HIERARCHICAL PITMAN-YOR LANGUAGE MODELS FOR ASR IN MEETINGS

Songfang Huang and Steve Renals

The Centre for Speech Technology Research
University of Edinburgh
Edinburgh, EH8 9LW, United Kingdom
{s.f.huang, s.renals}@ed.ac.uk

ABSTRACT

In this paper we investigate the application of a hierarchical Bayesian language model (LM) based on the Pitman-Yor process for automatic speech recognition (ASR) of multiparty meetings. The hierarchical Pitman-Yor language model (HPY-LM) provides a Bayesian interpretation of LM smoothing. An approximation to the HPYLM recovers the exact formulation of the interpolated Kneser-Ney smoothing method in n-gram models. This paper focuses on the application and scalability of HPYLM on a practical large vocabulary ASR system. Experimental results on NIST RT06s evaluation meeting data verify that HPYLM is a competitive and promising language modeling technique, which consistently performs better than interpolated Kneser-Ney and modified Kneser-Ney n-gram LMs in terms of both perplexity and word error rate.

***Index Terms*—** Language Model, Pitman-Yor Process, Hierarchical Bayesian Model, Meetings

1. INTRODUCTION

A statistical language model (LM) is an essential component of speech and language processing for human-computer interaction, used in automatic speech recognition (ASR), statistical machine translation, parsing, and information retrieval. The goal of an LM is to provide a predictive probability distribution for the next word conditioned on the words seen so far, approximated as the immediately preceding $n-1$ words in a conventional n-gram model.

There has been a considerable amount of research aimed at improving n-gram language models. For example, a number of smoothing methods [1] have been introduced to overcome the overfitting problem in the maximum-likelihood estimated (MLE) n-gram models. New approaches for language modeling, such as neural network LMs [2] and distributed LMs [3], have also been proposed to provide smoother LM probability estimates. In addition, much attention has been paid to the incorporation of richer knowledge into LMs, e.g., using factored LMs [4] to exploit morphological information, using structured LMs [5] for syntactic knowledge, and using Bayesian models for semantic knowledge such as topic information [6].

We are concerned with LMs for multiparty meetings. The multimodal and interactive nature of group meetings demands a more comprehensive modeling framework for LMs to accommodate multimodal cues other than lexical information in LMs. Conventional n-gram models can be considered as a *flat* model since they rely on short-span lexical context. There are additional cues available in multiparty meetings, such as prosodic features, semantic context, participant roles and visual focus of attention. Our intuition is that these multimodal cues may augment the lexical context in an n-gram LM, and consequently be helpful for predicting the next word in an LM. This leads to our proposal of a *structured* multimodal language model for meetings. However, there are several difficulties to be overcome. Firstly, multimodal cues in meetings are normally heterogeneous, with different types and different scales. This makes the modeling problem difficult, because directly using MLE-based n-gram models for this task tends to overfit and make data sparsity more acute. Secondly, we require unsupervised methods to automatically extract some multimodal cues such as semantic context. These factors motivate us to investigate a novel approach based on a Bayesian framework.

Hierarchical Bayesian models [7] offer several advantages for our task. Additional knowledge sources can be expressed as prior distributions in the Bayesian framework, which in turn has internally coherent mechanisms for learning and inference. Blei *et al.* [8] have successfully demonstrated that Bayesian models can be used in an unsupervised way to learn the latent structures that connect modalities of different scales such as text and images. It is also straightforward to incorporate Bayesian models into larger models in a principled manner. More specifically to language modeling, Bayesian language models, which have comparable performance to the state-of-the-art n-gram models [9], have been introduced recently. These facts suggest that it is natural and promising for us to investigate a Bayesian language model.

This paper reimplements the hierarchical Pitman-Yor language model (HPYLM), a Bayesian language model based on

978-1-4244-1746-9/07/$25.00 ©2007 IEEE

the Pitman-Yor process, which was initially proposed by Teh [9]. We will emphasize its application and scalability to large vocabulary ASR systems, trying to answer several questions concerning it. First, the performance of HPYLM was tested only in terms of perplexity (PPL). Will a reduction in PPL lead to a reduction in the word error rate (WER) of a practical ASR system? Second, we are interested in ASR applications that normally need to deal with a large vocabulary size and a large amount of training data. Will the HPYLM scale to large training data sets?

2. RELATED WORK

The idea of placing a prior distribution over parameters of LMs and learning point estimates of parameters from training data was investigated by Nadas in 1984 [10]. However, this was an "empirical Bayes" perspective in which parameters of the prior were point estimates learned by maximizing the likelihood on the training data rather than by full Bayesian inference.

MacKay *et al.* [11] introduced a full Bayesian approach for language modeling, which extended the empirical Bayes framework of Nadas to a hierarchical Dirichlet LM. The predictions of hierarchical Dirichlet LMs are similar to those of a traditionally smoothed LM. MacKay *et al.* in this way demonstrated, on a small corpus, that a hierarchical Dirichlet language model had comparable performance to a bigram model smoothed by deleted interpolation with specific values of interpolation weight.

Goldwater *et al.* argued in [12] that a Pitman-Yor process is more suitable as a prior distribution than a Dirichlet distribution to applications in natural language processing, as the power-law distributions of word frequencies produced by Pitman-Yor processes more closely resemble the heavy-tailed distributions observed in natural language.

Another more recent work along the line of using Pitman-Yor processes in hierarchical Bayesian models for language modeling was independently proposed by Teh [9], which can be considered as a natural generalization of the hierarchical Dirichlet language model [11], using a Pitman-Yor process rather than the Dirichlet distribution. Teh provided both hierarchical and experimental extensions to the Pitman-Yor language model of Goldwater *et al.* Experiments on an APNews corpus showed that the novel hierarchical Pitman-Yor language model produces results superior to hierarchical Dirichlet language models and n-gram smoothed by interpolated Kneser-Ney (IKN), and comparable to those smoothed by modified Kneser-Ney (MKN) [1].

On other hand, latent Dirichlet allocation (LDA) [8] is an unsupervised model to discover the latent structures in a large amount of data. Extensions of LDA have also been used for multimodal combination. Wallach proposed a hierarchical generative model that incorporates both n-gram statistics of a hierarchical Dirichlet bigram language model [11] and latent topics of a LDA [6]. This integration is totally within the hierarchical Bayesian framework. Both these extensions to LDA imply that it is possible for the marriage of language models, and topic models that go beyond the "bag-of-words" assumption, either by placing a Markov chain constrain over word sequences, or by full Bayesian ways.

3. HIERARCHICAL PITMAN-YOR LM

We introduce a Bayesian language model based on the Pitman-Yor process using a hierarchical framework. This section briefly summarizes the original work on HPYLM. For detailed information we refer to [9, 12].

3.1. Pitman-Yor Process

The Pitman-Yor process [13] PY(d, θ, G_0) is a three-parametric distribution over distributions, where d is a discount parameter, θ a strength parameter, and G_0 a base distribution that can be understood as a mean of draws from PY(d, θ, G_0). When $d = 0$, the Pitman-Yor process reverts to the Dirichlet process Dir(θG_0). In this sense, the Pitman-Yor process is a generalization of the Dirichlet process.

The procedure for generating draws G $\sim$ PY(d, θ, G_0) from a Pitman-Yor process can be described using the Chinese restaurant metaphor. Imagine a Chinese restaurant containing an infinite number of tables, each with infinite seating capacity. Customers enter the restaurant and seat themselves. The first customer sits at the first available table, while each of the subsequent customers sits at an occupied table with probability proportional to the number of customers already sitting there $c_k - d$, or at a new unoccupied table with probability proportional to $\theta + dt_.$. That is, if z_i is the index of the table chosen by the ith customer, then the ith customer sits at table k given the seating arrangement of previously $i-1$ customers $\mathbf{z}_{-i} = \{z_1, \ldots, z_{i-1}\}$ with probability

$$P(z_i = k|\mathbf{z}_{-i}) = \begin{cases} \frac{c_k - d}{\theta + c_.} & 1 \leq k \leq t_. \\ \frac{\theta + dt_.}{\theta + c_.} & k = t_. + 1 \end{cases} \qquad (1)$$

where $t_.$ is the current number of occupied tables, c_k the number of customers sitting at table k, and $c_. = \sum_k c_k$ the total number of customers. The Pitman-Yor process produces a power-law distribution over the number of customers seated at each table. The power-law distribution — a few outcomes have very high probability and most outcomes occur with low probability — has been found to be one of the most striking statistical properties of word frequencies in natural language.

3.2. Language Model based on Pitman-Yor Process

The Pitman-Yor process can be used to create a power-law distribution over integers, but to create a distribution over words for language modeling we need to combine it with a lexicon generator to make a full two-stage modeling framework [12] with a *generator* and an *adaptor*. The two-stage model can be viewed as a restaurant in which each table has a label with a word w generated by $G_0(w)$. Each customer represents a word token, so that the number of customers at a table corresponds to the frequency of the lexical word labelling that table. A customer may only be assigned to a table whose label matches that word token. The adaptor then 'adapts' the word frequencies produced by the generator to follow a power-law distribution.

Consider a vocabulary $\mathcal{W}$ with V word types. Let $G(w)$ be the unigram probability of w, and $G = [G(w)]_{w \in \mathcal{W}} = [G(w_1), G(w_2), G(w_3), \ldots, G(w_V)]$ represent the vector of word probability estimates for unigrams. A Pitman-Yor prior is placed over $G \sim \text{PY}(d, \theta, G_0)$ with uninformative mean distribution $G_0(w) = 1/V$ for all $w \in \mathcal{W}$. According to the Chinese restaurant metaphor, customers enter the restaurant and seat themselves at tables. Given the seating arrangement $\mathcal{S}$ of customers, the predictive probability of a new word is given by (2).

$$P(w|\mathcal{S}) = \frac{c_w - dt_w}{\theta + c_{\cdot}} + \frac{\theta + dt_w}{\theta + c_{\cdot}} G_0(w) \tag{2}$$

Averaging over the posterior probability over seating arrangements, we can get the actual prediction probability $P(w)$ for unigram LMs.

Similarly we can generalize the above unigram example to the n-gram case. An n-gram LM defines a probability distribution over the current word given a context $\mathbf{u}$ consisting of $n-1$ words. Let $G_{\mathbf{u}}(w)$ be the probability of the current word w and $G = [G_{\mathbf{u}}(w)]_{w \in \mathcal{W}}$ be the target probability distribution for n-gram. A Pitman-Yor process is served as the prior over $G_{\mathbf{u}}$, with discounting parameter $d_{|\mathbf{u}|}$ and strength parameter $\theta_{|\mathbf{u}|}$ specific to the length of the context. The mean distribution is $G_{\pi(\mathbf{u})}$, the lower order model of probabilities of the current word given all but the earliest word in the context. That is,

$$G_{\mathbf{u}} \sim \text{PY}(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}) \tag{3}$$

Since $G_{\pi(\mathbf{u})}$ is still an unknown probability distribution, a Pitman-Yor process is recursively placed over it with parameters specific to $\pi(\mathbf{u})$, $G_{\pi}(\mathbf{u}) \sim \text{PY}(d_{|\pi(\mathbf{u})|}, \theta_{|\pi(\mathbf{u})|}, G_{\pi(\pi(\mathbf{u}))})$. This is repeated until we reach $G_{\emptyset}$ for a unigram model discussed above. This results in a hierarchical prior (Fig. 1). Using this hierarchical prior setup, we generalize from the unigram model to the n-gram case. By using the hierarchical framework of Pitman-Yor priors, different orders of n-gram can thus share information with each other, similar to the traditional interpolation of higher order n-grams with lower order n-grams.

Based on this high-level framework for an HPYLM, one central task is the inference of seating arrangements in each restaurant and the estimation of the context-specific parameters from the training data. Given training data $\mathcal{D}$, we know the number of co-occurrences $c_{\mathbf{u}w}$ of a word w after a context $\mathbf{u}$ of length $n-1$. Actually this is the only information we need to train an HPYLM. A Markov chain Monte Carlo sampling based scheme can be used to infer the posterior of seating arrangements. In this paper Gibbs sampling is used to keep track of which table each customer sits at, by iterating over all customers present in each restaurant — first removing a customer from the restaurant, and then resampling the table at which that customer sits. After a sufficient number of iterations, the states of variables of interest in the seating arrangements will converge to the required samples from the posterior distribution. In the HPYLM the more frequent a word token, the more likely it is there are more tables corresponding to that word token.

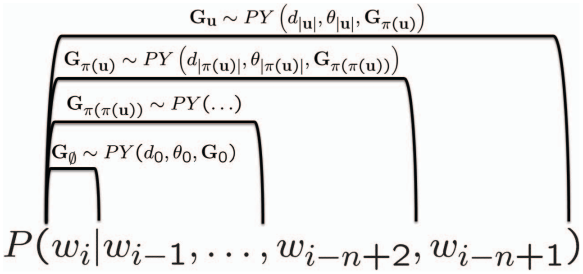
$\mathbf{G_u} \sim PY\left(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, \mathbf{G}_{\pi(\mathbf{u})}\right)$
$\mathbf{G}_{\pi(\mathbf{u})} \sim PY\left(d_{|\pi(\mathbf{u})|}, \theta_{|\pi(\mathbf{u})|}, \mathbf{G}_{\pi(\pi(\mathbf{u}))}\right)$
$\mathbf{G}_{\pi(\pi(\mathbf{u}))} \sim PY(\ldots)$
$\mathbf{G}_{\emptyset} \sim PY(d_0, \theta_0, \mathbf{G}_0)$
$P(w_i|w_{i-1}, \ldots, w_{i-n+2}, w_{i-n+1})$

Fig. 1. The hierarchy of Pitman-Yor priors for n-gram LMs.

For a n-gram LM, there are $2n$ parameters $\Theta = \{d_m, \theta_m : 0 \le m \le n-1\}$ to be estimated in total. In this paper, we use the auxiliary variable sampling method [9], which assumes that each discount parameter d_m has a Beta prior distribution $d_m \sim \text{Beta}(a_m, b_m)$ while each concentration parameter θ_m has a Gamma prior distribution $\theta_m \sim \text{Gamma}(\alpha_m, \beta_m)$.

Under a particular setting of seating arrangements $\mathcal{S}$ and parameters Θ, the predictive probability $P(w|\mathbf{u}, \mathcal{S}, \Theta)$ is:

$$\begin{aligned} P(w|\mathbf{u}, \mathcal{S}, \Theta) = & \frac{c_{\mathbf{u}w\cdot} - d_{|\mathbf{u}|} t_{\mathbf{u}w}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} \\ & + \frac{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|} t_{\mathbf{u}\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} P(w|\pi(\mathbf{u}), \mathcal{S}, \Theta) \end{aligned} \tag{4}$$

The overall predictive probability can be approximately obtained by collecting I samples from the posterior over $\mathcal{S}$ and Θ, and then averaging (4) to approximate the integral with samples, as shown in (5).

$$P(w|\mathbf{u}) \approx \sum_{i=1}^{I} P(w|\mathbf{u}, \mathcal{S}^{(i)}, \Theta^{(i)})/I \tag{5}$$

If we assume that the strength parameters $\theta_{|\mathbf{u}|} = 0$ for all $\mathbf{u}$, and restrict $t_{\mathbf{u}w}$ to be at most 1 (i.e., all customers representing the same word token should only sit on the same table together), then the predictive probability in (4) directly reduces to the predictive probability given by the IKN LM. We can thus interpret IKN as an approximate inference scheme for the hierarchical Pitman-Yor language model.

3.3. Implementation

We implemented the hierarchical Pitman-Yor language model within the SRILM toolkit [14], as an extended tool for Bayesian language models. We highlight four characteristics of this implementation. Firstly, it is consistent and coherent with the SRILM toolkit. We inherited the HPYLM classes from the base SRILM classes, and provided the same interfaces (i.e., `WordProb()` function) for language modeling. Secondly, it has efficient memory management and computational performance by directly using the data structures available in SRILM. Thirdly, it is a flexible framework for Bayesian language modeling. We can, for example, train a language model with Kneser-Ney smoothing for unigrams, modified Kneser-Ney smoothing for bigrams, and Pitman-Yor process smoothing for trigrams. Finally, this implementation is extensible for future developments: e.g., taking into accounts the combination with topic models like LDA.

A standard ARPA-format LM is output, with the exact format of a conventional n-gram LM. This makes it easy to test the HPYLM in a typical ASR system.

4. EXPERIMENTS ON ASR FOR MEETINGS

4.1. Data

The experiments reported in this paper were performed using the U.S. National Institute of Standard and Technology (NIST) rich transcription (RT) 2006 spring meeting recognition evaluations (RT06s). We tested only on those audio data recorded from individual head microphones (IHM), consisting of meeting data collecting by the AMI project, CMU, NIST, and VT (Virginia Tech).

The training data sets for language models used in this paper are listed in Table 1.

Table 1. The statistics of the training data sets for language models used throughout the experiments.

No.	LM Data Set	#Sentences	#Words
1	AMI data from rt05s	68,806	801,710
2	ICSI meeting corpus	79,307	650,171
3	ISL meeting corpus	17,854	119,093
4	NIST meeting corpus-2	21,840	156,238
5	NIST meeting corpus-a	18,007	119,989
6	Fisher (fisher-03-p1)	1,076,063	10,593,403
7	webdata (meetings)	3,218,066	36,073,718

All the following experiments were using a common vocabulary with $50,000$ word types, unless it is explicitly indicated otherwise.

4.2. PPL Experiments

We took the LM data sets from No.1 to No.5 in Table 1 as a core training set, which consists of 205,814 sentences and 1,847,201 words. We trained trigram IKN, MKN, and HPY LMs using this training data. For the HPYLM, we ran 200 iterations for inference, and collected 100 samples from the posterior over seating arrangements and parameters.

The test data for PPL estimation was extracted from the reference transcriptions for *rt06seval*. The final test data consisted of 3,597 sentences and 31,810 words. Four different experimental conditions were considered and are shown in Table 2: the combination of whether or not a closed vocabulary was used (*-vobab*) and/or mapping unknown words to a special symbol 'unk' (*-unk*) during training.

Table 2 shows the PPL results. We can see that in all the four experiment conditions, HPYLM has a lower PPL than both IKNLM and MKNLM.

Table 2. The PPL results on *rt06seval* testing data.

	-vocab	-unk	IKNLM	MKNLM	HPYLM
EC1	no	no	95.7	93.5	**88.6**
EC2	no	yes	122.0	119.2	**111.9**
EC3	yes	no	110.1	106.5	**101.2**
EC4	yes	yes	110.5	106.8	**102.6**

4.3. WER Experiments

We used the AMI-ASR system [15] as the baseline platform for our ASR experiments. The feature stream comprised of 12 MF-PLP features and raw log energy and first and second order derivatives are added. Cepstral mean and variance normalisation was performed on a per channel basis. The acoustic models were taken from the second pass of AMI-ASR system, which were trained on 108 hours speech data from ICSI, ISL, NIST, and AMI, using vocal tract length normalisation, heteroscedastic linear discrimant analysis, speaker adaptive training and minimum phone error. They are adapted using the transcripts of the first pass and a single constrained maximum likelihood regression transform. We only tested LMs trained under the EC3 in Table 2, that is, we used a 50k vocabulary but without setting *-unk* during training. For HPYLM, we output an ARPA-format LM. Different LMs were then used in the first pass decoding using `HDecode`.

Table 3 shows the WER results. The HPYLM also results in a lower WER than both IKNLM and MKNLM, although this is not statistically significant. However, this is an encouraging result, since it is the first time that the HPYLM has been tested using a state-of-the-art large vocabulary ASR system on standard NIST evaluation data.

Table 3. The WER results on *rt06seval* testing data.

LMS	SUB	DEL	INS	TOT
IKNLM	15.7	9.9	2.9	28.5
MKNLM	15.6	10.0	2.8	28.4
HPYLM	**15.3**	**10.1**	**2.7**	**28.1**

4.4. Scalability

To investigate the scalability of the HPYLM, we gradually increased the size of training data for the HPYLM, ranging from DS1 to DS3 as shown in Table 4. DS1 includes the training data sets No.1–5. DS2 consists of DS1 and the data set No.6. Finally further adding the data set No.7 to DS2 we get DS3. The following experiments were carried out on a machine with dual quad-core Intel Xeon 2.8GHz processors and 12GB of memory. Table 4 shows the different computational time per iteration and memory requirements when we change the size of training data, or vary the size of vocabulary. From the results in Table 4, we can see that the training time for each iteration scales linearly with the size of training data when having a common vocabulary. The vocabulary size is another factor that also affects the computational time and memory requirement. The smaller the size of the vocabulary, the quicker each iteration and the lower the memory requirement. For IKNLM and MKNLM trained on DS3, the memory requirement is around 1GB.

Table 4. The comparison of computational time and memory requirement of the HPYLM on different training data sets.

Data	#Words	Vocab	Time/Iter	Memory
DS1	1,847,201	50k	∼10sec	∼150MB
DS2	12,440,604	50k	∼120sec	∼600MB
DS3	48,514,322	50k	∼600sec	∼2400MB
		18k	∼300sec	∼2000MB
		8k	∼200sec	∼1400MB

We also evaluated PPL performance over these three data sets to investigate the scalability of PPL. As we found in PPL results on Table 5, the HPYLM generalizes well to larger training data. We obtained a consistent reduction in PPL over both IKNLM and MKNLM. This further strengthens the PPL results of Section 4.2.

Finally we trained three ARPA-format trigram LMs — IKNLM, MKNLM, and HPYLM — all on the DS3 training data set, which is a corpus of around 50 millions of words. The lower discounting cutoff of trigram counts is set to 2 in these LMs (*-gt3min* 2 in SRILM). Table 6 shows the WER results of these three different LMs in the first decoding using `HDecode`. Again we see the HPYLM performs slightly better than IKNLM and MKNLM. It should be noted, however, that the WER reductions are even smaller than those in Table 3, suggesting that the HPYLM estimates a better smoothed LM on smaller training data, while it tends to converge to the MLE-based n-gram LMs and produce similar results as IKNLM and MKNLM with a sufficiently larger amount of training data.

Table 5. The PPL results on *rt06seval* using different scale sizes of training data.

Data	IKNLM	MKNLM	HPYLM
DS1	110.1	106.5	**101.2**
DS2	106.7	103.6	**97.6**
DS3	102.9	100.8	**95.1**

Table 6. The WER results on *rt06seval* using different scale sizes of training data.

LMS	SUB	DEL	INS	TOT
IKNLM	14.6	10.0	2.6	27.3
MKNLM	14.6	9.9	2.5	27.0
HPYLM	**14.4**	**10.0**	**2.6**	**26.9**

5. ANALYSIS AND DISCUSSION

In this paper we carried out a set of experiments to verify the use and scalability of the HPYLM on ASR for multiparty meetings. The PPL and WER results seem very promising. We discuss further the behaviour of the HPYLM, such as effects of parameters, and its convergence.

The HPYLM can be considered as a novel smoothing method for language modeling. Even though each Pitman-Yor process $G_{\mathbf{u}}$ for each context only has one shared discount parameter $0 \leq d_{|\mathbf{u}|} < 1$ (4), different words w have different discount values $d_{|\mathbf{u}|}t_{\mathbf{u}w}$, since $t_{\mathbf{u}w}$ can take on different values. Discounts in the HPYLM grow gradually as a function of n-gram counts. In this sense, we say that the HPYLM estimates a better smoothed model than than IKNLM and MKNLM. This partly explains why HPYLM performs better in PPL and WER than IKNLM and MKNLM.

It is sometimes expensive to train an HPYLM, especially when working with large training data as demonstrated in Table 4. Therefore the convergence of HPYLM is an important factor. We trained a special HPYLM, which takes only 10 iterations for burning in to infer the seating arrangements, and then collects 300 samples from the posterior and at each iteration evaluates the PPL on the *rt06seval* test data. Fig. 2 shows the convergence of PPL. From this we can see that after several tens of iterations, the PPL has quickly converged

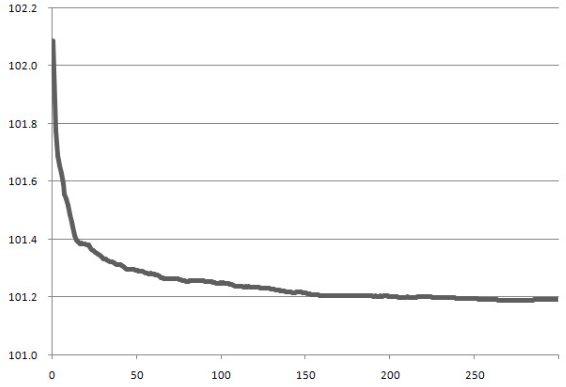

Fig. 2. The convergence of PPL on *rt06seval* test data using the HPYLM.

to a lower PPL value. On the other hand, although it is slow to train an HPYLM on large data, we only need to train the model once and output an ARPA-format LM, then apply it in a ASR system as a standard n-gram LM.

We are pleased to observe that the HPYLM has better performance in terms of both PPL and WER compared with IKN and MKN smoothing. This encourages us to incorporate the HPYLM and other probabilistic topic models such as LDA within a hierarchical Bayesian framework. As a future work, we plan to further investigate the approach of combining the HPYLM and LDA to incorporate multimodal cues into LMs for meetings.

The main contribution of this work is the introduction of a novel Bayesian language modeling technique in ASR, experimentally verified on the task of large vocabulary ASR for meetings. To conclude, the HPYLM is a promising approach to language modeling, which deserves further investigation.

6. ACKNOWLEDGEMENTS

We thank Yee Whye Teh for helpful discussions on HPYLM, and the AMI-ASR team for providing the baseline ASR system for experiments. This work is jointly supported by the Wolfson Microelectronics Scholarship and the European IST Programme Project FP6-033812 (AMIDA). This paper only reflects the authors' views and funding agencies are not liable for any use that may be made of the information contained herein.

7. REFERENCES

[1] Stanley F. Chen and Joshua Goodman, "An empirical study of smoothing techniques for language modeling," Tech. Rep., Computer Science Group, Harvard University, Cambridge, Massachusetts, August 1998.

[2] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. Feb, pp. 1137–1155, February 2003.

[3] John Blitzer, Amir Globerson, and Fernando Pereira, "Distributed latent variable models of lexical co-occurrences," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[4] J. A. Bilmes and K. Kirchhoff, "Factored language models and generalized parallel backoff," in *Proceedings of HLT/NACCL*, 2003, pp. 4–6.

[5] Peng Xu, Ahmad Emami, and Frederick Jelinek, "Training connectionist models for the structured language model," in *Empirical Methods in Natural Language Processing, EMNLP'2003*, 2003.

[6] Hanna M. Wallach, "Topic modeling: Beyond bag-of-words," in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

[7] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin, *Bayesian Data Analysis*, Chapman & Hall/CRC, London, first edition, July 1995.

[8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, 2003.

[9] Yee Whye Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," in *Proceedings of the Annual Meeting of the ACL*, 2006, vol. 44.

[10] A. Nadas, "Estimation of probabilities in the language model of the IBM speech recognition system," *IEEE Trans ASSP*, vol. 32, no. 4, pp. 859–861, 1984.

[11] David J. C. MacKay and Linda C. Bauman Peto, "A hierarchical Dirichlet language model," *Natural Language Engineering*, vol. 1, no. 3, pp. 1–19, 1994.

[12] Sharon J. Goldwater, Thomas L. Griffiths, and Mark Johnson, "Interpolating between types and tokens by estimating power-law generators," in *Advances in Neural Information Processing Systems 18*, 2006.

[13] Jim Pitman, "Exchangeable and partially exchangeable random partitions," *Probability Theory and Related Fields*, vol. 102, pp. 145–158, 1995.

[14] Andreas Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of ICSLP'02*, Denver, Colorado, September 2002.

[15] Thomas Hain, Lukas Burget, John Dines, Giulia Garau, Vincent Wan, Martin Karafiat, Jithendra Vepa, and Mike Lincoln, "The AMI system for the transcription of speech in meetings," in *Proceedings of ICASSP'07*, Hawaii, USA, April 15-20 2007.