Hierarchical Latent Words Language Models for Robust Modeling to Out-Of Domain Tasks

Ryo Masumura†‡**, Taichi Asami**†**, Takanobu Oba**†**,**
Hirokazu Masataki†**, Sumitaka Sakauchi**†**, Akinori Ito**‡
† NTT Media Intelligence Laboratories, NTT Corporation, Japan
‡ Graduate School of Engineering, Tohoku University, Japan
†{masumura.ryo, asami.taichi, oba.takanobu, masataki.hirokazu, sakauchi.sumitaka} @lab.ntt.co.jp, ‡ aito@spcom.ecei.tohoku.ac.jp

Abstract

This paper focuses on language modeling with adequate robustness to support different domain tasks. To this end, we propose a hierarchical latent word language model (h-LWLM). The proposed model can be regarded as a generalized form of the standard LWLMs. The key advance is introducing a multiple latent variable space with hierarchical structure. The structure can flexibly take account of linguistic phenomena not present in the training data. This paper details the definition as well as a training method based on layer-wise inference and a practical usage in natural language processing tasks with an approximation technique. Experiments on speech recognition show the effectiveness of h-LWLM in out-of domain tasks.

1 Introduction

Language models (LMs) are essential for automatic speech recognition or statistical machine translation (Rosenfeld, 2000). The performance of LMs strongly depends on quality and quantity of their training data. Superior performance is usually obtained by using enormous domain-matched training data sets to construct LMs (Brants et al., 2007). Unfortunately, in many cases, large amounts of domain-matched training data sets are not available. Therefore, LM technology that can robustly work for domains that differ from that of the training data is needed (Goodman, 2001).

For robust language modeling, several technologies have been proposed. Fundamental techniques are smoothing (Chen and Goodman, 1999) and clustering (Brown et al., 1992). Other solutions are Bayesian modeling (Teh, 2006) and ensemble modeling (Xu and Jelinek, 2004; Emami and Jelinek, 2005). Moreover, continuous representation of words in neural network LMs can also support robust modeling (Bengio et al., 2003; Mikolov et al., 2010). However, previous works are focused on maximizing performance in the same domain as that of the training data. In other words, it is uncertain that these technologies robustly support out-of domain tasks.

In contrast, latent words LMs (LWLMs) (Deschacht et al., 2012) are clearly effective for out-of domain tasks. We employed the LWLM to speech recognition and the resulting performance was significantly superior in out-of domain tasks while the performance was comparable in domain-matched task to conventional LMs (Masumura et al., 2013a; Masumura et al., 2013b). LWLMs are generative models that employ a latent word space. The latent space can flexibly take into account relationships between words and the modeling helps to efficiently increase the robustness to out-of domain tasks (Sec. 2).

In this paper, we focus on LWLMs and aim to make them more flexible for greater robustness to out-of domain tasks. To this end, this paper takes note of a fact that standard LWLM simply represents the latent space as n-gram model of latent words. However, function and meaning of words are essentially hierarchical and upper layers ought to be useful to increase the robustness to out-of domain tasks. The conventional LWLMs do not model the hierarchy, while the latent words are used to represent function and meaning of words. Thus, we tried to model the hierarchy in the latent space by estimating a latent word of a latent word recursively.

This paper proposes a novel LWLM with multiple latent word spaces that are hierarchically structured; we call it the hierarchical LWLM (h-LWLM). The proposed model can be regarded as a generalized form of the standard LWLMs. The hierarchical structure can take into account the abstraction process of function and meaning of words. Therefore, it can be expected that h-

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1896–1901,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

LWLMs flexibly calculate generative probability for unseen words unlike non-hierarchical LWLMs. To create the hierarchical latent word structure from training data sets, we also propose a layer-wise inference. The inference is inspired by a deep Boltzmann machine (Salakhutdinov and Hinton, 2009) that stacks up restricted Boltzmann machines (Hinton et al., 2006). In addition, we detail an n-gram approximation technique to apply the proposed model to practical natural language processing tasks (see Sec. 3).

In experiments, we construct LMs from spontaneous lecture task data and apply them to a contact center dialogue task and a voice mail task as out-of domain tasks. The effectiveness of the proposed method is shown by perplexity and speech recognition evaluation (Sec. 4).

2 Latent Words Language Models

LWLMs are generative models with single latent word space (Deschacht et al., 2012). The latent word is represented as a specific word that is selected from the entire vocabulary. Thus, the number of latent words equals the number of observed words.

Bayesian modeling of LWLM produces the generative probability of observed word sequence $\boldsymbol{w} = w_1, \cdots, w_K$ as:

$$P(\boldsymbol{w}) = \int_{\boldsymbol{\theta}} \prod_{k=1}^{K} \sum_{h_k} P(w_k|h_k, \boldsymbol{\theta}) P(h_k|\boldsymbol{l}_k, \boldsymbol{\theta}) P(\boldsymbol{\theta}) d\boldsymbol{\theta}, \quad (1)$$

where $\boldsymbol{\theta}$ indicates a model parameter of the LWLM, $\boldsymbol{h} = h_1, \cdots, h_K$ denotes a latent word sequence and $\boldsymbol{l}_k$ denotes context latent words $h_{k-n+1}, \cdots, h_{k-1}$. $P(h_k|\boldsymbol{l}_k, \boldsymbol{\theta})$ represents the transition probability which can be expressed by an n-gram model for latent words, and $P(w_k|h_k, \boldsymbol{\theta})$ represents the emission probability that models the dependency between the observed word and the latent word. More details are shown in previous works (Deschacht et al., 2012; Masumura et al., 2013a; Masumura et al., 2013b).

3 Hierarchical LWLMs

3.1 Definition

This paper introduces h-LWLM. The proposed model has multiple latent word spaces in a hierarchical structure. Thus, it assumes that there is a latent word behind a latent word. The proposed model can be regarded as a generalized form of the standard LWLM. Thus, standard LWLMs correspond to h-LWLMs with just one layer. The latent words in all layers are represented as a specific word that is selected from the entire vocabulary.

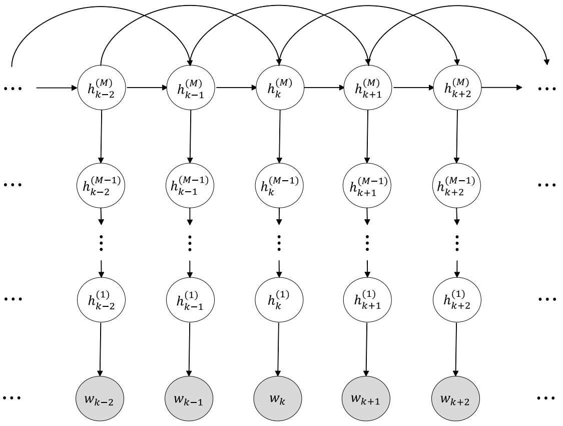

Figure 1: Graphical representation of h-LWLM.

A graphic rendering of h-LWLM is shown in Figure 1. In a generative process of the h-LWLM, a latent word in the highest layer is first generated depending on its context latent words. Next, a latent word in a lower layer is recursively generated depending on the latent word in the upper layer. Finally, an observed word is generated depending on the latent word in the lowest layer.

Bayesian modeling of h-LWLM produces the following generative probability:

$$P(\boldsymbol{w}) = \int_{\boldsymbol{\Theta}} \prod_{k=1}^{K} \sum_{h_k^{(1)}} \cdots \sum_{h_k^{(M)}} P(w_k|h_k^{(1)}, \boldsymbol{\Theta}) \cdots P(h_k^{(M-1)}|h_k^{(M)}, \boldsymbol{\Theta}) P(h_k^{(M)}|\boldsymbol{l}_k^{(M)}, \boldsymbol{\Theta}) P(\boldsymbol{\Theta}) d\boldsymbol{\Theta}, \quad (2)$$

where M is the number of layers and $\boldsymbol{\Theta}$ indicates a model parameter of h-LWLM. $\boldsymbol{h}^{(m)} = h_1^{(m)}, \cdots, h_K^{(m)}$ denotes a latent word sequence in the m-th layer. $P(h_k^{(M)}|\boldsymbol{l}_k^{(M)}, \boldsymbol{\Theta})$ represents the transition probability which is expressed by n-gram model for latent words in the highest layer. $P(h_k^{(m)}|h_k^{(m+1)}, \boldsymbol{\Theta})$ and $P(w_k|h_k^{(1)}, \boldsymbol{\Theta})$ represent the emission probabilities that respectively model the dependency between latent words in two layers and the dependency between the observed word and the latent word in the lowest layer.

As the integral with respect to $\boldsymbol{\Theta}$ is analytically

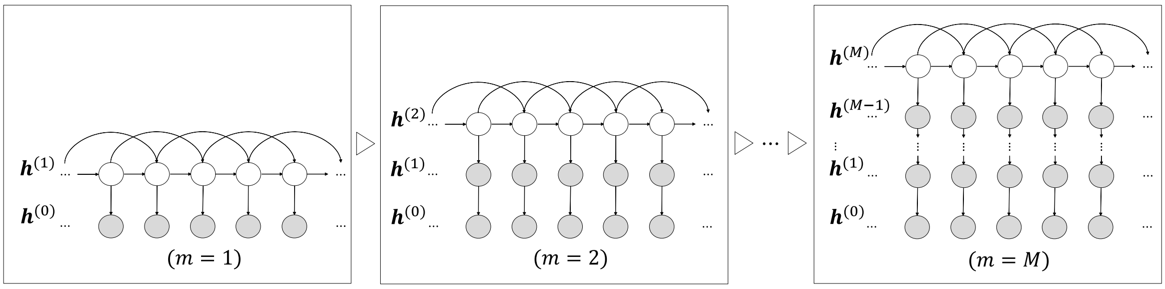

Figure 2: Layer-wise inference procedure.

Algorithm 1 :
Inference procedure for h-LWLM.

```
Input: Training data $\boldsymbol{w}$, number of instances $T$,
    number of layers $M$
Output: Model parameters $\boldsymbol{\Theta}_1, \cdots, \boldsymbol{\Theta}_T$
1: for $t = 1$ to $T$ do
2:     $\boldsymbol{h}^{(0)} = \boldsymbol{w}$
3:     for $m = 1$ to $M$ do
4:         $\boldsymbol{\theta}^{(m)}, \boldsymbol{h}^{(m)} \sim P(\boldsymbol{h}^{(m)}|\boldsymbol{h}^{(m-1)}, \boldsymbol{\theta}^{(m)})$
5:     end for
6:     $\boldsymbol{\Theta}_t = \boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(M)}$
7: end for
8: return $\boldsymbol{\Theta}_1, \cdots, \boldsymbol{\Theta}_T$
```

intractable, the equation can be approximated as:

$$P(\boldsymbol{w}) = \frac{1}{T} \prod_{k=1}^{K} \sum_{t=1}^{T} \sum_{h_k^{(1)}} \cdots \sum_{h_k^{(M)}} P(w_k|h_k^{(1)}, \boldsymbol{\Theta}_t) \cdots P(h_k^{(M-1)}|h_k^{(M)}, \boldsymbol{\Theta}_t) P(h_k^{(M)}|\boldsymbol{l}_k^{(M)}, \boldsymbol{\Theta}_t). \quad (3)$$

The probability distribution can be approximated using T instances of point estimated parameter; $\boldsymbol{\Theta}_t$ indicates the t-th point estimated parameter.

3.2 Parameter Inference

This paper proposes a layer-wise inference procedure for constructing h-LWLMs from training data. The detailed procedure is shown in **Algorithm 1**, and Figure 2 shows an image representation of the procedure as increased with the number of layers. In the procedure, LWLM structure is recursively accumulated by estimating a latent word sequence in an upper layer from a latent word sequence in the lower layer.

Line 4 in **Algorithm 1** denotes the key procedure of estimating a latent word sequence in an upper layer from a latent word sequence in the lower layer. $\boldsymbol{\theta}^{(m)}$ denotes model parameter of LWLM structure in m-th layer; it can be defined from both $\boldsymbol{h}^{(m)}$ and $\boldsymbol{h}^{(m-1)}$. For the inference of $\boldsymbol{h}^{(m)}$ from $\boldsymbol{h}^{(m-1)}$, Gibbs sampling is suitable (Casella and George, 1992; Robert et al., 1993; Scott, 2002). Gibbs sampling picks a new value for $h_k^{(m)}$ according to its probability distribution which is estimated from both $\boldsymbol{h}_{-k}^{(m)}$ and $\boldsymbol{h}^{(m-1)}$. $\boldsymbol{h}_{-k}^{(m)}$ represents all latent words in the m-th layer except for $h_k^{(m)}$. The probability distribution is given by:

$$P(h_k^{(m)}|\boldsymbol{h}_{-k}^{(m)}, \boldsymbol{h}^{(m-1)}, \boldsymbol{\theta}^{(m)}) \propto P(h_k^{(m-1)}|h_k^{(m)}, \boldsymbol{\theta}^{(m)}) \prod_{j=k}^{k+n-1} P(h_j^{(m)}|\boldsymbol{l}_j^{(m)}, \boldsymbol{\theta}^{(m)}). \quad (4)$$

For the inference, the prior distribution is necessary for each probability distribution. Usually, a hierarchical Pitman-Yor prior (Teh, 2006) is used for each transition probability and a Dirichlet prior (MacKay and Peto, 1994) is used for each emission probability.

As shown in line 6, t-th point estimated parameter $\boldsymbol{\Theta}_t$ indicates parameters of each LWLM for all layers in t-th iteration. The transition probabilities except for M-th layer are only used in the layer-wise inference procedure.

3.3 Usage

It is impractical to directly apply the h-LWLM to natural language processing tasks since the proposed model has multiple latent word spaces and we have to consider all possible latent word assignment for calculating generative probabilities. Therefore, this paper introduces an n-gram approximation technique as well as that for standard LWLM (Masumura et al., 2013a).

Algorithm 2 :
Random sampling for trained h-LWLM.

```
Input: Model parameters $\mathbf{\Theta}_1, \cdots, \mathbf{\Theta}_T$,
    number of sampled words $K$
Output: Sampled data $\boldsymbol{w}$
1: for $k = 1$ to $K$ do
2:     $\mathbf{\Theta}_t \sim P(\mathbf{\Theta}_t) = \frac{1}{T}$
3:     $h_k^{(M)} \sim P(h_k^{(M)}|\boldsymbol{l}_k^{(M)}, \mathbf{\Theta}_t)$
4:     for $m = M - 1$ to 1 do
5:         $h_k^{(m)} \sim P(h_k^{(m)}|h_k^{(m+1)}, \mathbf{\Theta}_t)$
6:     end for
7:     $w_k \sim P(w_k|h_k^{(1)}, \mathbf{\Theta}_t)$
8: end for
9: return $\boldsymbol{w} = w_1, \cdots, w_K$
```

The n-gram approximation is conducted as following steps. First, a lot of text data that permit h-LWLMs to be approximated by n-gram structure is generated by random sampling using trained h-LWLM. Next, an n-gram model is constructed from the generated data. The random sampling is based on **Algorithm 2**. The sampled data $\boldsymbol{w}$ in line 9 is only used for n-gram model estimation.

4 Experiments

4.1 Experimental Conditions

Our basic assumption is domain-matched training data is not available. Thus, for LM training, we used the Corpus of Spontaneous Japanese (CSJ) whose domain is a spontaneous lecture task (Maekawa et al., 2000). We divided CSJ into a training set and a small validation set (Valid). The validation set was used for optimizing several hyper parameters of LMs. For evaluation, a contact center dialogue task (Test 1) and a voice mail task (Test 2) were prepared. In contact center dialogue task, two speakers, an operator and a customer, talked to each other as in call center dialogues. 24 phone calls (24 operator channels and 24 customer channels) were used in the evaluation. In the voice mail task, a person spoke small voice messages using a smart phone. 237 messages are used in the evaluation. The training data had about 7M words, the vocabulary size was about 80K. The validation data size and test data size (both tasks) were about 20K words.

For speech recognition evaluation, we prepared an acoustic model based on hidden Markov models with deep neural networks (DNN-HMM) (Hinton et al., 2012). The DNN-HMM had 8 hidden layers with 2048 nodes. The speech recognizer was a weighted finite state transducer (WFST) decoder (Mohri et al., 2001; Hori et al., 2007).

As a baseline, 3-gram LM with interpolated Kneser-Ney smoothing (MKN) (Kneser and Ney, 1995) and 3-gram hierarchical Pitman-Yor LM (HPY) (Huang and Yor, 2007) were constructed from the training data. We also trained a class-based recurrent neural network LM with 500 hidden nodes and 500 classes (RNN) for comparison to state-of-the art language modeling (Mikolov et al., 2011). In addition, we constructed 3-gram standard LWLM and 3-gram h-LWLMs (LW). LW with 1 layer represents standard LWLM, and LW with 2-5 layers represent proposed h-LWLMs. The number of instances was set to 10 for each LW. For their n-gram approximation, we generated one billion words and approximated each as a 3-gram HPYLM. Moreover, we constructed interpolated model with LW and HPY (LW+HPY).

4.2 Results

Figure 3 shows the relation between number of layers in h-LWLM and perplexity (PPL) reduction for each condition. In addition, Table 1 shows speech recognition results in terms of word error rate (WER) for each condition. RNN was only tested in PPL evaluation as RNN cannot be converted into WFST format.

For the validation set (same domain as that of training set), PPL was not improved by the hierarchical structure in LW. LW is comparable to MKN and HPY, and inferior to RNN in terms of PPL. On the other hand, in test sets (out-of domain tasks), PPL improved with the increase in the number of layers in LW. LW with 5 layers was superior to 1 layer in terms of PPL and WER. The best results were obtained by LW+HPY with 5 layers. In fact, when we generated one billion words using a trained LWLM or trained h-LWLM, the number of observed trigrams in h-LWLM with 5 layers was 101M while the number of observed trigrams in non-hierarchical LWLM was 94M. Thus, h-LWLM can generate unseen words unlike non-hierarchical LWLM. Moreover, trigram coverage in each test data slightly increased with number of layers. These results show that h-LWLM with multiple layers offers robust performance not possible with other models while its performance in the same domain as that of training data was not improved. As a result, LW+HPY with 5 layers

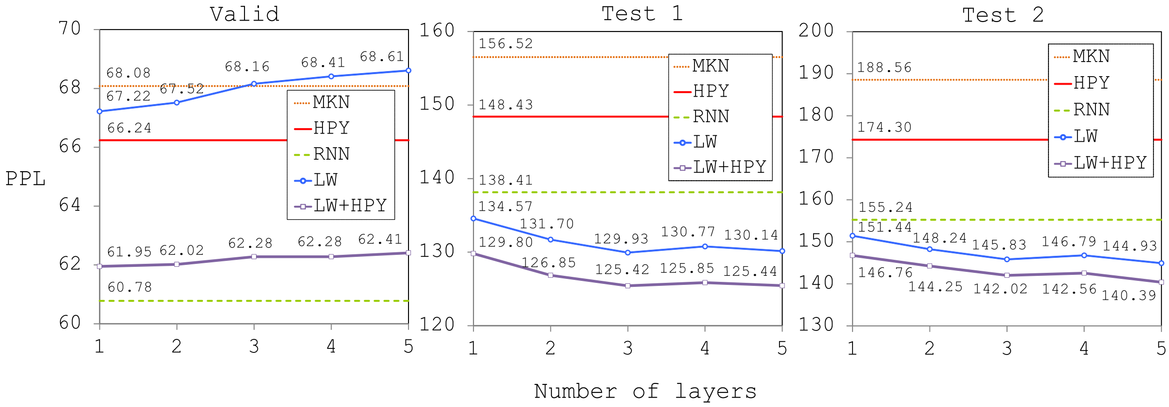

Figure 3: Perplexity (PPL) results.

Setup	Layer	Valid	Test 1	Test 2
MKN	-	24.79	38.67	32.31
HPY	-	24.67	38.29	32.00
LW	1	24.54	36.93	30.26
LW	5	24.60	36.49	29.57
LW+HPY	1	**23.62**	36.49	29.76
LW+HPY	5	23.68	**36.03**	**29.21**

Table 1: Word error rate (WER) results (%).

performed significantly better than MKN, HPY and RNN in the out-of domain tasks.

5 Conclusions

This paper proposed h-LWLM for robust modeling and detailed its definition, inference procedure, and approximation method. The proposed model has a hierarchical latent word space and it can flexibly handle linguistic phenomena not present in the training data. Our experiments showed that h-LWLM offers improved robustness to out-of domain tasks; h-LWLM is also superior to standard LWLM in terms of PPL and WER. Furthermore, our approach is significantly superior to the conventional n-gram models or the recurrent neural network LM in out-of domain tasks.

References

Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Thorsten Brants, AShok C. Popat, Peng Xu, Ftanz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. *In Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 858–867.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

George Casella and Edward I George. 1992. Explaining the Gibbs sampler. *The American Statistician*, 46:167–174.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13:359–383.

Koen Deschacht, Jan De Belder, and Marie-Francine Moens. 2012. The latent words language model. *Computer Speech & Language*, 26:384–409.

Ahmad Emami and Frederick Jelinek. 2005. Random clusterings for language modeling. *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1:581–584.

Joshua T. Goodman. 2001. A bit of progress in language modeling. *Computer Speech & Language*, 15:403–434.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep bilief nets. *Neural Computation*, 18:1527–1554.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. 2012. Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine*, pages 1–27.

Takaaki Hori, Chiori Hori, Yasuhiro Minami, and Atsushi Nakamura. 2007. Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1352–1365.

Songfang Huang and Marc Yor. 2007. Hierarchical Pitman-Yor language models for ASR in meetings. *In Proc IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 124–129.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1:181–184.

David J. C. MacKay and Linda C. Peto. 1994. A hierarchical Dirichlet language model. *Natural language engineering*, 1:289–308.

Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, and Hitoshi Isahara. 2000. Spontaneous speech corpus of Japanese. *In Proc. International Conference on Language Resources and Evaluation (LREC)*, pages 947–952.

Ryo Masumura, Hirokazu Masataki, Takanobu Oba, Osamu Yoshioka, and Satoshi Takahashi. 2013a. Use of latent words language models in ASR: a sampling-based implementation. *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8445–8449.

Ryo Masumura, Takanobu Oba, Hirokazu Masataki, Osamu Yoshioka, and Satoshi Takahashi. 2013b. Viterbi decoding for latent words language models using Gibbs sampling. *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 3429–3433.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1045–1048.

Tomas Mikolov, Stefan Kombrink Stefan, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2001. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16:69–88.

Christian P. Robert, Gilles Celeux, and Jean Diebolt. 1993. Bayesian estimation of hidden Markov chains: A stochastic implementation. *Statistics & Probability Letters*, 16:77–83.

Ronald Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here? *In Proc. IEEE*, 88:1270–1278.

Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Deep Boltzmann machines. *In Proc. the International Conference on Artificial Intelligence and Statistics*, 5:448–455.

Steven L. Scott. 2002. Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97:337–351.

Yee Whye Teh. 2006. A hierarchical bayesian language model based on Pitman-Yor processes. *In Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 985–992.

Peng Xu and Frederick Jelinek. 2004. Random forests in language modeling. *In Proc. Empirical Methods on Natural Language Processing (EMNLP)*, pages 325–332.