A Hierarchical Nonparametric Bayesian Approach to Statistical Language Model Domain Adaptation

Frank Wood and Yee Whye Teh
Gatsby Computational Neuroscience Unit
University College London
London WC1N 3AR, UK
{fwood, ywteh}@gatsby.ucl.ac.uk

Abstract

In this paper we present a *doubly* hierarchical Pitman-Yor process language model. Its bottom layer of hierarchy consists of multiple hierarchical Pitman-Yor process language models, one each for some number of domains. The novel top layer of hierarchy consists of a mechanism to couple together multiple language models such that they share statistical strength. Intuitively this sharing results in the "adaptation" of a latent shared language model to each domain. We introduce a general formalism capable of describing the overall model which we call the graphical Pitman-Yor process and explain how to perform Bayesian inference in it. We present encouraging language model domain adaptation results that both illustrate the potential benefits of our new model and suggest new avenues of inquiry.

1 INTRODUCTION

Consider the problem of statistical natural language model domain adaptation. Statistical language models typically have a very large number of parameters and thus need a large quantity of training data to produce good estimates of those parameters. If one requires a domain-specific model, obtaining a sufficient quantity of domain-specific training data can be both costly and logistically challenging. It is easy, however, to obtain non-domain-specific data, e.g. text from the world wide web. Unfortunately models trained using such data are often ill-suited for domain-specific applications (Rosenfeld 2000). The phrase domain adaptation describes procedures that take a model trained on a large amount of non-specific data and adapt it to work well for a specific domain for which less training data is available.

This paper describes a way to introduce another level of hierarchy to the already hierarchical Pitman-Yor process language model (HPYLM) (Teh 2006) such that a latent shared, non-domain-specific language model as well as domain specific models are estimated together . We call the resulting model the doubly hierarchical Pitman-Yor process language model (DHPYLM) (Section 3). Intuitively such a model is the natural hierarchical Bayesian approach to domain adaptation. Our first contribution is the development of a sensible construction for it. Our second contribution is the development of a new class of nonparametric Bayesian models which we call graphical Pitman-Yor processes and the derivation of generic inference algorithms for them (Section 4). The DHPYLM is a member of this class. Section 5 compares the DHPYLM to previous language model domain adaptation approaches and Section 6 reports on experiments showing the effectiveness of the new model. We start in the next section by reviewing language modeling and the HPYLM in particular.

2 LANGUAGE MODELING REVIEW

In this paper we focus on domain adaptation for Markovian (or n-gram) language models. Such n-gram models are characterized by assuming that the joint probability of a corpus $\mathcal{C} = [w_1 \cdots w_T]$ takes a simplified form

$$P(\mathcal{C}) = \prod_{t=1}^{T} P(w_t | [w_{t-1} \cdots w_{t-n+1}])$$

where the probability of word w_t is conditionally dependent on at most the $n-1$ preceding words. The

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5. Copyright 2009 by the authors.

maximum likelihood estimate of n-gram model parameters is likely to overfit, particularly when there are zero counts, thus regularization of the model through "smoothing" is usually necessary (Chen and Goodman 1998). Recently the best known n-gram smoothing approach, interpolated Kneser-Ney (Kneser and Ney 1995, Chen and Goodman 1998), was shown to be equivalent to approximate inference in the HPYLM (Teh 2006, Goldwater et al. 2007). Further, full posterior inference in the HPYLM was show to outperform interpolated Kneser-Ney. Because of this we chose to use the HPYLM as a building block in our model.

2.1 HIERARCHICAL PITMAN-YOR PROCESS LANGUAGE MODEL

The HPYLM is a hierarchical nonparametric Bayesian language model based on the hierarchical Pitman-Yor process (HPYP) (Teh 2006, Goldwater et al. 2007). The standard definition of a n-gram HPYLM assumes a fixed and finite sized dictionary of L unique words and has the following generative structure:

$$\begin{aligned}
\mathcal{G}_{[]} &\sim \mathrm{PY}(d_0, \alpha_0, \mathcal{U}) \\
\mathcal{G}_{[x_1]} &\sim \mathrm{PY}(d_1, \alpha_1, \mathcal{G}_{[]}) \\
&\vdots \\
\mathcal{G}_{[x_j \cdots x_1]} &\sim \mathrm{PY}(d_j, \alpha_j, \mathcal{G}_{[x_{j-1} \cdots x_1]}) \\
w_t | w_{t-n+1} \cdots w_{t-1} &\sim \mathcal{G}_{[w_{t-n+1} \cdots w_{t-1}]} \qquad (1)
\end{aligned}$$

where the w's are the observed instances of words ("tokens") and the x's range over the unique words ("types") in the dictionary. The notation $\mathcal{G} \sim \mathrm{PY}(d, \alpha, \mathcal{F})$ means that $\mathcal{G}$ is a random distribution drawn from a Pitman-Yor process with concentration parameter α, discount parameter d, and base measure $\mathcal{F}$. One can think of $\mathcal{F}$ as being the mean distribution on which $\mathcal{G}$ is "centered" in the sense that $E[\mathcal{G}(x)] = \mathcal{F}(x)$. Lastly $\mathcal{U}$ is a uniform distribution over word types. In (1) and in other equations like it later in the paper we omit conditioning variables for reasons of readability. For instance $\mathcal{G}_{[]}$ is conditionally dependent on d_0, α_0, and $\mathcal{U}$.

Each $\mathcal{G}_h$ is a distribution over words following a particular context h. It also can be thought of as a parameter vector that fully parameterizes a distribution over words. So, in a slight abuse of notation, we will refer to $\mathcal{G}$ as a parameter (vector) and a distribution interchangeably. The subscripts on the $\mathcal{G}$'s indicate the preceding context, i.e. $\mathcal{G}_{[\mathrm{the,United,States,of}]}$ is the distribution over words following the context "the United States of." In this case the most likely next word is almost certainly "America."

Starting from the top, (1) says that the distribution over words given no contextual information $\mathcal{G}_{[]}$ is centered on the uniform distribution. The remaining lines of (1) say that each distribution over words that follows a particular context is centered on a distribution over words following the same context with one word dropped. The directed graphical model with one vertex per $\mathcal{G}$ and edges to each $\mathcal{G}$ from the $\mathcal{G}$'s that appear in its base distribution forms a suffix tree. In Figure 1 this is the structure that appears in each of the schematic's triangles.

It becomes apparent that this model is an n-gram smoothing model only when one examines the form of the posterior predictive distribution for the next word to appear in a particular context given the entire training corpus $\mathcal{C}$. If we use h to denote context vector consisting of $n-1$ words then the predictive distribution of the word w appearing after h under this model is

$$\begin{aligned}
&P(w|h, \mathcal{C}) \\
&\quad = \quad E\left[\sum_{k=1}^{K} \frac{c_k - d_i}{\alpha + N} \delta(w - \phi_k) + \frac{\alpha + dK}{\alpha + N} P(w|h', \mathcal{C})\right]
\end{aligned}$$

where K, α, d, $[c_k]_{k=1}^{K}$, and $[\phi_k]_{k=1}^{K}$ are parameters and variables used in the Chinese restaurant franchise sampler for the HPYP (Teh et al. 2006), N is the number of times the context h occurs in the training data, h' is shorthand for removing one word from the context h, and $\delta(0) = 1, \delta(x) = 0 \, \forall \, x \neq 0$ is a standard indicator function. The correspondence of inference in the HPYLM to historical back-off schemes is established by considering a single sample approximation to this expectation (Teh 2006). The first term in the sum on the right hand side of this expression (sans expectation) is related to the count of the number of times w occurs after h in the training corpus. The second term corresponds to the "back-off" probability of w following a shorter-by-one-word context h'. This recursive form is similar to that of most back-off schemes.

3 DOUBLY HIERARCHICAL PITMAN-YOR PROCESS LANGUAGE MODEL

The DHPYLM consists of a collection of HPYLM's, one each for each domain, connected together through a "latent," shared HPYLM (see Fig. 1). The intuition behind this model architecture involves imagining a true, general generative process for text which is unobservable but affects the actual generation of all observed domain-specific corpora. Each estimated domain-specific model is then, as is typical in hiearchical Bayesian models, reflective of the general generative process, but sensitive to specific differences arising from each domain.

$$
\begin{aligned}
\mathcal{G}^{\mathcal{D}}_{[]} &\sim \mathrm{PY}(d^{\mathcal{D}}_0, \alpha^{\mathcal{D}}_0, \lambda^{\mathcal{D}}_0 \mathcal{U} + (1-\lambda^{\mathcal{D}}_0) G^{\mathcal{L}}_{[]}) \\
\mathcal{G}^{\mathcal{D}}_{[x_1]} &\sim \mathrm{PY}(d^{\mathcal{D}}_1, \alpha^{\mathcal{D}}_1, \lambda^{\mathcal{D}}_1 \mathcal{G}^{\mathcal{D}}_{[]} + (1-\lambda^{\mathcal{D}}_1) \mathcal{G}^{\mathcal{L}}_{[x_1]}) \\
&\vdots \\
\mathcal{G}^{\mathcal{D}}_{[x_j \cdots x_1]} &\sim \mathrm{PY}(d^{\mathcal{D}}_j, \alpha^{\mathcal{D}}_j, \lambda^{\mathcal{D}}_j \mathcal{G}^{\mathcal{D}}_{[x_{j-1} \cdots x_1]} + (1-\lambda^{\mathcal{D}}_j) \mathcal{G}^{\mathcal{L}}_{[x_j \cdots x_1]}) \\
w^{\mathcal{D}}_t | w^{\mathcal{D}}_{t-n+1} \cdots w^{\mathcal{D}}_{t-1} &\sim \mathcal{G}^{\mathcal{D}}_{[w^{\mathcal{D}}_{t-n+1} \cdots w^{\mathcal{D}}_{t-1}]} \qquad (2)
\end{aligned}
$$

The specific DHPYLM model structure we propose starts with a "latent" HPYLM

$$
\begin{aligned}
\mathcal{G}^{\mathcal{L}}_{[]} &\sim \mathrm{PY}(d^{\mathcal{L}}_0, \alpha^{\mathcal{L}}_0, \mathcal{U}) \\
\mathcal{G}^{\mathcal{L}}_{[x_1]} &\sim \mathrm{PY}(d^{\mathcal{L}}_1, \alpha^{\mathcal{L}}_1, \mathcal{G}^{\mathcal{L}}_{[]}) \\
&\vdots \\
\mathcal{G}^{\mathcal{L}}_{[x_j \cdots x_1]} &\sim \mathrm{PY}(d^{\mathcal{L}}_j, \alpha^{\mathcal{L}}_j, \mathcal{G}^{\mathcal{L}}_{[x_{j-1} \cdots x_1]}) \qquad (3)
\end{aligned}
$$

to which no observations are directly attributed. The formulae in (3) are exactly the same as those in (1) except that each variable has a superscript indicating its membership in the latent language model. In the graphical model shown in Figure 1 this latent language model runs down the left column.

Additionally the DHPYLM has a HPYLM for each domain $\mathcal{D}$ (Figure 1, graphical model, right column). The domain-specific HPYLM's in our model share the same HPYP suffix tree structure as the latent model, but they differ significantly in the way the base distribution for each PYP is specified.

The domain specific generative model is given in Eqn. 2. All but the last line in Eqn. 2 show a distribution over words following a particular context being "centered" on a mixture with two parts. This is key to how and why this model works for natural language model domain adaptation. The first member of each mixture is a distribution over words following a context that is one word shorter *from the same domain*; the second member is a distribution over words following the full context *from the latent language model* instead. This implies that the DHYPYLM can be understood as a model which "backs-off" by both dropping words from the context and by dropping domain specificity (retaining the full context). Further, as this mixture base distribution construction is used throughout all levels of the hierarchy, this "back-off" is recursive.

How to estimate such a model from data remains a question. Towards explaining this we first introduce a formalism called the graphical Pitman-Yor process. Models such as the DHPYLM, HPYLM, and others can be expressed as instances of graphical Pitman-Yor processes. Understanding estimation of graphical Pitman-Yor processes therefore will make clear how to estimate a DHPYLM from data.

4 GRAPHICAL PITMAN-YOR PROCESS

A graphical Pitman-Yor process (GPYP) is a directed acyclic graphical model, where each vertex $v \in V$ is labeled with a random distribution $\mathcal{G}_v$ and each has a PYP prior. An example GPYP is given in Fig. 1. Every edge $w \rightarrow v \in E$ in the GPYP has a non-negative weight $\lambda_{w \rightarrow v}$ and the weights are constrained such the sum of the weights on all of the incoming edges to a vertex equals one, i.e. $\sum_{w \in \mathrm{Pa}(v)} \lambda_{w \rightarrow v} = 1$ for all $v \in V$. Pa(v) is the set of parents of v in the DAG. The base distribution of each $\mathcal{G}_v$ is a mixture, with the edge weights being the mixing proportions and the $\mathcal{G}_w$'s on the parents $w \in \mathrm{Pa}(v)$ being the components. The generative model for such a GPYP is

$$
\mathcal{G}_v \sim \mathrm{PY}\left(d_v, \alpha_v, \sum_{w \in \mathrm{Pa}(v)} \lambda_{w \rightarrow v} \mathcal{G}_w\right) \quad \forall v \in V
$$

The parameters of a GPYP are $\Theta = \{d_v, \alpha_v, \lambda_{w \rightarrow v} : v \in V, w \in \mathrm{Pa}(v)\}$ each with its corresponding prior.

In most modeling situations we cannot directly observe the random distributions but observe draws from them instead. For instance, we may observe draws $\{x^n_v\}^{N_v}_{n=1} \sim \mathcal{F}(\phi^n_v)$ from a likelihood $\mathcal{F}$ whose parameter $\phi^n_v \sim \mathcal{G}_v$ is a draw from $\mathcal{G}_v$. In some modeling situations (like our language modeling application) the ϕ's are themselves directly observable.

As is usual in Bayesian modeling we are interested in generating posterior samples of the random distributions and GPYP parameters given observations. To do this we develop a representation of the GPYP in which the random distributions are integrated out. We call this representation for the GPYP the multi-floor Chinese restaurant franchise (MFCRF). It builds on the multi-floor Chinese restaurant process which we introduce next.

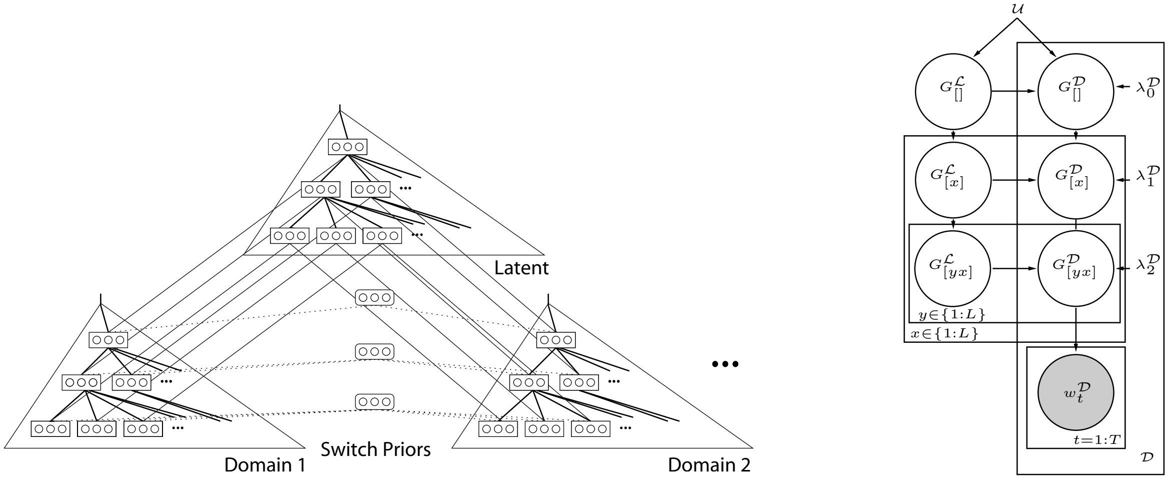


Figure 1: Left: a schematic of the DHPYLM. The triangles each surround a HPYLM model depicted in its Chinese restaurant franchise representation. Each PYP is itself depicted in its Chinese restaurant representation as a (rounded) rectangle with interior circles (tables). The seating arrangement of customers is not indicated. Solid edges going upwards from each PYP rectangle indicate which distribution(s) are members of the base distribution of that PYP. Thick solid lines indicate within-domain base distribution members whereas thin solid lines indicate out-of-domain base distribution members. Dotted lines indicate which switch variable prior is used for each of the PYP's. Right: a tri-gram DHPYLM graphical model. The $\mathcal{G}$'s are distributions over words following particular contexts (given in the subscript). There is one $\mathcal{G}$ in this graphical model for every rectangle in the schematic on the left. The context $\left[w^{\mathcal{D}}_{t-2}w^{\mathcal{D}}_{t-1}\right]$ of each observation w_t is not explicitly noted. Note that each such observation is attributed to the single corresponding $\mathcal{G}_{\left[w^{\mathcal{D}}_{t-2}w^{\mathcal{D}}_{t-1}\right]}$.

4.1 MULTI-FLOOR CHINESE RESTAURANT PROCESS

Consider a single random distribution $\mathcal{G}_v$ in the GPYP with a mixture base distribution $\sum_{w\in\mathrm{Pa}(v)}\lambda_{w\to v}\mathcal{G}_w$, and consider a sequence of i.i.d. draws $\phi^n_v \sim \mathcal{G}_v$ for $n = 1,\ldots,N_v$. The multi-floor Chinese restaurant process (MFCRP) is an extension of the Chinese restaurant process for normal PYPs which captures both the clustering structure of the draws as well as their association with components of the mixture base distribution.

The way that customers are assigned to tables in the MFCRP is the same as in the CRP. Each draw ϕ^n_v is identified with a customer entering the restaurant and sitting at a table z^n_v (denoting the cluster that ϕ^n_v belongs to). Let c^k_v be the number of customers sitting at the k^{th} table. If n customers have already seated themselves according to the MFCRP resulting in K_v tables being occupied then the probabilities of the next customer sitting at a currently occupied table or choosing a new table are given by

$$\begin{aligned} P(z^{n+1}_v = k|\{z^1_v\cdots z^n_v\}) &\propto c^k_v - d_v \\ P(z^{n+1}_v = K_v+1|\{z^1_v\cdots z^n_v\}) &\propto \alpha_v + d_vK_v \quad (4) \end{aligned}$$

If a new table is created (second line of Eqn. 4) then K_v is incremented. This means that customers entering the restaurant care about both how many other customers are sitting at a table and how many tables are in the restaurant.

As in the CRP, each table k in the MFCRP is given a label ψ^k_v which is an i.i.d. draw from the base distribution. Since this is a mixture we can achieve this by picking component w with probability $\lambda_{w\to v}$, and drawing from the chosen parent distribution $\mathcal{G}_w$. Let s^k_v be the chosen component. Each table is also labelled with this quantity. Metaphorically, this corresponds to table k being located on floor s^k_v of a multifloor restaurant.

4.2 MULTI-FLOOR CHINESE RESTAURANT FRANCHISE

Returning to the GPYP, we now consider marginalizing out all of the random distributions $\mathcal{G}_v$'s in the graph and replacing them with corresponding MFCRP representations. The MFCRP representation only requires being able to draw from the PYP base distribution. This means that we can directly use this representation for all $\mathcal{G}_v$'s that are leaf vertices in the graph because we can draw from the base distribution even if it is a mixture. Accordingly all of the table labels in the resulting MFCRP representations of leaf vertices arise from draws from their corresponding base distributions. The specific parent in the graph from which they were drawn is indicated by the corresponding floor indicator variables. This means that one can think of the table labels in a leaf node as being i.i.d. "observations" from the associated parents in the GPYP. With this insight it becomes apparent that we can repeat this procedure of switching from $\mathcal{G}_v$ to the MFCRP representation recursively up the graph because a table in any given restaurant must always correspond to a customer in one of its parent restaurants, the identity of which is determined by the state of the table-specific floor indicator variable.

The resulting representation, having replaced each $\mathcal{G}_v$ with a MFCRP, stipulates that customers in any given restaurant (indexed by vertex v) either must be associated with direct observations of draws from the underlying $\mathcal{G}_v$, or must have come from a table in a child restaurant. We call the resulting representation of the whole GPYP the multi-floor Chinese restaurant franchise (MFCRF). The MFCRF is a generalization of the Chinese restaurant franchise representation of the Pitman-Yor process (Teh 2006) to the situation here where each restaurant can have multiple parent restaurants. The distinctive characteristic of the MFCRF representation is that each table corresponds to a customer in one of a *set* of parent restaurants rather than a single parent restaurant and that each table maintains a label of the identity of the parent restaurant.

4.3 GPYP GIBBS SAMPLER

It is straightforward to derive a Gibbs sampler for the posterior of a GPYP in the MFCRF representation. Let X_v^n be the *set* of observations from all restaurants that can be traced to customer n in restaurant v. Let $\mathcal{F}(X_v^n|\psi)$ be the probability of observing X_v^n given parameter ψ. As before, let z_v^n indicate the table at which customer n sits. The update equations for the indicator variables associated with a single vertex are

$$\begin{aligned} &P(z_v^n = k|\{z_v^1 \cdots z_v^{N_v}\}\backslash z_v^n, X_v^n, \psi_v^k, \Theta) \\ &\quad \propto \quad \max((c_v^{k-} - d_v), 0)\mathcal{F}(X_v^n|\psi_v^k) \end{aligned}$$

$$\begin{aligned} &P(z_v^n = K_v^- + 1, s_v^{K_v^- +1} = w|\{z_v^1 \cdots z_v^{N_v}\}\backslash z_v^n, X_v^n, \Theta) \\ &\quad \propto \quad (\alpha_v + d_v K_v^-)\lambda_{w\to v} \int \mathcal{F}(X_v^n|\phi)\mathcal{G}_w(\phi)d\phi. \quad (5) \end{aligned}$$

Here the number of customers sitting at each table (c_v^{k-}) and the total number of occupied tables (K_v^-) are tallied with the current customer n "unseated". Note that the second equation above is a joint distribution over the parameter and floor indicator variables and includes the term $\lambda_{w\to v}$. The value of $\lambda_{w\to v}$ strongly influences the "floor" on which the table ends up. The floor variable is implicit in the first equation since s_v^k is fixed.

These update equations describe how to unseat and reseat customers in a single restaurant. As in the Chinese restaurant franchise sampler for the HDP (Teh et al. 2006) and as described in the preceeding section, the internal state of all of the restaurants must be consistent. For instance, if in sampling one of the restaurants in the GPYP a new table is created, then we know that its label had to have been a draw from one of its base distributions. This must be reflected in the MFCRF representation by recursively adding a customer (and table if necessary) to the corresponding parent restaurant. Conversely, if a table becomes empty its associated customer (and table if necessary) must be removed from the chosen parent restaurant. These updates propagate changes to the restaurant to the rest of the franchise. The complete MFCRF sampler then consists of visiting every restaurant in the GPYP and unseating and reseating every customer in all of the restaurants, maintaining consistency throughout the hierarchy by adding or removing customers from parent restaurants when tables become occupied or empty in child restaurants. The main difference between the MFCRF and the Chinese restaurant franchise samplers is that floor variables must be maintained in order to keep track of the parent restaurants from which each table came.

A complete posterior sampler for the GPYP requires sampling the parameters α_v's, d_v's, $\lambda_{w\to v}$'s, and the ψ_v^k's at the top of the GPYP as well. Next we describe how these variables are sampled in the specific case of the DHPYLM.

4.4 DHPYLM ESTIMATION

Note that the DHPYLM language model as previously described is a GPYP with a likelihood of the form $\mathcal{F}(x;\psi) = \delta(x-\psi)$ where x is a token (word instance) and ψ is a type (unique word identity). To be concrete we restrict ourselves to describing the specific model used in our experiments. This is a DHPYLM model with context length equal to two (corresponding to a trigram model) and with the $\Lambda_j^{\mathcal{D}}$'s shared across restaurants on the same level of the tree (see the dotted lines in the schematic on the left in Fig. 1). This clearly is not a restriction imposed by the underlying model; greater depths and different tying of the back-off mixtures are certainly possible. The prior we use is $\mathcal{S}_j = \mathrm{PYP}(d_j^{\mathcal{S}}, \alpha_j^{\mathcal{S}}, \mathcal{U}_2)$ where $\mathcal{U}_2$ is the uniform distribution over $\{0, 1\}$. By choosing this prior we are able to marginalize out the Λ's and utilize the general GPYP process estimation machinery to sample the switch variables as well. This is a difference worth highlighting between our approach to language model domain adaption and the prior art. We do not learn a single set of back-off parameters but instead marginalize them out through sampling. By placing a prior on the λ's, explicitly representing the "floor" in the MFCRP, and sampling floor indicator variables implicitly averages over various degrees of in- and out-of-domain back-off.

Tying the λ's together results in sharing of back-off behavior between different contexts (i.e. choosing whether to "back-off to an in-domain distribution over words given a context with one fewer tokens of history" or to "switch to the latent domain with the full context."). The characteristics of this sharing structure can easily vary from the independent style of Kneser and Steinbiss (1993) to the single back-off style of Bacchiani et al. (2006).

Metropolis updates were used to sample all of the α's and d's in the model. A Gamma$(1, 1)$ prior was placed on the α's. A uniform prior was placed on the d's.

5 RELATED WORK

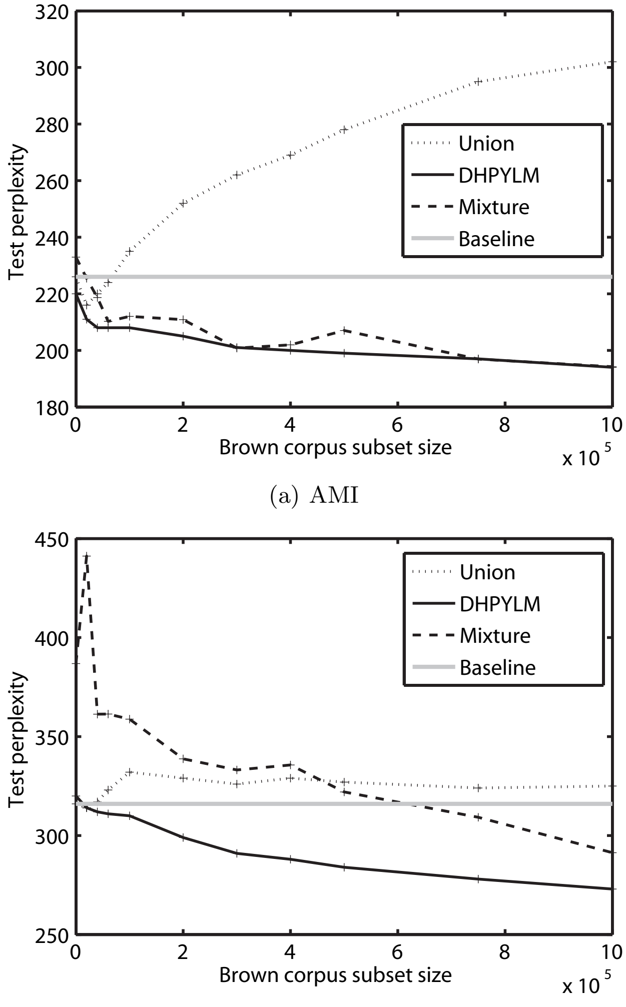
Many different domain adaptation approaches have been studied including mixtures of n-gram languge models in which the mixing weights themselves can be a function of preceding words (Kneser and Steinbiss 1993), various methods of estimating single n-gram models by combining count statistics from multiple corpora (Iyer et al. 1997), and using world wide web search engine query results to improve trigram models (Zhu and Rosenfeld 2001). Such count merging and model iterpolation approaches can be expressed in a unified mathematical framework and have been shown to be reasonably similar in terms of end-to-end performance in, among others, an automated speech recognition engine (Bacchiani et al. 2006).

For readers familiar with these types of language model domain adaptation approaches, ours may at first seem more similar to existing interpolation approaches than it in fact is. First, as ours is a Bayesian approach, we do not learn a single best set of parameters but instead show how to estimate a posterior over model parameters. Further, even if we were to restrict ourselves to using a point estimate of the posterior (for instance the mode), ours is a unified model in which all parameters, smoothing and domain adapting, are specified together and estimated at once. The prior art starts with a pre-trained set of smoothed language models and then tacks on additional domain adaptation smoothing parameters which are either "empirically determined" (Bacchiani et al. 2006) or learned through cross-validation (Kneser and Steinbiss 1993). Additionally, the non-domain-specific language model is latent in our model and learned from data. This is substantially different than all prior art and allows for straightforward generalization of the model to one that simultaneously adapts to multiple domains. Lastly, the actual predictive distribution that arises from our model utilizes back-off counts in a way that is different than all of these models.

It should be noted that substantially different approaches exist as well (Bellegarda 2004). For instance, marginally constrained exponential family language models (contrained to match observed low order n-gram statistics) can be used for adaptation by simultaneously maximizing the entropy of a constrained specific model while minimizing the divergence between it and another more general model of the same form (Della Pietra et al. 1992). In this work we restrict our comparisons, both theoretical and empirical, to the more closely related interpolation and count merging models because they have been shown to perform as well or better than these (Bellegarda 2004).

6 EXPERIMENTS

To evaluate the DHPYLM's language modeling and domain adaptation characteristics we performed experiments using two different "specific" in-domain corpora along with one "general" out-of-domain corpus. The general design of our experiments is intentionally antagonistic to our own performance for purposes of clearer comparison with prior art. Most importantly we do not sequentially incorporate the test corpus data into the model although in principle it is possible to do this due to the Bayesian nature of our model. This discards one of the greatest advantages of a Bayesian approach. For the general corpus we used the Brown corpus (Kucera and Francis 1967) (ignoring part of speech tags). The Brown corpus consists of slightly more than one million words of written English (49,743 unique words) collected from varied sources (news, literature, science, etc.) in the early 1960's. We used both a subset of the set of state of the union addresses (SOU) (SOUCorpus) and a subset of the AMI meeting corpus (Carletta 2007) as in-domain corpora in different experiments. The SOU corpus consists of a collection of "State of the Union" addresses (370,828 words, 12,914 unique words, 10,808 of which were the same as words in the Brown corpus) given by the President to the Congress of the United States. The first such address in our corpus was given by Harry Truman in 1945 while the last was given by George W. Bush in 2006. Throughout all of our experiments we withheld the speeches given by Lyndon Johnson (1963-1969, 37102 words) from the SOU corpus for use as test data. The SOU training corpus subsets used in the experiments primarily included speeches made before 1963 but also included some speeches made after 1969. The AMI corpus we used consisted of 801,710 words of tran-



(a) AMI

(b) SOU

Figure 2: Relative performance of various domain adaptation approaches for two different corpora. Both graphs show test perplexity as a function of Brown training corpus size. Lower test perplexity is better.

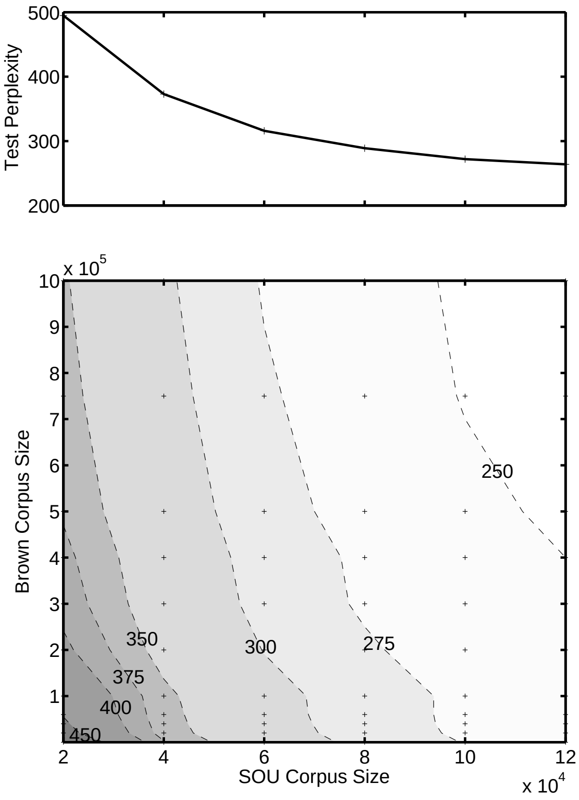


Figure 3: Illustration of the relative rates of test perplexity improvement that can be obtained by using more out-of-domain data versus using more in-domain data. Top: Baseline test perplexity of Lyndon Johnson's state of the union addresses as a function of training corpus size. Bottom: test perplexity for the DHPYLM. Lower test perplexity is better. The entire lower graph is below the baseline which means that in our model adding out of domain data always helps.

scribed meeting conversation (10,890 unique words, 8,057 of which were the same as words in the Brown corpus). The AMI test data used in our experiments consisted of the last 20,000 words of this corpus.

Figure 2 shows our DHPYLM domain adaptation approach in comparison to three others. The first, "baseline," is the performance given by training a hierarchical Pitman-Yor process language model (HPYLM) on the specific domain training data alone. The second, "union," is the performance of an HPYLM trained on the union of the specific and general training data corpora. The third, "mixture," is the MAP approach of Bacchiani et al. (2006) where two smoothed n-gram models (interpolated Kneser-Ney smoothing) are trained, one each for the specific and general domain corpora and then a smoothing parameter is learned by cross-validation. Both the AMI results shown in Fig. 2(a) and the SOU results shown in Fig. 2(b) suggest that our DHPYLM approach to domain adaptation works as well or better than these alternatives.

The second set of experiments starts to address the most pertinent domain adaptation question directly: How does adding more general domain data compare to acquiring more specific domain training data? In Figure 3 we explore how much the test perplexity of an in-domain test dataset is improved both by adding more out-of-domain data versus adding more in-domain data. In a real application where such a domain adaptation approach were being considered there would likely be a computational cost associated with adding more data and another (likely more expensive) resource cost to acquire more in-domain training data. While the costs specific to each application domain are different, Fig. 3 suggests that both adding more in-domain data and adding more out-of-domain data monotonically improves test perplexity. At the top of Fig. 3 we plot the "baseline" test perplexity for a HPYLM model trained on SOU corpus data alone (this is the same baseline as was established in Fig. 2). Tests were run for each combination of Brown and SOU training corpus sizes shown by the small crosses and absolute test perplexity improvement was interpolated between these points to produce isosurfaces of test perplexity improvement. An example reading from this figure indicates that with a SOU training corpus of 20,000 words, adding one million words of Brown data will reduce the SOU test corpus perplexity from near 500 to somewhat below 360. Equivalent test corpus perplexity could be had using a SOU-only HPYLM model by more than doubling the amount of SOU domain specific training words. In application domains where adding more out-of-domain data is significantly cheaper than acquiring more in-domain training data this could result in substantial savings.

7 SUMMARY

In this paper we have introduced a new approach to statistical language model domain adaptation. This approach achieves encouraging domain adaptation results; results that suggest a more thorough and data intensive comparison of it to other existing domain adaptation approaches. Additionally, we defined a graphical Pitman-Yor process, a generalization of the hierarchical Dirichlet process, and outlined a so-called multi-floor Chinese restaurant representation for sampling from such a process. Graphical Pitman-Yor processes form a general framework within which to explore a large variety of language models while retaining the same inference engine. We intend to undertake a more detailed and thorough theoretical treatment of graphical Pitman-Yor processes.

Lastly, there are a number of generalizations of this model which we intend to develop and demonstrate. First, requiring no modification to the model, but potentially further improving test performance, we will experiment with mutiple domain adaptation by adding more than two corpora into the DHPYLM. Secondly the DHPYLM can integrated into topic models such that the bag-of-words assumption can be avoided.

References

M. Bacchiani, M. Riley, B. Roark, and R. Sproat. MAP adaptation of stochastic grammars. *Computer Speech and Language*, 20:41–68, 2006.

J. R. Bellegarda. Statistical language model adaptation: review and perspectives. *Speech Communication*, 42:93–108, 2004.

J. Carletta. Unleashing the killer corpus: experiences in creating the multi-everything AMI meeting corpus. *Language Resources and Evaluation Journal*, 41:181–190, 2007.

S. F. Chen and J. T. Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Dept. of Comp. Sci., Harvard, 1998.

S. Della Pietra, V. Della Pietra, R. Mercer, and S. Roukos. Adaptive language model estimation using minimum discriminatioon estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 633–636, 1992.

S. Goldwater, T. L. Griffiths, and M. Johnson. Interpolating between types and tokens by estimating power law generators. In *Advances in Neural Information Processing Systems 19*, pages 459–466. MIT Press, 2007.

R. Iyer, M. Ostendorf, and H. Gish. Using out-of-domain data to improve in-domain language models. *IEEE Signal processing letters*, 4:221–223, 1997.

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, volume 1, pages 181–184, 1995.

R. Kneser and V. Steinbiss. On the dynamic adaptation of stochastic language models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 586–589, 1993.

H. Kucera and W. N. Francis. *Computational analysis of present-day American English*. Brown University Press, Providence, RI, 1967.

R. Rosenfeld. Two decades of statistical language modeling: where do we go from here? In *Proceedings of the IEEE*, volume 88, pages 1270–1278, 2000.

SOUCorpus. `http://www.c-span.org/executive/stateoftheunion.asp`.

Y. W. Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proc. of the Association for Computational Linguistics*, pages 985–992, 2006.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

X. Zhu and R. Rosenfeld. Improving trigram language modeling with the world wide web. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 533–536, 2001.