THE NATIONAL UNIVERSITY
of SINGAPORE

School of Computing
Lower Kent Ridge Road, Singapore 119260

TRA2/06

A Bayesian Interpretation of Interpolated Kneser-Ney

Yee Whye TEH

February 2006

Technical Report

Foreword

This technical report contains a research paper, development or tutorial article, which has been submitted for publication in a journal or for consideration by the commissioning organization. The report represents the ideas of its author, and should not be taken as the official views of the School or the University. Any discussion of the content of the report should be sent to the author, at the address shown on the cover.

JAFFAR, Joxan
Dean of School

A Bayesian Interpretation of Interpolated Kneser-Ney
NUS School of Computing Technical Report TRA2/06

Yee Whye Teh
tehyw@comp.nus.edu.sg
School of Computing,
National University of Singapore,
3 Science Drive 2, Singapore 117543.

Abstract

Interpolated Kneser-Ney is one of the best smoothing methods for n-gram language models. Previous explanations for its superiority have been based on intuitive and empirical justifications of specific properties of the method. We propose a novel interpretation of interpolated Kneser-Ney as approximate inference in a hierarchical Bayesian model consisting of Pitman-Yor processes. As opposed to past explanations, our interpretation can recover exactly the formulation of interpolated Kneser-Ney, and performs better than interpolated Kneser-Ney when a better inference procedure is used.

1 Introduction

Probabilistic language models are used extensively in a variety of linguistic applications. Standard examples include speech recognition, handwriting recognition, machine translation and spelling correction. The basic task is to model the probability distribution over sentences. Most researchers take the approach of modelling the conditional distribution of words given their histories, and piecing these together to form the joint distribution over the whole sentence,

$$P(\text{word}_1, \text{word}_2, \ldots, \text{word}_t) = \prod_{i=1}^{t} P(\text{word}_i \,|\, \text{word}_1, \ldots, \text{word}_{i-1}) \,. \tag{1}$$

The class of n-gram models form the bulk of such models. The basic assumption here is that the conditional probability of a word given its history can be simplified to its probability given a reduced context consisting of only the past $n-1$ words,

$$P(\text{word}_i \,|\, \text{word}_1, \ldots, \text{word}_{i-1}) = P(\text{word}_i \,|\, \text{word}_{i-N+1}, \ldots, \text{word}_{i-1}) \tag{2}$$

Even for modest values of n the number of parameters involved in n-gram models is still tremendous. For example typical applications use $n = 3$ and has a $O(50000)$ word vocabulary, leading to $O(50000^3)$ parameters. As a result direct maximum-likelihood parameter fitting will severely overfit to our training data, and smoothing techniques are indispensible for the proper training of n-gram models. A large number of smoothing techniques have been proposed in the literature; see Chen and Goodman (1998); Goodman (2001); Rosenfeld (2000) for overviews, while more recent proposals include Charniak (2001); Bilmes and Kirchhoff (2003); Bengio et al. (2003); Xu and Jelinek (2004) and Blitzer et al. (2005).

In an extensive and systematic survey of smoothing techniques for n-grams, Chen and Goodman (1998) showed that interpolated Kneser-Ney and its variants were the most successful smoothing techniques at the time. Although more recent techniques have exhibited better performance, interpolated Kneser-Ney is still an important technique now as the better performances have only been achieved by combining more elaborate models with it. Interpolated Kneser-Ney involves three concepts: it interpolates linearly between higher and lower order n-grams, it alters positive word counts by subtracting a constant amount (absolute discounting), and it uses an unusual estimate of lower order n-grams.

A number of explanations for why interpolated Kneser-Ney works so well has been given in the literature. Kneser and Ney (1995), Chen and Goodman (1998) and Goodman (2001) showed that the unusual estimate of lower order n-grams follows from interpolation, absolute discounting, and a constraint on word marginal distributions. Goodman (2001) further showed that n-gram models which does not preserve these word marginal distributions cannot be optimal. Empirical results in Chen and Goodman (1998) demonstrated that interpolation works better than other ways of combining higher and lower order n-grams and that absolute discounting is a good approximation to the optimal discount. Finally, a different approach by Goodman (2004) showed that back-off Kneser-Ney is similar to a maximum-entropy model with exponential priors.

We will give a new interpretation of interpolated Kneser-Ney as an approximate inference method in a Bayesian model. The model we propose is a straightforward hierarchical Bayesian model (Gelman et al. 1995), where each hidden variable represents the distribution over next words given a particular context. These variables are related hierarchically such that the prior mean of a hidden variable corresponding to a context is the word distribution given the context consisting of all but the earliest word (we will make clear what we mean by this in the later parts of the paper). The hidden variables are distributed according to a well-studied nonparametric generalization of the Dirichlet distribution variously known as the two-parameter Poisson-Dirichlet process or the Pitman-Yor process (Pitman and Yor 1997; Ishwaran and James 2001; Pitman 2002) (in this paper we shall refer to this as the Pitman-Yor process for succinctness).

As we shall show in this paper, this hierarchical structure corresponds exactly to the technique of interpolating between higher and lower order n-grams. Our interpretation has the advantage over past interpretations in that we can recover the exact form of interpolated Kneser-Ney. In addition, in comparison with the maximum-entropy view, where interpolated Kneser-Ney in fact does better than the model in which it is supposed to approximate, we show in experiments that our model works better than interpolated Kneser-Ney if we use more accurate inference procedures. As our model is fully Bayesian, we also reap other advantages of Bayesian methods, e.g. we can easily use the model as part of a more elaborate model.

Bayesian techniques are not new in natural language processing and language modelling given the probabilistic nature of most approaches. Maximum-entropy models have found many uses relating features of inputs to distributions over outputs (Rosenfeld 1994; Berger et al. 1996; McCallum et al. 2000; Lafferty et al. 2001). Use of priors is widespread and a number of studies have been conducted comparing different types of priors (Brand 1999; Chen and Rosenfeld 2000; Goodman 2004). Even hierarchical Bayesian models have been applied to language modelling—MacKay and Peto (1994) have proposed one based on Dirichlet distributions. Our model is a natural generalization of this model using Pitman-Yor processes rather than Dirichlet distributions.

Bayesian models have not been more widely adopted in the language modelling community because the models proposed so far have performed poorly in comparison to other smoothing techniques. The major contributions of our work are in proposing a Bayesian model with excellent performance, and in establishing the direct correspondence between interpolated Kneser-Ney, a well-established smoothing technique, and the Bayesian approach. We expect this connection to be useful both in terms of giving a principled statistical

footing to smoothing techniques and in suggesting even better performing Bayesian models.

Goldwater et al. (2006) observed that Pitman-Yor processes generate power-law distributions and argued that since such distributions occur frequently in natural languages, they are more suited for natural languages processing. Is it thus perhaps unsurprising that our model has performance superior to the hierarchical Dirichlet language model of MacKay and Peto (1994). In fact, Goldwater et al. (2006) have independently noted this relationship between a hierarchical Pitman-Yor process and interpolated Kneser-Ney, but have not corroborated this with further investigations and experimental results.

In the following section we will give a detailed description of interpolated Kneser-Ney and modified Kneser-Ney. We review the Pitman-Yor process as it pertains to language modelling in Section 3. In Section 4 we propose the hierarchical Pitman-Yor language model and relate it to interpolated Kneser-Ney. Experimental results establishing the performance of the model in terms of cross-entropy is reported in Section 5, and we conclude with some discussion in Section 6. Finally we delegate details of some additional properties of the model and inference using Markov chain Monte Carlo sampling to the appendices.

2 Interpolated Kneser-Ney and its Variants

In this section we introduce notations and describe in detail the n-gram language modelling task, interpolated Kneser-Ney and a modified version which performs better. Our sources of information are Chen and Goodman (1998) and Goodman (2001) which are excellent reviews of state-of-the-art smoothing techniques and language models.

We assume that we have a closed set of words in our vocabolary W, which is of size V. For a word $w \in W$ and a context consisting of a sequence of $n-1$ words $\mathbf{u} \in W^{n-1}$ let $c_{\mathbf{u}w}$ be the number of occurrences of w following $\mathbf{u}$ in our training corpus. The naive maximum likelihood probability for a word w following $\mathbf{u}$ is

$$P_{\mathbf{u}}^{\text{ML}}(w) = \frac{c_{\mathbf{u}w}}{c_{\mathbf{u}\cdot}} \tag{3}$$

where $c_{\mathbf{u}\cdot} = \sum_{w'} c_{\mathbf{u}w'}$. Instead, interpolated Kneser-Ney estimates the probability of word w following context $\mathbf{u}$ by discounting the true count $c_{\mathbf{u}w}$ by a fixed amount $d_{|\mathbf{u}|}$ depending on the length $|\mathbf{u}w|$ if $c_{\mathbf{u}w} > 0$ (otherwise the count remains 0). Further, it interpolates the estimated probability of word w with lower order m-gram probabilities. This gives,

$$P_{\mathbf{u}}^{\text{IKN}}(w) = \frac{\max(0, c_{\mathbf{u}w} - d_{|\mathbf{u}|})}{c_{\mathbf{u}\cdot}} + \frac{d_{|\mathbf{u}|} t_{\mathbf{u}\cdot}}{c_{\mathbf{u}\cdot}} P_{\pi(\mathbf{u})}^{\text{IKN}}(w) \tag{4}$$

where $t_{\mathbf{u}\cdot} = \#\{w' \,|\, c_{\mathbf{u}w'} > 0\}$ is the number of distinct words w' following context $\mathbf{u}$ in the training corpus, $\pi(\mathbf{u})$ is the context consisting of all words in $\mathbf{u}$ except the first and $P_{\pi(\mathbf{u})}^{\text{IKN}}(w)$ are the lower order m-gram probabilities. The value of $t_{\mathbf{u}\cdot}$ is chosen simply to make the probability estimates sum to 1. Finally, interpolated Kneser-Ney uses modified sets of counts for the lower order m-gram probabilities. In particular, for a context $\mathbf{u}'$ of length $m < n-1$ and words w' and w, let

$$t_{w'\mathbf{u}'w} = \begin{cases} 1 & \text{if } c_{w'\mathbf{u}'w} > 0; \\ 0 & \text{if } c_{w'\mathbf{u}'w} = 0; \end{cases} \qquad c_{\mathbf{u}'w} = t_{\cdot\mathbf{u}'w} = \sum_{w'} t_{w'\mathbf{u}'w} \tag{5}$$

where $w'\mathbf{u}'$ is the context formed by concatenating w' and $\mathbf{u}'$. The lower order m-gram probabilities are estimated as in (4) using the modified counts of (5). A different value of discount d_{m-1} is used for each length m and these can either be estimated using formulas or by using cross-validation.

Modified Kneser-Ney is an improvement upon interpolated Kneser-Ney where the amount of discount is allowed more variability. In the empirical studies in Chen and Goodman (1998) and Church and Gale (1991) it was found that the optimal amount of discount that should be used changes slowly as a function of the counts $c_{\mathbf{u}w}$. This was used as one of the reasons for absolute discounting in Chen and Goodman (1998). In the same study it was also noticed that the optimal discounts for low values of $c_{\mathbf{u}w}$ differ substantially from those with higher values. Modified Kneser-Ney uses different values of discounts for different counts, one each for $c_{\mathbf{u}w} = 1, 2, \ldots, c^{(\max)} - 1$ and another for $c_{\mathbf{u}w} \geq c^{(\max)}$. The same formulas for (4) and (5) are used. Modified Kneser-Ney reduces to interpolated Kneser-Ney when $c^{(\max)} = 1$, while Chen and Goodman (1998) uses $c^{(\max)} = 3$ as a good compromise between diminishing improvements and increasing implementational complexity.

The unusual counts in interpolated Kneser-Ney can be derived by preserving marginal word distributions. let $P^{\text{emp}}(\mathbf{u})$ be the empirical probability of word sequence $\mathbf{u}$ among sequences of length $n-1$. Let w' and w be words and $\mathbf{u}'$ be a word sequence of length $m = n - 2$. Assuming the form of (4) and the following marginal constraints,

$$\sum_{w'} P^{\text{emp}}(w'\mathbf{u}')P^{\text{IKN}}_{w'\mathbf{u}'}(w) = P^{\text{emp}}(\mathbf{u}'w) \tag{6}$$

we can derive that

$$P^{\text{IKN}}_{\mathbf{u}'}(w) = \frac{c_{\mathbf{u}'w}}{c_{\mathbf{u}'\cdot}} \tag{7}$$

where $c_{\mathbf{u}'w}$ is as given in (5). Finally, rather than using (7) we should discount these new counts and interpolate with even lower order m-gram probabilities, i.e. recursively apply (4) and (5).

Satisfying the marginal constraints (6) is reasonable since the n-gram probabilities should be consistent with the statistics of the word counts. In fact Goodman (2001) showed that if these constraints are not satisfied then the n-gram probability estimates cannot be optimal (the converse is not true; satisfying these constraints does not imply optimality). Taking the marginal constraints view further, Goodman (2004) showed that a back-off version of Kneser-Ney can be seen as an approximation to a maximum-entropy model with approximately satisfied marginal constraints and an exponential prior on the parameters of the model. However this view of interpolated Kneser-Ney in terms of marginal constraints is limited in scope for a few reasons. Firstly, the maximum-entropy model of which back-off Kneser-Ney is supposed to approximate in fact performs worse than back-off Kneser-Ney which is in turn worse than interpolated Kneser-Ney. Secondly, modified Kneser-Ney, which performs better than interpolated Kneser-Ney does not satisfy these marginal constraints.

3 Pitman-Yor Processes

We go through the properties of the Pitman-Yor process relevant to language modelling in this section. For more in depth discussion we refer to Pitman and Yor (1997); Ishwaran and James (2001); Pitman (2002), while Jordan (2005) gives a high-level tutorial of this branch of statistics and probability theory from a machine learning perspective.

The Pitman-Yor process $\text{PY}(d, \theta, G_0)$ is a distribution over distributions over a probability space $\mathbf{X}$. It has three parameters: a discount parameter $0 \leq d < 1$, a strength parameter $\theta > -d$ and a base distribution G_0 over $\mathbf{X}$. The base distribution can be understood as a putative mean of draws from $\text{PY}(d, \theta, G_0)$, while both θ and d control the amount of variability around the base distribution G_0. An explicit construction of draws $G_1 \sim \text{PY}(d, \theta, G_0)$ from a Pitman-Yor process is given by the stick-breaking construction

(Sethuraman 1994; Ishwaran and James 2001). This construction shows that G_1 is a weighted sum of an infinite sequence of point masses (with probability one). Let $V_1, V_2, \ldots$ and $\phi_1, \phi_2, \ldots$ be two sequence of independent random variables with distributions,

$$V_k \sim \text{Beta}(1-d, \theta+kd) \qquad \phi_k \sim G_0 \qquad \text{for } k = 1, 2, \ldots, \tag{8}$$

Then the following construction gives a draw from $\text{PY}(d, \theta, G_0)$:

$$G_1 = \sum_{k=1}^{\infty} (1-V_1)\cdots(1-V_{k-1})V_k \delta_{\phi_k} \tag{9}$$

where δ_ϕ is a point mass located at ϕ. The stick-breaking construction is useful as it is mathematically elegant and it gives us a direct visualization of Pitman-Yor processes.

A different perspective on the Pitman-Yor process is given by the Chinese restaurant process. This describes the properties of the Pitman-Yor process in terms of distributions over draws from G_1, which is itself a distribution over $\mathbf{X}$. Though indirect, this perspective is more useful for our purpose of language modelling, since draws from G_1 will correspond to words whose distributions we wish to model. Let $x_1, x_2, \ldots$ be a sequence of identical and independent draws from G_1. The analogy is that of a sequence of customers (x_i's) visiting a restaurant (corresponding to G_1) with an unbounded number of tables. The Chinese restaurant process assigns a distribution over the seating arrangement of the customers. The first customer sits at the first available table, while each of the other customers sits at the k^{th} occupied table with probability proportional to $c_k - d$, where c_k is the number of customers already sitting there, and she sits at a new unoccupied table with probability proportional to $\theta + dt_\cdot$, where $t_\cdot$ is the current number of occupied tables. To generate draws for $x_1, x_2, \ldots$, associate with each table k an independent draw $\phi_k \sim G_0$ from the base distribution G_0 and set the drawn value of x_i to be ϕ_k if customer i sat at table k. The ϕ_k draws can be thought of as dishes, with customers sitting at each table eating the dish on the table. The resulting conditional distribution of the next draw after a sequence of $c_\cdot = \sum_k c_k$ draws is thus:

$$x_{c_\cdot+1} \,|\, x_1 \ldots, x_{c_\cdot}, \text{ seating arrangement } \sim \sum_{k=1}^{t_\cdot} \frac{c_k - d}{\theta + c_\cdot} \delta_{\phi_k} + \frac{\theta + dt_\cdot}{\theta + c_\cdot} G_0 \tag{10}$$

The sequence $x_1, x_2, \ldots$ as generated by the Chinese restaurant process can be shown to be exchangeable. That is, the distribution assigned by the Chinese restaurant process to $x_1, x_2, \ldots$ is invariant to permuting the order of the sequence. De Finetti's theorem on exchangeable sequences then shows that there must be a distribution over distributions G_1 such that $x_1, x_2, \ldots$ are conditionally independent and identical draws from G_1 (Pitman 2002). The Pitman-Yor process is one such distribution over G_1.

Consider using the Pitman-Yor process as a prior for unigram word distributions. We use a uniform distribution over our fixed vocabulary W of V words as the base distribution G_0, that is, each word in W is equiprobable under G_0, while the draw from the Pitman-Yor process G_1 is the desired unigram distribution over words. We have a training corpus consisting of c_w occurrences of word $w \in W$, which corresponds to knowing that c_w customers are eating dish w in the Chinese restaurant representation. Given this information, we infer the seating arrangement of the $c_\cdot = \sum_w c_w$ customers in the restaurant. In particular, let t_w be the number of tables serving dish w in the seating arrangement (since the vocabulary W is finite there is positive probability that multiple tables serve the same dish). The predictive probability of a new word given the seating arrangement is given by (10), which evaluates to

$$P(x_{c_\cdot+1} = w \,|\, \text{seating arrangement}) = \frac{c_w - dt_w}{\theta + c_\cdot} + \frac{\theta + dt_\cdot}{\theta + c_\cdot} G_0(w) \tag{11}$$

by collecting terms in (10) corresponding to each dish w. The actual predictive probability is then (11) averaged over the posterior probability over seating arrangements. We see that there are two opposing effects on word counts c_w in the Pitman-Yor process. The second term adds to word counts, while the discount term in the first fraction dt_w subtracts from word counts. When $d = 0$ the Pitman-Yor process reduces to a Dirichlet distribution, and we only have the usual additive pseudo-counts of the Dirichlet distribution. If $d > 0$, we have discounts, and the additive term can be understood as interpolation with the uniform distribution. Further assuming that $t_w = 1$, i.e. only one table serves dish w, we obtain absolute discounting. In the appendix we show that t_w's grow as $O(c_w^d)$ instead.

4 Hierarchical Pitman-Yor Language Models

In the previous section we already see how we can obtain absolute discounting and interpolation using the Pitman-Yor process. In this section we describe a language model based on a hierarchical extension of the Pitman-Yor process, and show that we can recover interpolated Kneser-Ney as approximate inference in the model. The hierarchical Pitman-Yor process is a generalization of the hierarchical Dirichlet process, and the derivation described here is a straightforward generalization of those in Teh et al. (2006).

We are interested building a model of distributions over the current word given various contexts. Given a context $\mathbf{u}$ consisting of a sequence of up to $n-1$ words, let $G_{\mathbf{u}}(w)$ be the distribution over the current word w. Since we wish to infer $G_{\mathbf{u}}(w)$ from our training corpus, the Bayesian nonparametric approach we take here is to assume that $G_{\mathbf{u}}(w)$ is itself a random variable. We use a Pitman-Yor process as the prior for $G_{\mathbf{u}}(w)$, in particular,

$$G_{\mathbf{u}}(w) \sim \mathrm{PY}(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}(w)) \tag{12}$$

where $\pi(\mathbf{u})$ is the suffix of $\mathbf{u}$ consisting of all but the first word. The strength and discount parameters depend on the length of the context, just as in interpolated Kneser-Ney where the same discount parameter is used for each length of context. The base distribution is $G_{\pi(\mathbf{u})}(w)$, the distribution over the current word given all but the earliest word in the context. That is, we believe that without observing any data the earliest word is the least important in determining the distribution over the current word. Since we do not know $G_{\pi(\mathbf{u})}(w)$ either, We recursively place a prior over $G_{\pi(\mathbf{u})}(w)$ using (12), but now with parameters $\theta_{|\pi(\mathbf{u})|}, d_{|\pi(\mathbf{u})|}$ and base distribution $G_{\pi(\pi(\mathbf{u}))}(w)$ and so on. Finally the prior for $G_{\emptyset}(w)$, the distribution over current word given the empty context $\emptyset$ is given a prior of

$$G_{\emptyset}(w) \sim \mathrm{PY}(d_0, \theta_0, G_0) \tag{13}$$

where G_0 is the global base distribution, which is assumed to be uniform over the vocabulary W of V words.

The structure of the prior is that of a suffix tree of depth n, where each node corresponds to a context consisting of up to $n-1$ words, and each child corresponds to adding a different word to the beginning of the context. As we shall see, this choice of the prior structure expresses our belief that words appearing later in a context have more influence over the distribution over the current word.

We can apply the Chinese restaurant representation to the hierarchical Pitman-Yor language model to draw words from the prior. The basic observation is that to draw words from $G_{\mathbf{u}}(w)$ using the Chinese restaurant representation the only operation we need of the base distribution $G_{\pi(\mathbf{u})}(w)$ is to draw words from it. Since $G_{\pi(\mathbf{u})}(w)$ is itself distributed according to a Pitman-Yor process, we can use another Chinese restaurant to draw words from that. This is recursively applied until we need a draw from the global base distribution G_0, which is easy since it assigns equal probability to each word in the vocabulary. In summary

we have a restaurant corresponding to each $G_{\mathbf{u}}(w)$, which has an unbounded number tables and has a sequence of customers corresponding to words drawn from $G_{\mathbf{u}}(w)$. Each table is served a dish (corresponds to a word drawn from the base distribution $G_{\pi(\mathbf{u})}(w)$), while each customer eats the dish served at the table she sat at (the word drawn for her is the same as the word drawn for the table). The dish served at the table is in turn generated by sending a customer to the parent restaurant in a recursive fashion. Notice that there are two types of customers in each restaurant, the "independent" ones arriving by themselves, and those sent by a child restaurant. Further, every table at every restaurant is associated with a customer in the parent restaurant, and every dish served in the restaurants can be traced to a draw from G_0 in this way.

In the rest of the paper we index restaurants (contexts) by $\mathbf{u}$, dishes (words in our vocabulary) by w, and tables by k. Let $c_{\mathbf{u}wk}$ be the number of customers in restaurant $\mathbf{u}$ sitting at table k eating dish w ($c_{\mathbf{u}wk} = 0$ if table k does not serve dish w), and let $t_{\mathbf{u}w}$ be the number of tables in restaurant $\mathbf{u}$ serving dish w. We denote marginal counts by dots, for example $c_{\mathbf{u}\cdot k}$ is the number of customers sitting around table k in restaurant $\mathbf{u}$, $c_{\mathbf{u}w\cdot}$ is the number eating dish w in restaurant $\mathbf{u}$ (number of occurrences of word w in context $\mathbf{u}$), and $t_{\mathbf{u}\cdot}$ is the number of tables in restaurant $\mathbf{u}$.

In language modelling, the training data consists knowing the number of occurrences of each word w after each context $\mathbf{u}$ of length $n-1$ (we pad the beginning of each sentence with begin-sentence symbols). This corresponds to knowing the number $c_{\mathbf{u}w\cdot}$ of customers eating dish w in restaurant $\mathbf{u}$, for each $\mathbf{u}$ with length $n-1$. These customers are the only "independent" ones in the restaurants, the others are all sent by child restaurants. As a result only the values of $c_{\mathbf{u}w\cdot}$ with $|\mathbf{u}| = n-1$ are fixed by the training data, other values vary depending on the seating arrangement in each restaurant, and we have the following relationships among the $c_{\mathbf{u}w\cdot}$'s and $t_{\mathbf{u}w}$:

$$\begin{cases} t_{\mathbf{u}w} = 0 & \text{if } c_{\mathbf{u}w\cdot} = 0; \\ 1 \leq t_{\mathbf{u}w} \leq c_{\mathbf{u}w\cdot} & \text{if } c_{\mathbf{u}w\cdot} > 0; \end{cases} \qquad c_{\mathbf{u}w\cdot} = \sum_{\mathbf{u}':\pi(\mathbf{u}')=\mathbf{u}} t_{\mathbf{u}'w} \tag{14}$$

Algorithm 1 gives details of how the Chinese restaurant representation can be used to generate words given contexts in terms of a function which draws a new word by calling itself recursively. Notice the self-reinforcing property of the hierarchical Pitman-Yor language model: the more a word w has been drawn in context $\mathbf{u}$, the more likely will we draw w again in context $\mathbf{u}$. In fact word w will be reinforced for other contexts that share a common suffix with $\mathbf{u}$, with the probability of drawing w increasing as the length of the common suffix increases. This is because w will be more likely under the context of the common suffix as well.

The Chinese restaurant representation can also be used for inference in the hierarchical Pitman-Yor language model. Appendix A.4 gives the joint distribution over seating arrangements in the restaurants,

Table 1: Routine to draw a new word given context $\mathbf{u}$ using the Chinese restaurant representation.

Function DrawWord($\mathbf{u}$):

- If $j = 0$, return word $w \in W$ with probability $G_0(w) = 1/V$.
- Else with probabilities proportional to:

  $\max(0, c_{\mathbf{u}wk} - d_{|\mathbf{u}|})$: sit customer at table k (increment $c_{\mathbf{u}wk}$); return word w.

  $\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|}t_{\mathbf{u}\cdot}$: let $w \leftarrow$ DrawWord($\pi(\mathbf{u})$); sit customer at an unoccupied table k^{new} serving dish w (increment $t_{\mathbf{u}w}$, set $c_{\mathbf{u}wk^{\text{new}}} = 1$); return w.

while Appendix B gives an inference routine based upon Gibbs sampling which returns samples from the posterior distribution over seating arrangements. Appendix C gives an auxiliary sampling routine for the strength and discount parameters. Given a sample from the posterior seating arrangement and parameters, the predictive probability of the next draw from $G_{\mathbf{u}}(w)$ is given by recursively applying (11). For the global base distribution the predictive probability is simply

$$P_0^{\text{HPY}}(w \,|\, \text{seating arrangement}) = G_0(w) \tag{15}$$

while for each context $\mathbf{u}$ the predictive probability of the next word after context $\mathbf{u}$ given the seating arrangement is

$$P_{\mathbf{u}}^{\text{HPY}}(w \,|\, \text{seating arrangement}) = \frac{c_{\mathbf{u}w\cdot} - d_{|\mathbf{u}|}t_{\mathbf{u}w}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} + \frac{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|}t_{\mathbf{u}\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} P_{\pi(\mathbf{u})}^{\text{HPY}}(w \,|\, \text{seating arrangement}) \tag{16}$$

To form our n-gram probability estimates, we simply average (16) over the posterior of the seating arrangements and parameters.

From (16) the correspondence to interpolated Kneser-Ney is now straightforward. Suppose that the strength parameters are all $\theta_{|\mathbf{u}|} = 0$. Consider an approximate inference scheme for the hierarchical Pitman-Yor language model where we simply set

$$t_{\mathbf{u}w} = \begin{cases} 0 & \text{if } c_{\mathbf{u}w\cdot} = 0; \\ 1 & \text{if } c_{\mathbf{u}w\cdot} \geq 1; \end{cases} \qquad\qquad c_{\mathbf{u}w\cdot} = \sum_{\mathbf{u}':\pi(\mathbf{u}')=\mathbf{u}} t_{\mathbf{u}'w} \tag{17}$$

(17) says that there is at most one table in each restaurant serving each dish. The predictive probabilities given by (16) now directly reduces to the predictive probabilities given by interpolated Kneser-Ney (4). As a result we can interpret interpolated Kneser-Ney as this particular approximate inference scheme in the hierarchical Pitman-Yor language model.

Appendix A describes some additional properties of the hierarchical Pitman-Yor language model.

5 Experimental Results

We performed experiments on the hierarchical Pitman-Yor language model under two circumstances: trigrams on a 16 million word corpus derived from APNews[1]. and bigrams on a 1 million word corpus derived from the Penn TreeBank portion of the WSJ dataset[2] On the trigram APNews dataset, we compared our model to interpolated and modified Kneser-Ney on cross-entropies and studied the growth of discounts as functions of trigram counts. On the simpler bigram WSJ dataset, we studied the effect on cross-entropies of varying the strength and discount parameters and related our results to the hierarchical Dirichlet language model. We also showed that our proposed sampler converges very quickly.

We compared the hierarchical Pitman-Yor language model against interpolated Kneser-Ney and modified Kneser-Ney with $c^{(\text{max})} = 2$ and 3 on the trigram APNews dataset. We varied the training set size between approximately 2 million and 14 million words by six equal increments. For all three versions of interpolated Kneser-Ney, we first determined the discount parameters by conjugate gradient descent in the

[1]This is the same dataset as in Bengio et al. (2003). The training, validation and test sets consist of about 14 million, 1 million and 1 million words respectively, while the vocabulary size is 17964.

[2]This is the same dataset as in Xu and Jelinek (2004) and Blitzer et al. (2005). We split the data into training, validation and test sets by randomly assigning bigrams to each with probabilities .6, .2, .2 respectively. The vocabulary size is 10000.

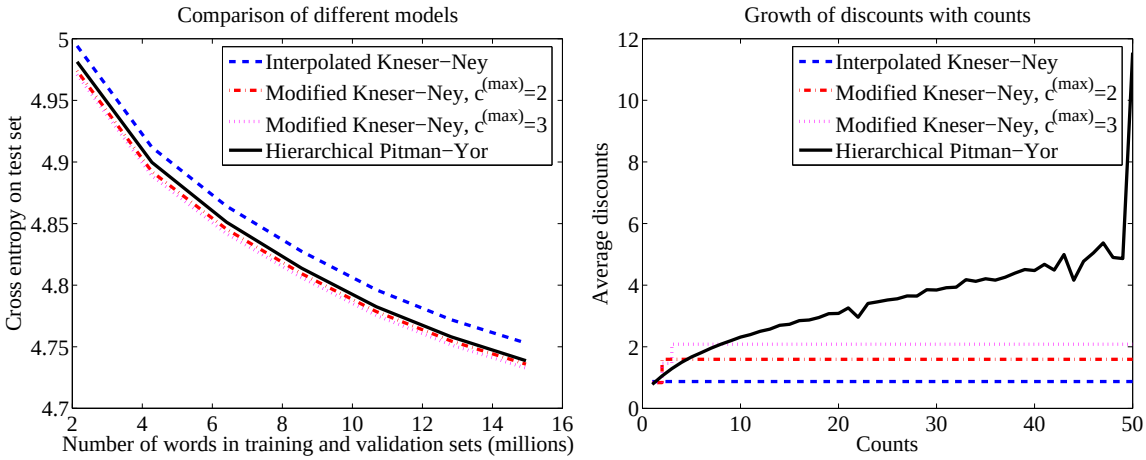

Figure 1: Left: Cross-entropy on test set (lower better). The training set size is varied on the x-axis while the y-axis shows the cross-entropy (in natural logarithm). Each line corresponds to a language model. Right: Average discount as a function of trigram counts. For the hierarchical Pitman-Yor language model the reported discount for a count c is d_2 times the number of tables averaged over posterior samples of seating arrangement and over all trigrams that occurred c times in the full training set. The last entry is averaged over all trigrams that occurred at least 50 times.

cross-entropy on the validation set (Chen and Goodman 1998). At the optimal values, we folded the validation set into the training set to obtain the final trigram probability estimates. For the hierarchical Pitman-Yor language model we inferred the posterior distribution over seating arrangement and the strength and discount parameters given both the training and validation set[3]. We used a sampling routine which alternates between updating the seating arrangement (Appendix B) and the parameters (Appendix C). Since the posterior is very well-behaved, we only used 125 iterations for burn-in, and 175 iterations to collect posterior samples. On the full 15 million word training set (includes data from the validation set) this took less than 2 hours on 1.4Ghz Pentium III's.

The cross-entropy results are given in Figure 1 (left). As expected the hierarchical Pitman-Yor language model performs better than interpolated Kneser-Ney, supporting our claim that interpolated Kneser-Ney is just an approximation inference scheme in the hierarchical Pitman-Yor language model. Interestingly, the hierarchical Pitman-Yor language model performs slightly worse than the modified versions of Kneser-Ney. In Figure 1 (right) we showed the average discounts returned by the hierarchical Pitman-Yor language model as a function of the observed count of trigrams in the training set. We also showed the discounts returned by the interpolated and modified Kneser-Ney models. We see that the average discounts returned by the hierarchical Pitman-Yor language model grows slowly as a function of the trigram counts. Appendix A.3 shows that the average discount grows as a power-law with index d_3 and this is reflected well by the figure. The growth of the average discounts also matches relatively closely with that of the optimal discounts in Figure 25 of Chen and Goodman (1998),

In the second set of experiments we investigated the effect of the strength and discount parameters on the

[3]This is one of the advantages of a Bayesian procedure, we need not use a separate validation set to determine parameters of the model. Instead we can include the validation set in the training set and infer both the hidden variables and parameters in a single phase of training.

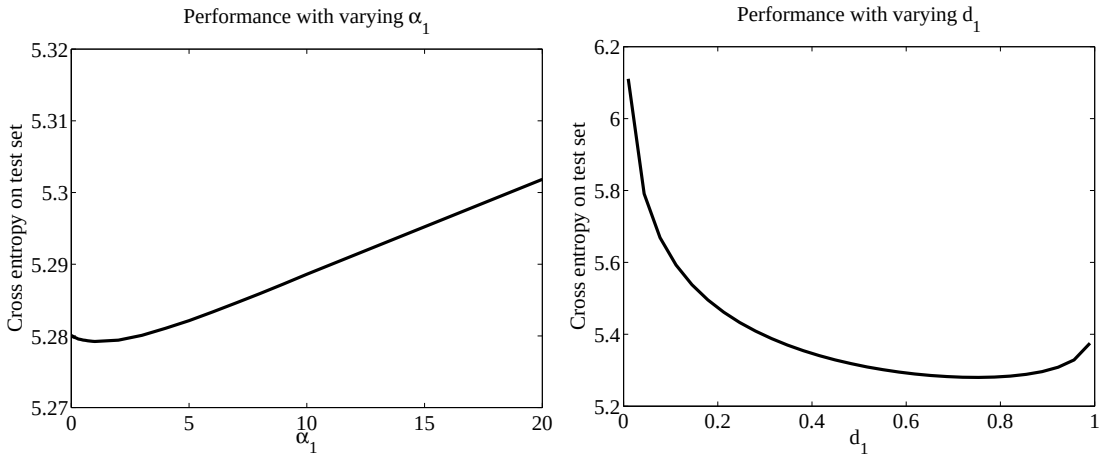

Figure 2: Left: Cross entropy on test data as θ_1 is varied and with other parameters held at the optimal settings found by interpolated Kneser-Ney. Right: Varying d_1 instead.

performance of the hierarchical Pitman-Yor language model in case of bigrams on a 1 million word dataset. We first found optimal settings for the four parameters θ_0, θ_1, d_0 and d_1 by optimizing the performance of interpolated Kneser-Ney on a validation set[4]. Then for each parameter we varied it while keeping the others fixed at its optimal. We found that the model is only sensitive to d_1 but is insensitive to d_0, θ_0 and θ_1. Results for θ_1 and d_1 are shown in Figure 2. The model is insensitive to the strength parameters because in most cases these are very small compared with the count and discount terms in the predictive probabilities (16). In fact, we had repeated both trigram and bigram experiments with θ_m set to 0 for each m, and the results were identical. The model is insensitive to d_0 for two reasons: its effect on the predictive probabilities (16) is small, and most values of $t_{\emptyset w} = 1$ or 2 so the discount term corresponding to d_0 in (16) is cancelled out by the additive term involving the uniform base distribution G_0 over the vocabulary. When $d_1 = 0$ the hierarchical Pitman-Yor language model reduces down to the hierarchical Dirichlet language model of MacKay and Peto (1994), and as seen in Figure 2 (right) this performs badly.

6 Discussion

We have described using a hierarchical Pitman-Yor process as a language model and derived estimates of n-gram probabilities based on this model that are generalizations of interpolated Kneser-Ney. Setting some variables and parameters to specific values reduces the formula for n-gram probabilities to those in interpolated Kneser-Ney, hence we may interpret interpolated Kneser-Ney as approximate inference in this model. In experiments we have also shown that cross-entropies attained by the model are better than those obtained by interpolated Kneser-Ney.

The hierarchical Dirichlet language model of MacKay and Peto (1994) was an inspiration for our work. Though MacKay and Peto (1994) had the right intuition to look at smoothing techniques as the outcome of hierarchical Bayesian models, the use of the Dirichlet distribution as a prior was shown to lead to non-competitive cross-entropy results. As a result the language modelling community seemed to have dismissed

[4]We can use average values of the parameters as returned by the hierarchical Pitman-Yor language model as well, the parameter values are similar and does not affect our results.

Bayesian methods as theoretically nice but impractical methods. Our model is a nontrivial but direct generalization of the hierarchical Dirichlet language model that gives state-of-the-art performance. We have shown that with a suitable choice of priors (namely the Pitman-Yor process), Bayesian methods can be competitive with the best smoothing techniques. In fact we have shown that one of the best smoothing techniques, namely interpolated Kneser-Ney, is a great approximation to a Bayesian model.

The hierarchical Pitman-Yor process is a natural generalization of the recently proposed hierarchical Dirichlet process (Teh et al. 2006). The hierarchical Dirichlet process was proposed to solve a clustering problem instead and it is interesting to note that such a direct generalization leads us to a well-established solution for a different problem, namely interpolated Kneser-Ney. This indicates the naturalness of this class of models. Both the hierarchical Dirichlet process and the hierarchical Pitman-Yor process are examples of Bayesian nonparametric processes. These have recently received much attention in the statistics and machine learning communities because they can relax previously strong assumptions on the parametric forms of Bayesian models yet retain computational efficiency, and because of the elegant way in which they handle the issues of model selection and structure learning in graphical models.

The hierarchical Pitman-Yor language model is only the first step towards comprehensive Bayesian solutions to many tasks in natural language processing. We envision that a variety of more sophisticated models which make use of the hierarchical Pitman-Yor process can be built to solve many problems. Foremost in our agenda are extensions of the current model that achieve better cross-entropy for language modelling, and verifying experimentally that this translates into reduced word error rates for speech recognition.

Acknowledgement

I wish to thank the Lee Kuan Yew Endowment Fund for funding, Joshua Goodman for answering many questions regarding interpolated Kneser-Ney and smoothing techniques, John Blitzer and Yoshua Bengio for help with datasets and Hal Daume III for comments on an earlier draft.

References

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Berger, A., Della Pietra, S., and Della Pietra, V. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1).

Bilmes, J. and Kirchhoff, K. (2003). Factored language models and generalized parallel backoff. In *Proceedings of the Human Language Technology Conference*.

Blitzer, J., Globerson, A., and Pereira, F. (2005). Distributed latent variable models of lexical cooccurrences. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, volume 10.

Brand, M. (1999). Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11(5).

Charniak, E. (2001). Immediate head parsing for language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, volume 39.

Chen, S. and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University.

Chen, S. and Rosenfeld, R. (2000). A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing*, 8(1).

Church, K. and Gale, W. (1991). A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer speech and Language*, 5:19–54.

Gelman, A., Carlin, J., Stern, H., and Rubin, D. (1995). *Bayesian data analysis*. Chapman & Hall, London.

Goldwater, S., Griffiths, T., and Johnson, M. (2006). Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems*, volume 18.

Goodman, J. (2001). A bit of progress in language modeling. Technical Report MSR-TR-2001-72, Microsoft Research.

Goodman, J. (2004). Exponential priors for maximum entropy models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Hsu, L. and Shiue, P.-S. (1998). A unified approach to generalized stirling numbers. *Advances in Applied Mathematics*, 20:366–384.

Ishwaran, H. and James, L. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173.

Jordan, M. (2005). Dirichlet processes, Chinese restaurant processes and all that. Tutorial presentation at the NIPS Conference.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Propabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, volume 18.

MacKay, D. and Peto, L. (1994). A hierarchical Dirichlet language model. *Natural Language Engineering*.

McCallum, A., Freitag, D., and Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591–598. Morgan Kaufmann, San Francisco, CA.

Pitman, J. (2002). Combinatorial stochastic processes. Technical Report 621, Department of Statistics, University of California at Berkeley. Lecture notes for St. Flour Summer School.

Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25:855–900.

Rosenfeld, R. (1994). *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, Computer Science Department, Carnegie Mellon University.

Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8).

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650.

Teh, Y., Jordan, M., Beal, M., and Blei, D. (2006). Hierarchical Dirichlet processes. *To appear in Journal of the American Statistical Association*.

Vinciarelli, A., Bengio, S., and Bunke, H. (2004). Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):709–720.

Xu, P. and Jelinek, F. (2004). Random forests in language modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

A Some Additional Properties

A.1 Out-of-Vocabulary Words

Following the conventional approach in language modelling we had limited our vocabulary to the words that appeared in our training and test corpora. When we do not have a closed vocabulary, it is possible to extend the hierarchical Pitman-Yor language model to handle previously unseen words. In fact, the only change required is to make the global base distribution G_0 be a distribution over all possible (seen or unseen) words. The rest of the model, the predictive probabilities and the inference algorithms do not need to be altered. An example of such a G_0 would be a hidden Markov model which emits finite sequences of letters (words). Such a model for words have been used in, e.g. Vinciarelli et al. (2004).

A.2 Marginal Constraints

Chen and Goodman (1998) derived the lower-order m-gram estimates of interpolated Knerser-Ney by satisfying the marginal word distribution constraints (6). We show that the hierarchical Pitman-Yor language model gives m-gram estimates that also satisfy these constraints when the strength parameters $\theta_m = 0$ for all $m < n$. In fact, the constraints are satisfied for every seating arrangement in the restaurants.

Let w' and w be words and $\mathbf{u}'$ be a word sequence of length $n-2$. The marginal constraints are

$$P^{\text{emp}}(\mathbf{u}'w) = \sum_{w'} P^{\text{emp}}(w'\mathbf{u}')P^{\text{HPY}}_{w'\mathbf{u}'}(w) \tag{18}$$

where $P^{\text{emp}}(\mathbf{u}) = \frac{c_{\mathbf{u}\cdot}}{\sum_{\mathbf{u}} c_{\mathbf{u}\cdot}}$ in our count notation. Plugging in the predictive probabilities (16) into (18) and simplifying,

$$\begin{aligned}
c_{\mathbf{u}'w\cdot} &= \sum_{w'}(c_{w'\mathbf{u}'w\cdot} - d_m t_{w'\mathbf{u}'w}) + d_m t_{w'\mathbf{u}'\cdot} P^{\text{HPY}}_{\pi(w'\mathbf{u}')}(w) \\
&= c_{\cdot\mathbf{u}'w\cdot} - d_m t_{\cdot\mathbf{u}'w} + d_m t_{\cdot\mathbf{u}'\cdot} P^{\text{HPY}}_{\mathbf{u}'}(w) \\
0 &= -t_{\cdot\mathbf{u}'w} + t_{\cdot\mathbf{u}'\cdot} P^{\text{HPY}}_{\mathbf{u}'}(w) \\
P^{\text{HPY}}_{\mathbf{u}'}(w) &= \frac{t_{\cdot\mathbf{u}'w}}{t_{\cdot\mathbf{u}'\cdot}} = \frac{c_{\mathbf{u}'w\cdot}}{c_{\mathbf{u}'\cdot\cdot}}
\end{aligned} \tag{19}$$

where $c_{\cdot \mathbf{u}' w \cdot} = c_{\mathbf{u}' w \cdot \cdot}$ since both count the number of occurrences of the word sequence $\mathbf{u}' w$ in the training corpus, and we have used (14). As in interpolated Kneser-Ney, the actual lower order m-gram probabilities used discounts the word counts $c_{\mathbf{u}' w \cdot}$ in (19) and interpolates with even lower order m-gram probabilities recursively.

A.3 Power-Law Discounting

Goldwater et al. (2006) noted that the numbers of customers sitting around tables in a Pitman-Yor process with parameters (θ, d) follows a power-law distribution with index $1 + d$. This is proposed as a general mechanism for linguistic models to produce power-law distributions commonly seen in natural languages. The Pitman-Yor process also produces another power-law behaviour that is useful in our situation: when $d > 0$ the expected number of tables in a Pitman-Yor process scales as $O(c^d)$ where c is the number of customers. This implies that the a priori amount of discounts used in the hierarchical Pitman-Yor language model follows a power-law growth. This contrasts with absolute discounting which assumes a fixed amount of discount, but still grows slowly compared with c, and is consistent with the findings in Chen and Goodman (1998) and Church and Gale (1991) that the amount of discount should increase slowly.

Here we will directly derive the $O(c^d)$ growth of the number of tables. Let $t(c)$ be the expected number of tables occupied in a restaurant with c customers. Clearly $t(1) = 1$. Given $t(c)$, the expected number of tables with $c + 1$ customers is the expected number occupied by the first c customers plus the probability of the last customer sitting at a new table. From (10) this gives:

$$t(c+1) = t(c) + \frac{\theta + dt(c)}{\theta + c} = \left(1 + \frac{d}{\theta + c}\right) t(c) + \frac{\theta}{\theta + c} \tag{20}$$

First we lower bound the growth of $t(c)$. dropping the last term in (20),

$$t(c+1) \geq \left(1 + \frac{d}{\theta + c}\right) t(c) = \prod_{i=1}^{c} \left(1 + \frac{d}{\theta + i}\right) \tag{21}$$

Taking logarithms, and the limit of large c,

$$\log(t(c+1)) \geq \sum_{i=1}^{c} \log\left(1 + \frac{d}{\theta + i}\right) \approx \sum_{i=1}^{c} \frac{d}{i} + constant \approx d \log c + constant \tag{22}$$

since $\log(1 + x) \approx x$ for small x, and $\frac{d}{\theta+i} \approx \frac{d}{i}$ for large i. Exponentiating back, we get $t(c) \geq O(c^d)$. In particular $t(c) \gg constant$ for large c so the last term of (20) is negligible to begin with and we conclude that $t(c) = O(c^d)$.

A.4 Joint Distribution over Seating Arrangements

In this appendix we give the joint distribution over seating arrangement in the Chinese restaurant representation of the hierarchical Pitman-Yor language model explicitly. This helps to clarify what we mean when we consider distributions over seating arrangement, and will be useful in deriving the sampling schemes.

The seating arrangement in each restaurant consists of the number of customers, the partitioning of customers into those sitting around each table, and the dish served at each table. From this we can derive other information, for example the number of occupied tables in each restaurant and the number of customers eating each dish. Recall that restaurants indexed by $\mathbf{u}$ correspond to contexts, dishes indexed by w correspond

to words from the vocabulary, $c_{\mathbf{u}wk}$ is the number of customers in restaurant $\mathbf{u}$ eating dish w at table k while $t_{\mathbf{u}w}$ is the number of tables serving dish w. We use dots to denote marginal counts.

The probability for a particular seating arrangement can be derived by accounting for each event as each customer visits her restaurant, sits at some table, and eat the dish assigned for the table. The probabilities of events are given by the terms in (10). Collecting like terms, we get

$$P(\text{seating arrangement}) = \prod_w G_0(w)^{c_{0w\cdot}} \prod_{\mathbf{u}} \frac{[\theta_{|\mathbf{u}|}]_{d_{|\mathbf{u}|}}^{(t_{\mathbf{u}\cdot})}}{[\theta_{|\mathbf{u}|}]_1^{(c_{\mathbf{u}\cdot\cdot})}} \prod_w \prod_{k=1}^{t_{\mathbf{u}\cdot}} [1 - d_{|\mathbf{u}|}]_1^{(c_{\mathbf{u}wk}-1)} \tag{23}$$

where the number $[a]_b^{(c)}$ is a generalized factorial:

$$[a]_b^{(0)} = [a]_b^{(-1)} = 1 \tag{24}$$

$$[a]_b^{(c)} = a(a+b)\cdots(a+(c-1)b) = \frac{\Gamma(a/b+c)}{b^c\Gamma(a/b)} \qquad \text{for } c > 0 \tag{25}$$

The G_0 term in (23) gives the probability of drawing each dish from G_0, with $c_{0w\cdot}$ the number of times dish w was drawn from G_0. The denominator in the fraction collects the denominator terms in (10), while the numerator collects terms corresponding to customers sitting at new tables. Finally the last term collects terms corresponding to customers sitting at already occupied tables. Notice that the denominator contains a $\theta_{|\mathbf{u}|}$ term which may be problematic when $\theta_{|\mathbf{u}|} = 0$. Since there is one such term each in the numerator and denominator we can cancel them out and (23) is still well-defined.

A.5 Joint Distribution over Numbers of Customers and Tables

Notice that as a function of the seating arrangement the predictive probabilities (16) of the hierarchical Pitman-Yor language model depend only on $c_{\mathbf{u}w\cdot}$ and $t_{\mathbf{u}w}$, the number of customers eating and the number of tables serving each dish w in each restaurant $\mathbf{u}$. Here we will derive a joint distribution over $c_{\mathbf{u}w\cdot}$ and $t_{\mathbf{u}w}$ only, summing out the specific seating arrangements of the customers.

It is sufficient to restrict ourselves to only consider the seating arrangements of those customers in a particular restaurant eating a particular dish. Let there be c such customers and t tables, and let $A(c,t)$ be the set of all seating arrangements of c customers among t tables. Given $a \in A(c,t)$ let c_{ak} be the number of customers sitting around the k^{th} table in the seating arrangement a. Let the discount parameter be simply d. We show that

$$\sum_{a \in A(c,t)} \prod_{k=1}^{t} [1-d]_1^{(c_{ak}-1)} = s_d(c,t) \tag{26}$$

where s_d are generalized Stirling numbers of type $(-1,-d,0)$ (Hsu and Shiue 1998), defined recursively as:

$$s_d(1,1) = s_d(0,0) = 1 \tag{27}$$

$$s_d(c,0) = s_d(0,t) = 0 \qquad \text{for } c,t > 0 \tag{28}$$

$$s_d(c,t) = 0 \qquad \text{for } t > c \tag{29}$$

$$s_d(c,t) = s_d(c-1,t-1) + (c-1-dt)s_d(c-1,t) \qquad \text{for } 0 < t \leq c \tag{30}$$

Now summing over seating arrangements for a particular restaurant $\mathbf{u}$ and a particular dish w in (23), and using (26) for each pair $c_{\mathbf{u}w\cdot}$ and $t_{\mathbf{u}w}$, this gives the following joint distribution over $c_{\mathbf{u}w\cdot}$'s and $t_{\mathbf{u}w}$'s:

$$P((c_{\mathbf{u}w\cdot}, t_{\mathbf{u}w} \,:\, \text{all } \mathbf{u}, w)) = \prod_w G_0(w)^{c_{0w\cdot}} \prod_{\mathbf{u}} \frac{[\theta_{|\mathbf{u}|}]_{d_{|\mathbf{u}|}}^{(t_{\mathbf{u}\cdot})}}{[\theta_{|\mathbf{u}|}]_1^{(c_{\mathbf{u}\cdot\cdot})}} \prod_w s_{d_{|\mathbf{u}|}}(c_{\mathbf{u}w\cdot}, t_{\mathbf{u}w}) \tag{31}$$

We will derive (26) by induction. The base cases in (27), (28) and (29) are easy to verify; for example, c customers cannot occupy $t > c$ tables so $s_d(c,t) = 0$ when $c > t$. For the general case of $0 < t \leq c$, assume that (26) holds for all $c' \leq c, t' \leq t$ with either $c' < c$ or $t' < t$. We split the set $A(c,t)$ into $t+1$ subsets depending on where the last customer sits:

- Let $A_0(c,t)$ be the subset consisting of those seating arrangements where the last customer sits by herself. Removing this customer leaves us with a seating arrangement of the other $c-1$ customers around $t-1$ tables. In fact it is easy to see that this operation of removing the last customer is a one-to-one correspondence between seating arrangements in $A_0(c,t)$ and $A(c-1,t-1)$. Further the last customer does not contribute any term to (26).
- For $k' = 1, \ldots, t$ let $A_{k'}(c,t)$ be the subset consisting of those seating arrangements where the last customer sits at table k', and table k' has at least two customers. Removing this customer does not make table k' unoccupied and leaves us with a seating arrangement of the other $c-1$ customers around t tables, and similarly we get a one-to-one correspondence between $A_{k'}(c,t)$ and $A(c-1,t)$. Further, this last customer contributes a term $c_{ak'} - d$ to (26) for each $a \in A(c-1,t)$.

Expanding (26) into these $t+1$ subsets, we get

$$\begin{aligned}
&\sum_{a \in A(c,t)} \prod_{k=1}^{t} [1-d]_1^{(c_{ak}-1)} \\
&= \sum_{a \in A_0(c,t)} \prod_{k=1}^{t} [1-d]_1^{(c_{ak}-1)} + \sum_{k'=1}^{t} \sum_{a \in A_{k'}(c,t)} \prod_{k=1}^{t} [1-d]_1^{(c_{ak}-1)} \\
&= \sum_{a \in A(c-1,t-1)} \prod_{k=1}^{t-1} [1-d]_1^{(c_{ak}-1)} + \sum_{k'=1}^{t} \sum_{a \in A(c-1,t)} (c_{ak'} - d) \prod_{k=1}^{t-1} [1-d]_1^{(c_{ak}-1)} \\
&= \sum_{a \in A(c-1,t-1)} \prod_{k=1}^{t} [1-d]_1^{(c_{ak}-1)} + \sum_{a \in A(c-1,t)} \sum_{k'=1}^{t} (c_{ak'} - d) \prod_{k=1}^{t} [1-d]_1^{(c_{ak}-1)} \\
&= s_d(c-1,t-1) + (c-1-dt) s_d(c-1,t)
\end{aligned} \tag{32}$$

Based on (31), it is possible to construct either sampling methods or approximate inference methods to obtain posterior estimates for $c_{\mathbf{u}w\cdot}$'s and $t_{\mathbf{u}w}$'s directly. We expect these to converge quickly and give good estimates since it can be shown that (31) is log-concave as a function jointly in $c_{\mathbf{u}w\cdot}$'s and $t_{\mathbf{u}w}$'s. In small preliminary experiments we have found loopy belief propagation to converge within a few iterations. However each iteration is computationally intensive as each $c_{\mathbf{u}w\cdot}$ and $t_{\mathbf{u}w}$ can potentially take on many values, and it is expensive to compute the generalized Stirling numbers $s_d(c,t)$. As a result we chose to use a sampling method based on (23), which converges very quickly in our experiments as well.

B Sampling for Seating Arrangements

We can obtain a Gibbs sampler for the hierarchical Pitman-Yor language model directly using the Chinese restaurant representation (i.e. using (23)). This is just an extension of the Chinese restaurant franchise sampler in Teh et al. (2006). The seating arrangement for each restaurant consists of: the number of customers, the number of tables, the table at which each customer sits, the dish served at each table and the dish each customer eats. The Gibbs sampler only keeps track of which table each customer sits at, while the other pieces of information in the seating arrangement can be reconstructed from this. The sampler then iterates over all customers present in each restaurant, resampling the table at which each customer sits. This resampling can be performed most easily using two routines: a RemoveCustomer routine that removes a customer from the restaurant, and an AddCustomer routine which adds the customer back into the restaurant, sitting her at some random table using (10).

Unfortunately in case of a language model which needs to be trained on very large corpora, the above

Table 2: Operations for sampling seating arrangement in the hierarchical Pitman-Yor language model.

Function WordProbability($\mathbf{u}$,w):
Returns the probability $P^{HPY}_{\mathbf{u}}(w)$ that the next word after context $\mathbf{u}$ will be w (computes (16)).
• If $\mathbf{u} = 0$ then return $G_0(w)$.
• Else return $\frac{c_{\mathbf{u}w\cdot} - d_{|\mathbf{u}|}t_{\mathbf{u}w}}{\theta_{|\mathbf{u}|}+c_{\mathbf{u}\cdot\cdot}} + \frac{\theta_{|\mathbf{u}|}+d_{|\mathbf{u}|}t_{\mathbf{u}\cdot}}{\theta_{|\mathbf{u}|}+c_{\mathbf{u}\cdot\cdot}}$DishProbability($\pi(\mathbf{u})$,$w$).

Function AddCustomer($\mathbf{u}$,w):
Adds a new customer eating dish w into restaurant $\mathbf{u}$.
• If $\mathbf{u} = 0$ then increment $c_{0w\cdot}$.
• Else with probabilities proportional to:
$\max(0, c_{\mathbf{u}wk} - d_{|\mathbf{u}|})$: sit customer at k^{th} table in restaurant $\mathbf{u}$ (increment $c_{\mathbf{u}wk}$).
$(\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|}t_{\mathbf{u}\cdot})$DishProbability($\pi(\mathbf{u})$,$w$): sit customer at a new table k^{new} serving dish w in restaurant $\mathbf{u}$ (increment $t_{\mathbf{u}w}$, set $c_{\mathbf{u}wk^{\text{new}}} = 1$); AddCustomer($\pi(\mathbf{u})$,$w$).

Function RemoveCustomer($\mathbf{u}$,w):
Removes a customer eating dish w from restaurant $\mathbf{u}$.
• If $\mathbf{u} = 0$ then decrement $c_{0w\cdot}$.
• Else with probabilities proportional to:
$c_{\mathbf{u}wk}$: remove a customer from k^{th} table in restaurant $\mathbf{u}$ (decrement $c_{\mathbf{u}wk}$).
• If as a result the k^{th} table becomes unoccupied then RemoveCustomer($\pi(\mathbf{u})$,w).

sampler requires far too much storage space since it needs to represent each customer explicitly. We use an alternative sampler which requires much less storage space since it does not represent the actual table at which each customer sits, but only the number of customers sitting around each table in each restaurant serving each dish.

The routines for sampling the seating arrangement are outlined in Algorithm 2. Notice that in language modelling we always know the dish served to each customer (since we always know the identity of each word in our corpus), and the only piece of information we need to infer is which table, among those serving that dish, did the customer sat at. The key insight to our algorithm is that given the dish a customer eats,

the actual identity of the table at which a customer sits has no effect on the likelihood of the data. Thus during the AddCustomer routine, after we chose a table for the customer and increment the number of customers sitting there, we may discards the identity of the table that the customer sits at. Then during the RemoveCustomer routine we reconstruct the identity of the table at which this customer sits at by sampling before removing this customer by decrementing the number of customers sitting at the table.

C Sampling for Parameters

In this appendix we give a simple to implement routine for sampling the strength and discount parameters of the hierarchical Pitman-Yor language model. The routine is based on the joint distribution (23) over seating arrangements. Cancelling out the $\theta_{|\mathbf{u}|}$ terms from the numerator and denominator, this gives, which we shall repeat here:

$$P(\text{seating arrangement}) = \prod_w G_0(w)^{c_{0w\cdot}} \prod_{\mathbf{u}} \frac{[\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|}]_{d_{|\mathbf{u}|}}^{(t_{\mathbf{u}\cdot}-1)}}{[\theta_{|\mathbf{u}|} + 1]_1^{(c_{\mathbf{u}\cdot\cdot}-1)}} \prod_w \prod_{k=1}^{t_{\mathbf{u}\cdot}} [1 - d_{|\mathbf{u}|}]_1^{(c_{\mathbf{u}wk}-1)} \tag{33}$$

Since we can evaluate (33) efficiently for different values of the parameters and for different seating arrangements, a variety of sampling routines can be used, e.g. Metropolis-Hastings or adaptive rejection Metropolis sampling. Instead we used one based on auxiliary variables that is easy to implement using basic operations (the only complex operation required is to sample from a Gamma distribution). We do not believe this sampling routine is better than others, and used it simply because of familiarity.

Our auxiliary variable sampling routine assumes that each discount parameter has prior distribution $d_m \sim \text{Beta}(a_m, b_m)$ while each strength parameter has prior $\theta_m \sim \text{Gamma}(\alpha_m, \beta_m)$. Notice that we have assumed $\theta_m \geq 0$ (rather than $\theta_m > -d_m$). This does not affect our results since it turns out the model is insensitive to θ_m's anyway. When $c_{\mathbf{u}\cdot\cdot} = 0, 1$ the denominator term in (33) is trivial. When $c_{\mathbf{u}\cdot\cdot} \geq 2$ the denominator is

$$\frac{1}{[\theta_{|\mathbf{u}|} + 1]_1^{(c_{\mathbf{u}\cdot\cdot}-1)}} = \frac{\Gamma(\theta_{|\mathbf{u}|} + 1)}{\Gamma(\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot})} = \frac{1}{\Gamma(c_{\mathbf{u}\cdot\cdot} - 1)} \int_0^1 x_{\mathbf{u}}^{\theta_{|\mathbf{u}|}} (1 - x_{\mathbf{u}})^{c_{\mathbf{u}\cdot\cdot}-2}\, dx \tag{34}$$

So we can introduce $x_{\mathbf{u}}$ as an auxiliary variable with conditional distribution,

$$x_{\mathbf{u}} \sim \text{Beta}(\theta_{|\mathbf{u}|} + 1, c_{\mathbf{u}\cdot\cdot} - 1) \tag{35}$$

When $t_{\mathbf{u}\cdot} \geq 2$ the numerator is

$$[\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|}]_{d_{|\mathbf{u}|}}^{(t_{\mathbf{u}\cdot}-1)} = \prod_{i=1}^{t_{\mathbf{u}\cdot}-1} (\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|} i) = \prod_{i=1}^{t_{\mathbf{u}\cdot}-1} \sum_{y_{\mathbf{u}i}=0,1} \theta_{|\mathbf{u}|}^{y_{\mathbf{u}i}} (d_{|\mathbf{u}|} i)^{1-y_{\mathbf{u}i}} \tag{36}$$

so we can introduce $y_{\mathbf{u}i}$ as Bernoulli auxiliary variables with conditional distributions,

$$y_{\mathbf{u}i} \sim \text{Bernoulli}\left(\frac{\theta_{|\mathbf{u}|}}{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|} i}\right) \tag{37}$$

where $\text{Bernoulli}(p)$ is a binary variable taking value 1 with probability p and 0 with probability $1 - p$. Finally the rightmost term in (33), when $c_{\mathbf{u}wk} \geq 2$, is

$$[1 - d_{|\mathbf{u}|}]_1^{(c_{\mathbf{u}wk}-1)} = \prod_{j=1}^{c_{\mathbf{u}wk}-1} (j - d_{|\mathbf{u}|}) = \prod_{j=1}^{c_{\mathbf{u}wk}-1} \prod_{z_{\mathbf{u}wkj}=0,1} (j-1)^{z_{\mathbf{u}wkj}} (1 - d_{|\mathbf{u}|})^{1-z_{\mathbf{u}wkj}} \tag{38}$$

so we can introduce $z_{\mathbf{u}wkj}$ as Bernoulli auxiliary variables with conditional distributions,

$$z_{\mathbf{u}wkj} \sim \text{Bernoulli}\left(\frac{j-1}{j-d_{|\mathbf{u}|}}\right) \tag{39}$$

Given sampled values of all the auxiliary variables, we can now sample the parameters according to their conditional distributions,

$$d_m \sim \text{Beta}\left(a_m + \sum_{\mathbf{u}:|\mathbf{u}|=m, t_{\mathbf{u}\cdot}\geq 2} \sum_{i=1}^{t_{\mathbf{u}\cdot}-1} (1-y_{\mathbf{u}i}), b_m + \sum_{\mathbf{u},w,k:|\mathbf{u}|=m, c_{\mathbf{u}wk}\geq 2} \sum_{j=1}^{c_{\mathbf{u}wk}-1} (1-z_{\mathbf{u}wkj})\right) \tag{40}$$

$$\theta_m \sim \text{Gamma}\left(\alpha_m + \sum_{\mathbf{u}:|\mathbf{u}|=m, t_{\mathbf{u}\cdot}\geq 2} \sum_{i=1}^{t_{\mathbf{u}\cdot}-1} y_{\mathbf{u}i}, \beta_m - \sum_{\mathbf{u}:|\mathbf{u}|=m, t_{\mathbf{u}\cdot}\geq 2} \log x_{\mathbf{u}}\right) \tag{41}$$