

3.1 Motivating example

Consider the following labeled Chinese-to-English PSCFG rules:

(4) S → 我 在 哪 能 VB_1 #
a place where I can VB_1
(3) S → 我 在 哪 能 VP_1 #
a place where I can VP_1
(2) SBAR → 我 在 哪 能 VP_1 #
a place where I can VP_1
(1) FRAG → 我 在 哪 能 AUX_1 #
a place where I can AUX_1

(8) VB → 吃饭 # eat
(1) VP → 吃饭 # eat
(1) NP → 吃饭 # eat

(10) NN → 吃饭 # dish

where the numbers are frequencies of the rule from the training corpus. In classical PSCFG we can think of the nonterminals mentioned in the rules as hard constraints on which rules can be used to expand a particular node; e.g., a VP can only be expanded by a VP rule. In Equation 2, $p_{\text{syn}}(d)$ explicitly enforces this hard constraint. Instead, we propose softening these constraints. In the rules below, labels are represented as soft preferences.

(10) X → 我 在 哪 能 X_1 #
a place where I can X_1

$$\left\{ \begin{array}{rcl} p(H_0 = \text{S}, H_1 = \text{VB} \mid r) & = & 0.4 \\ p(H_0 = \text{S}, H_1 = \text{VP} \mid r) & = & 0.3 \\ p(H_0 = \text{SBAR}, H_1 = \text{VP} \mid r) & = & 0.2 \\ p(H_0 = \text{FRAG}, H_1 = \text{AUX} \mid r) & = & 0.1 \end{array} \right\}$$

(10) X → 吃饭 # eat

$$\left\{ \begin{array}{rcl} p(H_0 = \text{VB} \mid r) & = & 0.8 \\ p(H_0 = \text{VP} \mid r) & = & 0.1 \\ p(H_0 = \text{NP} \mid r) & = & 0.1 \end{array} \right\}$$

(10) X → 吃饭 # dish

$$\left\{ \; p(H_0 = \text{NN} \mid r) \quad = \quad 1.0 \; \right\}$$

Each unlabeled form of the rule has an associated distribution over labels for the nonterminals referenced in the rule; the labels are random variables H_i, with H_0 the left-hand-side label. These unlabeled rule forms are simply packed representations of the original labeled PSCFG rules. In addition to the usual features $h_i(r)$ for each rule, estimated based on unlabeled rule frequencies, we now

have label preference distributions. These are estimated as relative frequencies from the labelings of the base, unlabeled rule. Our primary contribution is how we compute $p_{\text{syn}}(d)$—the probability that an unlabeled derivation adheres to traditional syntactic constraints—for derivations built from preference grammar rules. By using $p_{\text{syn}}(d)$ as a feature in the log-linear model, we allow the MERT framework to evaluate the importance of syntactic structure relative to other features.

The example rules above highlight the potential for $p_{\text{syn}}(d)$ to affect the choice of translation. The translation of the Chinese word sequence 我 在 哪 能 吃饭 can be performed by expanding the nonterminal in the rule "a place where I can X_1" with either "eat" or "dish." A hierarchical system (Chiang, 2005) would allow either expansion, relying on features like p_{LM} to select the best translation since both expansions occurred the same number of times in the data.

A richly-labeled PSCFG as in Zollmann and Venugopal (2006) would immediately reject the rule generating "dish" due to hard label matching constraints, but would produce three identical, competing derivations. Two of these derivations would produce S as a root symbol, while one derivation would produce SBAR. The two S-labeled derivations compete, rather than reinforce the choice of the word "eat," which they both make. They will also compete for consideration by any decoder that prunes derivations to keep runtime down.

The rule preferences indicate that VB and VP are both valid labels for the rule translating to "eat", and both of these labels are compatible with the arguments expected by "a place where I can X_1". Alternatively, "dish" produces a NN label which is not compatible with the arguments of this higher-up rule. We design $p_{\text{syn}}(d)$ to reflect compatibility between two rules (one expanding a right-hand side nonterminal in the other), based on label preference distributions.

3.2 Formal definition

Probabilistic synchronous context-free *preference* grammars are defined as PSCFGs with the following additional elements:

- $\mathcal{H}$: a set of implicit labels, not to be confused with the *explicit* label set $\mathcal{N}$.
- π: $\mathcal{H} \rightarrow \mathcal{N}$, a function that associates each implicit label with a single explicit label. We can therefore think of $\mathcal{H}$ symbols as refinements of the nonterminals in $\mathcal{N}$ (Matsusaki et al., 2005).
- For each rule r, we define a probability distribution over vectors $\vec{h}$ of implicit label bindings for its nonterminals, denoted $p_{\text{pref}}(\vec{h} \mid r)$. $\vec{h}$ includes bindings for the left-hand side nonterminal (h_0) as well as each right-hand side nonterminal ($h_1, ..., h_{|\vec{h}|}$). Each $h_i \in \mathcal{H}$.

When $\mathcal{N}, \mathcal{H}$ are defined to include just a single generic symbol as in (Chiang, 2005), we produce the unlabeled grammar discussed above. In this work, we define

- $\mathcal{N} = \{S, X\}$
- $\mathcal{H} = \{\text{NP}, \text{DT}, \text{NN} \cdots\} = \mathcal{N}_{\text{SAMT}}$

where $\mathcal{N}$ corresponds to the generic labels of Chiang (2005) and $\mathcal{H}$ corresponds to the syntactically motivated SAMT labels from (Zollmann and Venugopal, 2006), and π maps all elements of $\mathcal{H}$ to X. We will use $\text{hargs}(r)$ to denote the set of all $\vec{h} = \langle h_0, h_1, ..., h_k \rangle \in \mathcal{H}^{k+1}$ that are valid bindings for the rule with nonzero preference probability.

The preference distributions p_{pref} from each rule used in d are used to compute $p_{\text{syn}}(d)$ as described next.

4 Computing feature $p_{\text{syn}}(d)$

Let us view a derivation d as a collection of nonterminal tokens n_j, $j \in \{1, ..., |d|\}$. Each n_j takes an explicit label in $\mathcal{N}$. The score $p_{\text{syn}}(d)$ is a product, with one factor per n_j in the derivation d:

$$p_{\text{syn}}(d) = \prod_{j=1}^{|d|} \phi_j \tag{6}$$

Each ϕ_j factor considers the two rules that n_j participates in. We will refer to the rule above nonterminal token n_j as $\underline{r}_j$ (the nonterminal is a child in this rule) and the rule that expands nonterminal token j as $\overline{r}_j$.

The intuition is that derivations in which these two rules agree (at each j) about the implicit label

for n_j, in $\mathcal{H}$ are preferable to derivations in which they do not. Rather than making a decision about the implicit label, we want to reward p_{syn} when $\underline{r}_j$ and $\underline{r}_j$ are consistent. Our way of measuring this consistency is an inner product of preference distributions:

$$\phi_j \propto \sum_{h \in \mathcal{H}} p_{\text{pref}}(h \mid \underline{r}_j) p_{\text{pref}}(h \mid \overline{r}_j) \tag{7}$$

This is not quite the whole story, because $p_{\text{pref}}(\cdot \mid r)$ is defined as a joint distribution of *all* the implicit labels within a rule; the implicit labels are not independent of each other. Indeed, we want the implicit labels within each rule to be mutually consistent, i.e., to correspond to one of the rule's preferred labelings, for both $\text{hargs}(\underline{r})$ and $\text{hargs}(\overline{r})$.

Our approach to calculating p_{syn} within the dynamic programming algorithm is to recursively calculate preferences for each chart item based on (a) the smaller items used to construct the item and (b) the rule that permits combination of the smaller items into the larger one. We describe how the preferences for chart items are calculated. Let a chart item be denoted $[X, i, j, u, ...]$ where $X \in \mathcal{N}$ and i and j are positions in the source sentence, and

$$u : \{h \in \mathcal{H} \mid \pi(h) = X\} \rightarrow [0, 1]$$

(where $\sum_h u(h) = 1$) denotes a distribution over possible X-refinement labels. We will refer to it below as the *left-hand-side preference distribution*. Additional information (such as language model state) may also be included; it is not relevant here.

The simplest case is for a nonterminal token n_j that has no nonterminal children. Here the left-hand-side preference distribution is simply given by

$$u(h) = p_{\text{pref}}(h \mid \overline{r}_j) \ .$$

and we define the p_{syn} factor to be $\phi_j = 1$.

Now consider the dynamic programming step of combining an already-built item $[X, i, j, u, ...]$ rooted by explicit nonterminal X, spanning source sentence positions i to j, with left-hand-side preference distribution u, to build a larger item rooted by Y through a rule $r = Y \rightarrow \langle \gamma X_1 \gamma', \alpha X_1 \alpha', w \rangle$ with preferences $p_{\text{pref}}(\cdot \mid r)$.[2] The new item will have signature $[Y, i - |\gamma|, j + |\gamma'|, v, ...]$. The left-hand-side preferences v for the new item are calculated as follows:

$$v(h) = \frac{\tilde{v}(h)}{\sum_{h'} \tilde{v}(h')} \quad \text{where} \tag{8}$$

$$\tilde{v}(h) = \sum_{h' \in \mathcal{H} : \langle h, h' \rangle \in \text{hargs}(r)} p_{\text{pref}}(\langle h, h' \rangle \mid r) \times u(h')$$

Renormalizing keeps the preference vectors on the same scale as those in the rules. The p_{syn} factor ϕ, which is factored into the value of the new item, is calculated as:

$$\phi = \sum_{h' \in \mathcal{H} : \langle h, h' \rangle \in \text{hargs}(r)} u(h') \tag{9}$$

so that the value considered for the new item is $w \times \phi \times ...$, where factors relating to p_{LM}, for example, may also be included. Coming back to our example, if we let $\overline{r}$ be the leaf rule producing "eat" at shared nonterminal n_1, we generate an item with:

$$u = \langle u(\text{VB}) = 0.8, u(\text{VP}) = 0.1, u(\text{NP}) = 0.1 \rangle$$
$$\phi_1 = 1$$

Combining this item with X → ⟨ 我 在 哪 能 X_1 # a place where I can X_1 ⟩ as $\underline{r}_2$ at nonterminal n_2 generates a new target item with translation "a place where I can eat", $\phi_2 = 0.9$ and v as calculated in Fig. 1. In contrast, $\phi_2 = 0$ for the derivation where $\overline{r}$ is the leaf rule that produces "dish".

This calculation can be seen as a kind of single-pass, bottom-up message passing inference method embedded within the usual dynamic programming search.

5 Decoding Approximations

As defined above, accurately computing $p_{\text{syn}}(d)$ requires extending the chart item structure with u. For models that use the n-gram LM feature, the item structure would be:

$$[X, i, j, q(\alpha), u] : w \tag{10}$$

Since u effectively summarizes the choice of rules in a derivation, this extension would partition the

[2] We assume for the discussion that $\alpha, \alpha' \in \mathcal{T}_S^*$ and $\gamma, \gamma' \in \mathcal{T}_T^*$. If there are multiple nonterminals on the right-hand side of the rule, we sum over the longer sequences in $\text{hargs}(r)$ and include appropriate values from the additional "child" items' preference vectors in the product.

$\tilde{v}(\text{S}) = p_{pref}(\langle h = \text{S}, h' = \text{VB}\rangle \mid \underline{r})u(\text{VB}) + p_{pref}(\langle h = \text{S}, h' = \text{VP}\rangle \mid \underline{r})u(\text{VP}) = (0.4 \times 0.8) + (0.3 \times 0.1) = 0.35$
$\tilde{v}(\text{SBAR}) = p(\langle h = \text{SBAR}, h' = \text{VP}\rangle \mid \underline{r})u(\text{VP}) = (0.2 \times 0.1) = 0.02$
$v = \langle v(\text{S}) = 0.35/(\tilde{v}(\text{S}) + \tilde{v}(\text{SBAR})), v(\text{SBAR}) = 0.02/\tilde{v}(\text{S}) + \tilde{v}(\text{SBAR})\rangle = \langle v(\text{S}) = 0.35/0.37, v(\text{SBAR}) = 0.02/0.37\rangle$
$\phi_2 = u(\text{VB}) + u(\text{VP}) = 0.8 + 0.1 = 0.9$

Figure 1: Calculating v and ϕ_2 for the running example.

search space *further*. To prevent this partitioning, we follow the approach of Venugopal et al. (2007). We keep track of u for the best performing derivation from the set of derivations that share $[X, i, j, q(\alpha)]$ in a first-pass decoding. In a second top-down pass similar to Huang and Chiang (2007), we can recalculate $p_{syn}(d)$ for alternative derivations in the hypergraph; potentially correcting search errors made in the first pass.

We face another significant practical challenge during decoding. In real data conditions, the size of the preference vector for a single rule can be very high, especially for rules that include multiple nonterminal symbols that are located on the left and right boundaries of γ. For example, the Chinese-to-English rule X $\rightarrow \langle$ X$_1$ 的 X$_2$ # X$_1$'s X$_2$ $\rangle$ has over 24K elements in $\text{hargs}(r)$ when learned for the medium-sized NIST task used below. In order to limit the explosive growth of nonterminals during decoding for both memory and runtime reasons, we define the following label pruning parameters:

- β_R: This parameter limits the size of $\text{hargs}(r)$ to the β_R top-scoring preferences, defaulting other values to zero.
- β_L: This parameter is the same as β_R but applied only to rules with no nonterminals. The stricter of β_L and β_R is applied if both thresholds apply.
- β_P: This parameter limits the number labels in *item* preference vectors (Equation 8) to the β_P most likely labels during decoding, defaulting other preferences to zero.

6 Empirical Results

We evaluate our preference grammar model on small (IWSLT) and medium (NIST) data Chinese-to-English translation tasks (described in Table 1). IWSLT is a limited domain, limited resource task (Paul, 2006), while NIST is a broadcast news task with wide genre and domain coverage. We use a subset of the full training data (67M words of English text) from the annual NIST MT Evaluation. Development corpora are used to train model parameters via MERT. We use a variant of MERT that prefers sparse solutions where $\lambda_i = 0$ for as many features as possible. At each MERT iteration, a subset of features λ are assigned 0 weight and optimization is repeated. If the resulting BLEU score is not lower, these features are left at zero.

All systems are built on the SAMT framework described in Zollmann et al. (2008), using a trigram LM during search and the full-order LM during a second hypergraph rescoring pass. Reordering limits are set to 10 words for all systems. Pruning parameters during decoding limit the number of derivations at each source span to 300.

The system "Hier." uses a grammar with a single nonterminal label as in Chiang (2005). The system "Syntax" applies the grammar from Zollmann and Venugopal (2006) that generates a large number of syntactically motivated nonterminal labels. For the NIST task, rare rules are discarded based on their frequency in the training data. Purely lexical rules (that include no terminal symbols) that occur less than 2 times, or non-lexical rules that occur less than 4 times are discarded.

IWSLT task: We evaluate the preference grammar system "Pref." with parameters $\beta_R = 100$, $\beta_L = 5$, $\beta_P = 2$. Results comparing systems Pref. to Hier. and Syntax are shown in Table 2. Automatic evaluation results using the preference grammar translation model are positive. The preference grammar system shows improvements over both the Hier. and Syntax based systems on both unseen evaluation sets IWSLT 2007 and 2008. The improvements are clearest on the BLEU metric (matching the MERT training criteria). On 2007 test data, Pref. shows a 1.2-point improvement over Hier., while on the 2008 data, there is a 0.6-point improvement. For the IWSLT task, we report additional au-

System Name	Words in Target Text	LM singleton 1-n-grams (n)	Dev.	Test
IWSLT	632K	431K (5)	IWSLT06	IWSLT07,08
NIST	67M	102M (4)	MT05	MT06

Table 1: Training data configurations used to evaluate preference grammars. The number of words in the target text and the number of singleton 1-n-grams represented in the complete model are the defining statistics that characterize the scale of each task. For each LM we also indicate the order of the n-gram model.

System	Dev BLEU (lpen) ↑	2007 BLEU (lpen) ↑	2008 BLEU (lpen) ↑	2008 WER ↓	2008 PER ↓	2008 MET. ↑	2008 GTM ↑
Hier.	28.0 (0.89)	37.0 (0.89)	45.9 (0.91)	44.5	39.9	61.8	70.7
Syntax	30.9 (0.96)	35.5 (0.94)	45.3 (0.95)	45.7	40.4	62.1	71.5
Pref.	28.3 (0.88)	38.2 (0.90)	46.3 (0.91)	43.8	40.0	61.7	71.2

Table 2: Translation quality metrics on the IWSLT translation task, with IWSLT 2006 as the development corpora, and IWSLT 2007 and 2008 as test corpora. Each metric is annotated with an ↑ if increases in the metric value correspond to increase in translation quality and a ↓ if the opposite is true. We also list length penalties for the BLEU metric to show that improvements are not due to length optimizations alone.

tomatic evaluation metrics that generally rank the Pref. system higher than Hier. and Syntax. As a further confirmation, our feature selection based MERT chooses to retain λ_{m+1} in the model. While the IWSLT results are promising, we perform a more complete evaluation on the NIST translation task.

NIST task: This task generates much larger rule preference vectors than the IWSLT task simply due to the size of the training corpora. We build systems with both $\beta_R = 100, 10$ varying β_P. Varying β_P isolates the relative impact of propagating alternative nonterminal labels within the preference grammar model. $\beta_L = 5$ for all NIST systems. Parameters λ are trained via MERT on the $\beta_R = 100$, $\beta_L = 5$, $\beta_P = 2$ system. BLEU scores for each preference grammar and baseline system are shown in Table 3, along with translation times on the test corpus. We also report length penalties to show that improvements are not simply due to better tuning of output length.

The preference grammar systems outperform the Hier. baseline by 0.5 points on development data, and upto 0.8 points on unseen test data. While systems with $\beta_R = 100$ take significantly longer to translate the test data than Hier., setting $\beta_R = 10$ takes approximately as long as the Syntax based system but produces better slightly better results (0.3 points).

The improvements in translation quality with the preference grammar are encouraging, but how much of this improvement can simply be attributed to MERT finding a better local optimum for parameters λ? To answer this question, we use parameters $\lambda^\star$ optimized by MERT for the preference grammar system to run a purely hierarchical system, denoted Hier.($\lambda^\star$), which ignores the value of λ_{m+1} during decoding. While almost half of the improvement comes from better parameters learned via MERT for the preference grammar systems, 0.5 points can be still be attributed purely to the feature p_{syn}. In addition, MERT does not set parameter λ_{m+1} to 0, corroborating the value of the p_{syn} feature again. Note that Hier.($\lambda^\star$) achieves better scores than the Hier. system which was trained via MERT without p_{syn}. This highlights the local nature of MERT parameter search, but also points to the possibility that training with the feature p_{syn} produced a more diverse derivation space, resulting in better parameters λ. We see a very small improvement (0.1 point) by allowing the runtime propagation of more than 1 nonterminal label in the left-hand side posterior distribution, but the improvement doesn't extend to $\beta_P = 5$. Improved integration of the feature $p_{syn}(d)$ into decoding might help to widen this gap.

System	Dev. BLEU (lpen)	Test BLEU (lpen)	Test time (h:mm)
Baseline Systems			
Hier.	34.1 (0.99)	31.8 (0.95)	0:12
Syntax	34.7 (0.99)	32.3 (0.95)	0:45
Hier.($\lambda^\star$)	-	32.1 (0.95)	0:12
Preference Grammar: $\beta_R = 100$			
$\beta_P = 1$	-	32.5 (0.96)	3:00
$\beta_P = 2$	34.6 (0.99)	32.6 (0.95)	3:00
$\beta_P = 5$	-	32.5 (0.95)	3:20
Preference Grammar: $\beta_R = 10$			
$\beta_P = 1$	-	32.5 (0.95)	1:03
$\beta_P = 2$	-	32.6 (0.95)	1:10
$\beta_P = 5$	-	32.5 (0.95)	1:10

Table 3: Translation quality and test set translation time (using 50 machines with 2 tasks per machine) measured by the BLEU metric for the NIST task. NIST 2006 is used as the development (Dev.) corpus and NIST 2007 is used as the unseen evaluation corpus (Test). Dev. scores are reported for systems that have been separately MERT trained, Pref. systems share parameters from a single MERT training. Systems are described in the text.

7 Related Work

There have been significant efforts in the both the monolingual parsing and machine translation literature to address the impact of the MAP approximation and the choice of labels in their respective models; we survey the work most closely related to our approach. May and Knight (2006) extract n-best lists containing unique translations rather than unique derivations, while Kumar and Byrne (2004) use the Minimum Bayes Risk decision rule to select the lowest risk (highest BLEU score) translation rather than derivation from an n-best list. Tromble et al. (2008) extend this work to lattice structures. All of these approaches only marginalize over alternative candidate derivations generated by a MAP-driven decoding process. More recently, work by Blunsom et al. (2007) propose a purely discriminative model whose decoding step approximates the selection of the most likely translation via beam search. Matsusaki et al. (2005) and Petrov et al. (2006) propose automatically learning annotations that add information to categories to improve monolingual parsing quality. Since the parsing task requires selecting the most non-annotated tree, the annotations add an additional level of structure that must be marginalized during search. They demonstrate improvements in parse quality only when a variational approximation is used to select the most likely *unannotated* tree rather than simply stripping annotations from the MAP annotated tree. In our work, we focused on approximating the selection of the most likely unlabeled derivation during search, rather than as a post-processing operation; the methods described above might improve this approximation, at some computational expense.

8 Conclusions and Future Work

We have proposed a novel grammar formalism that replaces hard syntactic constraints with "soft" preferences. These preferences are used to compute a machine translation feature ($p_{\text{syn}}(d)$) that scores unlabeled derivations, taking into account traditional syntactic constraints. Representing syntactic constraints as a feature allows MERT to train the corresponding weight for this feature relative to others in the model, allowing systems to learn the relative importance of labels for particular resource and language scenarios as well as for alternative approaches to labeling PSCFG rules.

This approach takes a step toward addressing the fragmentation problems of decoding based on maximum-weighted derivations, by summing the contributions of compatible label configurations rather than forcing them to compete. We have suggested an efficient technique to approximate $p_{\text{syn}}(d)$ that takes advantage of a natural factoring of derivation scores. Our approach results in improvements in translation quality on small and medium resource translation tasks. In future work we plan to focus on methods to improve on the integration of the $p_{\text{syn}}(d)$ feature during decoding and techniques that allow us consider more of the search space through less pruning.

Acknowledgements

We appreciate helpful comments from three anonymous reviewers. Venugopal and Zollmann were supported by a Google Research Award. Smith was supported by NSF grant IIS-0836431.

References

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2007. A discriminative latent variable model for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Francisco Casacuberta and Colin de la Higuera. 2000. Computational complexity of problems on probabilistic grammars and transducers. In *Proc. of the 5th International Colloquium on Grammatical Inference: Algorithms and Applications*.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Compuational Linguistics (ACL)*.

David Chiang. 2007. Hierarchical phrase based translation. *Computational Linguistics*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inferences and training of context-rich syntax translation models. In *Proceedings of the Annual Meeting of the Association for Compuational Linguistics (ACL)*, Sydney, Australia.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the Annual Meeting of the Association for Compuational Linguistics (ACL)*.

Kevin Knight. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics, Squibs and Discussion*.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Boston,MA, May 27-June 1.

Takuya Matsusaki, Yusuke Miyao, and Junichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jonathan May and Kevin Knight. 2006. A better N-best list: Practical determinization of weighted finite tree automata. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL)*.

Franz J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Compuational Linguistics (ACL)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Compuational Linguistics (ACL)*.

Michael Paul. 2006. Overview of the IWSLT 2006 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the Annual Meeting of the Association for Compuational Linguistics (ACL)*.

Khalil Sima'an. 2002. Computational complexity of probabilistic disambiguation. *Grammars*, 5(2):125–151.

Roy Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ashish Venugopal, Andreas Zollmann, and Stephan Vogel. 2007. An efficient two-pass approach to Synchronous-CFG driven statistical MT. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL)*.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation, HLT/NAACL*, New York, June.

Andreas Zollmann, Ashish Venugopal, Franz J. Och, and Jay Ponte. 2008. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In *Proceedings of the Conference on Computational Linguistics (COLING)*.