Part-of-Speech Induction in Dependency Trees for Statistical Machine Translation

Akihiro Tamura[†,‡], **Taro Watanabe**[†], **Eiichiro Sumita**[†],
Hiroya Takamura[‡], **Manabu Okumura**[‡]
† National Institute of Information and Communications Technology
{akihiro.tamura, taro.watanabe, eiichiro.sumita}@nict.go.jp
† Precision and Intelligence Laboratory, Tokyo Institute of Technology
{takamura, oku}@pi.titech.ac.jp

Abstract

This paper proposes a nonparametric Bayesian method for inducing Part-of-Speech (POS) tags in dependency trees to improve the performance of statistical machine translation (SMT). In particular, we extend the monolingual infinite tree model (Finkel et al., 2007) to a bilingual scenario: each hidden state (POS tag) of a source-side dependency tree emits a source word together with its aligned target word, either jointly (joint model), or independently (independent model). Evaluations of Japanese-to-English translation on the NTCIR-9 data show that our induced Japanese POS tags for dependency trees improve the performance of a forest-to-string SMT system. Our independent model gains over 1 point in BLEU by resolving the sparseness problem introduced in the joint model.

1 Introduction

In recent years, syntax-based SMT has made promising progress by employing either dependency parsing (Lin, 2004; Ding and Palmer, 2005; Quirk et al., 2005; Shen et al., 2008; Mi and Liu, 2010) or constituency parsing (Huang et al., 2006; Liu et al., 2006; Galley et al., 2006; Mi and Huang, 2008; Zhang et al., 2008; Cohn and Blunsom, 2009; Liu et al., 2009; Mi and Liu, 2010; Zhang et al., 2011) on the source side, the target side, or both. However, dependency parsing, which is a popular choice for Japanese, can incorporate only shallow syntactic information, i.e., POS tags, compared with the richer syntactic phrasal categories in constituency parsing. Moreover, existing POS tagsets might not be optimal for SMT because they are constructed without considering the language in the other side. Consider the examples in Figure 1. The Japanese noun "利用" in

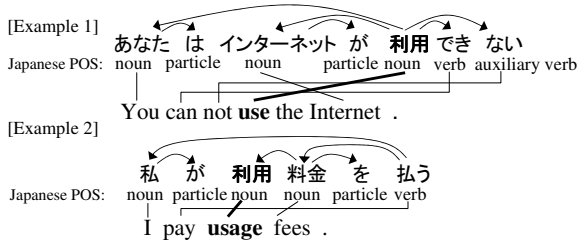

Figure 1: Examples of Existing Japanese POS Tags and Dependency Structures

Example 1 corresponds to the English verb "use", while that in Example 2 corresponds to the English noun "usage". Thus, Japanese nouns act like verbs in English in one situation, and nouns in English in another. If we could discriminate POS tags for two cases, we might improve the performance of a Japanese-to-English SMT system.

In the face of the above situations, this paper proposes an unsupervised method for inducing POS tags for SMT, and aims to improve the performance of syntax-based SMT by utilizing the induced POS tagset. The proposed method is based on the infinite tree model proposed by Finkel et al. (2007), which is a nonparametric Bayesian method for inducing POS tags from syntactic dependency structures. In this model, hidden states represent POS tags, the observations they generate represent the words themselves, and tree structures represent syntactic dependencies between pairs of POS tags.

The proposed method builds on this model by incorporating the aligned words in the other language into the observations. We investigate two types of models: (i) a joint model and (ii) an independent model. In the joint model, each hidden state jointly emits both a source word and its aligned target word as an observation. The independent model separately emits words in two languages from hidden states. By inferring POS

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 841–851,
Sofia, Bulgaria, August 4-9 2013.

tags based on bilingual observations, both models can induce POS tags by incorporating information from the other language. Consider, for example, inducing a POS tag for the Japanese word "利用" in Figure 1. Under a monolingual induction method (e.g., the infinite tree model), the "利用" in Example 1 and 2 would both be assigned the same POS tag since they share the same observation. However, our models would assign separate tags for the two different instances since the "利用" in Example 1 and Example 2 could be disambiguated by encoding the target-side information, either "use" or "usage", in the observations.

Inference is efficiently carried out by beam sampling (Gael et al., 2008), which combines slice sampling and dynamic programming. Experiments are carried out on the NTCIR-9 Japanese-to-English task using a binarized forest-to-string SMT system with dependency trees as its source side. Our bilingually-induced tagset significantly outperforms the original tagset and the monolingually-induced tagset. Further, our independent model achieves a more than 1 point gain in BLEU, which resolves the sparseness problem introduced by the bi-word observations.

2 Related Work

A number of unsupervised methods have been proposed for inducing POS tags. Early methods have the problem that the number of possible POS tags must be provided preliminarily. This limitation has been overcome by automatically adjusting the number of possible POS tags using nonparametric Bayesian methods (Finkel et al., 2007; Gael et al., 2009; Blunsom and Cohn, 2011; Sirts and Alumäe, 2012). Gael et al. (2009) applied infinite HMM (iHMM) (Beal et al., 2001; Teh et al., 2006), a nonparametric version of HMM, to POS induction. Blunsom and Cohn (2011) used a hierarchical Pitman-Yor process prior to the transition and emission distribution for sophisticated smoothing. Sirts and Alumäe (2012) built a model that combines POS induction and morphological segmentation into a single learning problem. Finkel et al. (2007) proposed the infinite tree model, which represents recursive branching structures over infinite hidden states and induces POS tags from syntactic dependency structures. In the following, we overview the infinite tree model, which is the basis of our proposed model. In particular, we will describe the independent children model (Finkel et al., 2007), where children are dependent only on their parents, used in our proposed model[1].

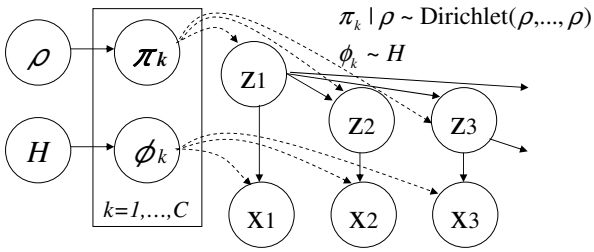

Figure 2: A Graphical Representation of the Finite Tree Model

2.1 Finite Tree Model

We first review the finite tree model, which can be graphically represented in Figure 2. Let T_t denote the tree whose root node is t. A node t has a hidden state z_t (the POS tag) and an observation x_t (the word). The probability of a tree T_t, $p_T(T_t)$, is recursively defined: $p_T(T_t) = p(x_t|z_t) \prod_{t' \in c(t)} p(z_{t'}|z_t) p_T(T_{t'})$, where $c(t)$ is the set of the children of t.

Let each hidden state variable have C possible values indexed by k. For each state k, there is a parameter ϕ_k which parameterizes the observation distribution for that state: $x_t|z_t \sim F(\phi_{z_t})$. ϕ_k is distributed according to a prior distribution H: $\phi_k \sim H$.

Transitions between states are governed by Markov dynamics parameterized by $\boldsymbol{\pi}$, where $\pi_{ij} = p(z_{c(t)} = j|z_t = i)$ and $\boldsymbol{\pi}_k$ are the transition probabilities from the parent's state k. $\boldsymbol{\pi}_k$ is distributed according to a Dirichlet distribution with parameter ρ: $\boldsymbol{\pi}_k|\rho \sim \text{Dirichlet}(\rho, \ldots, \rho)$. The hidden state of each child $z_{t'}$ is distributed according to a multinomial distribution $\boldsymbol{\pi}_{z_t}$ specific to the parent's state z_t: $z_{t'}|z_t \sim \text{Multinomial}(\boldsymbol{\pi}_{z_t})$.

2.2 Infinite Tree Model

In the infinite tree model, the number of possible hidden states is potentially infinite. The infinite model is formed by extending the finite tree model using a hierarchical Dirichlet process (HDP) (Teh et al., 2006). The reason for using an HDP rather

[1]Finkel et al. (2007) originally proposed three types of models: besides the independent children model, the simultaneous children model and the markov children model. Although we could apply the other two models, we leave this for future work.

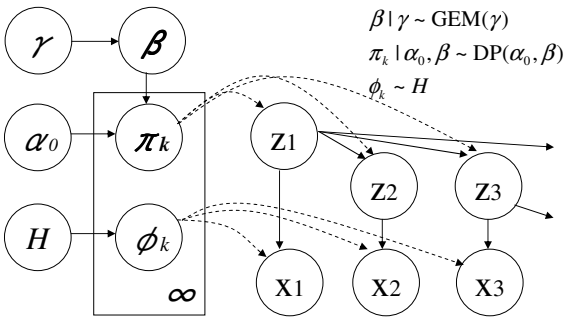

Figure 3: A Graphical Representation of the Infinite Tree Model

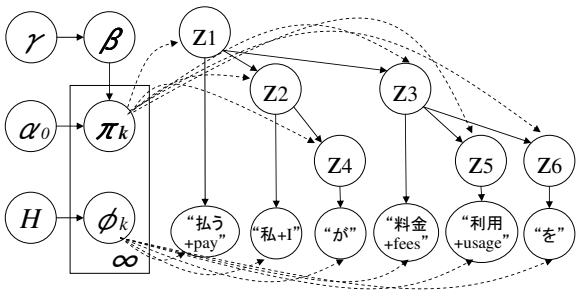

Figure 4: An Example of the Joint Model

than a simple Dirichlet process (DP)[2] (Ferguson, 1973) is that we have to introduce coupling across transitions from different parent's states. A similar measure was adopted in iHMM (Beal et al., 2001).

HDP is a set of DPs coupled through a shared random base measure which is itself drawn from a DP: each $G_k \sim \mathrm{DP}(\alpha_0, G_0)$ with a shared base measure G_0, and $G_0 \sim \mathrm{DP}(\gamma, H)$ with a global base measure H. From the viewpoint of the stick-breaking construction[3] (Sethuraman, 1994), the HDP is interpreted as follows: $G_0 = \sum_{k'=1}^{\infty} \boldsymbol{\beta}_{k'} \delta_{\phi_{k'}}$ and $G_k = \sum_{k'=1}^{\infty} \pi_{kk'} \delta_{\phi_{k'}}$, where $\boldsymbol{\beta} \sim \mathrm{GEM}(\gamma)$, $\boldsymbol{\pi}_k \sim \mathrm{DP}(\alpha_0, \boldsymbol{\beta})$, and $\phi_{k'} \sim H$.

We regard each G_k as two coindexed distributions: $\boldsymbol{\pi}_k$, a distribution over the transition probabilities from the parent's state k, and $\phi_{k'}$, an observation distribution for the state k'. Then, the infinite tree model is formally defined as follows:

$$
\begin{aligned}
&\boldsymbol{\beta}|\gamma \sim \mathrm{GEM}(\gamma), \\
&\boldsymbol{\pi}_k|\alpha_0, \boldsymbol{\beta} \sim \mathrm{DP}(\alpha_0, \boldsymbol{\beta}), \\
&\phi_k \sim H, \\
&z_{t'}|z_t \sim \mathrm{Multinomial}(\boldsymbol{\pi}_{z_t}), \\
&x_t|z_t \sim F(\phi_{z_t}).
\end{aligned}
$$

Figure 3 shows the graphical representation of the infinite tree model. The primary difference between Figure 2 and Figure 3 is whether the number of copies of the state is finite or not.

3 Bilingual Infinite Tree Model

We propose a bilingual variant of the infinite tree model, the bilingual infinite tree model, which utilizes information from the other language. Specifically, the proposed model introduces bilingual observations by embedding the aligned target words in the source-side dependency trees. This paper proposes two types of models that differ in their processes for generating observations: the joint model and the independent model.

3.1 Joint Model

The joint model is a simple application of the infinite tree model under a bilingual scenario. The model is formally defined in the same way as in Section 2.2 and is graphically represented similarly to Figure 3. The only difference from the infinite tree model is the instances of observations (x_t). Observations in the joint model are the combination of source words and their aligned target words[4], while observations in the monolingual infinite tree model represent only source words. For each source word, all the aligned target words are copied and sorted in alphabetical order, and then concatenated into a single observation. Therefore, a single target word may be emitted multiple times if the target word is aligned with multiple source words. Likewise, there may be target words which may not be emitted by our model, if the target words are not aligned.

Figure 4 shows the process of generating Example 2 in Figure 1 through the joint model, where aligned words are jointly emitted as observations. In Figure 4, the POS tag of "利用" (z_5) generates

[2] DP is a measure on measures. It has two parameters, a scaling parameter α and a base measure H: $DP(\alpha, H)$.

[3] Sethuraman (1994) showed a definition of a measure $G \sim \mathrm{DP}(\alpha_0, G_0)$. First, infinite sequences of i.i.d variables $(\pi'_k)_{k=1}^{\infty}$ and $(\phi_k)_{k=1}^{\infty}$ are generated: $\pi'_k|\alpha_0 \sim \mathrm{Beta}(1, \alpha_0)$, $\phi_k \sim G_0$. Then, G is defined as: $\pi_k = \pi'_k \prod_{l=1}^{k-1}(1 - \pi'_l)$, $G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$. If $\boldsymbol{\pi}$ is defined by this process, then we write $\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha_0)$.

[4] When no target words are aligned, we simply add a NULL target word.

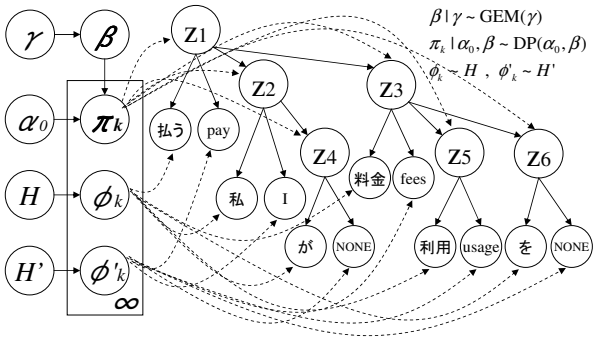

Figure 5: A Graphical Representation of the Independent Model

the string "利用+usage" as the observation (x_5). Similarly, the POS tag of "利用" in Example 1 would generate the string "利用+use". Hence, this model can assign different POS tags to the two different instances of the word "利用", based on the different observation distributions in inference.

3.2 Independent Model

The joint model is prone to a data sparseness problem, since each observation is a combination of a source word and its aligned target word. Thus, we propose an independent model, where each hidden state generates a source word and its aligned target word separately. For the aligned target side, we introduce an observation variable x'_t for each z_t and a parameter ϕ'_k for each state k, which parameterizes a distinct distribution over the observations x'_t for that state. ϕ'_k is distributed according to a prior distribution H'. Specifically, the independent model is formally defined as follows:

$$\begin{aligned}
&\boldsymbol{\beta}|\gamma \sim \text{GEM}(\gamma),\\
&\boldsymbol{\pi}_k|\alpha_0, \boldsymbol{\beta} \sim \text{DP}(\alpha_0, \boldsymbol{\beta}),\\
&\phi_k \sim H,\ \ \phi'_k \sim H',\\
&z_{t'}|z_t \sim \text{Multinomial}(\boldsymbol{\pi}_{z_t}),\\
&x_t|z_t \sim F(\phi_{z_t}),\ \ x'_t|z_t \sim F'(\phi'_{z_t}).
\end{aligned}$$

When multiple target words are aligned to a single source word, each aligned word is generated separately from observation distribution parameterized by ϕ'_k.

Figure 5 graphs the process of generating Example 2 in Figure 1 using the independent model. x'_t and ϕ'_k are introduced for aligned target words. The state of "利用" (z_5) generates the Japanese word "利用" as x_5 and the English word "usage" as x'_5. Due to this factorization, the independent model is less subject to the sparseness problem.

3.3 Introduction of Other Factors

We assumed the surface form of aligned target words as additional observations in previous sections. Here, we introduce additional factors, i.e., the POS of aligned target words, in the observations. Note that POSs of target words are assigned by a POS tagger in the target language and are not inferred in the proposed model.

First, we can simply replace surface forms of target words with their POSs to overcome the sparseness problem. Second, we can incorporate both information from the target language as observations. In the joint model, two pieces of information are concatenated into a single observation. In the independent model, we introduce observation variables (e.g., x'_t and x''_t) and parameters (e.g., ϕ'_k and ϕ''_k) for each information. Specifically, x'_t and ϕ'_k are introduced for the surface form of aligned words, and x''_t and ϕ''_k for the POS of aligned words. Consider, for example, Example 1 in Figure 1. The POS tag of "利用" generates the string "利用+use+verb" as the observation in the joint model, while it generates "利用", "use", and "verb" independently in the independent model.

3.4 POS Refinement

We have assumed a completely unsupervised way of inducing POS tags in dependency trees. Another realistic scenario is to refine the existing POS tags (Finkel et al., 2007; Liang et al., 2007) so that each refined sub-POS tag may reflect the information from the aligned words while preserving the handcrafted distinction from original POS tagset. Major difference is that we introduce separate transition probabilities $\boldsymbol{\pi}^s_k$ and observation distributions (ϕ^s_k, ϕ'^s_k) for each existing POS tag s. Then, each node t is constrained to follow the distributions indicated by the initially assigned POS tag s_t, and we use the pair (s_t, z_t) as a state representation.

3.5 Inference

In inference, we find the state set that maximizes the posterior probability of state transitions given observations (i.e., $P(z_{1:n}|x_{1:n})$). However, we cannot evaluate the probability for all possible states because the number of states is infinite. Finkel et al. (2007) presented a sampling algorithm for the infinite tree model, which is based on the Gibbs sampling in the direct assignment representation for iHMM (Teh et al., 2006). In the

Gibbs sampling, individual hidden state variables are resampled conditioned on all other variables. Unfortunately, its convergence is slow in HMM settings because sequential data is likely to have a strong correlation between hidden states (Gael et al., 2008).

We present an inference procedure based on beam sampling (Gael et al., 2008) for the joint model and the independent model. Beam sampling limits the number of possible state transitions for each node to a finite number using slice sampling (Neal, 2003), and then efficiently samples whole hidden state transitions using dynamic programming. Beam sampling does not suffer from slow convergence as in Gibbs sampling by sampling the whole state variables at once. In addition, Gael et al. (2008) showed that beam sampling is more robust to initialization and hyperparameter choice than Gibbs sampling.

Specifically, we introduce an auxiliary variable u_t for each node in a dependency tree to limit the number of possible transitions. Our procedure alternates between sampling each of the following variables: the auxiliary variables $\boldsymbol{u}$, the state assignments $\boldsymbol{z}$, the transition probabilities $\boldsymbol{\pi}$, the shared DP parameters $\boldsymbol{\beta}$, and the hyperparameters α_0 and γ. We can parallelize procedures in sampling $\boldsymbol{u}$ and $\boldsymbol{z}$ because the slice sampling for $\boldsymbol{u}$ and the dynamic programing for $\boldsymbol{z}$ are independent for each sentence. See Gael el al. (2009) for details.

The only difference between inferences in the joint model and the independent model is in computing the posterior probability of state transitions given observations (e.g., $p(z_{1:n}|x_{1:n})$ and $p(z_{1:n}|x_{1:n}, x'_{1:n})$) in sampling $\boldsymbol{z}$. In the following, we describe each sampling stage. See Teh et al., (2006) for details of sampling $\boldsymbol{\pi}$, $\boldsymbol{\beta}$, α_0 and γ.

Sampling $\boldsymbol{u}$:

Each u_t is sampled from the uniform distribution on $[0, \pi_{z_{d(t)}z_t}]$, where $d(t)$ is the parent of t: $u_t \sim \text{Uniform}(0, \pi_{z_{d(t)}z_t})$. Note that u_t is a positive number, since each transition probability $\pi_{z_{d(t)}z_t}$ is larger than zero.

Sampling $\boldsymbol{z}$:

Possible values k of z_t are divided into the two sets using u_t: a finite set with $\pi_{z_{d(t)}k} > u_t$ and an infinite set with $\pi_{z_{d(t)}k} \leq u_t$. The beam sampling considers only the former set. Owing to the truncation of the latter set, we can compute the posterior probability of a state z_t given observations for all t ($t = 1, \ldots, T$) using dynamic programming as follows:
In the joint model, $p(z_t|x_{\sigma(t)}, u_{\sigma(t)}) \propto$
$p(x_t|z_t) \cdot \sum_{z_{d(t)}:\pi_{z_{d(t)}z_t} > u_t} p(z_{d(t)}|x_{\sigma(d(t))}, u_{\sigma(d(t))})$,
and in the independent model,
$p(z_t|x_{\sigma(t)}, x'_{\sigma(t)}, u_{\sigma(t)}) \propto p(x_t|z_t) \cdot p(x'_t|z_t)$
$\cdot \sum_{z_{d(t)}:\pi_{z_{d(t)}z_t} > u_t} p(z_{d(t)}|x_{\sigma(d(t))}, x'_{\sigma(d(t))}, u_{\sigma(d(t))})$,
where $x_{\sigma(t)}$ (or $u_{\sigma(t)}$) denotes the set of x_t (or u_t) on the path from the root node to the node t in a tree.

In our experiments, we assume that $F(\phi_k)$ is Multinomial(ϕ_k) and H is Dirichlet($\rho, \ldots, \rho$), which is the same in Finkel et al. (2007). Under this assumption, the posterior probability of an observation is as follows: $p(x_t|z_t) = \frac{\dot{n}_{x_t k} + \rho}{\dot{n}_{\cdot k} + N\rho}$, where $\dot{n}_{xk}$ is the number of observations x with state k, $\dot{n}_{\cdot k}$ is the number of hidden states whose values are k, and N is the total number of observations $\boldsymbol{x}$. Similarly, $p(x'_t|z_t) = \frac{\dot{n}_{x'_t k} + \rho'}{\dot{n}_{\cdot k} + N'\rho'}$, where N' is the total number of observations $\boldsymbol{x}'$.

When the posterior probability of a state z_t given observations for all t can be computed, we first sample the state of each leaf node and then perform backtrack sampling for every other z_t where the z_t is sampled given the sample for $z_{c(t)}$ as follows: $p(z_t|z_{c(t)}, x_{1:T}, u_{1:T}) \propto p(z_t|x_{\sigma(t)}, u_{\sigma(t)}) \prod_{t' \in c(t)} p(z_{t'}|z_t, u_{t'})$.

Sampling $\boldsymbol{\pi}$:

We introduce a count variable $n_{ij} \in \boldsymbol{n}$, which is the number of observations with state j whose parent's state is i. Then, we sample $\boldsymbol{\pi}$ using the Dirichlet distribution: $(\pi_{k1}, \ldots, \pi_{kK}, \sum_{k'=K+1}^{\infty} \pi_{kk'}) \sim \text{Dirichlet}(n_{k1} + \alpha_0\beta_1, \ldots, n_{kK} + \alpha_0\beta_K, \alpha_0 \sum_{k'=K+1}^{\infty} \beta_{k'})$, where K is the number of distinct states in $\boldsymbol{z}$.

Sampling $\boldsymbol{\beta}$:

We introduce a set of auxiliary variables $\boldsymbol{m}$, where $m_{ij} \in \boldsymbol{m}$ is the number of elements of $\boldsymbol{\pi}_j$ corresponding to β_i. The conditional distribution of each variable is $p(m_{ij} = m|\boldsymbol{z}, \boldsymbol{\beta}, \alpha_0) \propto S(n_{ij}, m)(\alpha_0\beta_j)^m$, where $S(n, m)$ are unsigned Stirling numbers of the first kind[5].

[5] $S(0,0) = S(1,1) = 1$, $S(n,0) = 0$ for $n > 0$, $S(n,m) = 0$ for $m > n$, and $S(n+1,m) = S(n,m-1) + nS(n,m)$ for others.

The parameters $\boldsymbol{\beta}$ are sampled using the Dirichlet distribution: $(\beta_1, \ldots, \beta_K, \sum_{k'=K+1}^{\infty} \beta_{k'}) \sim \text{Dirichlet}(m_{\cdot 1}, \ldots, m_{\cdot K}, \gamma)$, where $m_{\cdot k} = \sum_{k'=1}^{K} m_{k'k}$.

Sampling α_0:

α_0 is parameterized by a gamma hyperprior with hyperparameters α_a and α_b. We introduce two types of auxiliary variables for each state ($k = 1, \ldots, K$), $w_k \in [0, 1]$ and $v_k \in \{0, 1\}$. The conditional distribution of each w_k is $p(w_k|\alpha_0) \propto w_k^{\alpha_0}(1-w_k)^{n_{\cdot k}-1}$ and that of each v_k is $p(v_k|\alpha_0) \propto (\frac{n_{\cdot k}}{\alpha_0})^{v_k}$, where $n_{\cdot k} = \sum_{k'=1}^{K} n_{k'k}$. The conditional distribution of α_0 given w_k and v_k ($k = 1, \ldots, K$) is $p(\alpha_0|\boldsymbol{w}, \boldsymbol{v}) \propto \alpha_0^{\alpha_a - 1 + m_{\cdot\cdot} - \sum_{k=1}^{K} v_k} e^{-\alpha_0(\alpha_b - \sum_{k=1}^{K} \log w_k)}$, where $m_{\cdot\cdot} = \sum_{k'=1}^{K} \sum_{k''=1}^{K} m_{k'k''}$.

Sampling γ:

γ is parameterized by a gamma hyperprior with hyperparameters γ_a and γ_b. We introduce an auxiliary variable η, whose conditional distribution is $p(\eta|\gamma) \propto \eta^{\gamma}(1-\eta)^{m_{\cdot\cdot}-1}$. The conditional distribution of γ given η is $p(\gamma|\eta) \propto \gamma^{\gamma_a - 1 + K} e^{-\gamma(\gamma_b - \log\eta)}$.

4 Experiment

We tested our proposed models under the NTCIR-9 Japanese-to-English patent translation task (Goto et al., 2011), consisting of approximately 3.2 million bilingual sentences. Both the development data and the test data consist of 2,000 sentences. We also used the NTCIR-7 development data consisting of 2,741 sentences for development testing purposes.

4.1 Experimental Setup

We evaluated our bilingual infinite tree model for POS induction using an in-house developed syntax-based forest-to-string SMT system. In the training process, the following steps are performed sequentially: preprocessing, inducing a POS tagset for a source language, training a POS tagger and a dependency parser, and training a forest-to-string MT model.

Step 1. Preprocessing

We used the first 10,000 Japanese-English sentence pairs in the NTCIR-9 training data for inducing a POS tagset for Japanese[6]. The Japanese sentences were segmented using MeCab[7], and the English sentences were tokenized and POS tagged using TreeTagger (Schmid, 1994), where 43 and 58 types of POS tags are included in the Japanese sentences and the English sentences, respectively. The Japanese POS tags come from the second-level POS tags in the IPA POS tagset (Asahara and Matsumoto, 2003) and the English POS tags are derived from the Penn Treebank. Note that the Japanese POS tags are used for initialization of hidden states and the English POS tags are used as observations emitted by hidden states.

Word-by-word alignments for the sentence pairs are produced by first running GIZA++ (Och and Ney, 2003) in both directions and then combining the alignments using the "grow-diag-final-and" heuristic (Koehn et al., 2003). Note that we ran GIZA++ on all of the NTCIR-9 training data in order to obtain better alignements.

The Japanese sentences are parsed using CaboCha (Kudo and Matsumoto, 2002), which generates dependency structures using a phrasal unit called a $bunsetsu$[8], rather than a word unit as in English or Chinese dependency parsing. Since we focus on the word-level POS induction, each $bunsetsu$-based dependency tree is converted into its corresponding word-based dependency tree using the following heuristic[9]: first, the last function word inside each $bunsetsu$ is identified as the head word[10]; then, the remaining words are treated as dependents of the head word in the same $bunsetsu$; finally, a $bunsetsu$-based dependency structure is transformed to a word-based dependency structure by preserving the head/modifier relationships of the determined head words.

Step 2. POS Induction

A POS tag for each word in the Japanese sentences is inferred by our bilingual infinite tree model, ei-

[6] Due to the high computational cost, we did not use all the NTCIR-9 training data. We leave scaling up to a larger dataset for future work.

[7] `http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html`

[8] A $bunsetsu$ is the smallest meaningful sequence consisting of a content word and accompanying function words (e.g., a noun and a particle).

[9] We could use other word-based dependency trees such as trees by the infinite PCFG model (Liang et al., 2007) and syntactic-head or semantic-head dependency trees in Nakazawa and Kurohashi (2012), although it is not our major focus. We leave this for future work.

[10] If no function words exist in a $bunsetsu$, the last content word is treated as the head word.

ther jointly ($Joint$) or independently (Ind). We also performed monolingual induction of Finkel et al. (2007) for comparison ($Mono$). In each model, a sequence of sampling $\boldsymbol{u}$, $\boldsymbol{z}$, $\boldsymbol{\pi}$, $\boldsymbol{\beta}$, α_0, and γ is repeated 10,000 times. In sampling α_0 and γ, hyperparameters α_a, α_b, γ_a, and γ_b are set to 2, 1, 1, and 1, respectively, which is the same setting in Gael et al. (2008). In sampling $\boldsymbol{z}$, parameters ρ, ρ', ..., are set to 0.01. In the experiments, three types of factors for the aligned English words are compared: surface forms ('s'), POS tags ('P'), and the combination of both ('s+P'). Further, two types of inference frameworks are compared: $induction$ (IND) and $refinement$ (REF). In both frameworks, each hidden state z_t is first initialized to the POS tags assigned by MeCab (the IPA POS tagset), and then each state is updated through the inference procedure described in Section 3.5. Note that in REF, the sampling distribution over z_t is constrained to include only states that are a refinement of the initially assigned POS tag.

Step 3. Training a POS Tagger and a Dependency Parser

In this step, we train a Japanese dependency parser from the 10,000 Japanese dependency trees with the induced POS tags which are derived from Step 2. We employed a transition-based dependency parser which can jointly learn POS tagging and dependency parsing (Hatori et al., 2011) under an incremental framework[11]. Note that the learned parser can identify dependencies between words and attach an induced POS tag for each word.

Step 4. Training a Forest-to-String MT

In this step, we train a forest-to-string MT model based on the learned dependency parser in Step 3. We use an in-house developed hypergraph-based toolkit, *cicada*, for training and decoding with a tree-to-string model, which has been successfully employed in our previous work for system combination (Watanabe and Sumita, 2011) and online learning (Watanabe, 2012). All the Japanese and English sentences in the NTCIR-9 training data are segmented in the same way as in Step 1, and then each Japanese sentence is parsed by the dependency parser learned in Step 3, which simultaneously assigns induced POS tags and word dependencies. Finally, a forest-to-string MT model is learned with Zhang et al., (2011), which extracts translation rules by a forest-based variant of the GHKM algorithm (Mi and Huang, 2008) after each parse tree is restructured into a binarized packed forest. Parameters are tuned on the development data using xBLEU (Rosti et al., 2011) as an objective and L-BFGS (Liu and Nocedal, 1989) as an optimization toolkit, since it is stable and less prone to randomness, unlike MERT (Och, 2003) or PRO (Hopkins and May, 2011). The development test data is used to set up hyperparameters, i.e., to terminate tuning iterations.

When translating Japanese sentences, a parse tree for each sentence is constructed in the same way as described earlier in this step, and then the parse trees are translated into English sentences using the learned forest-to-string MT model.

[11] `http://triplet.cc/software/corbit/`

	IND	REF
BS	27.54	
$Mono$	27.66	26.83
$Joint$[s]	**28.00**	**28.00**
$Joint$[P]	26.36	26.72
$Joint$[s+P]	**27.99**	**27.82**
Ind[s]	**28.00**	**27.93**
Ind[P]	**28.11**	**28.63**
Ind[s+P]	**28.13**	**28.62**

Table 1: Performance on Japanese-to-English Translation Measured by BLEU (%)

4.2 Experimental Results

Table 1 shows the performance for the test data measured by case sensitive BLEU (Papineni et al., 2002). We also present the performance of our baseline forest-to-string MT system (BS) using the original IPA POS tags. In Table 1, numbers in bold indicate that the systems outperform the baselines, BS and $Mono$. Under the Moses phrase-based SMT system (Koehn et al., 2007) with the default settings, we achieved a 26.80% BLEU score.

Table 1 shows that the proposed systems outperform the baseline $Mono$. The differences between the performance of Ind[s+P] and $Mono$ are statistically significant in the bootstrap method (Koehn, 2004), with a 1% significance level both in IND and REF. The results indicate that integrating the aligned target-side information in POS induction makes inferred tagsets more suitable for SMT.

Table 1 also shows that the independent model is more effective for SMT than the joint model. This means that sparseness is a severe problem in

Model	IND	REF
$Joint$[s+P]	164	620
Ind[s+P]	102	517
IPA POS tags	42	

Table 2: The Number of POS Tags

POS induction when jointly encoding bilingual information into observations. Additionally, all the systems using the independent model outperform BS. The improvements are statistically significant in the bootstrap method (Koehn, 2004), with a 1% significance level. The results show that the proposed models can generate more favorable POS tagsets for SMT than an existing POS tagset.

In Table 1, REFs are at least comparable to, or better than, INDs except for $Mono$. This shows that REF achieves better performance by preserving the clues from the original POS tagset. However, REF may suffer sever overfitting problem for $Mono$ since no bilingual information was incorporated. Further, when the full-level IPA POS tags[12] were used in BS, the system achieved a 27.49% BLEU score, which is worse than the result using the second-level IPA POS tags. This means that manual refinement without bilingual information may also cause an overfitting problem in MT.

5 Discussion

5.1 Comparison to the IPA POS Tagset

Table 2 shows the number of the IPA POS tags used in the experiments and the POS tags induced by the proposed models. This table shows that each induced tagset contains more POS tags than the IPA POS tagset. In the experimental data, some of Japanese verbs correspond to genuine English verbs, some are nominalized, and others correspond to English past participle verbs or present participle verbs which modify other words. Respective examples are "I $\underline{use}$ a card.", "$\underline{Using}$ the index is faster.", and "I explain $\underline{using}$ an example.", where all the underlined words correspond to the same Japanese word, "**用い**", whose IPA POS tag is a verb. Ind[s+P] in REF generated the POS tagset where the three types are assigned to separate POS groups.

The Japanese particle "**に**" is sometimes attached to nouns to give them adverb roles. For example, "相互 (mutual) **に**" is translated as the adverb "mutually" in English. Other times, it is attached to words to make them the objects of verbs. For example, "彼 (he) **に** 与える (give)" is translated as "give him". The POS tags by Ind[s+P] in REF discriminated the two types.

These examples show that the proposed models can disambiguate POS tags that have different functions in English, whereas the IPA POS tagset treats them jointly. Thus, such discrimination improves the performance of a forest-to-string SMT.

[12] 377 types of full-level IPA POS tags were included in our experimental data.

	Tagging		Dependency	
	IND	REF	IND	REF
$Original$	90.37		93.62	
$Mono$	90.75	88.04	91.77	91.51
$Joint$[s]	89.08	86.73	91.55	91.14
$Joint$[P]	80.54	79.98	91.06	91.29
$Joint$[s+P]	87.56	84.92	91.31	91.10
Ind[s]	87.62	84.33	92.06	92.58
Ind[P]	90.21	88.50	92.85	93.03
Ind[s+P]	89.57	86.12	92.96	92.78

Table 3: Tagging and Dependency Accuracy (%)

5.2 Impact of Tagging and Dependency Accuracy

The performance of our methods depends not only on the quality of the induced tag sets but also on the performance of the dependency parser learned in Step 3 of Section 4.1. We cannot directly evaluate the tagging accuracy of the parser trained through Step 3 because we do not have any data with induced POS tags other than the 10,000-sentence data gained through Step 2. Thus we split the 10,000 data into the first 9,000 data for training and the remaining 1,000 for testing, and then a dependency parser was learned in the same way as in Step 3.

Table 3 shows the results. $Original$ is the performance of the parser learned from the training data with the original POS tagset. Note that the dependency accuracies are measured on the automatically parsed dependency trees, not on the syntactically correct gold standard trees. Thus $Original$ achieved the best dependency accuracy.

In Table 3, the performance for our bilingually-induced POSs, $Joint$ and Ind, are lower than $Original$ and $Mono$. It seems performing parsing and tagging with the bilingually-induced POS tagset is too difficult when only monolingual in-

formation is available to the parser. However, our bilingually-induced POSs, except for $Joint[P]$, with the lower accuracies are more effective for SMT than the monolingually-induced POSs and the original POSs, as indicated in Table 1. The tagging accuracies for $Joint[P]$ both in IND and REF are significantly lower than the others, while the dependency accuracies do not differ significantly. The lower tagging accuracies may directly reflect the lower translation qualities for $Joint[P]$ in Table 1.

6 Conclusion

We proposed a novel method for inducing POS tags for SMT. The proposed method is a nonparametric Bayesian method, which infers hidden states (i.e., POS tags) based on observations representing not only source words themselves but also aligned target words. Our experiments showed that a more favorable POS tagset can be induced by integrating aligned information, and furthermore, the POS tagset generated by the proposed method is more effective for SMT than an existing POS tagset (the IPA POS tagset).

Even though we employed word alignment from GIZA++ with potential errors, large gains were achieved using our proposed method. We would like to investigate the influence of alignment errors in the future. In addition, we are planning to prove the effectiveness of our proposed method for language pairs other than Japanese-to-English. We are also planning to introduce our proposed method to other syntax-based SMT, such as a string-to-tree SMT and a tree-to-tree SMT.

Acknowledgments

We thank Isao Goto for helpful discussions and anonymous reviewers for valuable comments. We also thank Jun Hatori for helping us to apply his software, Corbit, to our induced POS tagsets.

References

Masayuki Asahara and Yuji Matsumoto. 2003. IPADIC User Manual. Technical report, Japan.

Matthew J. Beal, Zoubin Ghahramani, and Carl E. Rasmussen. 2001. The Infinite Hidden Markov Model. In *Advances in Neural Information Processing Systems*, pages 577–584.

Phil Blunsom and Trevor Cohn. 2011. A Hierarchical Pitman-Yor Process HMM for Unsupervised Part of Speech Induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 865–874.

Trevor Cohn and Phil Blunsom. 2009. A Bayesian Model of Syntax-Directed Tree to String Grammar Induction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 352–361.

Yuan Ding and Martha Palmer. 2005. Machine Translation Using Probabilistic Synchronous Dependency Insertion Grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 541–548.

Thomas S. Ferguson. 1973. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2007. The Infinite Tree. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 272–279.

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.

Jurgen Van Gael, Andreas Vlachos, and Zoubin Ghahramani. 2009. The infinite HMM for unsupervised PoS tagging. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, pages 678–687.

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 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. 2011. Overview of the Patent Machine Translation Task at the NTCIR-9 Workshop. In *Proceedings of the 9th NTCIR Workshop*, pages 559–578.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2011. Incremental Joint POS Tagging and Dependency Parsing in Chinese. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1216–1224.

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.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. A Syntax-Directed Translator with Extended Domain of Locality. In *Proceedings of the Workshop on*

Computationally Hard Problemsand Joint Inference in Speech and Language Processing, pages 1–8.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Human Language Technology Conference: North American Chapter of the Association for Computational Linguistics*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constrantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, pages 177–180.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395.

Taku Kudo and Yuji Matsumoto. 2002. Japanese Dependency Analysis using Cascaded Chunking. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 63–69.

Percy Liang, Slav Petrov, Michael I. Jordan, and Dan Klein. 2007. The Infinite PCFG using Hierarchical Dirichlet Processes. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 688–697.

Dekang Lin. 2004. A Path-based Transfer Model for Machine Translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 625–630.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming B*, 45(3):503–528.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616.

Yang Liu, Yajuan Lü, and Qun Liu. 2009. Improving Tree-to-Tree Translation with Packed Forests. In *Proceedings 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*, pages 558–566.

Haitao Mi and Liang Huang. 2008. Forest-based Translation Rule Extraction. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 206–214.

Haitao Mi and Qun Liu. 2010. Constituency to Dependency Translation with Forests. In *Proceedings of the 48th Annual Conference of the Association for Computational Linguistics*, pages 1433–1442.

Toshiaki Nakazawa and Sadao Kurohashi. 2012. Alignment by Bilingual Generation and Monolingual Derivation. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1963–1978.

Radford M. Neal. 2003. Slice Sampling. *Annals of Statistics*, 31:705–767.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29:19–51.

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.

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.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. In *Proceedings of the 43rd Annual Conference of the Association for Computational Linguistics*, pages 271–279.

Antti-Veikko Rosti, Bing Zhang, Spyros Matsoukas, and Richard Schwartz. 2011. Expected BLEU Training for Graphs: BBN System Description for WMT11 System Combination Task. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 159–165.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

Jayaram Sethuraman. 1994. A Constructive Definition of Dirichlet Priors. *Statistica Sinica*, 4(2):639–650.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. In *Proceedings of the 46th Annual Conference of the Association for Computational Linguistics: Human Language Technologies*, pages 577–585.

Kairit Sirts and Tanel Alumäe. 2012. A Hierarchical Dirichlet Process Model for Joint Part-of-Speech and Morphology Induction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 407–416.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Taro Watanabe and Eiichiro Sumita. 2011. Machine Translation System Combination by Confusion Forest. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1249–1257.

Taro Watanabe. 2012. Optimized Online Rank Learning for Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 253–262.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A Tree Sequence Alignment-based Tree-to-Tree Translation Model. In *Proceedings of the 46th Annual Conference of the Association for Computational Linguistics: Human Language Technologies*, pages 559–567.

Hao Zhang, Licheng Fang, Peng Xu, and Xiaoyun Wu. 2011. Binarized Forest to String Translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 19–24.