Latent Variable Synchronous CFGs for Hierarchical Translation

Abstract

Data-driven refinement of non-terminal categories has been demonstrated to be a reliable technique for improving monolingual parsing with PCFGs. In this paper, we extend these techniques to learn latent refinements of single-category synchronous grammars, so as to improve translation performance. We compare two estimators for this latent variable model: one based on EM and the other on the method of moments, and evaluate their performance on a Chinese–English translation task. The results indicate that we can achieve significant gains over the baseline with both approaches, but in particular the moments-based estimator is both faster and performs better than EM.

1 Introduction

Translation models based on synchronous context-free grammars (SCFGs) treat the translation problem as a context-free parsing problem. A parser constructs trees over the input sentence by parsing with the source language projection of a synchronous CFG, and each derivation induces translations in the target language (Chiang, 2007). However, in contrast to syntactic parsing, where linguistic intuitions can help elucidate the "right" tree structure for a grammatical sentence, no such intuitions are available for synchronous derivations, and so learning the "right" grammars is a central challenge.

Of course, learning synchronous grammars from parallel data is a widely studied problem (Wu, 1997; Blunsom et al., 2008; Levenberg et al., 2012, *inter alia*). However, there has been less exploration of learning rich non-terminal categories, largely because previous efforts to learn such categories have been coupled with efforts to learn derivation structures—a computationally formidable challenge. One popular approach has been to derive categories from source and/or target monolingual grammars (Galley et al., 2004; Zollmann and Venugopal, 2006; Hanneman and Lavie, 2013). While often successful, accurate parsers are not available in many languages: a more appealing approach is therefore to learn the category structure from the data itself.

In this work, we take a slightly different approach to previous work in synchronous grammar induction by assuming that reasonable tree structures for a parallel corpus can be chosen heuristically, and then, fixing the trees (thereby enabling us to sidestep the worst of the computational issues), we learn non-terminal categories as latent variables to explain the distribution of these synchronous trees. This technique has a long history in monolingual parsing (Petrov et al., 2006; Liang et al., 2007; Cohen et al., 2012), where it reliably yields state-of-the-art phrase structure parsers based on generative models, but we are the first to apply it to translation.

We first generalize the concept of latent PCFGs to latent variable SCFGs (§2). We then follow by a presentation of the tensor-based formulation for our parameters, a representation that makes it convenient to marginalize over latent states. Subsequently, two methods for parameter estimation are presented (§4): a spectral approach based on the method of moments, and an EM-based likelihood maximization. Results on a Chinese–English evaluation set (§5) indicate significant gains over baselines and point to the promise of using latent variable synchronous grammars in conjunction with a smaller, simpler set of rules instead of unwieldy and bloated grammars extracted via existing heuristics, where a large number of context-independent but un-generalizable rules are utilized. Hence, the hope is that this work promotes the move towards translation models that directly

model the conditional likelihood of translation rules via (potentially feature-rich) latent variable models which leverage information contained in the synchronous tree structure, instead of relying on the heuristic relative frequency parameter estimates (Koehn et al., 2003) from non-hierarchical phrase-based translation.

2 Latent Variable SCFGs

Before discussing parameter learning, we introduce latent-variable synchronous context-free grammars (L-SCFGs) and discuss an inference algorithm for marginalizing over latent states.

We extend the definition of L-PCFGs (Matsuzaki et al., 2005; Petrov et al., 2006) to synchronous grammars as used in machine translation (Chiang, 2007). A latent variable SCFG (L-SCFG) is a 6-tuple $(\mathcal{N}, m, n_s, n_t, \pi, t)$ where:

- $\mathcal{N}$ is a set of non-terminal (NT) symbols in the grammar. For HPBT, the set consists of only two symbols, **X** and a goal symbol **S**.
- $[m]$ is the set of possible hidden states associated with NTs. Aligned pairs of NTs across the source and target languages share the same hidden state.
- $[n_s]$ is the set of source side words, i.e., the source-side vocabulary, with $[n_s] \cap \mathcal{N} = \emptyset$.
- $[n_t]$ is the set of target side words, i.e., the target-side vocabulary, with $[n_t] \cap \mathcal{N} = \emptyset$.
- The synchronous production rules compose a set $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1 \cup \mathcal{R}_2$:

  • Binary rules ($\mathcal{R}_2$):

  $$a(h_1) \rightarrow \langle \alpha_1 b(h_2) \alpha_2 c(h_3) \alpha_3, \beta_1 b(h_2) \beta_2 c(h_3) \beta_3 \rangle$$

  or

  $$a(h_1) \rightarrow \langle \alpha_1 b(h_2) \alpha_2 c(h_3) \alpha_3, \beta_1 c(h_2) \beta_2 b(h_3) \beta_3 \rangle$$

  where $a, b, c \in \mathcal{N}$, $h_1, h_2, h_3 \in [m]$, $\alpha_1, \alpha_2, \alpha_3 \in [n_s]^*$ and $\beta_1, \beta_2, \beta_3 \in [n_t]^*$.

  • Unary rules ($\mathcal{R}_1$):

  $$a(h_1) \rightarrow \langle \alpha_1 b(h_2) \alpha_2, \beta_1 b(h_2) \beta_2 \rangle$$

  where $a, b \in \mathcal{N}$, $h_1, h_2 \in [m]$, $\alpha_1, \alpha_2 \in [n_s]^*$ and $\beta, \beta_2 \in [n_t]^*$.

  • Pre-terminal rules ($\mathcal{R}_0$): $a(h_1) \rightarrow \langle \alpha, \beta \rangle$ where $a \in \mathcal{N}$, $\alpha \in [n_t]^*$ and $\beta \in [n_s]^*$.

  Each of these rules is associated with a probability $t(a(h_1) \rightarrow \gamma | a, h_1)$ where γ is the right-hand side (RHS) of the rule.

- For $a \in \mathcal{N}$, $h \in [m]$, $\pi(a, h)$ is a parameter specifying the root probability of $a(h)$.

A skeletal tree (s-tree) for a sentence is the set of rules in the synchronous derivation of that sentence, without any additional latent state information or decoration. A full tree consists of an s-tree $r_1, \ldots, r_N$ together with values $h_1, \ldots, h_N$ for every NT in the tree. An important point to keep in mind in comparison to L-PCFGs is that the right-hand side (RHS) non-terminals of synchronous rules are aligned pairs across the source and target languages.

In this work, we refine the one-category grammar introduced by Chiang (2007) for hierarchical phrase-based translation (HPBT) in order to learn additional latent NT categories. Thus, the following discussion is restricted to these kinds of grammars, although the method is equally applicable in other scenarios, e.g., the extended tree-to-string transducer (**xRs**) formalism (Huang et al., 2006; Graehl et al., 2008) commonly used in syntax-directed translation.

Marginal Inference with L-SCFGs. For a parameter t of rule r, the latent state h_1 attached to the left-hand side (LHS) NT of r is associated with the outside tree for the sub-tree rooted at the LHS, and the states attached to the RHS NTs are associated with the inside trees of that NT. Since we do not assume conditional independence of these states, we need to consider all possible interactions, which can be compactly represented as a 3rd-order tensor in the case of a binary rule, a matrix (i.e., a 2nd-order tensor) for unary rules, and a vector for pre-terminal (lexical) rules. Preferences for certain outside-inside tree combinations are reflected in the values contained in these tensor structures. In this manner, we intend to capture interactions between non-local context, as represented by the outside tree, and local context, through the inside trees. We refer to these tensor structures collectively as C^r for rules $r \in \mathcal{R}$, which encompass the parameters t.

For $r \in \mathcal{R}_0 : C^r \in \mathbb{R}^{1 \times m}$; similarly for $r \in \mathcal{R}_1 : C^r \in \mathbb{R}^{m \times m}$ and $r \in \mathcal{R}_2 : C^r \in \mathbb{R}^{m \times m \times m}$. We also maintain a vector $C^{\mathbf{S}} \in \mathbb{R}^{m \times 1}$ corresponding to the parameters $\pi(\mathbf{S}, h)$ for the goal node (root). These parameters participate in tensor-vector operations: a 3rd-order tensor C^{r_2} can be multiplied along each of its three modes ($\times_0, \times_1, \times_2$), and if multiplied by an $m \times 1$ vec-

Inputs: Sentence $f_1 \dots f_N$, L-SCFG $(\mathcal{N}, S, m, n)$, parameters $C^r \in \mathbb{R}^{(m \times m \times m)}$, $\in \mathbb{R}^{(m \times m)}$, or $\in \mathbb{R}^{(1 \times m)}$ for all $r \in \mathcal{R}$, $C^{\mathbf{S}} \in \mathbb{R}^{(m \times 1)}$, hypergraph $\mathcal{H}$.
Data structures:
For each node $q \in \mathcal{H}$:

- $\boldsymbol{\alpha}(q) \in \mathbb{R}^{1 \times m}$ is a row vector of inside terms.
- $\boldsymbol{\beta}(q) \in \mathbb{R}^{m \times 1}$ is a column vector of outside terms.
- For each incoming edge $e \in \mathbf{B}(q)$ to node q, $\mu(e)$ is a marginal probability for edge (rule) e.

Algorithm:
▷ *Inside Computation*
For nodes q in topological order in $\mathcal{H}$,
  $\boldsymbol{\alpha}(q) = \mathbf{0}$
  For each incoming edge $e \in \mathbf{B}(q)$,
    tail = **t**(e), rule = **r**(e)
    if $|\text{tail}| = 0$, then $\boldsymbol{\alpha}(q) = \boldsymbol{\alpha}(q) + C^{\text{rule}}$
    else if $|\text{tail}| = 1$, then $\boldsymbol{\alpha}(q) = \boldsymbol{\alpha}(q) + C^{\text{rule}} \times_1 \boldsymbol{\alpha}(\text{tail}_0)$
    else if $|\text{tail}| = 2$, then $\boldsymbol{\alpha}(q) = \boldsymbol{\alpha}(q) + C^{\text{rule}} \times_2 \boldsymbol{\alpha}(\text{tail}_1) \times_1 \boldsymbol{\alpha}(\text{tail}_0)$
▷ *Outside Computation*
For $q \in \mathcal{H}$,
  $\boldsymbol{\beta}(q) = \mathbf{0}$
$\boldsymbol{\beta}(\text{goal}) = C^{\mathbf{S}}$
For q in reverse topological order in $\mathcal{H}$,
  For each incoming edge $e \in \mathbf{B}(q)$,
    tail = **t**(e), rule = **r**(e)
    if $|\text{tail}| = 1$, then $\boldsymbol{\beta}(\text{tail}_0) = \boldsymbol{\beta}(q) \times_0 C^{\text{rule}}$
    else if $|\text{tail}| = 2$, then,
      $\boldsymbol{\beta}(\text{tail}_0) = \boldsymbol{\beta}(q) \times_0 C^{\text{rule}} \times_2 \boldsymbol{\alpha}(\text{tail}_1)$
      $\boldsymbol{\beta}(\text{tail}_1) = \boldsymbol{\beta}(q) \times_0 C^{\text{rule}} \times_1 \boldsymbol{\alpha}(\text{tail}_0)$
▷*Edge Marginals*
Sentence probability $g = \boldsymbol{\alpha}(\text{goal}) \times \boldsymbol{\beta}(\text{goal})$
For edge $e \in \mathcal{H}$,
  head = **h**(e), tail = **t**(e), rule = **r**(e)
  if $|\text{tail}| = 0$, then $\mu(e) = (\boldsymbol{\beta}(\text{head}) \times_0 C^{\text{rule}})/g$
  else if $|\text{tail}| = 1$, then $\mu(e) = (\boldsymbol{\beta}(\text{head}) \times_0 C^{\text{rule}} \times_1 \boldsymbol{\alpha}(\text{tail}_0))/g$
  else if $|\text{tail}| = 2$, then $\mu(e) = (\boldsymbol{\beta}(\text{head}) \times_0 C^{\text{rule}} \times_2 \boldsymbol{\alpha}(\text{tail}_1) \times_1 \boldsymbol{\alpha}(\text{tail}_0))/g$

Figure 1: The tensor form of the hypergraph inside-outside algorithm, for calculation of rule marginals $\mu(e)$. A slight simplification in the marginal computation yields NT marginals for spans $\mu(\mathbf{X}, i, j)$. **B**(q) returns the incoming hyperedges for node q, and **h**(e), **t**(e), **r**(e) return the head node, tail nodes, and rule for hyperedge e.

tor, will produce an $m \times m$ matrix.[1] Note that matrix multiplication can be represented by $\times_1$ when multiplying on the right and $\times_0$ when multiplying on the left of the matrix.

The decoder computes probabilities for each rule in the parse forest of a source sentence by marginalizing over the latent states, which in practice corresponds to simple tensor-vector products, and is not dependent on the manner in which the parameters were estimated. Figure 1 presents the tensor version of the inside-outside algorithm for decoding L-SCFGs. The algorithm takes as input the parse forest of the source sentence represented as a hypergraph (Klein and Manning, 2001), which is computed using a bottom-up parser with Earley-style rules similar to the algorithm in Chiang (2007). Then, the algorithm computes inside and outside probabilities over the hypergraph using the tensor representations, and converts these probabilities to marginal rule probabilities. It is similar to the version presented in Cohen et al. (2014), but adapted to hypergraph parse forests.

The complexity of this decoding algorithm is $\mathcal{O}(n^3 m^3 |G|)$ where n is the length of the input sentence, $|G|$ is the number of production rules in the grammar *without* latent-variable annotations (i.e., 1 in this paper), and m is the number of latent states. The bulk of the computation is a series of tensor-vector products of relatively small size (each dimension is of length m), which can be computed very quickly and in parallel. The tensor computations can be significantly sped up using techniques described by Cohen and Collins (2012), so that they are linear in m and not cubic.

3 Derivation Trees for Parallel Sentences

To estimate the parameters t and π of an L-SCFG (discussed in detail in the next section), we assume the existence of a dataset composed of synchronous s-trees, which can be acquired from word alignments. Normally in phrase-based translation models, we consider all possible phrase pairs consistent with the word alignments and estimate features based on surface statistics associated with the phrase pairs or rules. The weights of these features are then learned using some kind of discriminative training algorithm (Och, 2003; Chiang, 2012). In contrast, in this work we restrict the number of possible synchronous derivations for each sentence pair to just one; thus, derivation forests do not have to be considered, making parameter estimation more tractable.

To achieve this objective, for each sentence in the training data we extract the **minimal** set of synchronous rules consistent with the word alignments, as opposed to the **composed** set of rules. Composed rules are ones that can be formed out of smaller rules in the grammar; with these rules, there are multiple synchronous trees consistent with the alignments for a given sentence pair, and thus the total number of applicable rules can be

[1] This operation is sometimes called a contraction.

combinatorially larger than if we just consider the set of rules that cannot be formed from other rules, namely the minimal rules. The rule types across all sentence pairs are combined to form a minimal grammar.[2]

To extract a set of minimal rules for each word-aligned sentence pair, we use the linear-time extraction algorithm of Zhang et al. (2008). This algorithm generalizes an approach for finding all common intervals (a pair of phrases such that no word pair in the alignment links a word inside the phrase to a word outside the phrase) between two permutations to sequences with many-to-many alignment links between the two sides, as in word alignment. By using minimal rules as a starting point instead of the traditional heuristically-extracted rules (Chiang, 2007) or arbitrary compositions of minimal rules (Galley et al., 2006), we are also able to explore the transition from minimal rules to composed ones in a principled manner by encoding contextual information through the latent states. Thus, a beneficial side effect of our refinement process is the creation of more context-specific rules without increasing the overall size of the grammar.

4 Parameter Estimation for L-SCFGs

We explore two methods for estimating the parameters C^r of the model: a likelihood-maximization approach based on EM (Dempster et al., 1977), and a spectral approach based on the method of moments (Hsu et al., 2009; Cohen et al., 2014), where we identify a subspace using a singular value decomposition (SVD) of the cross-product feature space between inside and outside trees and estimate parameters in this subspace.

Figure 2 presents a side-by-side comparison of the two algorithms, which we discuss in this section. In the spectral approach, we base our parameter estimates on low-rank representations of moments of features, while EM explicitly maximizes a likelihood criterion. The parameter estimation algorithms are relatively similar, but in lieu of sparse feature functions in the spectral case, EM uses partial counts estimated with the current set of parameters. The nature of EM allows it to be susceptible to local optima, while the spectral approach comes with guarantees on obtaining the global optimum. Lastly, computing the SVD and estimating parameters in the low-rank space is a one-shot operation, as opposed to the iterative procedure of EM.

4.1 Estimation with Spectral Moments

We generalize the parameter estimation algorithm presented in Cohen et al. (2013) to the synchronous or bilingual case. The central concept of the spectral parameter estimation algorithm is to learn an m-dimensional representation of inside and outside trees by defining these trees in terms of features, in combination with a projection step (SVD), with the hope being that the lower-dimensional space captures the syntactic and semantic regularities among rules from the sparse feature space. Every NT in an s-tree has an associated inside and outside tree; the inside tree contains the entire sub-tree at and below the NT, and the outside tree is everything else in the synchronous s-tree except the inside tree. The inside feature function $\phi \in \mathbb{R}^d$ maps the domain of inside tree fragments to a d-dimensional Euclidean space, and the outside feature function $\psi \in \mathbb{R}^{d'}$ maps the domain of outside tree fragments to a d'-dimensional space. The specific features we used are discussed in §5.2.

Let $\mathcal{O}$ be the set of all tuples of inside-outside trees in our training corpus, whose size is equivalent to the number of rule tokens M, and let $\phi(t) \in \mathbb{R}^{d\times 1}$, $\psi(o) \in \mathbb{R}^{d'\times 1}$ be the inside and outside feature functions. By computing the outer product $\otimes$ between the inside and outside feature vectors for each pair and aggregating, we obtain the empirical inside-outside feature covariance matrix:

$$\hat{\Omega} = \frac{1}{|\mathcal{O}|} \sum_{(o,t)\in\mathcal{O}} \phi(t) \left(\psi(o)\right)^\top \qquad (1)$$

If m is the desired latent space dimension, we compute an m-rank truncated SVD of the empirical covariance matrix $\hat{\Omega} = U\Sigma V^\top$, where $U \in \mathbb{R}^{d\times m}$ and $V \in \mathbb{R}^{d'\times m}$ are the matrices containing the left and right singular vectors, and $\Sigma \in \mathbb{R}^{d\times d'}$ is a diagonal matrix containing the m-largest singular values along its diagonal.

Figure 2a provides the remaining steps in the algorithm. In step 1, for each inside and outside tree, we project its high-dimensional representation to the latent space. Using the lower-dimensional representations for inside and outside trees, in step 2 for each rule type r we compute the covari-

[2] In our experiments (§5.1), a grammar extracted using the traditional Hiero heuristics was more than 80 times larger than the minimal grammar.

Inputs:
Training examples $(r^{(i)}, t^{(i,1)}, t^{(i,2)}, t^{(i,3)}, o^{(i)}, b^{(i)})$ for $i \in \{1 \dots M\}$, where $r^{(i)}$ is a context free rule; $t^{(i,1)}$, $t^{(i,2)}$, and $t^{(i,3)}$ are inside trees; $o^{(i)}$ is an outside tree; and $b^{(i)} = 1$ if the rule is at the root of tree, 0 otherwise. A function ϕ that maps inside trees t to feature-vectors $\phi(t) \in \mathbb{R}^d$. A function ψ that maps outside trees o to feature-vectors $\psi(o) \in \mathbb{R}^{d'}$.

Algorithm:

▷ *Step 0: Singular Value Decomposition*

- Compute the SVD of Eq. 1 to calculate matrices $\hat{U} \in \mathbb{R}^{(d \times m)}$ and $\hat{V} \in \mathbb{R}^{(d' \times m)}$.

▷ *Step 1: Projection*

$$Y(t) = U^\top \phi(t)$$
$$Z(o) = \Sigma^{-1} V^\top \psi(o)$$

▷ *Step 2: Calculate Correlations*

$$\hat{E}^r = \begin{cases} \frac{\sum_{o \in Q^r} Z(o)}{|Q^r|} & \text{if } r \in \mathcal{R}_0 \\ \frac{\sum_{(o,t) \in Q^r} Z(o) \otimes Y(t)}{|Q^r|} & \text{if } r \in \mathcal{R}_1 \\ \frac{\sum_{(o,t^2,t^3) \in Q^r} Z(o) \otimes Y(t^2) \otimes Y(t^3)}{|Q^r|} & \text{if } r \in \mathcal{R}_2 \end{cases}$$

Q^r is the set of outside-inside tree triples for binary rules, outside-inside tree pairs for unary rules, and outside trees for pre-terminals.

▷ *Step 3: Compute Final Parameters*

- For all $r \in \mathcal{R}$,

$$\hat{C}^r = \frac{\text{count}(r)}{M} \times \hat{E}^r$$

- For all $r^{(i)} \in \{1, \dots, M\}$ such that $b^{(i)}$ is 1,

$$\hat{C}^{\mathbf{S}} = \hat{C}^{\mathbf{S}} + \frac{Y(t^{(i,1)})}{|Q^{\mathbf{S}}|}$$

$Q^{\mathbf{S}}$ is the set of trees at the root.

(a) The spectral learning algorithm for estimating parameters of an L-SCFG.

Inputs:
Training examples $(r^{(i)}, t^{(i,1)}, t^{(i,2)}, t^{(i,3)}, o^{(i)}, b^{(i)})$ for $i \in \{1 \dots M\}$, where $r^{(i)}$ is a context free rule; $t^{(i,1)}$, $t^{(i,2)}$, and $t^{(i,3)}$ are inside trees; $o^{(i)}$ is an outside tree; $b^{(i)} = 1$ if the rule is at the root of tree, 0 otherwise; and MAX_ITERATIONS.

Algorithm:

▷ *Step 0: Parameter Initialization*

For rule $r \in \mathcal{R}$,

- if $r \in \mathcal{R}_0$: initialize $\hat{C}^r \in \mathbb{R}^{1 \times m}$
- if $r \in \mathcal{R}_1$: initialize $\hat{C}^r \mathbb{R}^{m \times m}$
- if $r \in \mathcal{R}_2$: initialize $\hat{C}^r \mathbb{R}^{m \times m \times m}$

Initialize $\hat{C}^{\mathbf{S}} \in \mathbb{R}^{m \times 1}$

$\hat{C}^r_0 = \hat{C}^r, \hat{C}^{\mathbf{S}}_0 = \hat{C}^{\mathbf{S}}$

For iteration $t = 1, \dots,$ MAX_ITERATIONS,

- Expectation Step:

  ▷ *Estimate Y and Z*

  Compute partial counts and total tree probabilities g for all t and o using Fig. 1 and parameters $\hat{C}^r_{t-1}, \hat{C}^{\mathbf{S}}_{t-1}$.

  ▷ *Calculate Correlations*

$$\hat{E}^r = \begin{cases} \sum_{o,g \in Q^r} \frac{Z(o)}{g} & \text{if } r \in \mathcal{R}_0 \\ \sum_{(o,t,g) \in Q^r} \frac{Z(o) \otimes Y(t)}{g} & \text{if } r \in \mathcal{R}_1 \\ \sum_{(o,t^2,t^3,g) \in Q^r} \frac{Z(o) \otimes Y(t^2) \otimes Y(t^3)}{g} & \text{if } r \in \mathcal{R}_2 \end{cases}$$

  ▷ *Update Parameters*

  For all $r \in \mathcal{R}$, $\hat{C}^r_t = \hat{C}^r_{t-1} \odot \hat{E}^r$

  For all $r^{(i)} \in \{1, \dots, M\}$ such that $b^{(i)}$ is 1, $\hat{C}^{\mathbf{S}}_t = \hat{C}^{\mathbf{S}}_t + (\hat{C}^{\mathbf{S}}_{t-1} \odot Y(r^{(i)}))/g$

  $Q^{\mathbf{S}}$ is the set of trees at the root.

- Maximization Step

  if $r \in \mathcal{R}_0$: $\forall h_1 : \hat{C}^r(h_1) = \frac{\hat{C}^r(h_1)}{\sum_{r'=r} \sum_{h_1} \hat{C}^{r'}(h_1)}$

  if $r \in \mathcal{R}_1$: $\forall h_1, h_2 : \hat{C}^r(h_1, h_2) = \frac{\hat{C}^r(h_1,h_2)}{\sum_{r'=r} \sum_{h_2} \hat{C}^{r'}(h_1,h_2)}$

  if $r \in \mathcal{R}_2$: $\forall h_1, h_2, h_3 : \hat{C}^r(h_1, h_2, h_3) = \frac{\hat{C}^r(h_1,h_2,h_3)}{\sum_{r'=r} \sum_{h_2,h_3} \hat{C}^{r'}(h_1,h_2,h_3)}$

  if LHS$(r) = \mathbf{S}$: $\forall h_1 : \hat{C}^r(h_1) = \frac{\hat{C}^r(h_1)}{\sum_{r'=r} \sum_{h_1} \hat{C}^{r'}(h_1)}$

(b) The EM-based algorithm for estimating parameters of an L-SCFG.

Figure 2: The two parameter estimation algorithms proposed for L-SCFGs; (**a**) method of moments; (**b**) expectation maximization. $\odot$ is the element-wise multiplication operator.

ance between the inside tree vectors and the outside tree vector using the *tensor product*, a generalized outer product to compute covariances between more than two random vectors. For binary rules, with two child inside vectors and one outside vector, the result $\hat{E}^r$ is a 3-mode tensor; for unary rules, a regular matrix, and for pre-terminal rules with no right-hand side non-terminals, a vector. The final parameter estimate is then the associated tensor/matrix/vector, scaled by the maximum likelihood estimate of the rule r, as in step 3.

The corresponding theoretical guarantees from Cohen et al. (2014) can also be generalized to the synchronous case. $\hat{\Omega}$ is an empirical esti-

mate of the true covariance matrix Ω, and if Ω has rank m, then the marginals computed using the spectrally-estimated parameters will converge to the true marginals. The sample complexity for convergence is inversely proportional to the m^{th} largest singular value.

4.2 Estimation with EM

A likelihood maximization approach can also be used to learn the parameters of an L-SCFG. Parameters are initialized by sampling each parameter value $\hat{C}^r(h_1, h_2, h_3)$ from the interval $[0, 1]$ uniformly at random.[3] We first decode the training corpus using an existing set of parameters to compute the inside and outside probability vectors associated with NTs for every rule in each s-tree, constrained to the tree structure of the training example. These probabilities can be computed using the decoding algorithm in Figure 1 (where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ correspond to the inside and outside probabilities respectively), except the parse forest consists of a single tree only. Each of these vectors represents partial counts over latent states. We can then define functions Y and Z (analogous to the spectral case) which map inside and outside tree instances to m-dimensional vectors containing these partial counts. In the spectral case, Y and Z are estimated just once, while in the case of EM they have to be re-estimated at each iteration.

The expectation step thus consists of computing the partial counts of inside and outside trees t and o, i.e., recovering the functions Y and Z, and updating parameters C^r by computing correlations, which involves summing over partial counts (across all occurrences of a rule in the corpus). Each partial count's contribution is divided by a normalization factor g, which is the total probability of the tree which t or o is part of. Note that unlike the spectral case, there is a specific normalization factor for each inside-outside tuple. Lastly, the correlations are scaled by the existing parameter estimates.

To obtain the next set of parameters, in the maximization step we normalize $\hat{C}^r$ for $r \in \mathcal{R}$ such that for every h_1, $\sum_{r'=r,h_2,h_3} \hat{C}^{r'}(h_1, h_2, h_3) = 1$ for $r \in \mathcal{R}_2$, $\sum_{r'=r,h_2} \hat{C}^{r'}(h_1, h_2) = 1$ for $r \in \mathcal{R}_1$, and $\sum_{r'=r,h_2} \hat{C}^{r'}(h_2) = 1$ for $r \in \mathcal{R}_0$. We also normalize the root rule parameters $\hat{C}^r$ where $\text{LHS}(r) = \mathbf{S}$. It is also possible to add sparse, overlapping features to an EM-based estimation procedure (Berg-Kirkpatrick et al., 2010) and leave this for future work.

[3] In our experiments, we also tried the initialization scheme described in Matsuzaki et al. (2005), but found that it provided little benefit.

5 Experiments

The goal of the experimental section is to evaluate the performance of the latent variable SCFG in comparison to a baseline without any additional NT annotations (MIN-GRAMMAR), and to compare the performance of the two parameter estimation algorithms. We also compare L-SCFGs to a HIERO baseline (Chiang, 2007). The language pair of evaluation is Chinese–English (ZH-EN).

We score translations using BLEU (Papineni et al., 2002). The latent variable model is integrated into the standard MT pipeline by computing marginal probabilities for each rule in the parse forest of a source sentence using the algorithm in Figure 1 with the parameters estimated through the algorithms in Figure 2, and is added as a feature for the rule during MERT (Och, 2003). These probabilities are conditioned on the LHS ($\mathbf{X}$), and are thus joint probabilities for a source-target RHS pair. We also write out as features the conditional probabilities $P(e|f)$ and $P(f|e)$ as estimated by our latent variable model, i.e., conditioned on the source and target RHS.

Overall, we find that both the spectral and the EM-based estimators improve upon a minimal grammar baseline with only a single category, but the spectral approach does better. In fact, it matches the performance of the standard HIERO baseline, despite learning on top of a minimal grammar.

5.1 Data and Baselines

The ZH-EN data is the BTEC parallel corpus (Paul, 2009); we combine the first and second development sets in one, and evaluate on the third development set. The development and test sets are evaluated with 16 references. Statistics for the data are shown in Table 1. We used the CDEC decoder (Dyer et al., 2010) to extract word alignments and the baseline hierarchical grammars, for MERT tuning, and decoding. We used a 4-gram language model built from the target-side of the parallel training data. The Python-based implementation of the tensor-based decoder, as well as the parameter estimation algorithms is available at `www.github.com/X`.

	ZH-EN
TRAIN (SRC)	334K
TRAIN (TGT)	366K
DEV (SRC)	7K
DEV (TGT)	7.6K
TEST (SRC)	3.8K
TEST (TGT)	3.9K

Table 1: Corpus statistics (in words). For the target DEV and TEST statistics, we take the first reference.

The baseline HIERO system uses a grammar extracted by applying the commonly used heuristics (Chiang, 2007). Each rule is decorated with two lexical and phrasal features corresponding to the forward $P(e|f)$ and backward $P(f|e)$ probabilities, along with the joint probability $P(e, f)$, the marginal probability of the source phrase $P(f)$, and whether the phrase pair or the source phrase is a singleton. Weights for the language model (and language model OOV), glue rule, and word penalty are also tuned. The MIN-GRAMMAR baseline maintains the same set of weights.

5.2 Spectral Features

We use the following set of sparse, binary features in the spectral learning process:

- **Rule Indicator.** For the inside features, we consider the rule production containing the current non-terminal on the left-hand side, as well as the rules of the children (distinguishing between left and right children for binary rules). For the outside features, we consider the parent rule production along with the rule production of the sibling (if it exists).
- **Lexical.** for both the inside and outside features, any lexical items that appear in the rule productions are recorded. Furthermore, we consider the first and last words of spans (left and right child spans for inside features, distinguishing between the two if both exist, and sibling span for outside features). Source and target words are treated separately.
- **Length.** the span length of the tree and each of its children for inside features, and the span length of the parent and sibling for outside features.

In our experiments, we used a total of 170,000 rule indicator features, 155,000 lexical features, and 80 length features.

	Setup	BLEU Dev	BLEU Test
Baselines	HIERO	46.08	55.31
	Minimal Grammar	43.38	51.78
	MLE	43.24	52.80
Spectral	$m = 1$ RI	44.18	52.62
	$m = 8$ RI	44.60	53.63
	$m = 16$ RI	46.06	55.83
	$m = 16$ RI+Lex+Sm	46.08	55.22
	$m = 16$ RI+Lex+Len	45.70	55.29
	$m = 24$ RI+Lex	43.00	51.28
	$m = 32$ RI+Lex	43.06	52.16
EM	$m = 8$ 40 iterations	40.67	49.11
	$m = 16$ 30 iterations	42.69	52.91
	$m = 32$ 35 iterations	40.85	50.34

Table 2: Results for the ZH-EN corpus, comparing across the baselines and the two parameter estimation techniques. RI, Lex, and Len correspond to the rule indicator, lexical, and length features respectively, and Sm denotes smoothing. For the EM experiments, we selected the best scoring iteration by tuning weights for parameters obtained after 25 iterations and evaluating other parameters with these weights.

5.3 Chinese–English Experiments

Table 2 presents a comprehensive evaluation of the ZH-EN experimental setup. The first section consists of the various baselines we consider. In addition to the aforementioned baselines, we evaluated a setup where the spectral parameters simply consist of the joint maximum likelihood estimates of the rules. This baseline should perform *en par* with MIN-GRAMMAR, which we see is the case on the development set. The performance on the test set is better though, primarily because we also include the reverse probability $P(f|e)$ computed from the latent variable model as an additional feature in MERT. Furthermore, in line with previous work (Galley et al., 2006) which compares minimal and composed rules, we find that minimal grammars take a hit of more than 2.5 BLEU points on the development set, compared to composed (HIERO) grammars. The $m = 1$ spectral baseline with only rule indicator features performs slightly better than the minimal grammar baseline, since it overtly takes into account inside-outside tree combination preferences in the parameters, but improvement is minimal with one latent state naturally and the performance on the test set is in line with the MLE baseline.

On top of the baselines, we looked at a number of feature combinations and latent states for the spectral and EM-estimated latent variable models. For the spectral models, we tuned MERT parameters separately for each rank on a set of parameters estimated from rule indicator features only; subse-

quent variations within a given rank, e.g., the addition of lexical or length features or smoothing, were evaluated with the same set of rank-specific weights from MERT. For EM, we ran parameter estimation for 50 iterations, and tuned the MERT parameters after 25 iterations. Similar to the spectral experiments, we fixed the MERT weight values and evaluated BLEU performance with parameters after every 5 iterations and chose the iteration with the highest score on the development set.

Firstly, we can see a clear dependence on rank, with peak performance for the spectral and EM models occurring at $m = 16$. In this instance, the spectral model roughly matches the performance of the HIERO baseline, but it only uses rules extracted from a minimal grammar, whose size is a fraction of the HIERO grammar. The gains seem to level off at this rank; additional ranks seem to add noise to the parameters. Feature-wise, additional lexical and length features add little, probably because much of this information is encapsulated in the rule indicator features. For EM, $m = 16$ outperforms the minimal grammar baseline, but is not at the level of the spectral results.

The two estimation algorithms differ significantly in their estimation time. The spectral algorithm is one or two orders magnitude faster: it completes the entire estimation process in the same time that EM takes for one or two iterations.

5.4 Discussion & Analysis

Figure 3 presents a comparison of the nonterminal span marginals for two sentences in the development set. We visualize these differences through a heat map of the CKY parse chart, where the starting word of the span is on the rows, and the span end index is on the columns. Each cell is shaded to represent the marginal of that particular non-terminal span, with higher likelihoods in blue and lower likelihoods in red.

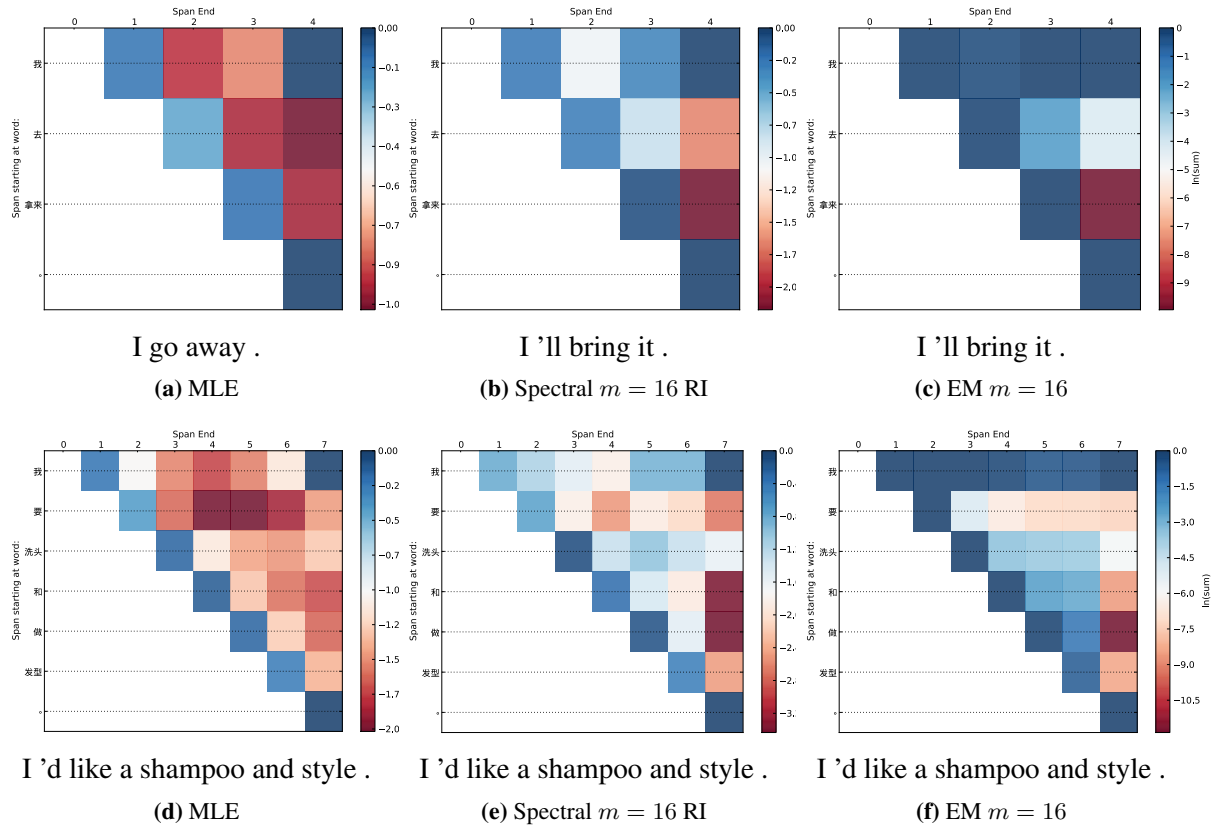
For the most part, marginals at the leaves (i.e., preterminal marginals) tend to score relatively similarly across different setups. Higher up in the chart, the latent SCFG marginals look quite different than the MLE parameters. Most noticeably, spans starting at the beginning of the sentence are much more favored. It is these rules that allow the right translation to be preferred since the MLE chooses not to place the object of the sentence in the subject's span. However, the spectral parameters seem to discriminate between these higher-level rules better than EM, which scores spans starting with the first word uniformly highly. Another interesting point is that the range of likelihoods is much larger in the EM case compared to the MLE and spectral variants. For the second sentence (row), the hypotheses produced by all systems are the same, but the heat map accentuates the previous observation.

6 Related Work

The goal of refining single-category HPBT grammars or automatically learning the NT categories in a grammar, instead of relying on noisy parser outputs, has been explored from several different angles in the MT literature. Blunsom et al. (2008) present a Bayesian model for synchronous grammar induction, and place an appropriate nonparametric prior on the parameters. However, their starting point is to estimate a synchronous grammar with multiple categories from parallel data (using the word alignments as a prior), while we aim to refine a fixed grammar with additional latent states. Furthermore, their estimation procedure is extremely expensive and is restricted to learning up to five NT categories, via a series of mean-field approximations.

Another approach is to explicitly attach a real-valued vector to each NT: Huang et al. (2010) use an external source-language parser for this purpose and score rules based on the similarity between a source sentence parse and the information contained in this vector, which explicitly requires the integration of a good-quality source-language parser. The EM-based algorithm that we propose here is similar to what they propose, except that we need to handle tensor structures. Mylonakis and Sima'an (2011) select among linguistically motivated nonterminal labels with a cross-validated version of EM. Although they consider a restricted hypothesis space, they do marginalize over different derivations therefore their inside-outside algorithm is $\mathcal{O}(n^6)$.

The idea of automatically learned grammar refinements comes from the monolingual parsing literature, where phenomena like head lexicalization can be modeled through latent variables. Matsuzaki et al. (2005) look at a likelihood-based method to split the NT categories of a grammar into a fixed number of sub-categories, while Petrov et al. (2006) learn a variable number of subcategories per NT. The latter's extension is not par-



(a) MLE **(b)** Spectral $m = 16$ RI **(c)** EM $m = 16$

(d) MLE **(e)** Spectral $m = 16$ RI **(f)** EM $m = 16$

Figure 3: A comparison of the CKY charts containing marginal probabilities of non-terminal spans $\mu(\mathbf{X}, i, j)$ for the MLE, spectral $m = 16$ with rule indicator features, and EM $m = 16$, for the two Chinese sentences. Higher likelihoods are in blue, lower likelihoods in red. The hypotheses produced by each setup are below the heat maps.

ticularly applicable to single-category grammars, but may be useful for finding the optimal number of latent states from the data.

Hsu et al. (2009) presented one of the initial efforts at spectral-based parameter estimation (using SVD) of observed moments for latent variable models, in the case of Hidden Markov models. This idea was extended to L-PCFGs (Cohen et al., 2014), and our approach can be seen as a bilingual or synchronous generalization, particularly the tensor formulation. We base our parameter estimation algorithm on the follow-up work of Cohen et al. (2013).

The question of whether we can incorporate additional contextual information in minimal rule grammars in MT via auxiliary models instead of using longer, composed rules has been investigated before as well. n-gram translation models (Mariño et al., 2006; Durrani et al., 2011) seek to model long-distance dependencies and reorderings through n-grams. Similarly, Vaswani et al. (2011) use a Markov model in the context of tree-to-string translation, where the parameters are Kneser-Ney smoothed (Kneser and Ney, 1993), while in our instance we capture this smoothing effect through low rank or latent states. Feng and Cohn (2013) also utilize a Markov model for MT, but learn the parameters through a more sophisticated estimation technique that makes use of Pitman-Yor hierarchical priors.

7 Conclusion

In this work, we presented an approach to refine synchronous grammars used in MT by inferring the latent categories for the single nonterminal in our grammar rules, and proposed two algorithms to estimate parameters for our latent variable model. By fixing the synchronous derivations of each parallel sentence in the training data, it is possible to avoid many of the computation issues associated with synchronous grammar induction. Improvements over a minimal grammar baseline and equivalent performance to a hierarchical phrase-based baseline are achieved by the spectral approach. For future work, we will seek to relax this consideration and jointly reason about nonterminal categories and derivation structures.

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 582–590, Stroudsburg, PA, USA. Association for Computational Linguistics.

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. Bayesian Synchronous Grammar Induction. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*, NIPS 2008.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, June.

David Chiang. 2012. Hope and Fear for Discriminative Training of Statistical Translation Models. *J. Mach. Learn. Res.*, pages 1159–1187.

S. B. Cohen and M. Collins. 2012. Tensor decomposition for fast parsing with latent-variable PCFGs. In *Proceedings of NIPS*.

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar. 2012. Spectral learning of latent-variable PCFGs. In *Proceedings of ACL*.

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar. 2013. Experiments with spectral learning of latent-variable PCFGs. In *Proceedings of NAACL*.

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar. 2014. Spectral learning of latent-variable PCFGs: Algorithms and sample complexity. *Journal of Machine Learning Research*.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1045–1054, Portland, Oregon, USA, June. Association for Computational Linguistics.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of ACL*.

Yang Feng and Trevor Cohn. 2013. A markov model of machine translation using non-parametric bayesian inference. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 333–342, Sofia, Bulgaria, August. Association for Computational Linguistics.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 961–968, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jonathan Graehl, Kevin Knight, and Jonathan May. 2008. Training tree transducers. *Comput. Linguist.*, 34(3):391–427, September.

Greg Hanneman and Alon Lavie. 2013. Improving syntax-augmented machine translation by coarsening the label set. In *Proc. of NAACL*.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. 2009. A Spectral Algorithm for Learning Hidden Markov Models. In *COLT*.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of AMTA*, August.

Zhongqiang Huang, Martin Čmejrek, and Bowen Zhou. 2010. Soft syntactic constraints for hierarchical phrase-based translation using latent syntactic distributions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 138–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2001. Parsing and hypergraphs. In *Proceedings of the Seventh International Workshop on Parsing Technologies (IWPT-2001), 17-19 October 2001, Beijing, China*.

Reinhard Kneser and Hermann Ney. 1993. Improved clustering techniques for class-based statistical language modeling. In *In Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational*

Linguistics on Human Language Technology - Volume 1, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Abby Levenberg, Chris Dyer, and Phil Blunsom. 2012. A Bayesian model for learning SCFGs with discontiguous rules. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 223–232, Stroudsburg, PA, USA. Association for Computational Linguistics.

Percy Liang, Slav Petrov, Michael I. Jordan, and Dan Klein. 2007. The infinite PCFG using hierarchical dirichlet processes. In *Proc. of EMNLP*.

José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José A. R. Fonollosa, and Marta R. Costa-jussà. 2006. N-gram-based machine translation. *Comput. Linguist.*, 32(4):527–549, December.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 75–82, Stroudsburg, PA, USA. Association for Computational Linguistics.

Markos Mylonakis and Khalil Sima'an. 2011. Learning Hierarchical Translation Structure with Linguistic Annotations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 642–652, Portland, Oregon, USA, June. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics*, pages 160–167, July.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michael Paul. 2009. Overview of the iwslt 2009 evaluation campaign. In *Proceedings of IWSLT 2009*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 433–440, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ashish Vaswani, Haitao Mi, Liang Huang, and David Chiang. 2011. Rule markov models for fast tree-to-string translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 856–864, Portland, Oregon, USA, June. Association for Computational Linguistics.

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

Hao Zhang, Daniel Gildea, and David Chiang. 2008. Extracting synchronous grammar rules from word-level alignments in linear time. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 1081–1088. Association for Computational Linguistics.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, StatMT '06, pages 138–141, Stroudsburg, PA, USA. Association for Computational Linguistics.