A Bayesian Model of Syntax-Directed Tree to String Grammar Induction

Trevor Cohn and **Phil Blunsom**
School of Informatics
University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
Scotland, United Kingdom
{tcohn,pblunsom}@inf.ed.ac.uk

Abstract

Tree based translation models are a compelling means of integrating linguistic information into machine translation. Syntax can inform lexical selection and reordering choices and thereby improve translation quality. Research to date has focussed primarily on decoding with such models, but less on the difficult problem of inducing the bilingual grammar from data. We propose a generative Bayesian model of tree-to-string translation which induces grammars that are both smaller and produce better translations than the previous heuristic two-stage approach which employs a separate word alignment step.

1 Introduction

Many recent advances in statistical machine translation (SMT) are a result of the incorporation of syntactic knowledge into the translation process (Marcu et al., 2006; Zollmann and Venugopal, 2006). This has been facilitated by the use of synchronous grammars to model translation as a generative process over pairs of strings in two languages. Such models are particularly attractive for translating between languages with divergent word orders, such as Chinese and English, where syntax-inspired translation rules can succinctly describe the requisite reordering operations. In contrast, standard phrase-based models (Koehn et al., 2003) assume a mostly monotone mapping between source and target, and therefore cannot adequately model these phenomena. Currently the most successful paradigm for the use of synchronous grammars in translation is that of string-to-tree transduction (Galley et al., 2004; Zollmann and Venugopal, 2006; Galley et al., 2006; Marcu et al., 2006). In this case a grammar is extracted from a parallel corpus, with strings on its source side and syntax trees on its target side, which is then used to translate novel sentences by performing inference over the space of target syntax trees licensed by the grammar.

To date grammar-based translation models have relied on heuristics to extract a grammar from a word-aligned parallel corpus. These heuristics are extensions of those developed for phrase-based models (Koehn et al., 2003), and involve symmetrising two directional word alignments followed by a projection step which uses the alignments to find a mapping between source words and nodes in the target parse trees (Galley et al., 2004). However, such approaches leave much to be desired. Word-alignments rarely factorise cleanly with parse trees (i.e., alignment points cross constituent structures), resulting in large and implausible translation rules which generalise poorly to unseen data (Fossum et al., 2008). The principal reason for employing a grammar based formalism is to induce rules which capture long-range reorderings between source and target. However if the grammar itself is extracted using word-alignments induced with models that are unable to capture such reorderings, it is unlikely that the grammar will live up to expectations.

In this work we draw on recent advances in Bayesian modelling of grammar induction (Johnson et al., 2007; Cohn et al., 2009) to propose a non-parametric model of synchronous tree substitution grammar (STSG), continuing a recent trend in SMT to seek principled probabilistic formulations for heuristic translation models (Zhang et al., 2008; DeNero et al., 2008; Blunsom et al., 2009b; Blunsom et al., 2009a). This model leverages a hierarchical Bayesian prior to induce a compact translation grammar directly from a parsed parallel corpus, unconstrained by word-alignments. We show that the induced grammars are more plausible and improve translation output.

This paper is structured as follows: In Section

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 352–361,
Singapore, 6-7 August 2009.

2 we introduce the STSG formalism and describe current heuristic approaches to grammar induction. We define a principled Bayesian model of string-to-tree translation in Section 3, and describe an inference technique using Gibbs sampling in Section 4. In Section 5 we analyse an induced grammar on a corpus of Chinese→English translation, comparing them with a heuristic grammar in terms of grammar size and translation quality.

2 Background

Current tree-to-string translation models are a form of *Synchronous Tree Substitution Grammar* (STSG; Eisner (2003)). Formally, a STSG is a 5-tuple, $G = (T, T', N, S, R)$, where T and T' are sets of *terminal symbols* in the target and source languages respectively, N is a set of *non-terminal symbols*, $S \in N$ is the distinguished *root non-terminal* and R is a set of productions (a.k.a. rules). Each production is a tuple comprising an *elementary tree* and a string, the former referring to a tree fragment of depth ≥ 1 where each internal node is labelled with a non-terminal and each leaf is labelled with either a terminal or a non-terminal. The string part of the rule describes the lexical component of the rule in the source language and includes a special variable for each *frontier non-terminal* in the elementary tree. These variables describe the reordering and form the recursion sites in the generative process of creating tree and string pairs with the grammar. For example, the rule

$$\langle (\text{NP NP}_{\boxed{1}} \text{ (PP (IN of) NP}_{\boxed{2}})), \boxed{2}\ 的\ \boxed{1} \rangle \quad (1)$$

rewrites a noun-phrase (NP) as a NP and prepositional phrase (PP) headed by 'of' in the target language. The rule generates the token '的' in the source and reverses the order of the two child noun-phrases, indicated by the numbering of the variables in the string part.

A *derivation* creates a (tree, string) pair by starting with the root non-terminal and an empty string, then choosing a rule to rewrite (substitute) the non-terminal and expand the string. This process repeats by rewriting all frontier non-terminals until there are none remaining. A *Probabilistic STSG* assigns a probability to each rule in the grammar. The probability of a derivation is the product of the probabilities of its component rules, and the probability of a (tree, string) pair is the sum of the probabilities over all its derivations.

2.1 Heuristic Grammar Induction

Grammar based SMT models almost exclusively follow the same two-stage approach to grammar induction developed for phrase-based methods (Koehn et al., 2003). In this approach they induce a finite-state grammar with phrase-pairs as rules by taking a sentence aligned parallel corpus and 1) predicting word alignments before 2) extracting transduction rules that are 'consistent' with the word aligned data. Although empirically effective, this two stage approach is less than ideal due to the disconnect between the word-based models used for alignment and the phrase-based translation model. This is problematic as the word-based model cannot recognise phrase-based phenomena. Moreover, it raises the problem of identifying and weighting the rules from the word alignment.

The same criticisms levied at the phrase-based models apply equally to the two-stage technique used for synchronous grammar induction (Galley et al., 2004; Zollmann and Venugopal, 2006; Galley et al., 2006; Marcu et al., 2006). Namely that the word alignment models typically do not use any syntax and therefore will not be able to model, e.g., consistent syntactic reordering effects, or the impact of the syntactic category on phrase translations. The identification and estimation of grammar rules from word aligned data is also non-trivial. Galley et al. (2004) describe an algorithm for inducing a string-to-tree grammar using a parallel corpus with syntax trees on target side. Their method projects the source strings onto nodes of the target tree using the word alignment, and then extracts the minimal transduction rules as well as rules composed of adjacent minimal units. The production weights are estimated either by heuristic counting (Koehn et al., 2003) or using the EM algorithm. Both estimation techniques are flawed. The heuristic method is inconsistent in the limit (Johnson, 2002) while EM is *degenerate*, placing disproportionate probability mass on the largest rules in order to describe the data with as few a rules as possible (DeNero et al., 2006). With no limit on rule size this method will learn a single rule for every training instance, and therefore will not generalise to unseen sentences. These problems can be ameliorated by imposing limits on rule size or early stopping of EM training, however neither of these techniques addresses the underlying problems.

In contrast, our model is trained in a single step, i.e., the alignment model *is* the translation model. This allows syntax to directly inform the alignments. We infer a grammar without resorting to word alignment constraints or limits on rule size. The model uses a prior to bias towards a compact grammar with small rules, thus solving the degeneracy problem.

3 Model

Our training data comprises parallel target trees and source strings and our aim is to induce a STSG that best describes this data. This is achieved by inferring a distribution over the derivations for each training instance, where the set of derivations collectively specify the grammar. In the following, we denote the source trees as $\mathbf{t}$, target strings $\mathbf{s}$, and derivations $\mathbf{r}$ which are sequences of grammar rules, r.

As described in section 2.1, previous methods for estimating a STSG have suffered from degeneracy. A principled way to correct for such degenerated behaviour is to use a *prior* over rules which biases towards small rules. This matches our intuition: we expect good translation rules to be small, with few internal nodes, frontier non-terminals and terminal strings. However, we recognise that on occasion larger rules will be necessary; we allow such rules when there is sufficient support in the data.

We model the grammar as a set of distributions, G_c, over the productions for each non-terminal symbol, c. We adopt a non-parametric Bayesian approach by treating each G_c as a random variable with a Dirichlet process (DP) prior,

$$\begin{aligned} r|c &\sim G_c \\ G_c|\alpha_c, P_0 &\sim \text{DP}(\alpha_c, P_0(\cdot|c))\,, \end{aligned}$$

where $P_0(\cdot|c)$ (the *base distribution*) is a distribution over the infinite space of trees rooted with c, and α_c (the *concentration parameter*) controls the model's tendency towards either reusing existing rules or creating novel ones as each training instance is encountered (and consequently, the tendency to infer larger or smaller grammars). We discuss the base distribution in more detail below.

Rather than representing the distribution G_c explicitly, we integrate over all possible values of G_c. This leads to the following predictive distribution for the rule r_i given the previously observed rules $\mathbf{r}^{-i} = r_1 \dots r_{i-1}$,

$$p(r_i|\mathbf{r}^{-i}, c, \alpha_c, P_0) = \frac{n_{r_i}^{-i} + \alpha_c P_0(r_i|c)}{n_c^{-i} + \alpha_c}\,, \quad (2)$$

where $n_{r_i}^{-i}$ is the number number of times r_i has been used to rewrite c in $\mathbf{r}^{-i}$, and $n_c^{-i} = \sum_{r,R(r)=c} n_r^{-i}$ is the total count of rewriting c (here $R(r)$ is the root non-terminal of r). The distribution is *exchangeable*, meaning that all permutations of the input sequence are assigned the same probability. This allows us to treat any item as being the last, which is fundamental for efficient Gibbs sampling. Henceforth we adopt the notation $\mathbf{r}^-$ and n^- to refer to the rules and counts for the whole data set excluding the current rules under consideration, irrespective of their location in the corpus.

The base distribution, P_0, assigns a prior probability to an infinite number of rules, where each rule is an (elementary tree, source string) pair denoted $r = (\mathbf{e}, \mathbf{w})$. While there are a myriad of possible distributions, we developed a very simple one. We decompose the probability into two factors,

$$P_0(\mathbf{e}, \mathbf{w}|c) = P(\mathbf{e}|c)P(\mathbf{w}|\mathbf{e})\,, \quad (3)$$

the probability of the target elementary tree and the probability of the source string, where $c = R(r)$.

The tree probability, $P(\mathbf{e}|c)$ in (3), is modelled using generative process whereby the root category c is expanded into a sequence of child non-terminals, then each of these are either expanded or left as-is. This process continues until there are no unprocessed children. The number of child nodes for each expansion is drawn from a geometric prior with parameter p_{child}, except in the case of pre-terminals where the number of children is always one. The binary expansion decisions are drawn from a Bernoulli prior with parameter p_{expand}, and non-terminals and terminals are drawn uniformly from N and T respectively. For example, the source side of rule (1) was generated as follows: 1) the NP was rewritten as two children; 2) an NP; and 4) a PP; 5) the NP child was not expanded; 6) the PP child was expanded; 7) as an IN; and 8) a NP; 9) the IN was expanded to the terminal 'of'; and 10) the final NP was not expanded. Each of these steps is a draw from the relevant distribution, and the total probability is the product of the probabilities for each step.

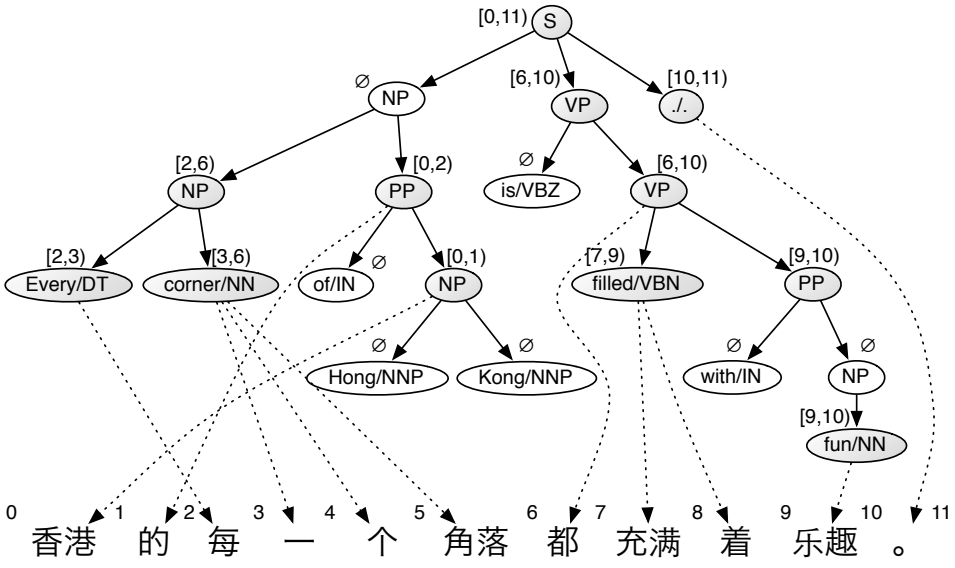

Figure 1: Example derivation. Each node is annotated with their span in the target string (aligned nodes are shaded). The dotted edges show the implied alignments. Preterminals are displayed with their child terminal in the leaf nodes.

The second factor in (3) is $P(\mathbf{w}|\mathbf{e})$, the probability of the source string (a sequence of source terminals and variables). We assume that the elementary tree is generated first, and condition the string probability on $l = F(\mathbf{e})$, its number of frontier nodes (i.e., variables). The string is then created by choosing a number of terminals from a geometric prior with parameter p_{term} then drawing each terminal from a uniform distribution over T'. Finally each of the l variables are inserted into the string one at a time using a uniform distribution over the possible placements. For the example rule in (1) the generative process corresponds to 1) deciding to create one terminal; 2) with value 的; 3) inserting the first variable after the terminal; and 4) inserting the second variable before the terminal. Again, the probability of the string is simply the product of the probabilities for each step.

Together both the factors in the base distribution penalise large trees with many nodes and long strings with many terminals and variables. P_0 decreases exponentially with rule size, thus discouraging the model from using larger rules; for this to occur the rules must significantly increase the likelihood.

4 Training

To train our model we use Gibbs sampling (Geman and Geman, 1984), a Markov chain Monte Carlo method (Gilks et al., 1996) in which variables are repeatedly sampled conditioned on the values of all other variables in the model.[1] After a period of burn-in, each sampler state (set of variable assignments) is a sample from the posterior distribution of the model. In our case, we wish to sample from the posterior over the grammar, $P(\mathbf{r}|\mathbf{t}, \mathbf{s}, \alpha)$.

To simplify matters we associate an alignment variable, a, with every internal node of the trees in the training set. This variable specifies the span of source tokens to which node is aligned. Alternatively, the node can be unaligned, which is encoded as an empty span. I.e. $a \in (J \times J) \cup \emptyset$ where J is the set of target word indices. Spans are written $[i, j)$: inclusive of i and exclusive of j. Each aligned node ($a \neq \emptyset$) forms the root of a rule as well as being a frontier non-terminal of an ancestor rule, while unaligned nodes form part of an ancestor rule.[2] The set of valid alignments are constrained by the tree in a number of ways. Child nodes can be aligned only to subspans of their ancestor nodes' alignments and no two nodes' alignments can overlap. Finally, the root node of the tree must be aligned to the full

[1] Previous approaches to bilingual grammar induction have used variational inference to optimise a bound on the data log-likelihood (Zhang et al., 2008; Blunsom et al., 2009b). Both these approaches truncated the grammar a priori in order to permit tractable inference. In contrast our Gibbs sampler can perform inference over the full space of grammars. See also Blunsom et al. (2009a) where we present a Gibbs sampler for inducing SCFGs without truncation.

[2] The Gibbs sampler is an extension of our sampler for monolingual tree-substitution grammar (Cohn et al., 2009), which used a binary substitution variable at each node to encode the segmentation of a training tree into elementary trees.

⟨(S (NP NP[1] PP[2]) VP[3] .[4]), [2] [1] [3] [4]⟩
⟨(NP DT[1] NN[2]), [1] [2]⟩
⟨(DT Every), 每⟩
⟨(NN corner), 一 个 角落⟩
⟨(PP (IN of) NP[1]), [1] 的⟩
⟨(NP (NNP Hong) (NNP Kong)), 香港⟩
⟨(VP (VBZ is) VP[1]), [1]⟩
⟨(VP VBN[1] PP[2]), 都 [1] [2]⟩
⟨(VBN filled), 充 着⟩
⟨(PP (IN with) (NP NN[1])), [1]⟩
⟨(NN fun), 趣⟩
⟨(. .), 。⟩

Table 1: Grammar rules specified by the derivation in Figure 1. Each rule is shown as a tuple comprising a target elementary tree and a source string. Boxed numbers show the alignment between string variables and frontier nonterminals.

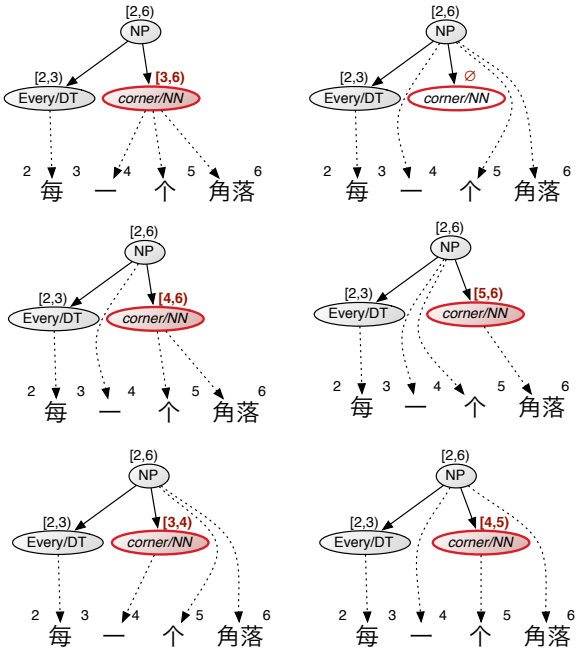

Figure 2: Possible state updates for the (NN corner) node using the EXPAND operator.

span of source words.

Collectively, the training trees and alignment variables specify the sequence of rules $\mathbf{r}$, which in turn specify the grammar. Figure 1 shows an example derivation with alignment variables. The corresponding STSG rules are shown in Table 1.

4.1 Gibbs operators

The Gibbs sampler works by sampling new values of the alignment variables, using two different Gibbs operators to make the updates. The first operator, EXPAND, takes a tree node, v, and samples a new alignment, a_v, given the alignments of all other nodes in the same tree and all other trees in the corpus, denoted $\mathbf{a}^-$. The set of valid labels is constrained by the other alignments in the tree, specifically that of the node's closest aligned ancestor, a_p, its closest aligned descendants, $\mathbf{a}_d$, and its aligned siblings, $\mathbf{a}_s$ (the aligned descendants of a). The alignment variable may be empty, $a_v = \emptyset$, while non-empty values must obey the tree constraints. Specifically the span must be a subspan of its ancestor, $a_v \subseteq a_p$, subsume its descendants, $a_v \supseteq \bigcup \mathbf{a}_d$, and not overlap its siblings, $j \notin \bigcup \mathbf{a}_s, \forall j \in a_v$. Figure 2 shows an example with the range of valid values for *corner/NN*'s alignment variable and the corresponding alignments that these encode.

Each alignment in the set of valid outcomes defines a set of grammar rules. The non-aligned outcome results in a single rule r_p rooted at ancestor node p. While the various aligned outcomes result in a pair of rules, $r_{p'}$ and r_v, rooted at p and v respectively. In the example in Figure 2, the top-right outcome has $a_v = \emptyset$ and

$$r_p = \langle (\text{NP DT}_{[1]}\ (\text{NN corner})), [1]\ 一\ 个\ 角落 \rangle .$$

The bottom-right outcome, $a_v = [4, 5)$, describes the pair of rules:

$$r_{p'} = \langle (\text{NP DT}_{[1]}\ \text{NN}_{[2]}), [1]\ 一\ [2]\ 角落 \rangle \text{ and}$$
$$r_v = \langle (\text{NN corner}), 个 \rangle .$$

The set of valid options are then scored according to the probability of their rules as follows:

$$P(r_p|\mathbf{r}^-) = \frac{n^-_{r_p} + \alpha P_0(r_p|c_p)}{n^-_{c_p} + \alpha} \quad (4)$$

$$\begin{aligned} P(r_{p'}, r_v|\mathbf{r}^-) &= P(r_{p'}|\mathbf{r}^-)P(r_v|\mathbf{r}^-, r_{p'}) \\ &= \frac{n^-_{r_{p'}} + \alpha P_0(r_{p'}|c_p)}{n^-_{c_p} + \alpha} \times \\ &\quad \frac{n^-_{r_v} + \delta(r_{p'}, r_v) + \alpha P_0(r_v|c_v)}{n^-_{c_v} + \delta(c_p, c_v) + \alpha} \end{aligned} \quad (5)$$

where c_p is the non-terminal at node p (similarly for c_v), n^- denote counts of trees (e.g., $n^-_{r_p}$) or the sum over all trees expanding a non-terminal (e.g., $n^-_{c_v}$) in the conditioning context, $\mathbf{r}^-$, and $\delta(\cdot, \cdot)$ is the Kronecker delta function, which returns 1 when its arguments are identical and 0 otherwise. For clarity, we have omitted some items from the conditioning context

in (4) and (5), namely $\mathbf{t}, \mathbf{s}$ and hyper-parameters $\alpha, p_{\text{child}}, p_{\text{expand}}, p_{\text{term}}$. The δ terms in the second factor of (5) account for the changes to n^- that would occur after observing $r_{p'}$, which forms part of the conditioning context for r_v. If the rules $r_{p'}$ and r_v are identical, then the count $n^-_{r_v}$ would increase by one, and if the rules expand the same root non-terminal, then $n^-_{c_v}$ would increase by one. Equation (4) is evaluated once for the unaligned outcome, $a_v = \emptyset$, and (5) is evaluated for each valid alignment. The probabilities are normalised and an outcome sampled.

The EXPAND operator is sufficient to move from one derivation to any other valid derivation, however it may take many steps to do so. These intermediate steps may require the sampler to pass through highly improbable regions of the state space, and consequently such moves are unlikely. The second operator, SWAP, is designed to help address this problem by increasing the mobility of the sampler, allowing it to mix more quickly. The operator considers pairs of nodes, v, w, in one tree and attempts to swap their alignment values.[3] This is illustrated in the example in Figure 3. There are two options being compared: preserving the alignments (left) or swapping them (right). This can change three rules implied by the derivation: that rooted at the nodes' common aligned ancestor, p, and those rooted at v and w. For the example, the left option implies rules

$$\begin{aligned}\{r_p &= \langle(\text{NP DT}\boxed{1}\ \text{NN}\boxed{2}), \boxed{1}\ \boxed{2}\rangle,\\ r_v &= \langle(\text{DT Every}), 每\rangle,\\ r_w &= \langle(\text{NN corner}), 一\ 个\ 角落\rangle\}\,,\end{aligned}$$

and the right option implies rules

$$\begin{aligned}\{r_p &= \langle(\text{NP DT}\boxed{1}\ \text{NN}\boxed{2}), \boxed{2}\ \boxed{1}\rangle,\\ r_v &= \langle(\text{DT Every}), 一\ 个\ 角落\rangle,\\ r_w &= \langle(\text{NN corner}), 每\rangle\}\,.\end{aligned}$$

We simply evaluate the probability of both triples of rules under our model, $P(r_p, r_v, r_w|\mathbf{r}^-) = P(r_p|\mathbf{r}^-)P(r_v|\mathbf{r}^-, r_p)$ $P(r_w|\mathbf{r}^-, r_p, r_v)$, where the additional rules in the conditioning context signify their inclusion in the counts $\mathbf{r}^-$ before applying (2) to evaluate the probability (much the same as in (5) where the δ functions encode the changes to the counts). An outcome is then sampled according to the normalised probabilities of the preserve vs. swap rules.

[3] We rarely need to consider the full quadratic space of node pairs, as the validity constraints mean that the only candidates for swapping are siblings (i.e., share the closest aligned ancestor) which do not have any aligned descendants.

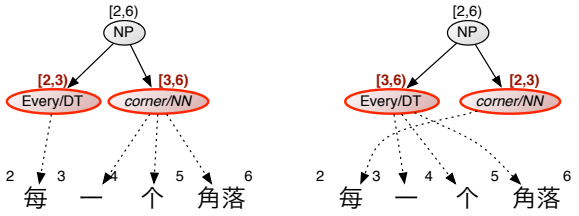

Figure 3: Possible state updates for the pair of nodes (DT every) and (NN corner) using the SWAP operator.

	English ← Chinese	
Sentences	300k	
Words or Segments	11.0M	8.6M
Avg. Sent. Length	36	28
Longest Sent.	80	80

Table 2: NIST Chinese-English corpora statistics (LDC2003E14, LDC2005E47).

The Gibbs sampler makes use of both operators. The algorithm visits each (tree, string) pair in the training set in random order and applies the EXPAND operator to every node in the tree. After the tree has been processed, the SWAP operator is applied to all candidate pairs of nodes. Visiting all sentence pairs in this way constitutes a single sample from the Gibbs sampler.

5 Experiments

We evaluate our non-parametric model of grammar induction on a subset of the NIST Chinese-English translation evaluation, representing a realistic SMT experiment with millions of words and long sentences. The Chinese-English training data consists of the FBIS corpus (LDC2003E14) and the first 100k sentence pairs from the Sinorama corpus (LDC2005E47). The Chinese text was segmented with a CRF-based Chinese segmenter optimized for MT (Chang et al., 2008), and the English text was parsed using the Stanford parser (Klein and Manning, 2003).

As a baseline we implemented the heuristic grammar extraction technique of Galley et al. (2004) (henceforth GHKM). This method finds the minimum sized translation rules which are consistent with a word-aligned sentence pair, as

described in section 2.1. The rules are then treated as events in a relative frequency estimate.[4] We used Giza++ Model 4 to obtain word alignments (Och and Ney, 2003), using the `grow-diag-final-and` heuristic to symmetrise the two directional predictions (Koehn et al., 2003).

The model was sampled for 300 iterations to 'burn-in', where in each iteration we applied both sampling operators to all nodes (or node pairs) of all training instances. We initialised the sampler using the GHKM derivation of the training data (the baseline system). The final state of the sampler was used to extract the grammar. The hyperparameters were set by hand to $\alpha = 10^6$, $p_{\text{child}} = 0.5$, $p_{\text{expand}} = 0.5$, and $p_{\text{term}} = 0.5$.[5] Overall the model took on average 2,218s per full iteration of Gibbs sampling and 1 week in total to train, using a single core of a 2.3Ghz AMD Opteron machine.

5.1 Grammar Analysis

The resulting grammar had 1.62M rules, almost identical to the GHKM grammar which had 1.63M. Despite their similarity in size the grammars were quite different, as illustrated in Figure 4, which shows histograms over various measures of rule size for the two grammars. Under each measure the sampled grammar finds many more simple rules – shallower with fewer internal nodes, fewer variables and fewer terminals – than the GHKM method. This demonstrates that the prior is effective in shifting mass away from complex rules. To show how the rules themselves differ, Table 3 lists rules in the sampled grammar that are not in the GHKM grammar. Note that many of these rules are highly plausible, describing regular tree structures and lexicalisation. These rules have not been specified to the same extent in the GHKM grammar. For example the first rule incorporates

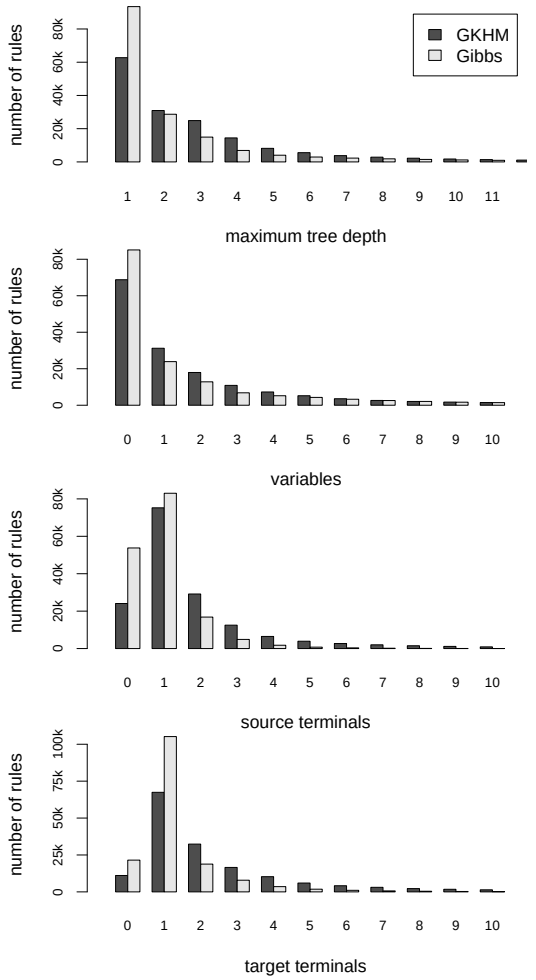

Figure 4: Histograms over rule statistics comparing the heuristic grammar (GHKM) and learnt grammar (Gibbs).

the TOP symbol, while the GHKM grammar instead relies on the rule $\langle(\text{TOP S}\boxed{1}), \boxed{1}\rangle$ to produce the same fragment. The model has learnt to distinguish between sentence-spanning and subsentential S constituents, which typically do not include final punctuation. The third and ninth (last) rule are particularly interesting. These rules encode reordering effects relating to noun phrases and subordinate prepositional phrases, in particular that Chinese prepositional modifiers precede the nouns they modify. Differences in word order such as these are quite common in Chinese-English corpora, so it is imperative that they are modelled accurately.

The rules in the GHKM grammar that do not appear in the sampled grammar are shown in Table 4. In contrast to the rules only present in the sampled grammar, these have much lower counts, i.e., are less probable. Each of these rules has been specified further by the Bayesian model.

[4] Our implementation of the GHKM algorithm attaches unaligned source words to the highest possible node in the source tree, rather than allowing all attachment points as in the original presentation (Galley et al., 2004). Allowing all attachments made no difference to translation performance, but did make the grammar considerably larger. We implemented only the minimal rule extraction, i.e., with no rule composition (Galley et al., 2006). Consequently there is no derivational ambiguity, obviating the need for expectation maximisation or similar for rule estimation.

[5] Note that although α seems large, it still encourages sparse distributions as the P_0 values are typically much smaller than its reciprocal, 10^{-6}, especially if the rule is large. $\alpha P_0 < 1$ implies a sparse Dirichlet prior.

⟨(TOP (S NP[1] VP[2] .[3])), [1] [2] [3]⟩
⟨(S (VP (TO to) VP[1])), [1]⟩
⟨(NP NP[1] (PP (IN of) NP[2])), [2] [1]⟩
⟨(PP (IN in) NP[1]), 在 [1]⟩
⟨(NP NP[1] (PP (IN of) NP[2])), [1] [2]⟩
⟨(NP (DT the) NN[1]), 的 [1]⟩
⟨(S (VP TO[1] VP[2])), [1] [2]⟩
⟨(VP (VBZ is) NP[1]), 是 [1]⟩
⟨(NP (NP (DT the) NN[1]) (PP (IN of) NP[2])), [2] [1]⟩

Table 3: Top ten rules in the sampled grammar that do not appear in the GHKM grammar. All the above rules are quite high probability, with counts between 37,118 and 7,275 from first to last.

⟨(PP (IN at) (NP DT[1] (NNS levels))), [1] 級⟩
⟨(NP NP[1] ,[2] NP[3] (, ,) CC[4] NP[5]), [1] [2] [3] [4] [5]⟩
⟨(NP NP[1] ,[2] NP[3] ,[4] NP[5] (, ,) (CC and) NP[6]), [1] [2] [3] [4] [5] , [6]⟩
⟨(S S[1] (NP (PRP They)) VP[2] .[3]), [1] [2] [3]⟩
⟨(S PP[1] ,[2] NP[3] VP[4] .[5] "[6]), [1] [2] [3] [4] [6] [5]⟩
⟨(S PP[1] ,[2] NP[3] VP[4] .[5]), [1] 中 [2] [3] [4] [5]⟩
⟨(NP (NNP Foreign) (NNP Ministry) NN[1] (NNP Zhu) (NNP Bangzao)), 外交部 [1] 朱邦造⟩
⟨(S S[1] S[2]), [1] [2]⟩
⟨(S S[1] (NP (PRP We)) VP[2] .[3]), [1] [2] [3]⟩
⟨(NP (DT the) (NNS people) POS[1]), 人民 [1]⟩

Table 4: Top ten rules in the GHKM grammar that do not appear in the sampled grammar. These are quite low probability rules: their counts range from 1,137 to 103.

For example, every instance of the first rule had the same determiner and target translation, ⟨(PP (IN at) (NP (DT all) (NNS levels))), 各 級⟩, and therefore the model specified the determiner, resulting in a single rule. The model has correctly learnt that other translations for (DT all) are not appropriate in this context (e.g., 都, 所有 or 一切). In a number of the remaining rules the commas were lexicalised, or S rules were extended to include the TOP symbol.

To further illustrate the differences between the grammars, Table 5 shows the rules which include the possessive particle, 的, and at least one variable. In both grammars there are many fully lexicalised rules which translate the token to, e.g., a determiner or a preposition. The grammars differ on the complex rules which combine lexicalisation and frontier non-terminals. The GHKM rules are all very simple depth-1 SCFG rules, containing minimal information. In contrast, the sampled rules are more lexicalised, licensing the insertion of various English tokens and tree substructure. Note particularly the second and forth rule which succinctly describe the reordering of prepositional phrases with an noun phrase.

Sampled Grammar
⟨(NP (DT the) NN[1]), 的 [1]⟩
⟨(NP (NP (DT the) NN[1]) (PP (IN of) NP[2])), [2] 的 [1]⟩
⟨(NP (DT the) NN[1]), [1] 的⟩
⟨(NP (NP (DT the) JJ[1] NN[2]) (PP (IN of) NP[3])), [3] 的 [1] [2]⟩
⟨(PP (IN of) NP[1]), [1] 的⟩
GHKM Grammar
⟨(NP JJ[1] NNS[2]), [1] 的 [2]⟩
⟨(NP JJ[1] NN[2]), [1] 的 [2]⟩
⟨(NP DT[1] JJ[2] NN[3]), [1] [2] 的 [3]⟩
⟨(NP PRP$[1] NN[2]), [1] 的 [2]⟩
⟨(NP NP[1] PP[2]), [2] 的 [1]⟩

Table 5: Top five rules which include the possessive particle 的 and at least one variable.

5.2 Translation

In order to test the translation performance of the grammars induced by our model and the GHKM method[6] we report BLEU (Papineni et al., 2002) scores on sentences of up to twenty words in length from the MT03 NIST evaluation. We built a synchronous beam search decoder to find the maximum scoring derivation, based on the CYK+ chart parsing algorithm and the cube-pruning method of Chiang (2007). Parse edges for all constituents spanning a given chart cell were cube-pruned together using a beam of width 1000, and only edges from the top ten constituents in each cell were retained. No artificial glue-rules or rule span limits were employed.[7] The parameters of the translation system were trained to maximize BLEU on the MT02 test set (Och, 2003). Decoding took roughly 10s per sentence for both grammars, using a 8-core 2.6Ghz Intel Xeon machine.

Table 6 shows the BLEU scores for the baseline using the GHKM rule induction algorithm, and our non-parametric Bayesian grammar induction method. We see a small increase in generalisation performance from our model. Our previous anal-

[6]Our decoder was unable to process unary rules (those which consume nothing in the source). Monolingual parsing with unary productions is fairly straightforward (Stolcke, 1995), however in the transductive setting these rules can licence infinite insertions in the target string. This is further complicated by the language model integration. Therefore we composed each unary rule instance with its descendant rule(s) to create a non-unary rule.

[7]Our decoder lacks certain features shown to be beneficial to synchronous grammar decoding, in particular rule binarisation (Zhang et al., 2006). As such the reported results for MT03 lag the state-of-the-art: the Moses phrase-based decoder (Koehn et al., 2007) achieves 26.8. We believe that improvements from a better decoder implementation would be orthogonal to the improvements presented here (and would allow us to relax the length restriction on the test set).

Model	BLEU score
GHKM	26.0
Our model	26.6

Table 6: Translation results on the NIST test set MT03 for sentences of length ≤ 20.

ysis (Section 5.1) of the grammars produced by the two approaches showed our method produced better lexicalised rules than those induced by the GHKM algorithm. Galley et al. (2006) noted that the GHKM algorithm often over generalised and proposed combining minimal rules to form composed rules as a solution. Although composing rules was effective at improving BLEU scores, the result was a massive expansion in the size of the grammar. By learning the appropriate level of lexicalisation we believe that our inference algorithm is having a similar effect as composing rules (Galley et al., 2006), however the resulting grammar remains compact, a significant advantage of our approach.

6 Conclusion

In this paper we have presented a method for inducing a tree-to-string grammar which removes the need for various heuristics and constraints from models of word alignment. Instead the model is capable of directly inferring a grammar in one step, using the syntactic fragments that it has learnt to better align the source and target data. Using a prior which favours sparse distributions and simpler rules, we demonstrate that the model finds a more parsimonious grammar than the heuristic technique. Moreover, this grammar results in improved translations on a standard evaluation set.

We expect that various extensions to the model would improve its performance. One avenue is to develop a more sophisticated prior over rules, e.g., one that recognises common types of rule via the shape of the tree and ordering pattern in the target. A second avenue is to develop better means of inference under the grammar, in order to ensure faster mixing and a means to escape from local optima. Finally, we wish to develop a method for decoding under the full Bayesian model, instead of the current beam search. With these extensions we expect that our model of grammar induction has the potential to greatly improve translation output.

Acknowledgements

The authors acknowledge the support of the EPSRC (grants GR/T04557/01 and EP/D074959/1). This work has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF). The ECDF is partially supported by the eDIKT initiative.

References

P. Blunsom, T. Cohn, C. Dyer, M. Osborne. 2009a. A Gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, Singapore. To appear.

P. Blunsom, T. Cohn, M. Osborne. 2009b. Bayesian synchronous grammar induction. In D. Koller, D. Schuurmans, Y. Bengio, L. Bottou, eds., *Advances in Neural Information Processing Systems 21*, 161–168. MIT Press, Cambridge, MA.

P.-C. Chang, M. Galley, C. D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, 224–232, Columbus, Ohio.

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

T. Cohn, S. Goldwater, P. Blunsom. 2009. Inducing compact but accurate tree-substitution grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 548–556, Boulder, Colorado.

J. DeNero, D. Gillick, J. Zhang, D. Klein. 2006. Why generative phrase models underperform surface heuristics. In *Proceedings on the Workshop on Statistical Machine Translation*, 31–38, New York City, NY.

J. DeNero, A. Bouchard-Côté, D. Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 314–323, Honolulu, Hawaii.

J. Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, 205–208, Sapporo, Japan.

V. Fossum, K. Knight, S. Abney. 2008. Using syntax to improve word alignment precision for syntax-based machine translation. In *Proceedings of the*

Third Workshop on Statistical Machine Translation, 44–52, Columbus, Ohio.

M. Galley, M. Hopkins, K. Knight, D. Marcu. 2004. What's in a translation rule? In *Proceedings of the 2004 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 273–280, Boston, MA.

M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, I. Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 961–968, Sydney, Australia.

S. Geman, D. Geman. 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

W. Gilks, S. Richardson, D. J. Spiegelhalter, eds. 1996. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, Suffolk.

M. Johnson, T. L. Griffiths, S. Goldwater. 2007. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In B. Schölkopf, J. Platt, T. Hoffman, eds., *Advances in Neural Information Processing Systems 19*, 641–648. MIT Press, Cambridge, MA.

M. Johnson. 2002. The DOP estimation method is biased and inconsistent. *Computational Lingusitics*, 28(1):71–76.

D. Klein, C. D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems 15*, 3–10. MIT Press.

P. Koehn, F. J. Och, D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 48–54, Edmonton, Canada.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, 177–180, Prague, Czech Republic.

D. Marcu, W. Wang, A. Echihabi, K. Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 44–52, Sydney, Australia.

F. J. Och, H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 160–167, Sapporo, Japan.

K. Papineni, S. Roukos, T. Ward, W.-J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318, Philadelphia, PA.

A. Stolcke. 1995. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2).

H. Zhang, L. Huang, D. Gildea, K. Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 256–263.

H. Zhang, C. Quirk, R. C. Moore, D. Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of ACL-08: HLT*, 97–105, Columbus, Ohio.

A. Zollmann, A. Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 138–141, New York City, NY.