THE SHARED DIRICHLET PRIORS FOR BAYESIAN LANGUAGE MODELING

Jen-Tzung Chien

Department of Electrical and Computer Engineering
National Chiao Tung University, Hsinchu, Taiwan 30010, ROC

ABSTRACT

We present a new full Bayesian approach for language modeling based on the shared Dirichlet priors. This model is constructed by introducing the Dirichlet distribution to represent the uncertainty of n-gram parameters in training phase as well as in test time. Given a set of training data, the marginal likelihood over n-gram probabilities is illustrated in a form of linearly-interpolated n-grams. The hyperparameters in Dirichlet distributions are interpreted as the prior backoff information which is shared for the group of n-gram histories. This study estimates the shared hyperparameters by maximizing the marginal distribution of n-gram given the training data. Such Bayesian language model is connected to the smoothed language model. Experimental results show the superiority of the proposed method to the other methods in terms of perplexity and word error rate.

***Index Terms*—** Bayesian learning, language model, model smoothing, optimal hyperparameter

1. INTRODUCTION

Statistical language model (LM) plays an important role in many signal and information processing systems including machine translation, document classification, writing correction, bio-informatics, and speech recognition. LM based on n-gram aims to calculate the probability of a word string by multiplying the probabilities of a word w_i conditional on its preceding $n-1$ words $w_{i-n+1}^{i-1} = \{w_{i-n+1}, \cdots, w_{i-1}\}$. Traditionally, the maximum likelihood (ML) is applied to estimate n-gram parameters $\boldsymbol{\theta}^{\mathsf{ML}} = \{p_{\mathsf{ML}}(w_i|w_{i-n+1}^{i-1})\}$. In general, the weaknesses of ML n-gram model are twofold [1, 2]: (1) insufficient training data for estimation of n-grams for so many word combinations in w_{i-n+1}^{i}, and (2) lack of long-distance information due to the n-gram window. In [3, 4], the modified Kneser-Ney LM (MKN-LM) was proposed to tackle the first weakness by backoff smoothing or interpolation smoothing based on the lower-order model. LM smoothing could be also implemented through the hierarchical Pitman-Yor language model [5, 6]. To deal with the second weakness, the recurrent neural network LM (RNN-LM) [7, 8] and the cache Dirichlet class LM (cDC-LM) [9] were constructed by combining the information from the recurrent hidden states or through the large-span topics [10], respectively.

Basically, the frequency estimator in higher-order LM has large variance, because there are so many possible word combinations in n-gram event $w_{i-n+1}^{i} = \{w_{i-n+1}^{i-1}, w_i\}$ that only a small fraction of them have been observed in the data. A simple linear interpolation scheme for an n-gram is performed by interpolating ML higher and lower models [11]

$$\lambda p_{\mathsf{ML}}(w_i|w_{i-n+1}^{i-1}) + (1-\lambda) p_{\mathsf{ML}}(w_i|w_{i-n+2}^{i-1}) \qquad (1)$$

where $0 \leq \lambda \leq 1$ denotes the interpolation weight which was determined for individual history w_{i-n+1}^{i-1} from validation data. In this study, we compensate for the variations in the estimated n-gram parameters due to insufficient training data accordance to the Bayesian framework [12, 13]. The uncertainty of n-gram parameters $\boldsymbol{\theta} = \{p(w_i|w_{i-n+1}^{i-1})\}$ is characterized by a Dirichlet prior with the shared hyperparameters $\boldsymbol{\alpha}$. The marginal n-gram over a Dirichlet prior of n-gram parameters is constructed [14]. The optimal hyperparameters are estimated by maximizing the marginal likelihood. This Bayesian LM can be interpreted as the smoothed n-gram. Different from the heuristic LM smoothing [3, 4], a *full Bayesian* language model is realized to carry out interpolation smoothing for LM without the need of validation data. A set of experiments are conducted to illustrate the performance of LM smoothing based on the Bayesian LM.

2. BAYESIAN LEARNING FOR N-GRAMS

2.1. Dirichlet prior and posterior

N-gram model parameters $\boldsymbol{\theta} = \{\theta_{i|h}\} = \{p(w_i|w_{i-n+1}^{i-1})\}$ are known as the multinomial parameters which are used to predict n-gram events of word w_i appearing after history words $h = w_{i-n+1}^{i-1}$. Assuming there are $\mathcal{V}$ multinomial parameters in $\boldsymbol{\theta} = \{\theta_{i|h}\}$ for $\mathcal{V}$ vocabulary words where $0 \leq \theta_{i|h} \leq 1$ and $\sum_{i=1}^{\mathcal{V}} \theta_{i|h} = 1$. We are interested in Bayesian language model where the prior density of multinomial parameters is introduced. The conjugate prior over multinomial parameters $\boldsymbol{\theta}$ is specified by a Dirichlet prior with hyperparameters $\boldsymbol{\alpha} = \{\alpha_{i|h}\}$ in a form of

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \mathrm{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{Z(\boldsymbol{\alpha})} \prod_h \prod_{i=1}^{\mathcal{V}} (\theta_{i|h})^{\alpha_{i|h}-1} \tag{2}$$

where the normalization term is expressed by

$$Z(\boldsymbol{\alpha}) = \prod_h \left[\frac{\prod_{i=1}^{\mathcal{V}} \Gamma(\alpha_{i|h})}{\Gamma(\sum_{i=1}^{\mathcal{V}} \alpha_{i|h})} \right]. \tag{3}$$

Dirichlet distribution was used to represent the multinomial topics for document modeling [15] and the multinomial classes for language modeling [9]. The mean vector of Dirichlet distribution is obtained by $\mathbb{E}(\boldsymbol{\theta}) = \frac{\boldsymbol{\alpha}}{\sum_h \sum_{i=1}^{\mathcal{V}} \alpha_{i|h}}$. Given the training corpus $\mathcal{D}$, the posterior distribution is derived as another Dirichlet distribution

$$\begin{aligned} p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\alpha}) &= \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{p(\mathcal{D}|\boldsymbol{\alpha})} \\ &= \frac{\prod_h \prod_{i=1}^{\mathcal{V}} (\theta_{i|h})^{c(\theta_{i|h})+\alpha_{i|h}-1}}{Z(\mathbf{c}+\boldsymbol{\alpha})} = \mathrm{Dir}(\boldsymbol{\theta}|\mathbf{c}+\boldsymbol{\alpha}) \end{aligned} \tag{4}$$

with the updated hyperparameters $\mathbf{c}+\boldsymbol{\alpha}$. In Eq. (4), each entry $c(\theta_{i|h})$ of $\mathbf{c} = \{c(\theta_{i|h})\}$ denotes the number of occurrences of the n-gram events $\theta_{i|h}$ observed in training data $\mathcal{D}$. In [16], the maximum *a posteriori* (MAP) estimation was applied for task adaptation where MAP *point estimates* $\boldsymbol{\theta}^{\mathsf{MAP}}$ were obtained by maximizing the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\alpha})$.

2.2. Marginal likelihood

To fulfill full Bayesian framework [13], we consider the uncertainty of n-gram parameter $\theta_{i|h} = p(w_i|w_{i-n+1}^{i-1})$ and calculate the marginal distribution from training data $\mathcal{D}$ over all values of parameter $\boldsymbol{\theta}$

$$\begin{aligned} p(w_i|w_{i-n+1}^{i-1}, \mathcal{D}, \boldsymbol{\alpha}) &= \int p(w_i|w_{i-n+1}^{i-1}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\alpha}) d\boldsymbol{\theta} \\ = \int \theta_{i|h} \cdot \mathrm{Dir}(\boldsymbol{\theta}|\mathbf{c}+\boldsymbol{\alpha}) d\boldsymbol{\theta} &= \frac{c(\theta_{i|h}) + \alpha_{i|h}}{\sum_{j=1}^{\mathcal{V}} [c(\theta_{j|h}) + \alpha_{j|h}]} \end{aligned} \tag{5}$$

which is also known as the *distribution estimate* of an n-gram. This distribution is equivalent to calculate the *mean* of n-gram parameter $\theta_{i|h}$ based on the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\alpha})$ as given in Eq. (4). In this study, we would like to reverse-engineer the language model smoothing method in Eq. (1) from Bayesian perspective by using the Bayesian language model in Eq. (5).

3. BAYESIAN LANGUAGE MODEL

3.1. Interpolation-smoothed language models

It is important to investigate the physical meaning of Bayesian LM. The hyperparameter $\alpha_{i|h}$ in Dirichlet prior is seen as a *pseudo-count* for an n-gram event $w_{i-n+1}^{i} = \{w_i, w_{i-n+1}^{i-1}\}$. Basically, the marginal distribution is an integration of the prior statistics $\boldsymbol{\alpha}$ and the training data $\mathcal{D}$ or equivalently the counts of occurrences $\mathbf{c}$ because

$$\begin{aligned} \frac{c(w_{i-n+1}^{i}) + \alpha(w_{i-n+1}^{i})}{\sum_{w_i} \left[c(w_{i-n+1}^{i}) + \alpha(w_{i-n+1}^{i})\right]} &= \lambda_{w_{i-n+1}^{i-1}} \\ \times p_{\mathsf{ML}}(w_i|w_{i-n+1}^{i-1}) + (1 - \lambda_{w_{i-n+1}^{i-1}}) &\frac{\alpha(w_{i-n+1}^{i})}{\sum_{w_i} \alpha(w_{i-n+1}^{i})} \end{aligned} \tag{6}$$

where $p_{\mathsf{ML}}(w_i|w_{i-n+1}^{i-1})$ is the ML model and

$$1 - \lambda_{w_{i-n+1}^{i-1}} = \frac{\sum_{w_i} \alpha(w_{i-n+1}^{i})}{\sum_{w_i} \left[c(w_{i-n+1}^{i}) + \alpha(w_{i-n+1}^{i})\right]} \tag{7}$$

implies the interpolation weight for prior statistics. We can see that marginal distribution is interpreted as the smoothed n-gram based on the *interpolation smoothing* as shown in Eq. (1). The relation between Bayesian LM and interpolation-smoothed LM is shown. A reasonable solution to $\alpha_{i|h}$ could be determined from the $(n-1)$-gram event w_{i-n+2}^{i}.

In addition, MKN-LM [3, 4] was realized as a kind of interpolation-smoothed LM by replacing the scaling factor $\lambda_{w_{i-n+1}^{i-1}}$ for higher-order ML model in Eq. (6) by using a fixed discount $0 \leq d \leq 1$

$$\frac{\max\{c(w_{i-n+1}^{i}) - d, 0\}}{\sum_{w_i} c(w_{i-n+1}^{i})} + (1 - \lambda_{w_{i-n+1}^{i-1}}) p_{\mathsf{ML}}(w_i|w_{i-n+2}^{i-1}). \tag{8}$$

The interpolation weight for lower-order ML model is empirically obtained so as to meet the distribution sum up to 1

$$1 - \lambda_{w_{i-n+1}^{i-1}} = \frac{d \cdot N_{1+}(w_{i-n+1}^{i-1} \bullet)}{\sum_{w_i} c(w_{i-n+1}^{i})} \tag{9}$$

where $N_{1+}(w_{i-n+1}^{i-1} \bullet) = |\{w_i : c(w_{i-n+1}^{i-1} w_i) > 0\}|$ represents the number of unique words that follow w_{i-n+1}^{i-1}. Bayesian LM in Eq. (6) and MKN-LM in Eq. (8) are both interpolation-smoothed LMs but in different realizations. Bayesian LM is known as a kind of *smoothed LM* where the hyperparameters $\boldsymbol{\alpha}$ are estimated from training data $\mathcal{D}$.

3.2. Estimation for the shared Dirichlet priors

We aim to find optimal hyperparameters or Dirichlet priors $\boldsymbol{\alpha}$ of Bayesian LM from training data $\mathcal{D}$. According to the evidence framework [12, 13], we maximize the evidence function to find

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}} \; p(\mathcal{D}|\boldsymbol{\alpha}) \tag{10}$$

where the evidence function is derived by referring to Eq. (4)

$$\begin{aligned} p(\mathcal{D}|\boldsymbol{\alpha}) = \frac{Z(\mathbf{c}+\boldsymbol{\alpha})}{Z(\boldsymbol{\alpha})} &= \prod_h \left[\frac{\prod_{i=1}^{\mathcal{V}} \Gamma(c(\theta_{i|h}) + \alpha_{i|h})}{\Gamma(\sum_{i=1}^{\mathcal{V}} c(\theta_{i|h}) + \alpha_{i|h})} \right. \\ &\left. \times \frac{\Gamma(\sum_{i=1}^{\mathcal{V}} \alpha_{i|h})}{\prod_{i=1}^{\mathcal{V}} \Gamma(\alpha_{i|h})} \right] \end{aligned} \tag{11}$$

which is arranged as a ratio of normalization constants of posterior probability $p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\alpha})$ over prior probability $p(\boldsymbol{\theta}|\boldsymbol{\alpha})$.

Similar to the cDC-LM in [9], we introduce latent class k for individual history h and estimate the shared Dirichlet priors $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_k\} = \{\alpha_{i|k}\}$ for the clusters or the classes of n-gram histories. We estimate the individual parameter $\alpha_{i|k}$ through solving

$$\begin{aligned}\frac{\partial}{\partial \alpha_{i|k}} \log p(\mathcal{D}|\boldsymbol{\alpha}) = \sum_{h(k)} \Bigg[& \Psi(c(\theta_{i|h}) + \alpha_{i|k}) - \Psi(\alpha_{i|k}) \\ & -\Psi\left(\sum_{i=1}^{\mathcal{V}} c(\theta_{i|h}) + \alpha_{i|k}\right) + \Psi\left(\sum_{i=1}^{\mathcal{V}} \alpha_{i|k}\right)\Bigg] = 0\end{aligned} \tag{12}$$

where $h(k)$ denotes all histories h corresponding to latent class k and $\Psi(x) \triangleq \frac{\partial}{\partial x} \log \Gamma(x)$ defines the digamma function. We may implement the conjugate gradient algorithm to estimate $\hat{\alpha}_{i|k}$ or apply some approximation to derive an explicit optimization algorithm.

In this optimization, we consider $\sum_{i=1}^{N} \alpha_{i|k} > 1$, $\alpha_{i|k} < 1$ and solve Eq. (12) by applying the recursive formula of digamma function $\Psi(x+1) = \Psi(x) + \frac{1}{x}$ to yield

$$\begin{aligned}
&\Psi(c(\theta_{i|h}) + \alpha_{i|k}) - \Psi(\alpha_{i|k}) = \frac{1}{c(\theta_{i|h}) - 1 + \alpha_{i|k}} \\
&+ \frac{1}{c(\theta_{i|h}) - 2 + \alpha_{i|k}} + \cdots + \frac{1}{2 + \alpha_{i|k}} + \frac{1}{1 + \alpha_{i|k}} + \frac{1}{\alpha_{i|k}} \\
&= \frac{1}{\alpha_{i|k}} + \sum_{c=2}^{c(\theta_{i|h})} \left[\frac{1}{c - 1 + \alpha_{i|k}}\right] \\
&\approx \frac{1}{\alpha_{i|k}} + \sum_{c=2}^{c(\theta_{i|h})} \left[\frac{1}{c-1} - \frac{\alpha_{i|k}}{(c-1)^2} + \mathcal{O}(\alpha_{i|k}^2)\right] \\
&= \frac{1}{\alpha_{i|k}} + \sum_{c=2}^{c(\theta_{i|h})} \frac{1}{c-1} - \alpha_{i|k} \sum_{c=2}^{c(\theta_{i|h})} \frac{1}{(c-1)^2} + \mathcal{O}(\alpha_{i|k}^2)\,.
\end{aligned} \tag{13}$$

Here, $c(\theta_{i|h}) \geq 2$ is assumed and a Taylor series is introduced to approximate the function $f(\alpha) = \frac{1}{c-1+\alpha}$ at point $\alpha = 0$.

Further, the remaining terms of Eq. (12) are approximated by $\Psi(x) \approx \log(x) - \frac{1}{2x} + \mathcal{O}\left(\frac{1}{x^2}\right)$ so as to obtain

$$\begin{aligned}
&\sum_{h(k)} \left[\Psi\left(\sum_{i=1}^{\mathcal{V}} c(\theta_{i|h}) + \alpha_{i|k}\right) - \Psi\left(\sum_{i=1}^{\mathcal{V}} \alpha_{i|k}\right)\right] \\
&\approx \sum_{h(k)} \log\left[\frac{\sum_{i=1}^{\mathcal{V}} c(\theta_{i|h}) + \alpha_{i|k}}{\sum_{i=1}^{\mathcal{V}} \alpha_{i|k}}\right] + \frac{1}{2} \\
&\times \sum_{h(k)} \left[\frac{\sum_{i=1}^{\mathcal{V}} c(\theta_{i|h})}{\left(\sum_{i=1}^{\mathcal{V}} \alpha_{i|k}\right)\left(\sum_{i=1}^{\mathcal{V}} c(\theta_{i|h}) + \alpha_{i|k}\right)}\right] = S(\boldsymbol{\alpha}_k).
\end{aligned} \tag{14}$$

Denote the number of contexts h of class k occurring before word i by $V_{i|k} = \sum_{h(k)}(1)$. We compute the quantities:

$$G_{i|k} = \sum_{h(k)} \sum_{c=2}^{c(\theta_{i|h})} \frac{1}{c-1}, \; H_{i|k} = \sum_{h(k)} \sum_{c=2}^{c(\theta_{i|h})} \frac{1}{(c-1)^2}. \tag{15}$$

Finally, the optimal hyperparameter corresponding to word i with history h in class k should satisfy [14]

$$\begin{aligned}\hat{\alpha}_{i|k} &= \frac{2V_{i|k}}{S(\boldsymbol{\alpha}_k) - G_{i|k} + \sqrt{(S(\boldsymbol{\alpha}_k) - G_{i|k})^2 + 4H_{i|k}V_{i|k}}} \\ &= \mathcal{F}_i(\boldsymbol{\alpha}_k)\end{aligned} \tag{16}$$

which is derived as a root of a quadratic function of $\alpha_{i|k}^{-1}$ from Eqs. (12)(13). This is an *implicit solution* because the right-hand-side of Eq. (16) is also function of $\boldsymbol{\alpha}_k = \{\alpha_{i|k}\}$. Such an implicit solution converges fast in the implementation.

4. DISCUSSION AND EVALUATION

4.1. Discussion

We illustrate five properties of the Bayesian LM. First of all, the sum of hyperparameters $\gamma_k = \sum_i \alpha_{i|k}$ measures the sharpness of the distribution. A large γ_k produces a distribution of components $\{\theta_{i|h}\}_{i=1}^{\mathcal{V}}$ which have similar values. For small γ_k, few components in $\{\theta_{i|h}\}_{i=1}^{\mathcal{V}}$ receive overwhelming share of the probability mass. Second, the estimated hyperparameter $\alpha_{i|k}$ is closely related to $V_{i|k}$, the number of histories h in class k for prediction of word i. Three, instead of finding interpolation weight $\lambda_{w_{i-n+1}^{i-1}}$ or λ_h via cross-validation, the Bayesian LM only adopts the training data in model construction. Four, finding $\alpha_{i|k}$ in Bayesian LM requires computation time linear in the size of vocabulary $\mathcal{V}$, but finding λ_h in [11] requires time linear in the size of training corpus. Five, the shared hyperparameter $\alpha_{i|k}$ in Eq. (16) is calculated by using the higher-order information from the histories under the same class $h(k)$. This is different from the standard interpolation-smoothed LM in Eqs. (1) and (8) based on the lower-order LM where the higher-order information is not considered. Notably, the proposed LM is also different from the hierarchical Dirichlet language model (HD-LM) in [14] where the individual hyperparameters $\alpha_{i|h}$ are estimated for different n-gram parameters $\theta_{i|h}$. HD-LM could not be implemented for unseen n-grams $\theta_{i|h}$ in training data $\mathcal{D}$ which definitely happen in real-world applications.

4.2. Experimental setup

The 1987-1989 Wall Street Journal (WSJ) corpus containing 86K documents with 38M words was utilized to evaluate different methods for continuous speech recognition (CSR). The metrics of perplexity and word error rate (WER) (%) were evaluated. The SI-84 training set with 15.3 hours was used to estimate HMM parameters based on 39-dimensional MFCC-based feature vectors. Triphone models were built for 39 phones and one background silence. Each triphone model had three states and eight Gaussians. The HTK toolkit [17] was used for HMM training and lattice generation. The SRI toolkit [18] was applied to train baseline MKN trigrams [3, 4].

The 100-best lists were generated by using baseline trigrams. Various LMs were linearly interpolated with a baseline trigram system and were employed for N-best rescoring. The 5K and 20K non-verbalized pronunciation, closed vocabularies were adopted. A total of 330 and 333 test sentences were sampled from November 1992 ARPA CSR benchmark test data for vocabularies of 5K and 20K words, respectively. A small set of sentences sampled from the WSJ development set were used to select the interpolation weight λ_h. For comparative study, we carried out the cDC-LM [9] and the MKN-LM [3, 4] which tackled the issues of insufficient training data and long-distance information. However, the proposed Bayesian LM did not consider long-distance information. To compensate this weakness, we combined it with RNN-LM by using linear interpolation as denoted by B-LM. Number of hidden neurons in RNN-LM was 300. We also implemented the neural network LM (NN-LM) [19] with 100 and 200 neurons in the first and the second hidden layers, respectively. The sharing of Dirichlet prior parameters $\alpha_{i|k}$ in B-LM was done by following the same Dirichlet classes k for trigram histories h we learned by using cDC-LM.

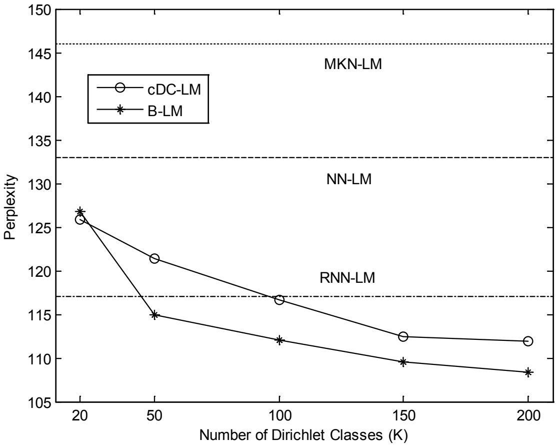

Fig. 1. Perplexity vs. number of classes using different LMs.

4.3. Evaluation for model perplexity

Figure 1 demonstrates the perplexities of using baseline MKN-LM, NN-LM, RNN-LM, cDC-LM and B-LM with different number of Dirichlet classes K. WSJ corpus is used. The results of cDC-LM and RNN-LM are better than that of NN-LM since NN-LM does not learn long-distance information. Also, cDC-LM attains lower perplexity than RNN-LM when $K = 150$ and $K = 200$. Among different LMs, the lowest perplexity is achieved by B-LM for most cases of K. B-LM performs well for LM smoothing. The shared hyperparameters in B-LM has the potential to deal with the overfitted LM due to abundant samples.

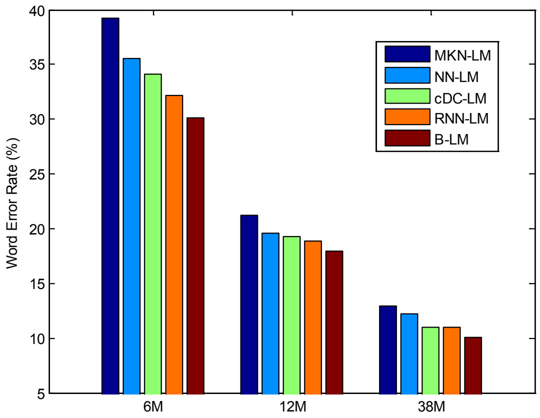

Fig. 2. Comparison of WERs (%) for different LMs with different sizes of training data.

Table 1. Comparison of WERs (%) for different LMs with different vocabulary sizes $\mathcal{V}$.

	MKN-LM	NN-LM	cDC-LM	RNN-LM	B-LM
$\mathcal{V}$=5K	5.38	5.18	4.93	4.92	4.89
$\mathcal{V}$=20K	12.89	12.20	11.01	10.95	10.08

4.4. Evaluation for word error rate

Figure 2 shows the WERs of different LMs with different sizes of training data. We fix $K = 200$ for cDC-LM and B-LM. WSJ is adopted. The issue of small sample size is examined. WERs were reduced when increasing the amount of training data. The reduction of WER of using cDC-LM, RNN-LM and B-LM is significant compared to that of using MKN-LM and NN-LM when the amount of data is as small as 6M words. Table 1 further reports the WERs of different methods by using vocabulary sizes of 5K and 20K. We do consistently see the improvement of using B-LM when compared with NN-LM, cDC-LM and RNN-LM.

5. CONCLUSIONS

We have presented the Bayesian LM with the shared Dirichlet priors and connected its relation to the interpolation-smoothed LM. This model was derived as a distribution estimate of n-grams which considered the randomness of n-gram parameters in language modeling. We tackled the issue of model regularization and compensated for the variations or uncertainties in n-gram estimation due to real-world condition in natural language. The shared prior parameters were estimated to implement the smoothed Bayesian LM for speech recognition. The computation cost was limited. Experiments on different conditions show consistent improvement over the state-of-art LMs.

6. REFERENCES

[1] G. Saon and J.-T. Chien, "Large-vocabulary continuous speech recognition systems - a look at some recent advances," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 18–33, 2012.

[2] J.-T. Chien, "Association pattern language modeling," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1719–1728, 2006.

[3] R. Kneser and H. Ney, "Improved backing-off for *m*-gram language modeling," *Proceedings of International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, pp. 181–184, 1995.

[4] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech and Language*, vol. 13, no. 4, pp. 359–394, 1999.

[5] S. Huang and S. Renals, "Hierarchical Bayesian language models for conversational speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 8, pp. 1941–1954, 2010.

[6] Y. W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," *Proceeding of Annual Meeting of the Association for Computational Linguistics*, pp. 985–992, 2006.

[7] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," *Proceedings of Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1045–1048, 2010.

[8] J.-T. Chien and Y.-C. Ku, "Bayesian recurrent neural network language model," *Proceeding of IEEE Spoken Language Technology Workshop (SLT)*, pp. 206–211, 2014.

[9] J.-T. Chien and C.-H. Chueh, "Dirichlet class language models for speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 3, pp. 482–495, 2011.

[10] J. R. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1279–1296, 2000.

[11] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," *Proceedings of the Workshop on Pattern Recognition in Practice*, pp. 381–397, 1980.

[12] D. J. C. MacKay, "The evidence framework applied to classification networks," *Neural Computation*, vol. 4, no. 5, pp. 720–736, 1992.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science, 2006.

[14] D. J. C. MacKay and L. C. B. Peto, "A hierarchical Dirichlet language model," *Natural Language Engineering*, vol. 1, no. 3, pp. 289–308, 1995.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 5, pp. 993–1022, 2003.

[16] H. Masataki, Y. Sagisaka, K. Hisaki, and T. Kawahara, "Task adaptation using MAP estimation in *n*-gram language modeling," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, pp. 783–786, 1997.

[17] S. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. J. Odell, D. Ollason, D. Povey, V. Valchev, and P. Woodland, *The HTK Book, Version 3.4*, Cambridge University Engineering Department, 2006.

[18] A. Stolcke, "SRILM-an extensible language modeling toolkit," *Proceedings of International Conference on Spoken Language Processing*, pp. 901–904, 2002.

[19] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.