A Method for Structure Estimation of Weighted Finite-State Transducers and Its Application To Grapheme-to-Phoneme Conversion

Yotaro Kubo, Takaaki Hori, Atsushi Nakamura

NTT Communication Science Laboratories, NTT Corporation, Kyoto 619–0237, Japan
{kubo.yotaro, hori.t, nakamura.atsushi}@lab.ntt.co.jp

Abstract

Weighted finite-state transducers (WFSTs) are widely used as a fundamental data structure in several spoken language processing systems since they can provide a unified representation of many types of probabilistic models. Even though the use of accurate WFSTs is important in many spoken language systems, WFSTs are conventionally obtained by transforming probabilistic models that are not estimated in terms of WFST accuracy. Several recent techniques have enabled the direct optimization of weight parameters in WFSTs; however, these techniques do not optimize the structures of WFSTs directly. In this paper, with the goal of achieving a direct estimation of WFST structures from a dataset, we introduce a Bayesian method for structure inference. The proposed method employs the hierarchical Dirichlet process (HDP) as a prior process of generative processes of arcs in the WFSTs. Thanks to the flexibility of the HDP that enables the handling of countably infinite entities, the proposed method can potentially generate the infinite number of arcs in the WFSTs. The efficiency of the proposed method is verified by estimating WFSTs for grapheme-to-phoneme (G2P) conversion. We confirmed that the WFST obtained by the proposed method realized a compact representation of G2P conversion compared with the conventional N-gram-based G2P models.

Index Terms: Weighted finite-state transducers, Bayesian inference, grapheme-to-phoneme conversion

1. Introduction

Weighted finite-state transducers (WFSTs) have been widely used as fundamental building blocks of systems related to spoken language processing, including speech recognition, machine translation, and grapheme-to-phoneme (G2P) conversion [1, 2]. Since the probabilistic models used in such systems can be converted into WFSTs, the WFSTs in these systems are conventionally obtained by estimating the corresponding probabilistic models, such as hidden Markov models (HMMs), N-gram models, and rule-based replacement models. Thus, conventionally, the structures and weight parameters of the WFSTs are not estimated so that the estimated transducers are efficient in terms of transduction.

However, recent developments have enabled the direct optimization of the weight parameters in WFSTs. For example, Eisner introduced an expectation maximization (EM)-based learning algorithm for the WFST weights [3]. Chiang *et al.* extended the above EM-based method to a Bayesian method [4]. Lin and Yvon employed a discriminative training criterion to optimize the WFST weights [5]. These methods are basically aiming at an optimization of the arc weights under the given structure of WFSTs. Therefore, these methods have to import a WFST structure from other probabilistic models.

Optimizing the structures of WFSTs in a certain criterion is important since WFSTs are generalizations of many kinds of probabilistic distributions including HMMs and N-gram models, and able to represent a wider range of probability distributions by modifying structures. Several methods have been introduced for estimating the structures of WFSTs. Lucas and Reynolds introduced a method based on genetic algorithms to search for an optimal structure for WFSTs [6]. Rybach and Riley employed a greedy optimization with a split and merge strategy and applied to the estimation problem of triphone WFSTs [7]. However, since these methods lack a probabilistic interpretation of the estimated result, it is difficult to extend them to reflect prior knowledge about the structures of WFSTs.

In this paper, with the aim of realizing the probabilistic estimation of WFST structures, we introduce a non-parametric Bayesian method for inferring WFST structure. Since WFSTs and HMMs share several analogies, e.g. latent sequences are assumed to be generated from a first order Markov chain, the proposed method is based on hierarchical Dirichlet processes (HDPs) that are already successfully applied for HMMs [8]. By leveraging an HDP-based formulation, we can assume that WFSTs comprise a countably infinite number of arcs, and only a limited number of arcs are actually used for transducing training examples. Thanks to the sparseness introduced by using Dirichlet process priors, the estimated WFSTs are expected to learn a succinct representation from the input and output examples of WFSTs.

Furthermore, we propose a Markov chain Monte Carlo (MCMC) sampler for the Bayesian inference using the proposed model. Due to the alignment uncertainty of the input and output sequences, the MCMC sampler of the proposed model has to consider all possible arc sequences. This paper proposes a sampling algorithm based on the WFST composition to handle such arc sequences. The composition algorithms in the sampler also allow the WFST-based representations of the observed input and output sequences, which are advantageous as regards expressing uncertainty in the training dataset.

To verify the efficiency of the proposed method, we estimated WFSTs for G2P conversion. G2P conversion processes are conventionally modeled by using N-gram models of segmental pairs of graphemes and phonemes (graphones) [9]. However, N-gram models of graphones are known to be huge in terms of memory consumption and difficult to interpret so that human experts modify the behavior of the model. By applying the proposed method to G2P conversion, we show that succinct models, which can easily be modified by the experts, are obtained automatically.

2. Distribution over WFSTs

In this section, we formulate a probabilistic distribution over WFSTs based on HDPs. Thanks to the flexibility of HDPs, it is possible to design an arbitrary deep hierarchy in prior processes. However, in this paper, we introduce a simple example as a first attempt.

First, we describe the mathematical notations used to express WFSTs. WFSTs are used to represent a process of rewriting input sequences $\boldsymbol{x}$ to output sequences $\boldsymbol{y}$ by using a finite state model that consumes one or zero symbols in the input sequence and generates one or zero symbols in the output sequence on each state transition. Formally, WFST T is defined as 7-tuple as $T \equiv (\mathcal{Q}, \mathcal{X}, \mathcal{Y}, \mathcal{I}, \mathcal{F}, \mathcal{K}, \mathcal{A})$ where $\mathcal{Q}$ is a finite set of states, $\mathcal{X}$ is a set of input symbols, $\mathcal{Y}$ is a set of output symbols, $\mathcal{I} \subset \mathcal{Q}$ is a set of initial states, $\mathcal{F} \subset (\mathcal{Q} \times \mathcal{K})$ is a set of pairs of final states and final weights, $\mathcal{K}$ is a semiring used to represent transition weights, and $\mathcal{A} \subset (\mathcal{Q} \times (\mathcal{I} \cup \{\epsilon\}) \times (\mathcal{O} \cup \{\epsilon\}) \times \mathcal{Q} \times \mathcal{K})$ is a set of arcs that represent state transitions. The general formulation of a WFST allows the use of arbitrary semiring as a representation of transition weights $\mathcal{K}$. However, in the proposed method, we focused on a probability semiring since this semiring is a natural choice for representing probabilistic models by WFSTs. Therefore, $\mathcal{K}$ is defined as a set of positive real numbers $\mathcal{K} \overset{\text{def}}{=} \mathbb{R}^+$.

The objective of this study is to estimate a set of arcs $\mathcal{A}$ from a given dataset. Each arc $(p_{s,k}, i_{s,k}, o_{s,k}, q_{s,k}, \theta_{s,k}) \overset{\text{def}}{=} \pi_{s,k} \in \mathcal{A}$ represents a state transition in the WFSTs where s is an index for source states, k is an index for arcs in the state, $p_{s,k} = s$ is a corresponding source state that always equals s, $i_{s,k}$ is a symbol to be consumed during the transition where $i_{s,k} = \epsilon$ indicates that the transition does not consume any symbol, $o_{s,k}$ is a symbol to be generated during the transition (ϵ is also used as in the input case), $q_{s,k}$ is a destination state, and $\theta_{s,k}$ is the probability of the transition. $\Pi[\boldsymbol{x}, \boldsymbol{y}]$ denotes a set of sequences of states and arc indices that represent possible state transitions with a given input $\boldsymbol{x}$ and output $\boldsymbol{y}$.

With the abovementioned notation, the following generative model for the input and output sequences can be defined.

$$P(\boldsymbol{x}, \boldsymbol{y}|T) \overset{\text{def}}{=} \sum_{\boldsymbol{\pi} \in \Pi[\boldsymbol{x}, \boldsymbol{y}]} \prod_{(s',k') \in \boldsymbol{\pi}} \theta_{s',k'}. \tag{1}$$

The sum-to-one constraint of the above probability can easily be satisfied by satisfying the following state-wise sum-to-one constraint.

$$\sum_k \theta_{s,k} = 1 \ \ (\forall s). \tag{2}$$

Although this generative probability of the input and output sequence is straightforward regarding the typical applications of WFSTs, the parameter estimation based on this formulation is not simple because of the complex relation between $(\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{\pi}$ induced by, for example, ϵ-symbols. This paper tackles this problem by introducing a sampling algorithm based on the WFST composition algorithm.

By considering $\theta_{s,k}$ as mixture proportions in the Dirichlet process mixture models (DPMMs), we define a generation process of arc parameters ($i_{s,k}$, $o_{s,k}$, and $q_{s,k}$) that depend on the source state s, the base measure of arc parameters G_s, and concentration parameter α_s:

$$i, o, q|s \sim DP(G_s, \alpha_s), \tag{3}$$

where the base measure is defined as a product of a state-dependent input/output symbol distribution $G_s^{(\mathrm{IO})}$ and a destination state distribution $G_s^{(\mathrm{S})}$, as follows:

$$G_s(i, o, s') \overset{\text{def}}{=} G_s^{(\mathrm{IO})}(i, o) \times G_s^{(\mathrm{S})}(s'). \tag{4}$$

We assumed that these base measures are also generated by Dirichlet processes, as follows:

$$G_s^{(\mathrm{IO})} \sim DP(G_0^{(\mathrm{IO})}, \beta_0), \ \ G_s^{(\mathrm{S})} \sim DP(G_0^{(\mathrm{S})}, \gamma_0), \tag{5}$$

where β_0 and γ_0 are concentration parameters, and $G_0^{(\mathrm{IO})}$ and $G_0^{(\mathrm{S})}$ are base measures of base measures. The state-dependent concentration parameters α_s are assumed to be generated from a gamma distribution, as follows:

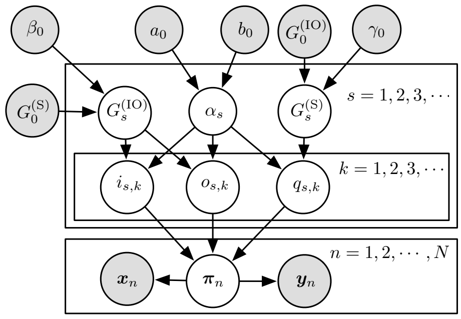

Figure 1: Plate notation of the proposed WFST model.

$$\alpha_s \sim \Gamma(a_0, b_0). \tag{6}$$

Recent advances on Bayesian methods have made it possible to introduce prior processes to $G_0^{(\mathrm{IO})}$, $G_0^{(\mathrm{S})}$, a_0, b_0, β_0, and γ_0 and to assume that these variables are random variables. However, in this paper, we fixed these values as hyperparameters to keep the formulation and algorithm simple. Fixing $G_0^{(\mathrm{S})}$ implies that the maximum number of states has to be fixed in advance; however, since the state transitions are estimated by using the dataset, the number of states can also be adjusted by deleting dead states that are not connected to other states.

Figure 1 summarizes the dependency between the variables introduced in this section by using the plate notation.

3. Markov chain Monte Carlo (MCMC) sampling

To compute the posteriors of the abovementioned distributions over WFSTs with the given examples of the input and output sequences, we developed an MCMC sampler based on a WFST framework. By adapting the Chinese restaurant process (CRP) method, we employed a method that marginalizes $\theta_{s,k}$ out, and samples arc sequences $\boldsymbol{\pi}_n$ directly.

In this section, we denote the training dataset as $\mathcal{Z} \overset{\text{def}}{=} \{(\boldsymbol{x}_n, \boldsymbol{y}_n)|n \in \{1, .., N\}\}$ where $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ are n-th input and output sequences in the dataset, respectively. The sampler introduced in this section generates samples of the posterior distribution $P(\mathcal{A}|\mathcal{Z})$ by using the Gibbs sampling method.

Here, in order to make the following MCMC procedure computationally efficient, we assume that a state is visited only once for each input and output pair. By introducing this assumption, the posterior probability of the arc sequence $\boldsymbol{\pi}_n$ used for the Gibbs sampling can be expressed similar to that of the CRPs, as follows:

$$P(\boldsymbol{\pi}_n|\mathrm{rest}) = \prod_{(s,k) \in \boldsymbol{\pi}_n} q(s, k),$$
$$q(s, k) = \begin{cases} \frac{C^{(\neg n)}(q,k)}{\alpha_s + \sum_{q',k'} C^{(\neg n)}(s',k')} & k \leq K_s^{(\neg n)}, \\ \frac{\alpha_s}{\alpha_s + \sum_{s',k'} C^{(\neg n)}(s',k')} & k = K_s^{(\neg n)} + 1, \end{cases} \tag{7}$$

where $K_s^{(\neg n)}$ is the current number of arcs from the state s, $C^{(\neg n)}(s, k)$ is the occupancy count of the arc (s, k), counted as follows:

$$\begin{aligned} K_s^{(\neg n)} &= \sum_k \mathbf{1}[C^{(\neg n)}(s,k) > 0], \\ C^{(\neg n)}(s,k) &= \sum_{n' \neq n} \sum_{(s',k') \in \boldsymbol{\pi}_{n'}} \mathbf{1}[s' = s \wedge k' = k], \end{aligned} \tag{8}$$

where $\mathbf{1}[.]$ is the indicator function that is 1 if the given predicate is true, and 0 otherwise. Although we introduced the additional constraint on state visits for defining the above posterior probability, the proposed method does not restrict the possible path in the actual sampling step. Even though the posterior probability introduced here is not accurate, we considered that the estimation bias introduced by this heuristic is small enough. Therefore, the additional rejection step is not introduced.

Unlike HDPs applied to HMMs, the arc parameters $i_{s,k}$, $o_{s,k}$, and $q_{s,k}$ for the existing $k \leq K_s^{(\neg n)}$ are not changed during the MCMC sampling processes due to the deterministic property of the emission probability of the input and output sequences (Eq. (1)). These variables are sampled from the base measures $G_s^{(\mathrm{IO})}$ and $G_s^{(\mathrm{S})}$ for each creation of new arcs and not changed until deletion.

We sampled the state-dependent base measures ($G_s^{(\mathrm{IO})}$ and $G_s^{(\mathrm{S})}$) for computational efficiency where the standard approaches marginalize these measures. The posterior for these base measures can be expressed as follows:

$$P(G^{(\mathrm{IO})}|\mathrm{rest}) = \mathrm{Dir}(\boldsymbol{\lambda}_s), \quad P(G^{(\mathrm{S})}|\mathrm{rest}) = \mathrm{Dir}(\boldsymbol{\eta}_s), \tag{9}$$

where Dir denotes the Dirichlet distribution, and the parameter vectors , $\boldsymbol{\lambda}_s = \{\lambda_{s,i',o'}\}_{i',o'}$ and $\boldsymbol{\eta}_s = \{\eta_{s,q'}\}_{q'}$, of the Dirichlet distributions can be computed as follows:

$$\begin{aligned} \lambda_{s,i',o'} =& \beta_0 G_0^{(\mathrm{IO})}(i',o') + \sum_k \mathbf{1}[i_{s,k} = i' \wedge o_{s,k} = o'], \\ \eta_{s,q'} =& \gamma_0 G_0^{(\mathrm{S})}(q') + \sum_k \mathbf{1}[q_{s,k} = q']. \end{aligned} \tag{10}$$

We can sample these base measures from the Dirichlet distribution by using the standard sampling technique. By using a concentration parameter that encourages sparse samples, i.e. $\beta_0 < 1, \gamma_0 < 1$, the sampled base measures become sparse. The sparse measures are advantageous in terms of keeping the computational efficiency high in the sampling procedures.

By incorporating the sampled base measures and the posteriors of the arc sequences (Eq. (7)), we could make weighted finite-state transducers $\Psi[T^{(\neg n)}]$ that represent all possible arc sequences including new arcs that do not appear in the current estimation of WFSTs. $\Psi[T^{(\neg n)}]$ is generated by applying the expansion operator $\Psi[.]$ to the current estimation of WFST $T^{(\neg n)}$. First, the current estimation of WFST $T^{(\neg n)}$ is constructed with reference to all observed arc sequences except for $\boldsymbol{\pi}_n$ where arc weights are given by Eq. (7). Then, the Ψ operator is applied by adding expansion arcs that represent the creation of new arcs for each possible combination of input symbol, output symbol, and destination state. The weights $\psi_{p,i,o,q}$ for the expansion arc from p to q with input and output symbols i, o are defined as follows:

$$\psi_{p,i,o,q} = \alpha_p G_p^{(\mathrm{IO})}(i,o) G_p^{(\mathrm{S})}(q). \tag{11}$$

The sparseness in the base measures ($G_p^{(\mathrm{IO})}$ and $G_p^{(\mathrm{S})}$) also encourages the sparseness of the resulting expanded WFSTs. The Ψ operation can be implemented in an on-the-fly manner for computational efficiency.

Algorithm 1 MCMC Algorithm

```
1: Input: current paths $\boldsymbol{\pi}_{(n)}$
2: loop
3:     Resample $\alpha_s$, $G_s^{(\mathrm{IO})}$ and $G_s^{(\mathrm{S})}$.
4:     Select the index of the training data $r$ randomly.
5:     Construct WFST $T^{(\neg r)}$.
6:     Obtain expanded WFST $\Phi = \Psi[T^{(\neg r)}]$.
7:     Sample $\hat{\pi}_r$ from the lattice $\boldsymbol{x}_r \circ \Phi \circ \boldsymbol{y}_r$.
8:     Create new arc according to expansion arcs in $\hat{\pi}$.
9: end loop
```

By using the expanded WFSTs, the posterior Φ of the arc sequences can be denoted by using the WFST composition operator ($\circ$), as follows:

$$\Phi = \boldsymbol{x} \circ \Psi[T^{(\neg n)}] \circ \boldsymbol{y}. \tag{12}$$

If we assume $G_0^{(\mathrm{IO})}(\epsilon,\epsilon) = 0$, Φ can be represented as a lattice, and the arc sequence $\boldsymbol{\pi}_n$ can be sampled by performing forward-backward sampling on the lattice. For each appearance of expansion arcs that are generated by the expansion operator, we create a new arc where the input and output symbols, and destination state are copied from these variables corresponding to the expansion arc. α_s is sampled by using the method described in [10]. Since β_0 and γ_0 are important for controlling the sparseness in the expanded WFST and the computational efficiency of the MCMC processes, these two concentration parameters are fixed at small values (0.1 is used in the following experiments).

Algorithm 1 summarizes the proposed MCMC sampling procedure. It should be noted that it is not necessary to sample (α_s, $G_s^{(\mathrm{IO})}$ and $G_s^{(\mathrm{S})}$; Line 3) every iteration.

4. Experiments

In this section, we describe two types of experiments that we conducted to verify the proposed method.

4.1. Synthetic data experiments

To verify the validity of the proposed method, an experiment was conducted based on a synthetic data.

In the experiments, we used a dataset that included 100 samples generated from the ideal WFST (Figure 2 (a)). The ideal WFST was designed to transduce the input alphabets {a, b, c} into the output alphabets {x, y} by replacing each appearance of "ab" and "abc" with "x" and "y", respectively. In the experiment, we assumed that we knew the number of states in the ideal WFST. Therefore, we used a simple uniform distribution $G_0^{(\mathrm{S})}(s) = 1/3$ as the base measure for the destination states. The base measure for the symbol pairs $G_0^{(\mathrm{IO})}$ is defined as follows.

$$G_0^{(\mathrm{IO})}(i,o) = P(i)P(o). \tag{13}$$

Here, $P(i)$ and $P(o)$ were estimated by counting the appearance of the alphabets where the counts of the epsilon symbols were defined by computing the differences in the length of the input and output sequences. As a preliminary experiment, β_0 and γ_0 is fixed to 0.1 to ensure a sparseness of the expanded WFST $\Psi[T^{(\neg n)}]$ during the sampling process. The shape parameter $a_0 = 1.0$ and the rate parameter $b_0 = 3.0$ are used in Eq. (6) to encourage sparseness in the number of arcs.

By performing the MCMC sampling process, we obtained a WFST as in Figure 2 (b). We found that the proposed method can estimate a similar WFST to the ideal WFST. Since the base measure that we estimated by the above procedure yielded $P(i = \epsilon) = 0$, the arc that outputs "x" was altered to include

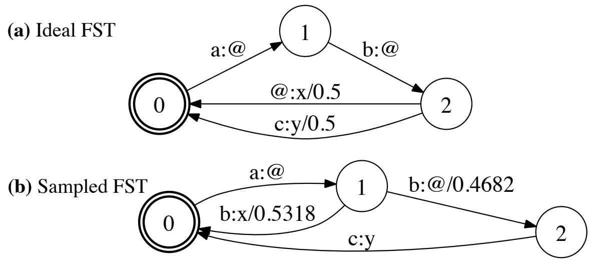

Figure 2: WFSTs (a) used to generate data, and (b) estimated from the generated data ("@" denotes the epsilon symbol).

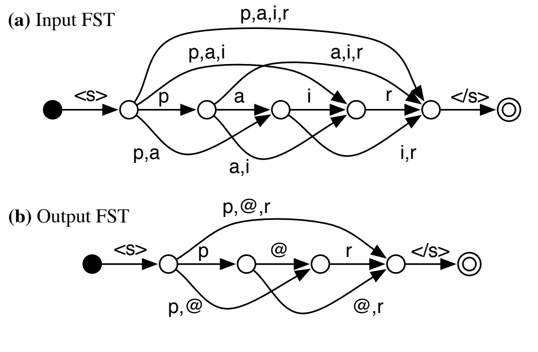

Figure 3: The input and output WFSTs corresponding to the word "pair".

the input symbol b in the estimated WFST. However, these two WFSTs can be considered equivalent if the weights terms are omitted. The weights estimated by the proposed method are also similar to the ideal patterns. Thus, we verified that the proposed method could estimate the correct WFST structure if sufficient data were given.

4.2. Grapheme-to-phoneme conversion

To confirm the efficiency of WFSTs obtained by the proposed method, we used the method to estimate G2P transducers.

In the G2P experiments, we used the "nettalk15k" corpus for training and evaluation. The training set in the corpus contains 15,006 pairs of grapheme and phoneme sequences, and the evaluation set contains 5,006 pairs. To represent multigram segments in both grapheme and phoneme sequences, we constructed the input and output WFST ($\boldsymbol{x}$ and $\boldsymbol{y}$) so that these FSTs represent all possible multigram sequences. The input and output WFSTs for the word "pair" are depicted in Figure 3.

The base measure for the symbol pairs $G_0^{(\mathrm{IO})}$ is defined as in Eq. (13) where $P(i)$ and $P(o)$ were directly estimated from the dataset $\mathcal{Z}$ by accumulating the symbol appearance in the input and output sequences; therefore, we did not use ϵ-symbols in the G2P experiments. As the base measure for the destination states, we used a simple uniform distribution $G_0^{(\mathrm{S})} = 1/\bar{S}$ by fixing the maximum number of states at $\bar{S}$. The other hyperparameters were set as in the previous experiments.

Furthermore, to verify compatibility with the discriminative training method for WFSTs such as [5], we also employed maximum mutual information (MMI) optimization for the WFST weights under the estimated WFST structure. The MMI optimization was performed by maximizing the following objective function

$$F(\Theta) = \sum_n \log P(\boldsymbol{y}_n|\boldsymbol{x}_n, \Theta), \tag{14}$$

where $\Theta \overset{\mathrm{def}}{=} \{\theta_{s,k}|\forall s, \forall k\}$ is the set of weight parameters.

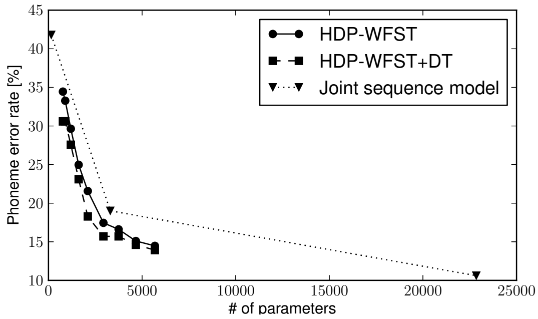

Figure 4: Phoneme error rates as functions of the numbers of parameters

Since MMI training generally requires a held out validation set, we sampled 5% of words randomly, and used them as a held out validation set.

Figure 4 shows the error rates of the phoneme-to-grapheme conversion where "HDP-WFST" denotes the proposed method and "HDP-WFST + DT" denotes the proposed method followed by the discriminative training of the weight parameters (Eq. (14)). The number of parameters in the proposed method was controlled by varying the number of maximum states as $\bar{S} = 1, 2, 4, 8, 16, 32, 64, 128, 256$. We compared the proposed method with the current standard G2P technique, which is called a joint sequence model [9]. The experimental results indicate that the proposed method is advantageous for obtaining succinct transduction models in terms of the number of parameters. However, we also confirmed that a large number of parameters cannot currently be leveraged owing to the computational inefficiency of MCMC sampling. We also confirmed that the efficiency of the proposed method can be improved by employing discriminative training. Thus, it is suggested that the WFST structures estimated by the proposed method did not lose discriminative information even though compact representations were obtained.

Furthermore, we verified that the N-gram approach was still applicable if accuracy was more important than succinctness. To apply the N-gram model, we consider the obtained WFST structure to be $P(\boldsymbol{x}|\boldsymbol{y})$. By composing a phoneme N-gram WFST $P(\boldsymbol{y})$ for this $P(\boldsymbol{x}|\boldsymbol{y})$, we obtained a WFST that derived the rich structures of the N-gram model. The $P(\boldsymbol{x}, \boldsymbol{y})$ obtained by this procedure followed by discriminative training achieved a phoneme error rate of 9.0%, which is comparable to the current state-of-the-art G2P convertor.

5. Conclusion

In this paper, we introduced a Bayesian generative model for weighted finite-state transducers (WFSTs) by adapting the hierarchical Dirichlet process. By running the proposed Markov chain Monte Carlo (MCMC) sampler of the proposed model, a Bayesian inference of WFST structures is realized. Furthermore, we introduced several techniques that accelerate the proposed MCMC sampler.

Future work will include a computationally efficient method for applying approximated inference to WFSTs. For example, the variational Bayesian method is a promising approach for the distributed computing of inference [11]. In this paper, we assumed that the sum-to-one constraints of arc weights are satisfied in all states; however, the general formulation of WFSTs allows arc weights that are not normalized. Therefore, extending the proposed method by considering unnormalized weight parameters would also be promising.

6. References

[1] D. Caseiro, L. Trancoso, L. Oliveira, and C. Viana, "Grapheme-to-phone using finite-state transducers," in *Proc. of IEEE Workshop on Speech Synthesis*, 2002, pp. 215–218.

[2] J. Novak, P. Dixon, N. Minematsu, K. Hirose, C. Hori, and H. Kashioka, "Improving WFST-based G2P conversion with alignment constraints and RNNLM N-best rescoring," in *Proc. of Interspeech*, 2012.

[3] J. Eisner, "Expectation semirings: Flexible EM for learning finite-state transducers," in *Proc. ESSLLI workshop on finite-state methods in NLP*, 2001.

[4] D. Chiang, J. Graehl, K. Knight, A. Pauls, and S. Ravi, "Bayesian inference for finite-state transducers," in *Proc. HLT-NAACL*, 2010, pp. 447–455.

[5] S.-S. Lin and F. Yvon, "Discriminative training of finite state decoding graphs," in *Proc. Interspeech*, 2005, pp. 733–736.

[6] S. M. Lucas and T. J. Reynolds, "Learning finite-state transducers: Evolution versus heuristic state merging," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 3, pp. 308–325, 2007.

[7] D. Rybach and M. Riley, "Direct construction of compact context-dependency transducers from data," in *Proc. of INTERSPEECH*, 2010, pp. 218–221.

[8] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[9] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[10] M. West, *Hyperparameter estimation in Dirichlet process mixture models*. Duke University, 1992.

[11] Y. W. Teh, K. Kurihara, and M. Welling, "Collapsed variational inference for HDP," *Advances in Neural Information Processing Systems*, vol. 20, no. 20, pp. 1481–1488, 2008.