Improvements to the Bayesian Topic N-gram Models

Hiroshi Noji†‡
noji@nii.ac.jp

Daichi Mochihashi†∗
daichi@ism.ac.jp

Yusuke Miyao†‡
yusuke@nii.ac.jp

†Graduate University for Advanced Studies
‡National Institute of Informatics, Tokyo, Japan
∗The Institute of Statistical Mathematics, Tokyo, Japan

Abstract

One of the language phenomena that n-gram language model fails to capture is the topic information of a given situation. We advance the previous study of the Bayesian topic language model by Wallach (2006) in two directions: one, investigating new priors to alleviate the sparseness problem caused by dividing all n-grams into exclusive topics, and two, developing a novel Gibbs sampler that enables moving multiple n-grams across different documents to another topic. Our blocked sampler can efficiently search for higher probability space even with higher order n-grams. In terms of modeling assumption, we found it is effective to assign a topic to only some parts of a document.

1 Introduction

N-gram language model is still ubiquitous in NLP, but due to its simplicity it fails to capture some important aspects of language, such as difference of word usage in different situations, sentence level syntactic correctness, and so on. Toward language model that can consider such a more global context, many extensions have been proposed from lexical pattern adaptation, e.g., adding cache (Jelinek et al., 1991) or topic information (Gildea and Hofmann, 1999; Wallach, 2006), to grammaticality aware models (Pauls and Klein, 2012).

Topic language models are important for use in e.g., *unsupervised language model adaptation*: we want a language model that can adapt to the domain or topic of the current situation (e.g., a document in SMT or a conversation in ASR) automatically and select the appropriate words using both topic and syntactic context. Wallach (2006) is one such model, which generate each word based on local context and global topic information to capture the difference of lexical usage among different topics.

However, Wallach's experiments were limited to bigrams, a toy setting for language models, and experiments with higher-order n-grams have not yet been sufficiently studied, which we investigate in this paper. In particular, we point out the two fundamental problems caused when extending Wallach's model to a higher-order: *sparseness* caused by dividing all n-grams into exclusive topics, and *local minima* caused by the deep hierarchy of the model. On resolving these problems, we make several contributions to both computational linguistics and machine learning.

To address the first problem, we investigate incorporating a global language model for ease of sparseness, along with some priors on a suffix tree to capture the difference of *topicality* for each context, which include an *unsupervised* extension of the doubly hierarchical Pitman-Yor language model (Wood and Teh, 2009), a Bayesian generative model for *supervised* language model adaptation. For the second inference problem, we develop a novel blocked Gibbs sampler. When the number of topics is K and vocabulary size is V, n-gram topic model has $O(KV^n)$ parameters, which grow exponentially to n, making the local minima problem even more severe. Our sampler resolves this problem by moving many customers in the hierarchical Chinese restaurant process at a time.

We evaluate various models by incremental calculation of test document perplexity on 3 types of corpora having different size and diversity. By combining the proposed prior and the sampling method, our Bayesian model achieve much higher accuracies than the naive extension of Wallach (2006) and shows results competitive with the unigram rescaling (Gildea and Hofmann, 1999), which require

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1180–1190,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

huge computational cost at prediction, with much faster prediction time.

2 Basic Models

All models presented in this paper are based on the Bayesian n-gram language model, the hierarchical Pitman-Yor process language model (HPYLM). In the following, we first introduce the HPYLM, and then discuss the topic model extension of Wallach (2006) with HPYLM.

2.1 HPYLM

Let us first define some notations. W is a vocabulary set, $V = |W|$ is the size of that set, and $u, v, w \in W$ represent the word type.

The HPYLM is a Bayesian treatment of the n-gram language model. The generative story starts with the unigram word distribution G_ϕ, which is a V-dimensional multinomial where $G_\phi(w)$ represents the probability of word w. The model first generates this distribution from the PYP as $G_\phi \sim \mathrm{PYP}(a, b, G_0)$, where G_0 is a V-dimensional uniform distribution ($G_0(u) = \frac{1}{V}; \forall u \in W$) and acts as a prior for G_ϕ and a, b are hyperparameters called discount and concentration, respectively. It then generates all bigram distributions $\{G_u\}_{u \in W}$ as $G_u \sim \mathrm{PYP}(a, b, G_\phi)$. Given this distributions, it successively generates 3-gram distributions $G_{uv} \sim \mathrm{PYP}(a, b, G_u)$ for all $(u, v) \in W^2$ pairs, which encode a natural assumption that contexts having common suffix have similar word distributions. For example, two contexts “he is” and “she is”, which share the suffix “is”, are generated from the same (bigram) distribution G_{is}, so they would have similar word distributions. This process continues until the context length reaches $n - 1$ where n is a prespecified n-gram order (if $n = 3$, the above example is a complete process). We often generalize this process using two contexts h and h' as

$$G_h \sim \mathrm{PYP}(a, b, G_{h'}), \tag{1}$$

where $h = ah'$, in which a is a leftmost word of h.

We are interested in the posterior word distribution following a context h. Our training corpus $\mathbf{w}$ is a collection of n-grams, from which we can calculate the posterior $p(w|h, \mathbf{w})$, which is often explained with the Chinese restaurant process (CRP):

$$p(w|h, \mathbf{w}) = \frac{c_{hw} - at_{hw}}{c_{h\cdot} + b} + \frac{at_{h\cdot} + b}{c_{h\cdot} + b} p(w|h', \mathbf{w}), \tag{2}$$

where c_{hw} is an observed count of n-gram hw called customers, while t_{hw} is a hidden variable called tables. $c_{h\cdot}$ and $t_{h\cdot}$ represents marginal counts: $c_{h\cdot} = \sum_w c_{hw}$ and $t_{h\cdot} = \sum_w t_{hw}$. This form is very similar to the well-known Kneser-Ney smoothing, and actually the Kneser-Ney can be understood as a heuristic approximation of the HPYLM. This characteristic enables us to build the state-of-the-art language model into a more complex generative model.

2.2 Wallach (2006) with HPYLM

Wallach (2006) is a generative model for a document collection that combines the topic model with a Bayesian n-gram language model. The latent Dirichlet allocation (LDA) (Blei et al., 2003) is the most basic topic model, which generates each word in a document based on a unigram word distribution defined by a topic allocated to that word. The bigram topic model of Wallach (2006) simply replaces this unigram word distribution (a multinomial) for each topic with a bigram word distribution [1]. In other words, ordinary LDA generates word conditioning *only* on the latent topic, whereas the bigram topic model generates conditioning on *both* the latent topic and the previous word, as in the bigram language model. Extending this model with a higher order n-gram is trivial; all we have to do is to replace the bigram language model for each topic with an n-gram language model.

The formal description of the generative story of this n-gram topic model is as follows. First, for each topic $k \in 1, \cdots, K$, where K is the number of topics, the model generates an n-gram language model G_h^k.[2] These n-gram models are generated by the PYP, so $G_h^k \sim \mathrm{PYP}(a, b, G_{h'}^k)$ holds. The model then generate a document collection. For each document $j \in 1, \cdots, D$, it generates a K-

[1] This is the model called *prior 2* in Wallach (2006); it consistently outperformed the other prior. Wallach used the Dirichlet language model as each topic, but we only explore the model with HPYLM because its superiority to the Dirichlet language model has been well studied (Teh, 2006b).

[2] We sometimes denote G_h^k to represent a language model of topic k, not a specific multinomial for some context h, depending on the context.

dimensional topic distribution θ_j by a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha})$ where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_K)$ is a prior. Finally, for each word position $i \in 1, \cdots, N_j$ where N_j is the number of words in document j, i-th word's topic assignment z_{ji} is chosen according to θ_j, then a word type w_{ji} is generated from $G^{z_{ji}}_{h_{ji}}$ where h_{ji} is the last $n-1$ words preceding w_{ji}. We can summarize this process as follows:

```
1. Generate topics:
   For each $h \in \phi, \{W\}, \cdots, \{W\}^{n-1}$:
     For each $k \in 1, \cdots, K$:
       $G^k_h \sim \mathrm{PYP}(a, b, G^k_{h'})$
2. Generate corpora:
   For each document $j \in 1, \cdots D$:
     $\theta_j \sim \mathrm{Dir}(\boldsymbol{\alpha})$
     For each word position $i \in 1, \cdots, N_j$:
       $z_{ji} \sim \theta_j$
       $w_{ji} \sim G^{z_{ji}}_{h_{ji}}$
```

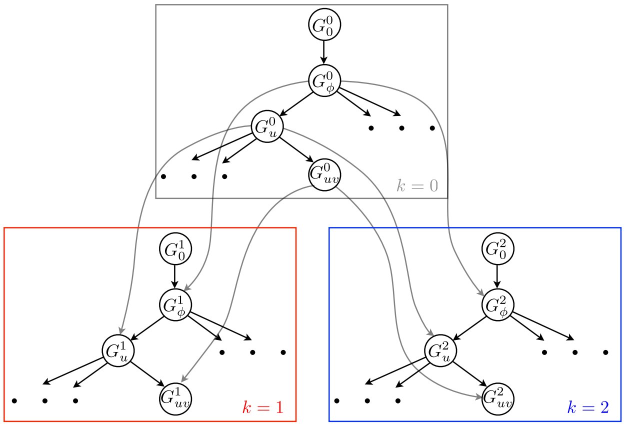


Figure 1: Variable dependencies of the HIERARCHICAL model. $\{u, v\}$ are word types, k is a topic and each G^k_h is a multinomial word distribution. For example, G^2_{uv} represents a word distribution following the context uv in topic 2.

3 Extended Models

One serious drawback of the n-gram topic model presented in the previous section is *sparseness*. At inference, as in LDA, we assign each n-gram a topic, resulting in an exclusive clustering of n-grams in the corpora. Roughly speaking, when the number of topics is K and the number of all n-grams in the training corpus is N, a language model of topic k, G^k_h is learned using only about $O(N/K)$ instances of the n-grams assigned the topic k, making each G^k_h much sparser and unreliable distribution.

One way to alleviate this problem is to place another n-gram model, say G^0_h, which is shared with all topic-specific n-gram models $\{G^k_h\}^K_{k=1}$. However, what is the best way to use this special distribution? We explore two different approaches to incorporate this distribution in the model presented in the previous section. In one model, the HIERARCHICAL model, G^0_h is used as a prior for all other n-gram models, where G^0_h exploits global statistics across all topics $\{G^k_h\}$. In the other model, the SWITCHING model, no statistics are shared across G^0_h and $\{G^k_h\}$, but some words are directly generated from G^0_h regardless of the topic distribution.

3.1 HIERARCHICAL Model

Informally, what we want to do is to establish hierarchies among the global G^0_h and other topics $\{G^k_h\}$. In Bayesian formalism, we can explain this using an abstract distribution $\mathcal{F}$ as $G^k_h \sim \mathcal{F}(G^0_h)$. The problem here is making the appropriate choice for the distribution $\mathcal{F}$. Each topic word distribution already has hierarchies among $n-1$-gram and n-gram contexts as $G^k_h \sim \mathrm{PYP}(a, b, G^k_{h'})$. A natural solution to this problem is the doubly hierarchical Pitman-Yor process (DHPYP) proposed in Wood and Teh (2009). Using this distribution, the new generative process of G^k_h is

$$G^k_h \sim \mathrm{PYP}(a, b, \lambda G^k_{h'} + (1-\lambda) G^0_h), \qquad (3)$$

where λ is a new hyperparameter that determines mixture weight. The dependencies among G^0_h and $\{G^k_h\}$ are shown in Figure 1. Note that the generative process of G^0_h is the same as the HPYLM (1).

Let us clarify the DHPYP usage differences between our model and the previous work of Wood and Teh (2009). A key difference is the problem setting: Wood and Teh (2009) is aimed at the *supervised* adaptation of a language model for a specific domain, whereas our goal is *unsupervised* adaptation. In Wood and Teh (2009), each G^k_h for $k \in 1, 2, \cdots$ corresponds to a language model of a specific domain and the training corpus for each k is pre-specified and fixed. For ease of data sparseness of domain-specific corpora, latent model G^0_h exploits shared statistics among G^k_h for $k = 1, 2, \cdots$. In contrast, with our model, each G^k_h is a topic, so it must perform the clustering of n-grams in addition to exploiting the latent G_h^0. This makes inference harder and requires more careful design of λ.

Modeling of λ We can better understand the role of λ in (3) by considering the posterior predictive form corresponds to (2), which is written as

$$p(w|h,k,\mathbf{w}) = \frac{c_{hw}^k - at_{hw}^k}{c_{h\cdot}^k + b} + \frac{at_{h\cdot}^k + b}{c_{h\cdot}^k + b} q(w|h,k,\mathbf{w}), \tag{4}$$

$$q(w|h,k,\mathbf{w}) = \lambda p(w|h',k,\mathbf{w}) + (1-\lambda) p(w|h,0,\mathbf{w}),$$

where c, t with superscript k corresponds to the count existing in topic k. This shows us that λ determines the back-off behavior: which probability we should take into account: the shorter context of the same topic $G_{h'}^k$ or the full context of the global model G_h^0. Wood and Teh (2009) shares this variable across all contexts of the same length, for each k, but this assumption may not be the best. For example, after the context "in order", we can predict the word "to" or "that", and this tendency is unaffected by the topic. We call this property of context the *topicality* and say that "in order" has *weak topicality*. Therefore, we place λ as a distinct value for each context h, which we share across all topics. We designate this λ determined by h λ_h in the following. Moreover, similar contexts may have similar values of λ_h. For example, the two contexts "of the" and "in the", which share the suffix "the", both have a *strong topicality*[3]. We encode this assumption by placing hierarchical Beta distributions on the suffix tree across all topics:

$$\lambda_h \sim \mathrm{Beta}(\gamma\lambda_{h'}, \gamma(1-\lambda_{h'})) = \mathrm{DP}(\gamma, \lambda_{h'}), \tag{5}$$

where DP is the hierarchical Dirichlet process (Teh et al., 2006), which has only two atoms in {0,1} and γ is a concentration parameter. As in HPYLM, we place a uniform prior $\lambda_0 = 1/2$ on the base distribution of the top node ($\lambda_\phi \sim \mathrm{DP}(\gamma, \lambda_0)$).

Having generated the topic component of the model, the corpus generating process is the same as the previous model because we only change the generating process of G_h^k for $k = 1, \cdots, K$.

3.2 SWITCHING Model

Our second extension also exploits the global G_h^0, albeit differently than the HIERARCHICAL model. In this model, the relationship of G_h^0 to the other $\{G_h^k\}$ is *flat*, not *hierarchical*: G_h^0 is a special topic that can generate a word. The model first generates each language model of $k = 0, 1, 2, \cdots, K$ independently as $G_h^k \sim \mathrm{PYP}(a, b, G_{h'}^k)$. When generating a word, it first determines whether to use global model G_h^0 or topic model $\{G_h^k\}_{k=1}^K$. Here, we use the λ_h introduced above in a similar way: the probability of selecting $k = 0$ for the next word is determined by the previous context. This assumption seems natural; we expect the G_h^0 to mainly generate common n-grams, and the topicality of each context determines how common that n-gram might be. The complete generative process of this model is written as follows:

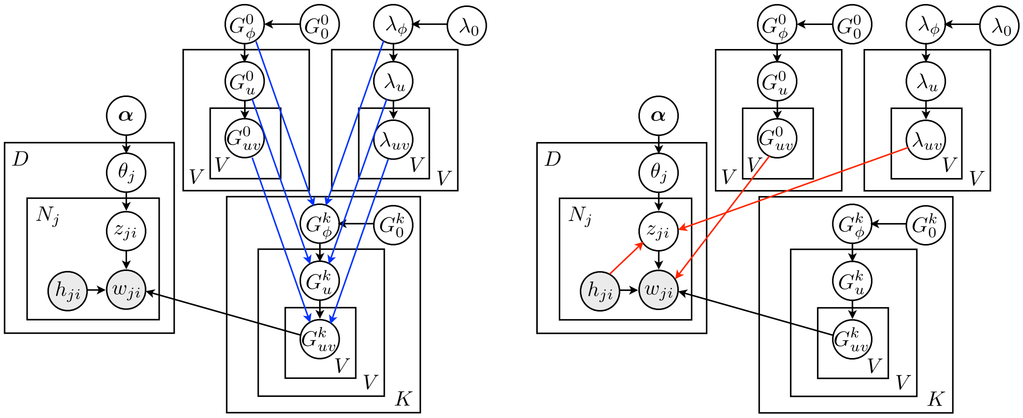
```
1. Generate topics:
   For each $h \in \phi, \{V\}, \cdots, \{V\}^{n-1}$:
     $\lambda_h \sim \mathrm{DP}(\gamma, \lambda'_h)$
     For each $k \in 0, \cdots, K$:
       $G_h^k \sim \mathrm{PYP}(a, b, G_{h'}^k)$
2. Generate corpora:
   For each document $j \in 1, \cdots D$:
     $\theta_j \sim \mathrm{Dir}(\boldsymbol{\alpha})$
     For each word position $i \in 1, \cdots, N_j$:
       $l_{ji} \sim \mathrm{Bern}(\lambda_{h_{ji}})$
       If $l_{ji} = 0$: $z_{ji} = 0$
       If $l_{ji} = 1$: $z_{ji} \sim \theta_j$
       $w_{ji} \sim G_{h_{ji}}^{z_{ji}}$
```

The difference between the two models is their usage of the global model G_h^0. For a better understanding of this, we provide a comparison of their graphical models in Figure 2.

4 Inference

For posterior inference, we use the collapsed Gibbs sampler. In our models, all the latent variables are $\{G_h^k, \lambda_h, \theta_j, \mathbf{z}, \Theta\}$, where $\mathbf{z}$ is the set of topic assignments and $\Theta = \{a, b, \gamma, \boldsymbol{\alpha}\}$ are hyperparameters, which are treated later. We collapse all multinomials in the model, i.e., $\{G_h^k, \lambda_h, \theta_j\}$, in which G_h^k and λ_h are replaced with the Chinese restaurant process of PYP and DP respectively. Given the training corpus $\mathbf{w}$, the target posterior distribution is $p(\mathbf{z}, \mathbf{S}|\mathbf{w}, \Theta)$, where $\mathbf{S}$ is the set of seating arrangements of all restaurants. To distinguish the two types of restaurant, in the following, we refer the *restaurant* to indi-

[3] These words can be used very differently depending on the context. For example, in a teen story, "in the room" or "in the school" seems more dominant than "in the corpora" or "in the topic", which is likely to appear in this paper.



(a) HIERARCHICAL (b) SWITCHING

Figure 2: Graphical model representations of our two models in the case of a 3-gram model. Edges that only exist in one model are colored.

cate the collapsed state of G^k_h (PYP), while we refer the *restaurant of* λ_h to indicates the collapsed state of λ_h (DP). We present two different types of sampler: a *token-based sampler* and a *table-based sampler*. For both samplers, we first explain in the case of our basic model (Section 2.2), and later discuss some notes on our extended models.

4.1 Token-based Sampler

The token-based sampler is almost identical to the collapsed sampler of the LDA (Griffiths and Steyvers, 2004). At each iteration, we consider the following conditional distribution of z_{ji} given all other topic assignments $\mathbf{z}^{-ji}$ and $\mathbf{S}^{-ji}$, which is the set of seating arrangements with a customer corresponds to w_{ji} removed, as

$$p(z_{ji}|\mathbf{z}^{-ji},\mathbf{S}^{-ji}) \propto p(z_{ji}|\mathbf{z}^{-ji})p(w_{ji}|z_{ji},h_{ji},\mathbf{S}^{-ji}), \tag{6}$$

where $p(w_{ji}|z_{ji},h_{ji},\mathbf{S}^{-ji}) =$

$$\frac{c^k_{hw} - at^k_{hw}}{c^k_{h\cdot}+b} + \frac{at^k_{h\cdot}+b}{c^k_{h\cdot}+b}p(w_{ji}|z_{ji},h_{ji},\mathbf{S}^{-ji}) \tag{7}$$

is a predictive word probability under the topic z_{ji}, and

$$p(z_{ji}|\mathbf{z}^{-ji}) = \frac{n^{-ji}_{jk}+\alpha_k}{N_j - 1 + \sum_{k'}\alpha_{k'}}, \tag{8}$$

where n^{-ji}_{jk} is the number of words that is assigned topic k in document j excluding w_{ji}, which is the same as the LDA. Given the sampled topic z_{ji}, we update the language model of topic z_{ji}, by adding customer w_{ji} to the restaurant specified by z_{ji} and context h_{ji}. See Teh (2006a) for details of these customer operations.

HIERARCHICAL Adding customer operation is slightly changed: When a new table is added to a restaurant, we must track the label $l \in \{0,1\}$ indicating the parent restaurant of that table, and add the customer corresponding to l to the restaurant of λ_h. See Wood and Teh (2009) for details of this operation.

SWITCHING We replace $p(z_{ji}|\mathbf{z}^{-ji})$ with

$$p(z_{ji}|\mathbf{z}^{-ji}) = \begin{cases} p(l_{ji}=0|h_{ji}) & (z_{ji}=0) \\ p(l_{ji}=1|h_{ji})\cdot\frac{n^{-ji}_{jk}+\alpha_k}{\sum_{k\neq 0} n^{-ji}_{jk}+\sum_{k'}\alpha_{k'}} & (z_{ji}\neq 0), \end{cases} \tag{9}$$

where $p(l_{ji}|h_{ji})$ is a predictive of l_{ji} given by the CRP of $\lambda_{h_{ji}}$. We need not assign a label to a new table, but rather we always add a customer to the restaurant of λ_h according to whether the sampled topic is 0 or not.

4.2 Table-based Sampler

One problem with the token-based sampler is that the seating arrangement of the internal restaurant would never be changed unless a new table is created (or an old table is removed) in its child restaurant. This probability is very low, particularly in the restaurants of shallow depth (e.g., unigram or

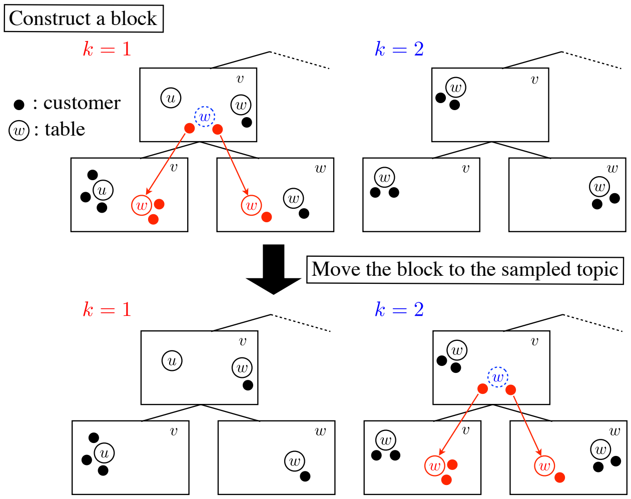


Figure 3: Transition of the state of restaurants in the table-based sampler when the number of topics is 2. $\{u, v, w\}$ are word types. Each box represents a restaurant where the type in the upper-right corner indicates the context. In this case, we can change the topic of the three 3-grams (vvw, vvw, uvw) in some documents from 1 to 2 at the same time.

Algorithm 1 Table-based sampler

```
for all table in all restaurants do
    Remove a customer from the parent restaurant.
    Construct a block of seating arrangement $S$ by descending the tree recursively.
    Sample topic assignment $z_S \sim p(z_S|S, \mathbf{S}^{-S}, \mathbf{z}^{-S})$.
    Move $S$ to sampled topic, and add a customer to the parent restaurant of the first selected table.
end for
```

bigram restaurants) because these restaurants have a larger number of customers and tables than those of deep depth, leading to get stack in undesirable local minima. For example, imagine a table in the restaurant of context "hidden" (depth is 2) and some topic, served "unit". This table is connected to tables in its child restaurants corresponding to some 3-grams (e.g., "of hidden unit" or "train hidden unit"), whereas similar n-grams, such as those of "of hidden units" or "train hidden units" might be gathered in another topic, but collecting these n-grams into the same topic might be difficult under the token-based sampler. The *table-based sampler* moves those different n-grams having common suffixes jointly into another topic.

Figure 3 shows a transition of state by the table-based sampler and Algorithm 4.2 depicts a high-level description of one iteration. First, we select a table in a restaurant, which is shown with a dotted line in the figure. Next, we descend the tree to collect the tables connected to the selected table, which are pointed by arrows. Because this connection cannot be preserved in common data structures for a restaurant described in Teh (2006a) or Blunsom et al. (2009), we select the child tables randomly. This is correct because customers in CRP are exchangeable, so we can restore the parent-child relations arbitrarily. We continue this process recursively until reaching the leaf nodes, obtaining a block of seating arrangement S. After calculating the conditional distribution, we sample new topic assignment for this block. Finally, we move this block to the sampled topic, which potentially changes the topic of many words across different documents, which are connected to customers in a block at leaf nodes (this connection is also arbitrary).

Conditional distribution Let $\mathbf{z}_S$ be the block of topic assignments connected to S and z_S be a variable indicating the topic assignment. Thanks to the exchangeability of all customers and tables in one restaurant (Teh, 2006a), we can imagine that customers and tables in S have been added to the restaurants last. We are interested in the following conditional distribution: (conditioning Θ is omitted)

$$p(z_S = k'|S, \mathbf{S}^{-S}, \mathbf{z}^{-S}) \propto p(S|\mathbf{S}^{-S}, k')p(z_S = k'|\mathbf{z}^{-S}),$$

where $p(S|\mathbf{S}^{-S}, k')$ is a product of customers' actions moving to another topic, which can be decomposed as:

$$p(S|\mathbf{S}^{-S}, k') = p(w|k', h) \prod_{s \in S} p(s|k') \tag{10}$$

$$p(s|k') = \frac{\prod_{i=0}^{t_s-1}(b+a(t_{h_s w}^{k'(-s)}+i)) \prod_{j=1}^{c_{si}}(j-a)}{(b+c_{h_s w\cdot}^{k'(-s)})^{\overline{c_{s\cdot}}}} \tag{11}$$

$$\propto \frac{\prod_{i=0}^{t_s-1}(b+a(t_{h_s w}^{k'(-s)}+i))}{(b+c_{h_s w\cdot}^{k'(-s)})^{\overline{c_{s\cdot}}}}. \tag{12}$$

Let us define some notations used above. Each $s \in S$ is a part of seating arrangements in a restaurant, there being t_s tables, i-th of which with c_{si} customers, with h_s as the corresponding context. A restaurant of context h and topic k has t_{hw}^k tables served dish w, i-th of which with c_{hwi}^k customers. Superscripts $-s$ indicate excluding the contribution of customers in s, and $x^{\overline{n}} = x(x+1)\cdots(x+n-1)$ is the ascending factorial. In (10) $p(w|k', h)$ is the parent distribution of the first selected table, and the other $p(s|k')$ is the seating arrangement of customers. The likelihood for changing topic assignments across documents must also be considered, which is $p(z_S = k'|\mathbf{z}^{-S})$ and decomposed as:

$$p(z_S = k'|\mathbf{z}^{-S}) = \prod_j \frac{(n_{jk'}^{-S}+\alpha_{k'})^{\overline{n_j(S)}}}{(N_j^{-S}+\sum_k \alpha_k)^{\overline{n_j(S)}}}, \qquad (13)$$

where $n_j(S)$ is the number of word tokens connected with S in document j.

HIERARCHICAL We skip tables on restaurants of $k = 0$, because these tables are all from other topics and we cannot construct a block. The effects of λ can be ignored because these are shared by all topics.

SWITCHING In the SWITCHING, $p(z_S = k'|\mathbf{z}^{-S})$ cannot be calculated in a closed form because $p(l_{ji}|h_{ji})$ in (9) would be changed dynamically when adding customers. This problem is the same one addressed by Blunsom and Cohn (2011), and we follow the same approximation in which, when we calculate the probability, we fractionally add tables and customers recursively.

4.3 Inference of Hyperparameters

We also place a prior on each hyperparameter and sample value from the posterior distribution for every iteration. As in Teh (2006a), we set different values of a and b for each depth of PYP, but share across all topics and sample values with an auxiliary variable method. We also set different value of γ for each depth, on which we place $\text{Gamma}(1, 1)$. We make the topic prior $\boldsymbol{\alpha}$ asymmetric: $\boldsymbol{\alpha} = \beta\boldsymbol{\alpha}_0; \beta \sim \text{Gamma}(1, 1), \boldsymbol{\alpha}_0 \sim \text{Dir}(\mathbf{1})$.

5 Related Work

HMM-LDA (Griffiths et al., 2005) is a composite model of HMM and LDA that assumes the words in a document are generated by HMM, where only one state has a document-specific topic distribution. Our SWITCHING model can be understood as a lexical extension of HMM-LDA. It models the topicality by context-specific binary random variables, not by hidden states. Other n-gram topic models have focused mainly on information retrieval. Wang et al. (2007) is a topic model on automatically segmented chunks. Lindsey et al. (2012) extended this model with the hierarchical Pitman-Yor prior. They also used switching variables, but for a different purpose: to determine the segmenting points. They treat these variables completely independently, while our model employs a hierarchical prior to share statistical strength among similar contexts.

	min.		training set		test set	
Corpus	appear	# types	# docs	# tokens	# docs	# tokens
Brown	4	19,759	470	1,157,225	30	70,795
NIPS	4	22,705	1500	5,088,786	50	167,730
BNC	10	33,071	6,162	12,783,130	100	202,994

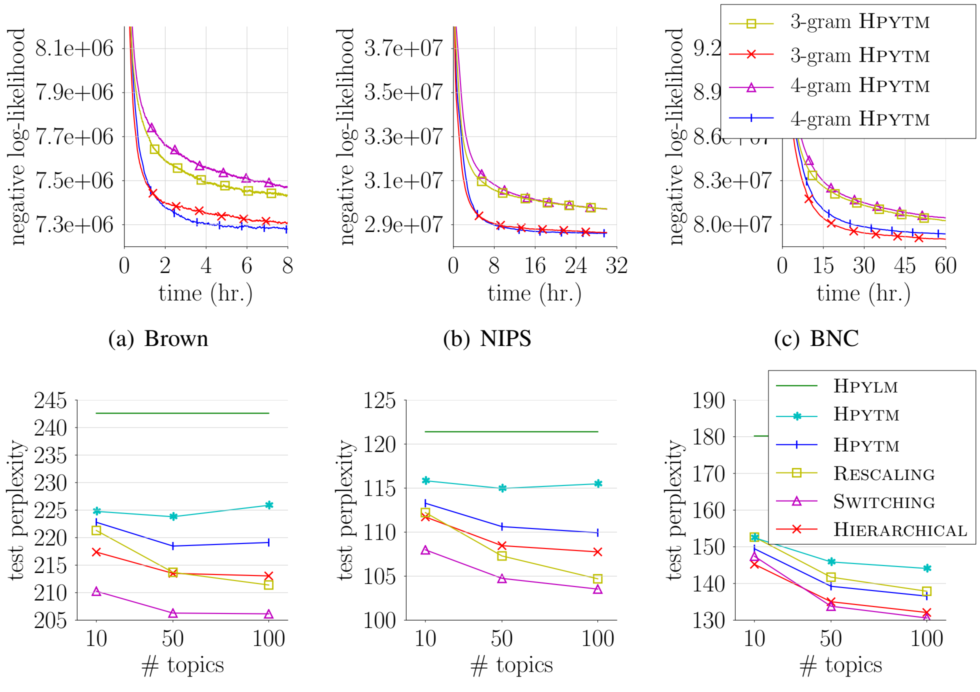
Table 1: Corpus statistics after the pre-processing: We replace words appearing less than min.appear times in training + test documents, or appearing only in a test set with an unknown token. All numbers are replaced with #, while punctuations are remained.

Our primary interest is language model adaptation, which has been studied mainly in the area of speech processing. Conventionally, this adaptation has relied on a heuristic combination of two separately trained models: an n-gram model $p(w|h)$ and a topic model $p(w|d)$. The unigram rescaling, which is a product model of these two models, perform better than more simpler models such as linear interpolation (Gildea and Hofmann, 1999). There are also some extensions to this method (Tam and Schultz, 2009; Huang and Renals, 2008), but these methods have one major drawback: at prediction, the rescaling-based method requires normalization across vocabulary at each word, which prohibits use on applications requiring dynamic (incremental) adaptation, e.g., settings where we have to update the topic distribution as new inputs come in. Tam and Schultz (2005) studied on this incremental settings, but they employ an interpolation. The practical interest here is whether our Bayesian models can rival the rescaling-based method in terms of prediction power. We evaluate this in the next section.

6 Experiments

6.1 Settings

We test the effectiveness of presented models and the blocked sampling method on unsupervised language model adaptation settings. Specifically we



(a) Brown (b) NIPS (c) BNC

(d) Brown (e) NIPS (f) BNC

Figure 4: (a)–(c): Comparison of negative log-likelihoods at training of Hpytm ($K = 50$). Lower is better. Hpytm is trained on both token- and table-based samplers, while Hpytm$_{token}$ is trained only on the token-based sampler. (d)–(f): Test perplexity of various 3-gram models as a function of number of topics on each corpus.

concentrate on the *dynamic* adaptation: We update the posterior of language model given previously observed contexts, which might be decoded transcripts at that point in ASR or MT.

We use three corpora: the Brown, BNC and NIPS. The Brown and BNC are balanced corpora that consist of documents of several genres from news to romance. The Brown corpus comprises 15 categories. We selected two documents from each category for the test set, and use other 470 documents for the training set. For the NIPS, we randomly select 1,500 papers for training and 50 papers for testing. For BNC, we first randomly selected 400 documents from a written corpus and then split each document into smaller documents every 100 sentences, leading to 6,262 documents, from which we randomly selected 100 documents for testing, and other are used for training. See Table 1 for the pre-processing of unknown types and the resulting corpus statistics.

For comparison, besides our proposed Hierarchical and Switching models, we prepare various models for baseline. Hpylm is a n-gram language model without any topics. We call the model without the global G_h^0 introduced in Section 2.2 Hpytm. To see the effect of the table-based sampler, we also prepare Hpytm$_{token}$, which is trained only on the token-based sampler. Rescaling is the unigram rescaling. This is a product model of an n-gram model $p(w|h)$ and a topic model $p(w|d)$, where we learn each model separately and then combine them by:

$$p(w|h,d) \propto \left(\frac{p(w|d)}{p(w)}\right)^{\beta} p(w|h). \qquad (14)$$

We set β in (14) to 0.7, which we tuned with the Brown corpus.

6.2 Effects of Table-based Sampler

We first evaluate the effects of our blocked sampler at training. For simplicity, we concentrate on the Hpytm with $K = 50$. Table 4(a)–(c) shows negative likelihoods of the model during training. On all corpora, the model with the table-based sampler reached the higher probability space with much faster speed on both 3-gram and 4-gram models.

6.3 Perplexity Results

Training For burn-in, we ran the sampler as follows: For HPYLM, we ran 100 Gibbs iterations. For RESCALING, we ran 900 iterations on LDA and 100 iterations on HPYLM. For all other models, we ran 500 iterations of the Gibbs; HPYTM$_{token}$ is trained only on the token-based sampler, while for other models, the table-based sampler is performed after the token-based sampler.

Evaluation We have to adapt to the topic distribution of unseen documents incrementally. Although previous works have employed incremental EM (Gildea and Hofmann, 1999; Tam and Schultz, 2005) because their inference is EM/VB-based, we use the left-to-right method (Wallach et al., 2009), which is a kind of particle filter updating the posterior topic distribution of a test document. We set the number of particles to 10 and resampled each particle every 10 words for all experiments. To get the final perplexity, after burn-in, we sampled 10 samples every 10 iterations of Gibbs, calculated a test perplexity for each sample, and averaged the results.

Comparison of 3-grams Figure 4(d)–(f) shows perplexities when varying the number of topics. Generally, compared to the HPYTM$_{token}$, the HPYTM got much perplexity gains, which again confirm the effectiveness of our blocked sampler. Both our proposed models, the HIERARCHICAL and the SWITCHING, got better performances than the HPYTM, which does not place the global model G_h^0. Our SWITCHING model consistently performed the best. The HIERARCHICAL performed somewhat worse than the RESCALING when K become large, but the SWITCHING outperformed that.

Comparison of 4-grams and beyond We summarize the results with higher order n-grams in Table 2, where we also show the time for prediction. We fixed the number of topics $K = 100$ because we saw that all models but HPYTM$_{token}$ performed best at $K = 100$ when $n = 3$. Generally, the results are consistent with those of $n = 3$. The models with $n = \infty$ indicate a model extension using the Bayesian variable-order language model (Mochihashi and Sumita, 2008), which can naturally be integrated with our generative models. By this extension, we can prune unnecessary nodes stochastically during training. We can see that this ∞-gram did not hurt performances, but the sampled model get much more compact; in BNC, the number of nodes of the SWITCHING with 4-gram is about 7.9M, while the one with ∞-gram is about 3.9M. Note that our models require no explicit normalization, thereby drastically reducing the time for prediction compared to the RESCALING. This difference is especially remarkable when the vocabulary size becomes large.

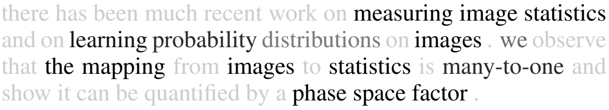
		NIPS		BNC	
Model	n	PPL	time	PPL	time
HPYLM	4	117.2	59	169.2	74
HPYLM	∞	117.9	61	173.1	59
RESCALING	4	101.4	19009	130.3	36323
HPYTM	4	107.0	1004	133.1	980
HPYTM	∞	107.2	1346	133.6	1232
HIERARCHICAL	4	106.3	1038	129.0	993
HIERARCHICAL	∞	105.7	1337	129.3	1001
SWITCHING	4	**100.0**	1059	**125.5**	991
SWITCHING	∞	100.4	1369	125.7	1006

Table 2: Comparison of perplexity and the time require for prediction (in seconds). The number of topics is fixed to 100 on all topic-based models.

We can see that our SWITCHING performed consistently better than the HIERARCHICAL. One reason for this result might be the mismatch of prediction of the topic distribution in the HIERARCHICAL. The HIERARCHICAL must allocate some (not global) topics to every word in a document, so even the words to which the SWITCHING might allocate the global topic (mainly function words; see below) must be allocated to some other topics, causing a mismatch of allocations of topic.

6.4 Qualitative Results

To observe the behavior in which the SWITCHING allocates some words to the global topic, in Figure 5, we show the posterior of allocating the topic 0 or not at each word in a part of the NIPS training corpus. We can see that the model elegantly identified content and function words, learning the topic distribution appropriately using only semantic contexts. These same results in the HIERARCHICAL are presented in Table 3, where we show some relations between λ_h and context h. Contexts that might be likely to precede nouns have a higher value of λ_h,



there has been much recent work on measuring image statistics and on learning probability distributions on images . we observe that the mapping from images to statistics is many-to-one and show it can be quantified by a phase space factor .

Figure 5: The posterior for assigning topic 0 or not in NIPS by the ∞-gram SWITCHING. Darker words indicate a higher probability of not being assigned topic 0.

λ_h	h
0.0–0.1	in spite, were unable, a sort, on behalf, . regardless
0.5–0.6	assumed it, rand mines, plans was, other excersises
0.9–1.0	that the, the existing, the new, their own, and spatial

Table 3: Some contexts h for various values of λ_h induced by the 3-gram HIERARCHICAL in BNC.

while prefixes of idioms have a lower value. The ∞-gram extension gives us the posterior of n-gram order $p(n|h)$, which can be used to calculate the probability of a word ordering composing a phrase in topic k as $p(w, n|k, h) \propto p(n|h)p(w|k, n, h)$. In Table 4, we show some higher probability topic-specific phrases from the model trained on the NIPS.

7 Conclusion

We have presented modeling and algorithmic contributions to the existing Bayesian n-gram topic model. We explored two different priors to incorporate a global model, and found the effectiveness of the flat structured model. We developed a novel blocked Gibbs move for these types of models to accelerate inference. We believe that this Gibbs operation can be incorporated with other models having a similar hierarchical structure. Empirically, we demonstrate that by a careful model design and efficient inference, a well-defined Bayesian model can rival the conventional heuristics.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Phil Blunsom and Trevor Cohn. 2011. A hierarchical pitman-yor process hmm for unsupervised part of speech induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 865–874, Portland, Oregon, USA, June. Association for Computational Linguistics.

0	46
according to ˆ (#) ˆ section # techniques such as ˆ (b)	ˆ support vectors in high dimentional as decision function set of # observations original data set
83	89
the hierarchical mixtures the rbf units the gating networks grown hme the modular architecture	ˆ linear discriminant images per class multi-class classification ˆ decision boundaries references per class

Table 4: Topical phrases from NIPS induced by the ∞-gram SWITCHING model. ˆ is a symbol for the beginning of a sentence and # represents a number.

Phil Blunsom, Trevor Cohn, Sharon Goldwater, and Mark Johnson. 2009. A note on the implementation of hierarchical dirichlet processes. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 337–340, Suntec, Singapore, August. Association for Computational Linguistics.

Daniel Gildea and Thomas Hofmann. 1999. Topic-based language models using em. In *In Proceedings of EUROSPEECH*, pages 2167–2170.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. 2005. Integrating topics and syntax. In *In Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press.

Songfang Huang and Steve Renals. 2008. Unsupervised language model adaptation based on topic and role information in multiparty meetings. In *in Proc. Interspeech08*, pages 833–836.

F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss. 1991. A dynamic language model for speech recognition. In *Proceedings of the workshop on Speech and Natural Language*, HLT '91, pages 293–295, Stroudsburg, PA, USA. Association for Computational Linguistics.

Robert Lindsey, William Headden, and Michael Stipicevic. 2012. A phrase-discovering topic model using hierarchical pitman-yor processes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 214–222, Jeju Island, Korea, July. Association for Computational Linguistics.

Daichi Mochihashi and Eiichiro Sumita. 2008. The infinite markov model. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1017–1024. MIT Press, Cambridge, MA.

Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, pages 959–968. Association for Computational Linguistics.

Yik-Cheung Tam and Tanja Schultz. 2005. Dynamic language model adaptation using variational bayes inference. In *INTERSPEECH*, pages 5–8.

Yik-Cheung Tam and Tanja Schultz. 2009. Correlated bigram lsa for unsupervised language model adaptation. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1633–1640.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Yee Whye Teh. 2006a. *A Bayesian Interpretation of Interpolated Kneser-Ney*. NUS School of Computing Technical Report TRA2/06.

Yee Whye Teh. 2006b. A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 985–992, Sydney, Australia, July. Association for Computational Linguistics.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA. ACM.

Hanna M. Wallach. 2006. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 977–984.

Xuerui Wang, Andrew McCallum, and Xing Wei. 2007. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 697–702, Washington, DC, USA. IEEE Computer Society.

Frank Wood and Yee Whye Teh. 2009. A hierarchical nonparametric Bayesian approach to statistical language model domain adaptation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 12.