Hierarchical Back-off Modeling of Hiero Grammar based on Non-parametric Bayesian Model

Hidetaka Kamigaito[1] kamigaito@lr.pi.titech.ac.jp
Taro Watanabe[2] tarow@google.com
Hiroya Takamura[1] takamura@pi.titech.ac.jp
Manabu Okumura[1] oku@pi.titech.ac.jp
Eiichiro Sumita[3] eiichiro.sumita@nict.go.jp
[1]Tokyo Institute of Technology [2]Google Japan Inc.
[3]National Institute of Information and Communication Technology

Abstract

In hierarchical phrase-based machine translation, a rule table is automatically learned by heuristically extracting synchronous rules from a parallel corpus. As a result, spuriously many rules are extracted which may be composed of various incorrect rules. The larger rule table incurs more run time for decoding and may result in lower translation quality. To resolve the problems, we propose a hierarchical back-off model for Hiero grammar, an instance of a synchronous context free grammar (SCFG), on the basis of the hierarchical Pitman-Yor process. The model can extract a compact rule and phrase table without resorting to any heuristics by hierarchically backing off to smaller phrases under SCFG. Inference is efficiently carried out using two-step synchronous parsing of Xiao et al., (2012) combined with slice sampling. In our experiments, the proposed model achieved higher or at least comparable translation quality against a previous Bayesian model on various language pairs; German/French/Spanish/Japanese-English. When compared against heuristic models, our model achieved comparable translation quality on a full size German-English language pair in Europarl v7 corpus with significantly smaller grammar size; less than 10% of that for heuristic model.

1 Introduction

Hierarchical phrase-based statistical machine translation (HPBSMT) (Chiang, 2007) is a popular alternative to phrase-based SMT (PBSMT), in which synchronous context free grammar (SCFG) is used as the basis of the machine translation model. With HPBSMT, a restricted form of an SCFG, i.e., Hiero grammar, is usually used and is especially suited for linguistically divergent language pairs, such as Japanese and English. However, a rule table, i.e., a synchronous grammar, may be composed of spuriously many rules with potential errors especially when it was automatically acquired from a parallel corpus. As a result, the increase in the rule table incurs a large amount of time for decoding and may result in lower translation quality.

Pruning a rule table either on the basis of significance test (Johnson et al., 2007) or entropy (Ling et al., 2012; Zens et al., 2012) used in PBSMT can be easily applied for HPBSMT. However, these methods still rely on a heuristically determined threshold parameter. Bayesian SCFG methods (Blunsom et al., 2009) solve the spurious rule extraction problem by directly inducing a compact rule table from a parallel corpus on the basis of a non-parametric Bayesian model without any heuristics. Training for Bayesian SCFG models infers a derivation tree for each training instance, which demands the time complexity of $O(|\mathbf{f}|^3|\mathbf{e}|^3)$ when we use dynamic programming SCFG biparsing (Wu, 1997). Gibbs sampling without biparsing (Levenberg et al., 2012) can avoid this problem, though the induced derivation trees may strongly depend on initial derivation trees. Even though we may learn a statistically sound model on the basis of non-parametric Bayesian methods, current approaches for an SCFG still rely on exhaustive heuristic rule extraction from the word-alignment decided by derivation trees since the learned models cannot handle rules and phrases of various granularities.

We propose a model on the basis of the previous work on the non-parametric Inversion Transduction Grammar (ITG) model (Neubig et al., 2011) wherein phrases of various granularities are

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1229–1239,
Lisbon, Portugal, 17-21 September 2015.

learned in a hierarchical back-off process. We extend it by incorporating arbitrary Hiero rules when backing off to smaller spans. For efficient inference, we use a fast two-step bi-parsing approach (Xiao et al., 2012) which basically runs in a time complexity of $O(|\mathbf{f}|^3)$. Slice sampling for an SCFG (Blunsom and Cohn, 2010) is used for efficiently sampling a derivation tree from a reduced space of possible derivations.

Our model achieved higher or at least comparable BLEU scores against the previous Bayesian SCFG model on language pairs; German/French/Spanish-English in the News-Commentary corpus, and Japanese-English in the NTCIR10 corpus. When compared against heuristically extracted model through the GIZA++ pipeline, our model achieved comparable score on a full size Germany-English language pair in Europarl v7 corpus with significantly less grammar size.

2 Related Work

Various criteria have been proposed to prune a phrase table without decreasing translation quality, e.g., Fisher's exact test (Johnson et al., 2007) or relative entropy (Ling et al., 2012; Zens et al., 2012). Although those methods are easily applied for pruning a rule table, they heavily rely on the heuristically determined threshold parameter to trade off the translation quality and decoding speed of an MT system.

Previously, EM-algorithm based generative models were exploited for generating compact phrase and rule tables. Joint phrase alignment model (Marcu and Wong, 2002) can directly express many-to-many word aligments without heuristic phrase extraction. DeNero et al. (2006) proposed IBM Model 3 based many-to-many alignment model. Rule arithmetic method (Cmejrek and Zhou, 2010) can generate SCFG rules by combining other rule pairs through an inside-outside algorithm. However, those previous attempts were restricted in that the rules and phrases were induced by heuristic combination.

Bayesian SCFG models can induce a compact model by incorporating sophisticated non-parametric Bayesian models for an SCFG, such as a dirichlet process (DeNero et al., 2008; Blunsom et al., 2009; Chung et al., 2014) or Pitman-Yor process (Levenberg et al., 2012; Peng and Gildea, 2014). A model is learned by sampling derivation trees in a parallel corpus and by accumulating the rules in the sampled trees into the model. Due to the $O(|\mathbf{f}|^3|\mathbf{e}|^3)$ time complexity for bi-parsing a bilingual sentence, previous studies relied on bi-parsing at the initialization step, and conducted Gibbs sampling by local operators (Blunsom et al., 2009; Levenberg et al., 2012) or sampling on fixed word alignments (Chung et al., 2014; Peng and Gildea, 2014). As a result, the inference can easily result in local optimum, wherein induced derivation trees may strongly depend on the initial trees.

Xiao et al. (2012) proposed a two-step approach for bi-parsing a bilingual sentence in $O(|\mathbf{f}|^3)$ in the context of inducing SCFG rules discriminatively; however, their approach violates the detailed balance due to its heuristic k-best pruning. Blunsom and Cohn (2010) proposed a slice sampling for an SCFG, in the same manner as that for Infinite Hiden Markov Model (iHMM) (Van Gael et al., 2008), which can efficiently prune a space of possible derivations on the basis of dynamic programming. Although slice sampling can prune spans without violating the detailed balance, its time complexity of $O(|\mathbf{f}|^3|\mathbf{e}|^3)$ is still impractical for a large-scale experiment. We efficiently carried out large-scale experiments on the basis of the two-step bi-parsing of Xiao et al. combined with slice sampling of Blunsom and Cohn.

After learning a Bayesian model, it is not directly used in a decoder since it is composed of only minimum rules without considering phrases of various granularities. As a consequence, it is a standard practice to obtain word alignment from derivation trees and to extract SCFG rules heuristically from the word-aligned data (Cohn and Haffari, 2013). The work by Neubig et al. (2011) was the first attempt to directly use the learned model on the basis of a Bayesian ITG in which phrases of many granularities were encoded in the model by employing a hierarchical back-off procedure. Our work is strongly motivated by their work, but greatly differs in that our model can incorporate many arbitrary Hiero rules, not limited to ITG-style binary branching rules.

3 Model

We use Hiero grammar (Chiang, 2007), an instance of an SCFG, which is defined as a context-free grammar for two languages. Let Σ denote a set of terminal symbols in the source language, Δ a set of terminal symbols in the target language,

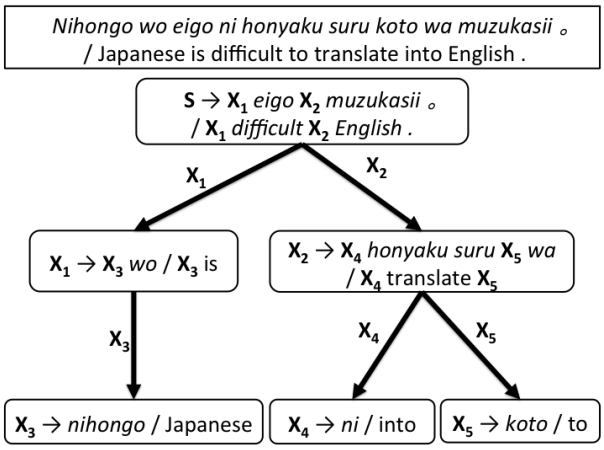

Figure 1: Derivation tree generated from Bayesian SCFG model

V a set of non-terminal symbols, S a start symbol and R a set of rewrite rules. An SCFG is denoted as a tuple of $\langle \Sigma, \Delta, V, S, R \rangle$. Each rewrite rule in R is represented as $X \rightarrow \langle \alpha / \beta \rangle$ in which α is a string of non-terminals and source side terminals $(V \cup \Sigma)^*$ and β is a string of non-terminals and target side terminals $(V \cup \Delta)^*$. An example derivation in an SCFG for the sentence pair "*nihongo wo eigo ni honyaku suru koto wa muzukasii* 。 / Japanese is difficult to translate into English ." is represented as follows:
$S \rightarrow X_1$ *eigo* X_2 *muzukasii* 。 / X_1 difficult X_2 English .
$X_1 \rightarrow X_3$ *wo* / X_3 is
$X_2 \rightarrow X_4$ *honyaku suru* X_5 *wa* / X_4 translate X_5
$X_3 \rightarrow$ *nihongo* / Japanese
$X_4 \rightarrow$ *ni* / into
$X_5 \rightarrow$ *koto* / to .

A Hiero grammar has additional constraints over a general SCFG; the number of terminal symbols in each rule for both source and target sides is limited to 5. Each rule may contain at most two non-terminal symbols; adjacent non-terminal symbols in the source side are prohibited. For details, refer to (Chiang, 2007).

3.1 Bayesian SCFG Models

Previous Bayesian SCFG Models, for instance a model proposed by Levenberg et al. (2012), are based on the Pitman-Yor process (Pitman and Yor, 1997) and learn SCFG rules by sampling a derivation tree for each bilingual sentence. Figure 1 shows an example derivation tree for our running example sentence pair under the model. The generative process is represented as follows:

$$\begin{aligned} G_X &\sim P_{rule}(d_r, \theta_r, G_{r_0}), \\ X \rightarrow \langle \alpha / \beta \rangle &\sim G_X, \end{aligned} \tag{1}$$

where G_X is a derivation tree and $P_{rule}(d_r, \theta_r, G_{r_0})$ is a Pitman-Yor process (Pitman and Yor, 1997), which is a generalization of a Dirichlet process parametrized by a discount parameter d_r, a strength parameter θ_r and a base measure G_{r_0}. The output probability of a Pitman-Yor process obeys the power-law distribution with the discount parameter, which is very common in standard NLP tasks.

The probability that a rule r_k is drawn from a model $P_{rule}(d_r, \theta_r, G_{r_0})$ is determined by a Chinese restaurant process which is decomposed into two probability distributions. If r_k already exists in a table, we draw r_k with probability

$$\frac{c_k - d_r \cdot |\varphi_{r_k}|}{\theta_r + n_r}, \tag{2}$$

where c_k is the number of customers of r_k, n_r is the number of all customers and φ_{r_k} is a number of r_k's tables. On the other hand, if r_k is a new rule, we draw r_k with probability

$$\frac{\theta_r + d_r \cdot |\varphi_r|}{\theta_r + n_r} \cdot G_{r_0}, \tag{3}$$

where $|\varphi_r|$ is the number of tables in the model.

3.2 Hierarchical Back-off Model

In the previous models, the generative process is represented as a rewrite process starting from the symbol S, which can incorporate only minimal rules. Following Neubig et al. (2011), our model reverses the process by recursively backing off to smaller phrase pairs as shown in Figure 2. First, our model attempts to generate a phrase pair, i.e., a sentence pair, as a derivation tree. If the model successfully generates the phrase pair, we will finish the generation process. Otherwise, a Hiero rule is generated to fallback to smaller spans represented in each non-terminal symbol X in the rule. Then, each phrase pair corresponding to each smaller span is recursively generated through our model. In Figure 2, a phrase pair with "nil" indicates those not in our model; therefore the phrase pair is forced to back-off either by generating a new phrase pair from a base measure (base) or by falling back to smaller phrases using a Hiero rule (back-off). The recursive procedure is done until

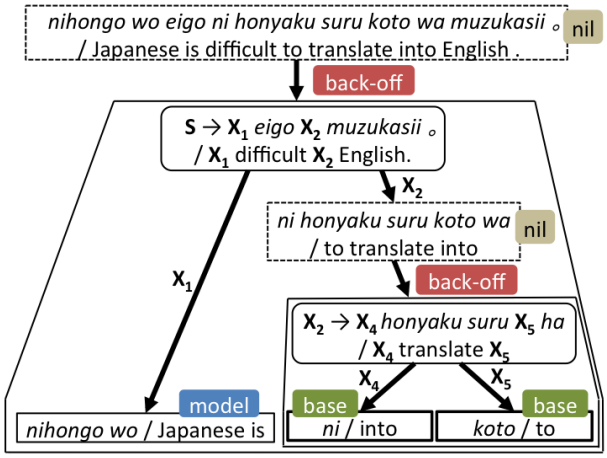

Figure 2: Derivation tree generated from the hierarchical back-off model

we reach phrase pairs which are generated without any back-offs. Let a discount parameter be d_p, a strength parameter be θ_p, and a base measure be G_{p_0}. More formally, the generative process is represented as follows:

$$\begin{aligned} G_X &\sim P_{rule}(d_r, \theta_r, G_{phrase}), \\ G_{phrase} &\sim P_{phrase}(d_p, \theta_p, G_X), \\ X \rightarrow \langle \mathbf{s}/\mathbf{t} \rangle &\sim G_{phrase}, \\ X \rightarrow \langle \alpha/\beta \rangle &\sim G_X, \end{aligned} \quad (4)$$

where $\mathbf{s}$ is source side terminals and $\mathbf{t}$ is target side terminals in phrase pair $\langle \mathbf{s}/\mathbf{t} \rangle$. P_{phrase} is composed of three states, i.e., model, back-off, and base, and follows a hierarchical Pitman-Yor process (Teh, 2006).

model: We draw a phrase pair $\langle \mathbf{s}/\mathbf{t} \rangle$ with the probability similar to Equation (2):

$$\frac{c_k - d_p \cdot |\varphi_{p_k}|}{\theta_p + n_p}, \quad (5)$$

where c_k is the numbers of customers of a phrase pair p_k and n_p is the number of all customers Note that this state is reachable when the phrase pair $\langle \mathbf{s}/\mathbf{t} \rangle$ exists in the model in the same manner as Equation (2).

back-off: We will back off to smaller phrases using a rule generated by P_{rule} as follows:

$$\begin{aligned} &\frac{\theta_p + d_p \cdot |\varphi_p|}{\theta_p + n_p} \cdot \frac{c_{back} + \gamma_b \cdot G_b}{c_{back} + c_{base} + \gamma_b} \\ &\cdot P_{rule}(d_r, \theta_r, G_{phrase}) \\ &\cdot \prod_{X \in \langle \alpha/\beta \rangle} P_{phrase}(d_p, \theta_p, G_X), \end{aligned} \quad (6)$$

where c_{back} and c_{base} are the number of customers sampled from the back-off and base phrases, respectively, with a base measure G_b and hyper-parameter γ_b. We use a uniform distribution for $G_b = 0.5$ since we consider only two states, back-off and base. Unlike the model state, P_{phrase} may reach this state even when a phrase pair is not in the model. The phrase pair is backed-off to smaller phrase pairs using P_{phrase} through the non-terminals in the generated rule $X \in \langle \alpha/\beta \rangle$.

base: As an alternative to the back-off state, we may reach the base state which follows the probability distribution on the basis of the base measure G_{p_0},

$$\frac{\theta_p + d_p \cdot |\varphi_p|}{\theta_p + n_p} \cdot \frac{c_{base} + \gamma_b \cdot G_b}{c_{back} + c_{base} + \gamma_b} \cdot G_{p_0}. \quad (7)$$

In summary, $P_{phrase}(d_p, \theta_p, G_X)$ is defined as a joint probability of Equations (5) through (7).

3.3 Base Measure

Similar to Levenberg et al. (2012), the base measure for rule probability G_{r_0} is composed of four generative processes. First, a number of symbols in a source side of a rule $|\alpha|$ is generated from a Poisson distribution, i.e., $|\alpha| \sim Poisson(0.1)$. Let $t(x)$ denote a function that returns terminals from a string x. The number of target side terminal symbols $|t(\beta)|$ is also generated from a Poisson distribution and represented as $|t(\beta)| \sim Poisson(\alpha + \lambda_0)$[1]. The type of symbol α_i in the source side, $type_i$, either terminal or non-terminal symbol, is determined by $type_i \sim Bernoulli(\phi^{|\alpha|})$ where ϕ is a hyper-parameter taking $0 < \phi < 1$. $\phi^{|\alpha|}$ is based on an intuition that shorter rules should be relatively more likely to contain terminal symbols than longer rules. Source and target terminal symbol pair $\langle t(\alpha), t(\beta) \rangle$ are generated from the geometric means of two directional IBM Model 1 word alignment probabilities and monolingual unigram probabilities for two languages, and represented as:

$$\begin{aligned} \langle t(\alpha), t(\beta) \rangle \sim (&P_{uni}(t(\alpha)) P_{\overrightarrow{M1}}(t(\alpha), t(\beta)) \cdot \\ &P_{uni}(t(\beta)) P_{\overleftarrow{M1}}(t(\alpha), t(\beta)))^{\frac{1}{2}}. \end{aligned} \quad (8)$$

When the $t(\alpha)$ or $t(\beta)$ is empty, we use the constant 0.01 instead of the Model1 probabilities.

[1]Note that λ_0 is a small constant for the input distribution greater than zero.

The base measure for phrases G_{p_0} is composed of three generative processes, in a similar manner as Levenberg et al. (2012), the number of terminal symbols in a phrase pair in the source side, $|\mathbf{s}|$, is generated from a Poisson distribution $|\mathbf{s}| \sim Poisson(0.1)$. The length for the target side $|\mathbf{t}|$ is generated in the same manner as the source side of the phrase pair. The alignments between $\mathbf{s}$ and $\mathbf{t}$ are also generated in the same manner as those for the base measure in a rule.

4 Inference

In inference, we use a sentence-wise block sampling of Blunsom and Cohn (2010), which has a better convergence property when compared with a step-wise Gibbs sampling. We repeat the following steps given a sentence pair.

1. Decrement customers of the rules and phrase pairs used in the current derivation for the sentence pair.
2. Bi-parse the sentence pair in a bottom up manner.
3. Sample a new derivation tree in a top-down manner.
4. Increment customers of the rules and phrase pairs in the sampled derivation tree.

The most time-consuming step during the inference procedure is bi-parsing of a sentence pair which essentially takes $O(|\mathbf{f}|^3|\mathbf{e}|^3)$ time using a bottom up dynamic programming algorithm (Wu, 1997). When a span is very large, it can easily suffer combinatorial explosion. To avoid this problem, we use a two-step slice sampling by performing the two-step bi-parsing of Xiao et al. (2012) and by pruning possible derivation space (Blunsom and Cohn, 2010) in each step (Algorithm 1). From lines 1 to 7, a set of word alignment is enumerated and put into $cube_a$. In addition to the arbitrary word alignment of $source_i$ to $target_j$, null word alignment is also merged into $cube_a$ (line 5). Note that word alignment considered in the algorithm is restricted to one-to-many. The set of word alignments in $cube_a$ is pruned and added to the $chart_a$ by SliceSampling. From lines 8 to 15, all possible phrases and rules for each span constrained by the pruned word alignment are enumerated and temporally stored into $cube$. The phrases and rules in $cube$ are pruned by SliceSampling and the remainders are added to $chart$. The

Algorithm 1 Two-step slice sampling

```
1:  for i ← 1, ··· , |source| do
2:    for j ← 1, ··· , |target| do
3:      cube_a ← {souce_i, target_j}
4:    end for
5:    cube_a ← {souce_i, null}
6:    chart_a ← SliceSampling(cube_a)
7:    clear cube_a
8:  end for
9:  for h ← 1, ··· , |source| do
10:   for all the i, j s.t j − i = h do
11:     for inferable rule, phrase from the sub-
        spans of [i, j] of all charts do
12:       cube ← rule, phrase
13:     end for
14:     chart ← SliceSampling(cube)
15:     clear cube
16:   end for
17: end for
```

time complexity for the word alignment enumeration from lines 1 to 7 is $O(|f||e|)$ and that for the phrase and rule enumeration from lines 8 to 15 is $O(|f|^3)$.

The key difference to the slice sampling of Blunsom and Cohn (2010) lies in lines 6 and 3 of Algorithm 1. Let $\mathbf{d}$ denote a set of derivation trees d and $\mathbf{u}$ be a set of slice variables u. In slice sampling, we prune the rules $\mathbf{r_{sp}}$ in each source span sp based on a slice variable u_{sp} corresponding to that sp. After pruning, we sample trees from the pruned space of $\mathbf{r}$. The above process is formally represented as:

$$\begin{aligned} \mathbf{u} &\sim P(\mathbf{u}|\mathbf{d}), \\ \mathbf{d} &\sim P(\mathbf{d}|\mathbf{u}), \end{aligned} \tag{9}$$

where $P(\mathbf{d}|\mathbf{u})$ is computed through sampling in a top-down manner after parsing in a bottom-up manner with Algorithm 1, and is equal to $\prod_d P(d|\mathbf{u})$. The probability $P(\mathbf{u}|\mathbf{d})$ is equal to $\prod_{sp} P(u_{sp}|\mathbf{d})$. Let r^*_{sp} denote a currently adopted rule in the span sp and $P(u_{sp}|d)$ be defined using a pruning score $Score(r^*_{sp})$ as follows:

$$Score(r_{sp_i}) = Inside(r_{sp_i}) \cdot Future(r_{sp_i}), \tag{10}$$

where $Inside(r_{sp})$ and $Future(r_{sp})$ are inside and outside probabilities for sp, respectively. Let $\mathbf{s}_{r_{sp}}$ denote a set of source side words in r_{sp}, $\mathbf{t}_{r_{sp}}$ a set of target side words in r_{sp}, $\overline{\mathbf{s}_{sp}}$ a set of words in a source sentence without $\mathbf{s}_{r_{sp}}$ and $\overline{\mathbf{t}_{sp}}$, a set of

words in a target sentence without $\mathbf{t}_{r_{sp}}$. By using IBM Model 1 probabilities in two directions, $Inside(r_{sp})$ is calculated by

$$(P_{\overrightarrow{M1}}(\mathbf{s}_{sp}, \mathbf{t}_{sp}) \cdot P_{\overleftarrow{M1}}(\mathbf{s}_{sp}, \mathbf{t}_{sp}))^{\frac{1}{2}}. \quad (11)$$

We use IBM Model1 outside probability for future score $Future(r_{sp})$. Similarly, the future score $Future(r_{sp})$ is computed using the two directional models:

$$(P_{\overrightarrow{M1}}(\overline{\mathbf{s}_{sp}}, \overline{\mathbf{t}_{sp}}) \cdot P_{\overleftarrow{M1}}(\overline{\mathbf{s}_{sp}}.\overline{\mathbf{t}_{sp}}))^{\frac{1}{2}}. \quad (12)$$

When sp is used in the current derivation $\mathbf{d}$, slice variable u_{sp} is sampled from a uniform distribution[2]:

$$P(u_{sp}|d) = \frac{\mathbb{I}(u_{sp} < Score(r^*_{sp}))}{Score(r^*_{sp})}, \quad (13)$$

otherwise, u_{sp} is sampled from a beta distribution if sp is not in the current derivation $\mathbf{d}$:

$$P(u_{sp}|d) = Beta(u_{sp}; a, 1.0), \quad (14)$$

where $a < 1$ is a parameter for the beta distribution. If the $Score(r_{sp_i})$ is less than u_{sp}, we prune the r_{sp_i} from $cube$. Similar to Blunsom and Cohn (2010), if the span sp is not in the current derivation, the rules with low probability are pruned according to Equation (14). Let r^d_{sp} denotes a rule in d with span sp, $P(d|\mathbf{u})$ is calculated by:

$$\prod_{sp \in d} \frac{P(r^d_{sp})}{\sum_{r_j \in \mathbf{r_{sp}}} P(r_j)\mathbb{I}(u_{sp} < Score(r_j))}. \quad (15)$$

In our experiments discussed in Section 6, slice sampling parameter a was set to 0.02 when incorporating the future score of Equation (12). In contrast, we used $a = 0.1$ when performing slice sampling without the future score. We empirically found that setting a lower value for a led to slower progress in learning due to a combinatorial explosion when inferencing a derivation for each sentence pair.

In the beginning of training, we do not have any derivation trees for given training data, although the derivation trees are required for estimating parameters for Bayesian models. We use the two-step parsing for generating initial derivation trees from only base measures. The k-best pruning is conducted against the score denoted by the equation 10 , which is very similar to Xiao et al. (2012).[3]

For faster bi-parsing, we run sampling in parallel in the same way as Zhao and Huang (2013), in which bi-parsing is performed in parallel among the bilingual sentences in a mini-batch. The updates to the model are synchronized by incrementing and decrementing customers for the bilingual sentences in the mini-batch. Note that the bi-parsing for each mini-batch is conducted on the fixed model parameters after the synchronised parameter updates.

In addition to the model parameters, hyperparameters are re-sampled after each training iteration following the discount and strength hyperparameter resampling in a hierarchical Pitman-Yor process (Teh, 2006). In particular, we resample $\langle d_p, \theta_p \rangle$, the pair of discount and strength parameters for phrases from a distribution:

$$\frac{[\theta_p]^{|\varphi_p|}_{d_p}}{[\theta_p]^{n_p}_1} \prod_{\langle \mathbf{s},\mathbf{t} \rangle} \prod_{k=1}^{|\varphi_p|} [1 - d_p]^{(c_{\langle \mathbf{s},\mathbf{t} \rangle} - 1)}_1 \quad (16)$$

where $[\,]$ denotes a generalized Pochhammer symbol, and $c_{\langle \mathbf{s},\mathbf{t} \rangle}$ the number of customers of phrase pair $\langle \mathbf{s}, \mathbf{t} \rangle$. We resample the pair $\langle d_r, \theta_r \rangle$ in the same way as $\langle d_p, \theta_p \rangle$. The hyperparameter γ_b is resampled from distribution:

$$\frac{(c_{back} + \gamma_b \cdot G_b)(c_{base} + \gamma_b \cdot G_b)}{(c_{back} + c_{base} + \gamma_b)^2}, \quad (17)$$

where ϕ, used in the generative process for either terminal or non-terminal symbol $type_i \sim Bernoulli(\phi^\alpha)$, is resampled from a distribution:

$$\prod_{\langle \alpha/\beta \rangle \in Base} Bernoulli(\phi^{|\alpha|})^{c_{\langle \alpha/\beta \rangle}}, \quad (18)$$

where $c_{\langle \alpha/\beta \rangle}$ denotes the number of customers of rule $\langle \alpha/\beta \rangle$, and $Base$ denotes a set of rules generated from the base measure. All the hyperparameters are inferred by slice sampling (Neal, 2003).

5 Extraction of Translation Model

In the previous work on Bayesian approaches (Blunsom and Cohn, 2010; Levenberg et al., 2012), it is a standard practice to heuristically extract rules and phrase pairs from the word alignment derived from the derivation trees sampled

[2] $\mathbb{I}(\cdot)$ is a function returns 1 if the condition is satisfied and 0 otherwise

[3] Note that we use $k = 30$ for k-best pruning.

from the Bayesian models. Instead of the heuristic method, we directly extract rules and phrase pairs from the learned models which are represented as Chinese restaurant tables. To limit grammar size, we include only phrase pairs that are selected at least once in the sample. During this extraction process, we limit the source or target terminal symbol size of phrase pairs to 5.

For each extracted rule or phase pair, we compute a set of feature scores used for a HPBSMT decoder; a weighted combination of multiple features is necessary in SMT since the model learned from training data may not fit well to translate an unseen test data (Och, 2003). We use the following six features; the joint model probability P_{model} is calculated by Equation (2) for rules and by Equation (5) for phrase pairs. The joint posterior probability $P_{posterior}(f, e)$ is estimated from the posterior probabilities for every rule and phrase pair in derivation trees through relative count estimation, motivated by Neubig et al. (2011) [4]. The joint posterior probability is considered as an approximation for those back-off scores. The conditional model probabilities in two directions, $P_{model}(f|e)$ and $P_{model}(e|f)$, are estimated by marginalizing the joint probability $P_{model}(f, e)$:

$$P_{model}(f|e) = \frac{P_{model}(f, e)}{\sum_{f'} P_{model}(f', e)}. \qquad (19)$$

The inverse direction $P_{model}(e|f)$ is estimated, similarly. The lexical probabilities in two directions, $P_{lex}(f|e)$ and $P_{lex}(e|f)$, are scored by IBM Model probabilities between the source and target terminal symbols in rules and phrase pairs. In addition to the above features, we use Word penalty for each rule and phrase pair used in the cdec decoder (Dyer et al., 2010).

As indicated in previous studies (Koehn et al., 2003; DeNero et al., 2006), the translation quality of generative models is lower than that of models with heuristically extracted rules and phrase pairs. DeNero et al. (2006) reported that considering multiple phrase boundaries is important for improving translation quality. The generative models, in particular Bayesian models, are strict in determining phrase boundaries since their models are usually estimated from sampled derivations. As a result, translation quality is poorer when compared with a model estimated using a heuristic method. The Hiero grammar severely suffers from the phrase granularity problem and can overfit to the training data due to the flexibility of the rules.

To alleviate this problem, Neubig et al. (2011) combined the derivation trees across training iterations by averaging the features for each rule and phrase pair. During the sampling process, each training iteration draws a different derivation tree for each sentence pair, and the combination of those different derivation trees can provide multiple possible phrase boundaries to the model. Inspired by the averaging over the models from different iterations, we combine them as a part of a sampling process; we treat the derivation trees acquired from different iterations as additional training data, and increment the corresponding customers into our model. Hyperparameters are resampled after the merging process. The new features are directly computed from the merged model.

6 Experiments

6.1 Comparison with Previous Bayesian Model

First, we compared the previous Bayesian model (Gen) with our hierarchical back-off model (Back). We used the first 100K sentence pairs of the WMT10 News-Commentary corpus for German/Spanish/French-to-English pairs (Callison-Burch et al., 2010) and NTCIR10 corpus for Japanese-English (Goto et al., 2013) for the translation model. All sentences are lowercased and filtered to preserve at most 40 words on both source and target sides. We sampled 20 iterations for Gen and Back and combined the last 10 iterations for extracting the translation model.[5] The batch size was set to 64. The language models were estimated from the all-English side of the WMT News-Commentary and europarl-v7. In NTCIR10, we simply used the all-English side of the training data. All the 5-gram language models were estimated using SRILM (Stolcke and others, 2002) with interpolated Kneser-Ney smoothing. The details of the corpus are presented in Table 2. For detailed analysis, we also evaluate Hiero grammars extracted from GIZA++ (Och and Ney, 2003) grow-diag-final bidirectional alignments using Moses (Koehn et al., 2007) with Hiero options.

[4] Note that the correct way to decode from our model is to score every phrase pair created during decoding with back-off states, which is computationally intractable

[5] Gen and Back took 1 day, Back+future took 1.5 days for inference.

		News-Commentary						NTCIR10	
		de-en		es-en		fr-en		ja-en	
Model	Sample	BLEU	SIZE	BLEU	SIZE	BLEU	SIZE	BLEU	SIZE
∗GIZA++	-	16.66	7.07M	23.16	6.07M	20.79	6.25M	26.08	3.45M
Gen	1	15.36	397.63k	21.10	295.69k	19.45	311.76k	25.73	262.45k
	10	15.39	529.46k	20.83	384.55k	19.24	419.33k	25.79	344.67k
Back	1	15.30	410.92k	*21.43*	314.95k	*19.74*	362.22k	25.69	294.90k
	10	15.42	563.80k	*21.53*	420.15k	19.51	497.51k	25.63	388.87k
Back + future	1	**15.49**	384.69k	**21.63**	296.30k	**19.97**	340.70k	**25.82**	268.38k
	10	**15.55**	579.12k	**21.74**	429.33k	**19.97**	513.41k	25.41	390.23k

Table 1: Results of translation evaluation in 100k corpus

	de-en	es-en	fr-en	ja-en
TM(en)	1.85M	1.67M	1.54M	1.80M
TM(other)	1.86M	1.86M	1.83M	2.03M
LM(en)	55.6M	55.6M	55.6M	27.8M
Dev(en)	65.5k	65.5k	65.5k	67.3k
Dev(other)	62.7k	68.1k	72.5k	73.0k
Test(en)	61.9k	61.9k	61.9k	310k
Test(other)	61.3k	65.5k	70.5k	333k

Table 2: The number of words in training data

	TM	LM	Dev	Test
de	31.3M	-	55.1k	59.4k
en	32.8M	50.5M	58.8k	55.5k

Table 3: The number of words in training data

We use GIZA++ and Moses default parameters for training. Decoding was carried out using the cdec decoder (**?**). Feature weights were tuned on the development data by running MIRA (Chiang, 2012) for 20 iterations with 16 parallel. For other parameters, we used cdec's default values. The numbers reported here are the average of three tuning runs (Hopkins and May, 2011).

Table 1 lists the results measured using BLEU (Papineni et al., 2002).The term Sample denotes the combination size for each model. The term SIZE in the table denotes the number of the extracted grammar types composed of Hiero rules and phrase pairs. The numbers in italic denotes the score of Back, significantly improved from the score of 1 sampled combinated Gen. The numbers in bold denotes the score of Back + future, significantly improved from the score of 1 sampled combinated Back. All significance test are performed using Clark et al. (2011) under p-value of 0.05. Back performed better than Gen on Spanish-English and French-English language pairs. Note that the gains were achieved with the comparable grammar size. When comparing German-English and Japanese-English language pairs, there are no significant differences between Back and Gen. The combination of our Back with future score during slice sampling (+future) achieved further gains over the slice sampling without future scores, and slightly decrese the grammar size, compared to Back. However, there are still no significant difference between Back+future and Gen on German-English and Japanese-English language pairs. Sample combination has no or slight gain on BLEU score, in spite of the increase in grammar size. From the results, using last one sample as a grammar is sufficient for translation quality. The performance of the Bayesian model did not match with that for the GIZA++ pipeline heuristic approach. In general, complex model, such as Gen and Back, demands larger corpus size for training, and the evaluation on such smaller corpus may not be a fair comparison, since the sampling approach can rely on only sampled derivations. Thus, we evaluate these methods on large size corpus in the next section.

6.2 Comparison with Heuristic Extraction

As reported in (Koehn et al., 2003; DeNero et al., 2006), the comparison against heuristic extraction is a challenging task. We compare the Back+future and a baseline extracted from grow-diag-final alignments of GIZA++ using Moses with Hiero options. We use GIZA++ and Moses default parameters for training. In addition, we present heuristic extraction from the last 1 sample of Back+future in +Exhaustive.

We used the full europarl-v7 German-English corpus as presented in Table 3. The experimental set up was similar to that in Section 6.1 with the following exceptions; Slice sampling parameter a was set to 0.05. Mini-batch size was set to 1024 and sampling was performed 5 iterations.[6] The translation model was extracted by last 1 iterations.

Table 4 lists the results[7]. Our Back+future can

[6]Inference took 5 days.

[7]The row mark up with ∗ indicate the model using word

Model	BLEU	SIZE
∗ GIZA++ Model 4	27.21	73.24M (×14.0)
GIZA++ Model 3	26.78	59.26M (×11.3)
Back + future	26.83	5.25M (×1.0)
Back + future + exhasustive	26.73	90.42M (×17.2)

Table 4: Results of translation evaluation in de-en full size corpus.

Gen	*gin* X *kamera* / silver X camera *en* / salt
Back + future	*gin en kamera* / silver salt camera

Table 5: Example of a grammar

decrease the grammar size against GIZA++ with comparable BLEU score. Surprisingly, exhaustive extraction had no gains, probably because of the word alignment in each Hiero rules relied on the IBM Model 1.

7 Analysis

Intuitively, the use of the hierarchical back-off increases the Hiero grammar size, since the phrases of all the granularities in the derivation trees are incorporated in the grammar. In contrast, our hierarchical back-off model achieved gains in translation quality without increasing the size of the extracted grammar when compared to the previous generative model. The major differences were the use of the minimal phrase pairs used in the previous work in which only minimal phrase pairs in the leaves of derivation trees were included in the model. As a result, larger phrase pairs were forced to be constructed from those minimal rules. On the other hand, our back-off model could directly express phrase pairs of multiple granularities. In particular, a complex noun may be composed of several Hiero rules in the previous model, but it can be directly expressed by a single phrase pair in our model. Table 5 gives an example of a Japanese-English phrase pair which is represented by two Hiero rules in the previous model; it is directly expressed by a single phrase pair in our model.

The BLEU score of Back+future was higher than the generative baseline with comparable grammar size. We observed that a very different word alignment was sampled in every training iteration; the tendency was very frequent for function words. Our future score for inferring the slice variables may take into account the context in a sentence better than those without the future score. As a result, Back+future infers better models by avoiding over pruning spans.

The BLEU score of our back-off model did not achieve gains over the heuristic baselines. The detail analysis of the learned Hiero grammar's CRP tables reveals that the grammar is very sparse and may have little generalization capacity. The expansion of back-off process and the use of word classes will solve the sparsity and increase the translation quality.

class informations. Model 3 and our Back-off model dose not use any word class informations.

8 Conclusion

We proposed a hierarchical back-off model for Hiero grammar. Our back-off model achieved higher or equal translation quality against a previous Bayesian model under BLEU score on various language pairs;German/French/Spanish/Japanese-English. In addition to the hierarchical back-off model, we also proposed a two-step slice sampling approach. We showed that the two-step slice sampling approach can avoid over-pruning by incorporating a future score for estimating slice variables, which led to increase in translation quality through the experiments. The joint use of hierarchical back-off model and two step slice sampling approach achieved comparable translation quality on a full size Germany-English language pair in Europarl v7 corpus with with significantly smaller grammar size; 10% less than that for he heuristic baseline.

For future work, we plan to embed a back-off feature to decoder which is computed for all the phrase pairs constructed in a derivation during the decoding process. We will reflect the change of a probability as a statefull feature for decoding step.

References

Phil Blunsom and Trevor Cohn. 2010. Inducing synchronous grammars with slice sampling. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 238–241, Los Angeles, California, June. Association for Computational Linguistics.

Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 782–790, Suntec, Singapore, August. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar F Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. *computational linguistics*, 33(2):201–228.

David Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *The Journal of Machine Learning Research*, 13(1):1159–1187.

Tagyoung Chung, Licheng Fang, Daniel Gildea, and Daniel Štefankovič. 2014. Sampling tree fragments from forests. *Computational Linguistics*, 40(1):203–229.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June. Association for Computational Linguistics.

Martin Cmejrek and Bowen Zhou. 2010. Two methods for extending hierarchical rules from the bilingual chart parsing. In *Coling 2010: Posters*, pages 180–188, Beijing, China, August. Coling 2010 Organizing Committee.

Trevor Cohn and Gholamreza Haffari. 2013. An infinite hierarchical bayesian model of phrasal translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 780–790, Sofia, Bulgaria, August. Association for Computational Linguistics.

John DeNero, Dan Gillick, James Zhang, and Dan Klein. 2006. Why generative phrase models underperform surface heuristics. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 31–38, New York City, June. Association for Computational Linguistics.

John DeNero, Alexandre Bouchard-Côté, and Dan Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 314–323, Honolulu, Hawaii, October. Association for Computational Linguistics.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan 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 the ACL 2010 System Demonstrations*, pages 7–12, Uppsala, Sweden, July. Association for Computational Linguistics.

Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin K Tsou. 2013. Overview of the patent machine translation task at the ntcir-10 workshop. In *Proceedings of the 10th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, NTCIR-10*.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic, June. Association for Computational Linguistics.

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*, pages 48–54. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. 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*, pages 223–232, Jeju Island, Korea, July. Association for Computational Linguistics.

Wang Ling, João Graça, Isabel Trancoso, and Alan Black. 2012. Entropy-based pruning for phrase-based machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 962–971, Jeju Island, Korea, July. Association for Computational Linguistics.

Daniel Marcu and Daniel Wong. 2002. A phrase-based,joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference*

on Empirical Methods in Natural Language Processing, pages 133–139. Association for Computational Linguistics, July.

Radford M Neal. 2003. Slice sampling. *Annals of statistics*, pages 705–741.

Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 632–641, Portland, Oregon, USA, June. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Xiaochang Peng and Daniel Gildea. 2014. Type-based mcmc for sampling tree fragments from forests. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1735–1745, Doha, Qatar, October. Association for Computational Linguistics.

Jim Pitman and Marc Yor. 1997. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, pages 855–900.

Andreas Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*.

Yee Whye Teh. 2006. A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 985–992, Sydney, Australia, July. Association for Computational Linguistics.

Jurgen Van Gael, Yunus Saatci, Yee Whye Teh, and Zoubin Ghahramani. 2008. Beam sampling for the infinite hidden markov model. In *Proceedings of the 25th international conference on Machine learning*, pages 1088–1095. ACM.

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

Xinyan Xiao, Deyi Xiong, Yang Liu, Qun Liu, and Shouxun Lin. 2012. Unsupervised discriminative induction of synchronous grammar for machine translation. In *Proceedings of COLING 2012*, pages 2883–2898, Mumbai, India, December. The COLING 2012 Organizing Committee.

Richard Zens, Daisy Stanton, and Peng Xu. 2012. A systematic comparison of phrase table pruning techniques. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 972–983, Jeju Island, Korea, July. Association for Computational Linguistics.

Kai Zhao and Liang Huang. 2013. Minibatch and parallelization for online large margin structured learning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 370–379, Atlanta, Georgia, June. Association for Computational Linguistics.