Smoothing for Bracketing Induction

Xiangyu Duan, Min Zhang*, Wenliang Chen
Soochow University, China
Institute for Infocomm Research, Singapore
{xiangyuduan, minzhang, wlchen}@suda.edu.cn
{xduan, mzhang, wechen}@i2r.a-star.edu.sg

Abstract

Bracketing induction is the unsupervised learning of hierarchical constituents without labeling their syntactic categories such as verb phrase (VP) from natural raw sentences. Constituent Context Model (CCM) is an effective generative model for the bracketing induction, but the CCM computes probability of a constituent in a very straightforward way no matter how long this constituent is. Such method causes severe data sparse problem because long constituents are more unlikely to appear in test set. To overcome the data sparse problem, this paper proposes to define a non-parametric Bayesian prior distribution, namely the Pitman-Yor Process (PYP) prior, over constituents for constituent smoothing. The PYP prior functions as a back-off smoothing method through using a hierarchical smoothing scheme (HSS). Various kinds of HSS are proposed in this paper. We find that two kinds of HSS are effective, attaining or significantly improving the state-of-the-art performance of the bracketing induction evaluated on standard treebanks of various languages, while another kind of HSS, which is commonly used for smoothing sequences by n-gram Markovization, is not effective for improving the performance of the CCM.

1 Introduction

Grammar induction, the unsupervised learning of hierarchical syntactic structure from natural language text, has long been of interest to computational linguists for a variety of reasons: to help to construct large treebanks, to study language acquisition by children, or to induce bilingual synchronous grammar for statistical machine translation. In recent years we have seen considerable improvement in the performance of grammar induction. Most of the researches target at inducing either *constituents* [Clark, 2001; Klein and Manning, 2002; Klein and Manning, 2004; Bod, 2007; Seginer, 2007; Hänig, 2010; Ponvert *et al*., 2011] or *dependencies* [Klein and Manning, 2004; Headden III *et al*., 2009; Cohen and Smith, 2009; Cohn *et al*., 2010; Spitkovsky *et al*., 2010].

This paper falls into the first category: inducing constituents. Apart from unsupervised constituent labeling of syntactic categories such as verb phrase (VP) given gold-standard constituent surface strings [Haghighi and Klein, 2006; Borensztajn and Zuidema, 2007], recent constituent induction researches mainly focus on **bracketing induction**, which is to induce all constituents without labeling their syntactic categories given raw texts [Klein and Manning, 2002; Klein and Manning, 2004; Bod, 2007; Seginer, 2007; Hänig, 2010; Ponvert *et al*., 2011]. Diverse techniques have been applied for the bracketing induction, ranging from probabilistic methods to non-probabilistic methods. This paper targets at the bracketing induction, and is based on one probabilistic model: Constituent Context Model (CCM) [Klein and Manning, 2002], which is an effective generative model for the bracketing induction.

In the CCM, probability of a constituent is computed in a straightforward way no matter how long this constituent is. Severe data sparse problem is caused because long constituents are more unlikely to appear in test set. This paper proposes a method for constituent smoothing by defining a non-parametric Bayesian prior distribution over constituents, namely the Pitman-Yor Process (PYP) prior [Pitman and Yor, 1997]. The PYP prior functions as an elegant back-off smoothing method through using a hierarchical smoothing scheme (HSS). Various kinds of HSS are proposed in this paper. We find that two kinds of HSS, both of which back-off a constituent's probability to its boundary's probability, significantly improve the performance of the CCM, and attain or significantly improve the state-of-the-art performance of the bracketing induction evaluated on standard treebanks of various languages. Another kind of HSS inspired by language modeling [Teh, 2006b] is proposed. It applies the commonly used *n*-gram Markovization on constituents, but it turns out ineffective for improving the performance of the CCM.

The rest of the paper is structured as follows: In section 2, we introduce the original CCM. In section 3, we present the proposed Bayesian method for the constituent smoothing. Experiments and results are presented in section 4. Conclusion is presented in section 5.

* Corresponding Author

2 Constituent Context Model (CCM)

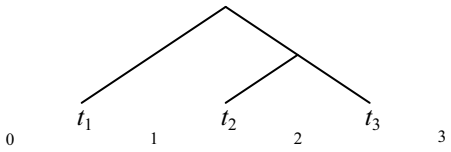
The CCM is an important breakthrough for bracketing induction [Klein and Manning, 2002], and is the first to outperform a right-branching baseline on English. Unlike Probabilistic Context Free Grammars which are defined on production rules over trees, the CCM deals with both tree spans and non-tree spans. All spans are represented by two kinds of strings: constituents and contexts. The CCM is a generative model defined over such representations.

Note that, the terminology "constituent" in the CCM stands for contiguous surface string of either tree span or non-tree span, and such usage of "constituent" will be adopted throughout the following parts of this paper. It distinguishes from the usual usage of "constituent", which only stands for tree spans.

2.1 Constituents and Contexts

Constituents are contiguous surface strings of sentence spans (subsequences), contexts are ordered pairs of tokens preceding and following the constituents. Each sentence span, either a tree span or a non-tree span, is represented by a constituent and a context.

Figure 1 shows examples of constituents and contexts of a sentence, which consists of t_1, t_2, and t_3. A latent bracketing tree with t_1, t_2, and t_3 as terminal nodes is illustrated at the top of Figure 1. The bottom numbers of the tree are indexes for denoting spans.



tree span	constituent	context
<0,1>	t_1	◇ - t_2
<1,2>	t_2	t_1 - t_3
<2,3>	t_3	t_2 - ◇
<1,3>	t_2 t_3	t_1 - ◇
<0,3>	t_1 t_2 t_3	◇ - ◇

non-tree span	constituent	context
<0,0>	*nil*	◇ – t_1
<1,1>	*nil*	t_1 – t_2
<2,2>	*nil*	t_2 - t_3
<3,3>	*nil*	t_3 - ◇
<0,2>	t_1 t_2	◇ - t_3

Figure 1. Illustration of constituents and contexts over a sentence "t_1 t_2 t_3"

Given the tree, two sets of constituents and contexts can be extracted as shown in tables of Figure 1. One is about the tree spans, the other is about the non-tree spans. ◇ appearing in the contexts denotes a sentence boundary. *nil* appearing in the constituents of the non-tree spans denotes an empty span, which is actually a space between two terminals (or between a terminal and ◇). In the CCM, the terminal nodes are part-of-speech (POS) tags, which are also set as terminals in most of our proposed methods, except one which uses both lexicons and POS tags as terminal nodes. Details are presented in section 3.

2.2 Generative Model

For each sentence span, the CCM treats its constituent and context as observed strings, and uses a Boolean variable to indicate whether this span is a tree span or not. The latent bracketing structure of whole sentence can be read off from these Boolean variables. Concretely, there are four kinds of conditional probabilities employed in the CCM: p(α |*true*), p(β |*true*), p(α |*false*) and p(β |*false*), where α denotes a constituent, β denotes a context, *true* indicates that the current considered span is a tree span, *false* for a non-tree span.

The probability of a sentence S and its bracketing tree B is defined as below:

$$\begin{aligned} p(S,B) &= p(B)p(S \mid B) \\ &= p(B) \prod_{\langle i,j \rangle \in T(B)} p(\alpha_{ij} \mid true) p(\beta_{ij} \mid true) \\ &\quad \prod_{\langle i,j \rangle \notin T(B)} p(\alpha_{ij} \mid false) p(\beta_{ij} \mid false) \end{aligned} \quad (1)$$

where $T(B)$ denotes the set of tree spans contained in B, i and j are span's boundary indexes over S. First, a bracketing tree B is chosen according to some distribution $p(B)$, then the sentence S is generated given that bracketing tree B. In the CCM, $p(B)$ is taken to be uniform over all possible binary bracketing trees (binary trees with no crossing brackets) of S and zero elsewhere. So, the CCM only induces binary bracketing trees. $p(S|B)$ is the product of the probabilities of all constituents and all contexts, given whether their spans are tree spans or not. An EM-like algorithm is applied on the estimation of this generative model [Klein, 2005].

3 Bayesian Modeling of the CCM for Constituent Smoothing

Bayesian modeling is proposed in this section for overcoming data sparse problem faced by the original CCM, which computes probability of a constituent by Maximum-Likelihood-Estimation (MLE, employed in the EM algorithm) no matter how long this constituent is. Long constituents tend to not appear in unseen data, and cause severe data sparse problem.

We propose to define a non-parametric Bayesian prior distribution, namely the PYP prior, over constituents for constituent smoothing. We use Markov Chain Monte Carlo (MCMC) sampling to infer smoothed constituents from posterior distributions over constituents.

At first, the Bayesian framework is introduced in this section. Then, under the Bayesian framework, the constituent smoothing by placing the PYP prior over the

constituents is presented, followed by the presentation of MCMC sampling for inference.

3.1 Bayesian Framework

The Bayesian framework for the original CCM is to compute the posterior probabilities of $p(\alpha|true)$, $p(\beta|true)$, $p(\alpha|false)$ and $p(\beta|false)$ after observing a raw corpus. For the ease of explanation, we take a raw sentence S and its bracketing tree B for example:

$$\begin{aligned} p(S \mid B) &= p(\boldsymbol{c})p(\boldsymbol{d})p(\boldsymbol{e})p(\boldsymbol{f}) \\ &= \prod_{c \in T(B)} p(c) \prod_{d \in T(B)} p(d) \prod_{e \notin T(B)} p(e) \prod_{f \notin T(B)} p(f) \\ &= \prod_{\langle i,j \rangle \in T(B)} p(\alpha_{ij} \mid true) p(\beta_{ij} \mid true) \\ &\quad \prod_{\langle i,j \rangle \notin T(B)} p(\alpha_{ij} \mid false) p(\beta_{ij} \mid false) \end{aligned}$$

where $\boldsymbol{c}=(c_1,c_2,\ldots)$ is a sequence of constituents composing the bracketing tree B, $c \in T(B)$ means c is a tree span constituent. Similarly, $e \notin T(B)$ means e is a non-tree span constituent. So do the context variables d and f.

Using the Bayes' rule, the posterior over $\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}$, and $\boldsymbol{f}$ is:

$$p(\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f} \mid S) \propto p(S \mid \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f})\, p(\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f})$$

where $p(S \mid \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f})$ is either equal to 1 (when S and $\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}$ are consistent) or 0 (otherwise). Furthermore, under the independent assumption of the original CCM, $p(\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}) = p(\boldsymbol{c})\, p(\boldsymbol{d})\, p(\boldsymbol{e})\, p(\boldsymbol{f})$. Therefore, the task of Bayesian modeling of the original CCM turns out to consist of modeling only the four prior distributions $p(\boldsymbol{c})$, $p(\boldsymbol{d})$, $p(\boldsymbol{e})$, and $p(\boldsymbol{f})$.

Various kinds of distributions can be set as the prior distribution over $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{e}$, and $\boldsymbol{f}$. Over constituents $\boldsymbol{c}$ and $\boldsymbol{e}$, we put the PYP distribution as the prior distribution for constituent smoothing due to its inherent back-off smoothing property [Teh, 2006b; Blunsom and Cohn, 2010; Cohn *et al*., 2010; Shindo *et al*., 2012]. Over contexts $\boldsymbol{d}$ and $\boldsymbol{f}$, we apply a simpler smoothing method by using the Dirichlet distribution as the prior distribution over $\boldsymbol{d}$ and $\boldsymbol{f}$ because contexts do not confront data sparse problem as seriously as constituents do.

3.2 Constituent Smoothing

In this section, we propose to use the PYP prior for smoothing $\boldsymbol{c}$ and $\boldsymbol{e}$. Taking $\boldsymbol{c}$ for example:

$$\begin{aligned} c &\sim G_p \\ G_p &\sim \mathrm{PYP}(a, b, p_{base}) \end{aligned} \tag{2}$$

An element of $\boldsymbol{c}$, the c, is drawn *iid* from the distribution G_p, which is drawn from the PYP distribution with a discount parameter a, a strength parameter b, and a base distribution p_{base}. By integrating out G_p, the predictive probability of c_{n+1} taking a specific tree span constituent surface string x given $\boldsymbol{c}=(c_1, \ldots, c_n)$ is:

$$p(c_{n+1} = x \mid \boldsymbol{c}) = \frac{n_x - K_x a + (Ka + b) p_{base}(x)}{n + b} \tag{3}$$

where n_x is the frequency of x collected from $\boldsymbol{c}$, which consists of n elements. Eq. (3) can be interpreted in a restaurant metaphor: n customers (c_1, …, c_n in this case) sequentially enter a restaurant. They are seated at K ($K \leqslant n$) tables. Customers at each table share only one menu. n_x is the number of customers choosing a specific menu x, K_x is the number of tables equipped with menu x. See (Goldwater et al., 2011) for details of the restaurant metaphor.

Compared to the MLE $p(c_{n+1}=x|\boldsymbol{c})=n_x/n$, eq. (3) manifests a back-off smoothing scheme. n_x is discounted by $K_x a$, and is back-off to the probability $p_{base}(x)$ with a back-off-weight $(Ka+b)/(n+b)$. For the ease of presentation in the following, we use a succinct form of eq. (3):

$$p_{predictive}(x) = p_{discounted}(x) + BOW \times p_{base}(x) \tag{4}$$

where BOW denotes the back-off-weight.

Through the above introduction of the PYP, we can see that the PYP prior shown in formula (2) actually encodes a back-off scheme: $p_{predictive} \rightarrow p_{base}$, where $\rightarrow$ denotes the back-off direction. Recursive calling of p_{base} can build a hierarchical smoothing scheme (HSS).

In this section, we propose various kinds of HSS for constituent smoothing. At first, we propose two kinds of HSS, both of which deal with the boundary representations of constituents. One is named Constituent- Boundary HSS, the other is named Lexicalized HSS. At last, we propose a kind of HSS inspired by language modeling [Teh, 2006b], which uses n-gram Markovization over sequences of words. Though n-gram Markovization is the commonly used smoothing method that is easy to come up with, it is proven ineffective for the CCM as presented in section 4.5.

Constituent-Boundary Hierarchical Smoothing Scheme (CB-HSS)

Taking $\boldsymbol{c}$ for example:

$$\begin{aligned} c &\sim G_{con} \\ G_{con} &\sim \mathrm{PYP}(a_{con}, b_{con}, p_{bound}(\cdot)) \end{aligned}$$

The above formula encodes a back-off scheme: $p_{predictive} \rightarrow p_{bound}$, where p_{bound} models the generation of a constituent c by first generating its boundary representation from a Dirichlet distribution, then generating its middle POS tags from a uniform distribution over POS tags:

$$\begin{aligned} p_{bound}(c) &= B(br(c)) \times (1 / |POS|^{|c|-|br(c)|}) \\ r &\sim B \\ B &\sim \mathrm{Dirichlet}(\tau) \end{aligned}$$

where br is a function returning a constituent's boundary representation, $|POS|$ is the size of POS tag set, $|c|-|br(c)|$ is the number of the POS tags of c excluding those in the boundary representation, and r donotes a boundary

representation of a constituent. Each r is drawn from the distribution B, which is drawn from the Dirichlet distribution with a parameter τ.

We use a constituent "$t_1\ t_2\ t_3\ t_4$" for the illustration of the boundary representation. We choose "$t_1\ t_2$" as left boundary, "t_4" as right boundary, if we decide the left boundary width is two and the right boundary width is one. Finally, the boundary representation of the constituent is "$t_1\ t_2\ t_4$". If a constituent spans terminals less than the sum of the left boundary width and the right boundary width, this constituent is represented by itself.

The first row of table 1 illustrates the back-off smoothing scheme of the CB-HSS. The probability of the example constituent "$t_1\ t_2\ t_3\ t_4$" is approximately backed-off to the probability of its boundary representation "$t_1\ t_2\ t_4$" if the left boundary width is 2 and the right boundary width is 1. Because in the CB-HSS, "t_3" is generated from a uniform distribution over POS tags, and has no disambiguating impact, we omit "t_3" in the illustration.

CB-HSS	$t_1\ t_2\ t_3\ t_4 \rightarrow t_1\ t_2\ t_4$
L-HSS	$l_1t_1\ l_2t_2\ l_3t_3\ l_4t_4 \rightarrow l_1t_1\ l_2t_2\ l_4t_4 \rightarrow t_1\ t_2\ t_4$
NB-HSS	$t_3 \mid t_2, t_1, true \rightarrow t_3 \mid t_2, true \rightarrow t_3 \mid true$

Table 1. Illustration of the various kinds of HSS. t_i denotes a POS tag in a constituent. l_i denotes the lexicon of t_i.

Lexicalized Hierarchical Smoothing Scheme (L-HSS)

L-HSS is the lexicalized version of CB-HSS. It enriches the representation of a constituent, which originally consists of POS tags, with corresponding lexicons. The second row of table 1 shows an example lexicalized constituent "$l_1t_1\ \ l_2t_2\ l_3t_3\ \ l_4t_4$", where l denotes a lexicon, t denotes a POS tag. The inclusion of lexicons makes the data sparse problem more serious. More deep hierarchy than the CB-HSS is applied in the L-HSS to the constituent smoothing. Taking tree span constituents for example:

$$\begin{aligned} c_{lex} &\sim G_{lex} \\ G_{lex} &\sim \mathrm{PYP}(a_{lex}, b_{lex}, p_{lex}(\cdot)) \end{aligned}$$

The above formula encodes a back-off scheme: $p_{predictive} \rightarrow p_{lex}$, where p_{lex} models the generation of a lexicalized constituent c_{lex} by first generating its lexicalized boundary from a PYP distribution, then generating its middle lexicons and POS tags from a uniform distribution over all pairs of lexicons and POS tags:

$$\begin{aligned} p_{lex}(c_{lex}) &= B_{lex\text{-}bound}(br(c_{lex})) \\ &\quad \times \frac{1}{|LEXPOS|^{|c_{lex}|-|br(c_{lex})|}} \\ lb &\sim B_{lex\text{-}bound} \\ B_{lex\text{-}bound} &\sim \mathrm{PYP}(a_{lex\text{-}bound}, b_{lex\text{-}bound}, p_{lex\text{-}bound}(\cdot)) \end{aligned}$$

where $br(c_{lex})$ returns the lexicalised boundary of c_{lex}. As illustrated in table 1, the lexicalised boundary of "$l_1t_1\ \ l_2t_2\ l_3t_3\ \ l_4t_4$" is "$l_1t_1\ \ l_2t_2\ \ l_4t_4$" if left boundary width is 2, and right boundary width is 1. $|LEXPOS|$ denotes the set size of all pairs of lexicons and POS tags, $|c_{lex}|-|br(c_{lex})|$ gets the number of middle elements of c_{lex}, lb denotes a lexicalized boundary.

The above formula encodes the second level of the back-off scheme: $p_{lex} \rightarrow p_{lex\text{-}bound}$, where the uniform distribution over pairs of lexicons and POS tags is neglected due to its inability of disambiguation. Furthermore, $p_{lex\text{-}bound}$ removes the lexicalization from lexicalized boundary by describing the generation of a lexicalised boundary lb by first generating its POS tags from a Dirichlet distribution, then generating its lexicons from a uniform distribution over vocabulary V:

$$\begin{aligned} p_{lex\text{-}bound}(lb) &= B_{pos\text{-}bound}(unlex(lb)) \times (1 / |V|^{|lb|}) \\ pb &\sim B_{pos\text{-}bound} \\ B_{pos\text{-}bound} &\sim \mathrm{Dirichlet}(\tau) \end{aligned}$$

where pb denotes a boundary representation without lexicalization, $unlex$ is the function removing the lexicalization and leaving only the POS tags. Thus, the third level of the back-off scheme is constructed: $p_{lex\text{-}bound} \rightarrow$ Dirichlet, where the Dirichlet is over POS boundary. The uniform distribution is neglected here again.

Overall, the L-HSS encodes the hierarchical back-off scheme: $p_{predictive} \rightarrow p_{lex} \rightarrow p_{lex\text{-}bound} \rightarrow$ Dirichlet. The second row of table 1 illustrates an example of the L-HSS.

N-gram Based Hierarchical Smoothing Scheme (NB-HSS)

Inspired by language modeling, NB-HSS performs smoothing in an n-gram Markov way. Take a tree span constituent "$t_1\ t_2\ t_3$" for example. Each t_i is a POS tag. If we use *tri*-gram computation,

$$\begin{aligned} p(t_1\ t_2\ t_3 \mid true) = {} & p(t_1 \mid \langle s\rangle, \langle s\rangle, true)p(t_2 \mid t_1, \langle s\rangle, true) \\ & p(t_3 \mid t_2, t_1, true)\ p(\langle /s\rangle \mid t_3, t_2, true) \end{aligned}$$

where <s> and </s> denote the start and end of a constituent, respectively. The probability of a constituent is smoothed by the product of probabilities of the component POS tags given their preceding tags and the constituent's span type (tree span in the above example). The hierarchical PYP [Teh, 2006a,b] is applied in NB-HSS for modeling n-grams. The third row of table 1 illustrates the NB-HSS for the *tri*-gram $p(t_3|t_2, t_1, true)$.

3.3 MCMC Sampling for Inferring the Latent Bracketing Trees

The inferring of the latent bracketing trees is based on the posteriors over $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{e}$, and $\boldsymbol{f}$, which is solely determined by the priors over $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{e}$, and $\boldsymbol{f}$ as presented in section 3.1. MCMC sampling directly makes use of the priors $p(\boldsymbol{c})$, $p(\boldsymbol{d})$, $p(\boldsymbol{e})$, and $p(\boldsymbol{f})$ to infer the latent bracketing trees from a raw text corpus. We apply a kind of MCMC sampling, the blocked Gibbs sampling, to perform the inference. Gibbs sampling is a widely applied approach for obtaining random

samples from a probabilitic distribution. The blocked Gibbs sampling is to obtain blocks of samples simultaneously. In our case, the blocked Gibbs sampling is to simultaneously obtain sentence-level samples of tree spans and non-tree spans' constituents and contexts.

The routine of our blocked Gibbs sampling is similar to that applied for inducing PCFGs [Johnson *et al*., 2007]. It consists of the following three steps: for each sentence, 1) calculate the inside score in a bottom-up manner, 2) sample a bracketing tree in a top-down manner based on the inside score obtained in the first step, and 3) accept or reject the sampled bracketing tree based on the Metropolis-Hastings (MH) test. These three steps are repeated until a pre-defined number of sampling iterations is attained.

The computation of the inside score [Klein 2005] in the first step is based on the predictive distributions that are derived from the priors over $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{e}$, and $\boldsymbol{f}$. For constituent $\boldsymbol{c}$ and $\boldsymbol{e}$, the predictive distribution shown in eq. (3) encodes a back-off smoothing scheme. Through using various kinds of the HSS proposed in section 3.2, $\boldsymbol{c}$ and $\boldsymbol{e}$ are smoothed in the process of MCMC sampling.

We exploit multiple sampled bracketing trees of the same raw sentence to generate the final bracketing tree. For example, given two kinds of sampled binary bracketings are obtained: $<<t_1\ t_2>\ t_3>$ and $<t_1\ <t_2\ t_3>>$, we extract the agreement and give up the disagreement. Finally we get $<t_1\ t_2\ t_3>$. Such method breaks the binary branching limitation imposed by using the inside chart during MCMC sampling.

4 Experiments

4.1 Experimental Setting

Experiments were carried out on English, Chinese, and German. Apart from some grammar induction works that include the test sentences into the training set, we adopted the experimental setting that keeps the evaluation sets blind to the models during training [Cohen *et al*., 2008; Headden III *et al*., 2009; Spitkovsky *et al*., 2010; Ponvert *et al*., 2011]. For English (WSJ) [Marcus *et al*., 1999], sections 00-22 was used for training, section 23 for testing and section 24 for development. For Chinese (CTB) [Palmer *et al*., 2006], the data split of Duan *et al*. [2007] was adopted. For German (NEGRA) [Krenn *et al*., 1998], the first 18602 sentences were used for training, the last 1000 sentences were used for development, the penultimate 1000 sentences were used for testing.

Evaluation metrics (Klein, 2005) are brackets' Unlabeled Precision (UP), Unlabeled Recall (UR), and the unlabeled f-score F1, which is the harmonic mean of UP and UR: F1=2*UP*UR/(UP+UR). Brackets spanning one word, multiplicity of brackets are ignored in the evaluation.

We re-implemented the original CCM[†] as the baseline. We used a tool implementing the seating arrangements in the restaurant of the PYP [Blunsom et al., 2009][‡].

[†] Same F1 score (71.9) to the original CCM using the same data.

We set burn-in as 20, and ran 200 iterations of blocked Gibbs sampling in all of our experiments.

4.2 Initialization

We build the initialization procedure based on a splitting process [Klein and Manning, 2002]. Given k terminals, we choose a split point at random, then recursively build trees on each side of the split point. $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{e}$, and $\boldsymbol{f}$ are initialized according to these bracketing trees.

4.3 Hyper-parameters

There are two categories of hyper-parameters in our work: one category is the discount parameter a_x and the strength parameter b_x of all the proposed PYP distributions, the other category is τ of all the proposed Dirichlet priors.

The values of a_x and b_x are automatically estimated by being sampled from their posterior distribution. Vague priors are applied: $a_x \sim \text{Beta}(1, 1)$, $b_x \sim \text{Gamma}(1, 1)$. A slice sampler[§] [Neal, 2003] is applied for this sampling. In the NB-HSS, restaurants of the same history length share the same a_x and b_x, special computation of the posterior over a_x and b_x is adopted (Teh, 2006a).

For the Dirichlet prior, the parameter τ is divided into two categories: τ_t for tree span, and τ_{nt} for non-tree span. They are tuned on the development sets.

In addition, boundary width used in the CB-HSS and the L-HSS also needs tuning. We set left/right boundary width as two/one for both English and Chinese, while set left/right boundary width as one/four for German.

4.4 Phrasal Punctuations

Punctuations are usually ignored in the grammar induction research, while some of them provide informative phrase boundaries within a sentence. We use the punctuation set shown in table 2 as phrasal punctuations for indicating phrase boundaries [Seginer, 2007; Ponvert *et al*., 2011]. Any tree span except the entire sentence must not cover a phrasal punctuation. Such constraint reduces the search space and bracketing errors.

Eng/WSJ	Chi/CTB	Ger/NEGRA
, . ! ? ;	， 。 ： ！ ？	, .

Table 2. Sets of phrasal punctuations of the different languages.

4.5 Evaluation

The proposed constituent smoothing methods were evaluated on sentences of at most 10 words after the removal of punctuations and null elements in the three treebanks [Klein and Manning, 2002]. Table 3 shows the evaluation results of the three languages.

[‡] *http://staffwww.dcs.shef.ac.uk/people/T.Cohn/restaurant.tar.gz*. This tool was originally developed for Chinese Restaurant Process (CRP). We modified it for the PYP applications.

[§] We made use of the slice sampler included in Mark Johnson's Adaptor Grammar implementation *http://www.cog.brown.edu/~mj/Software.htm*

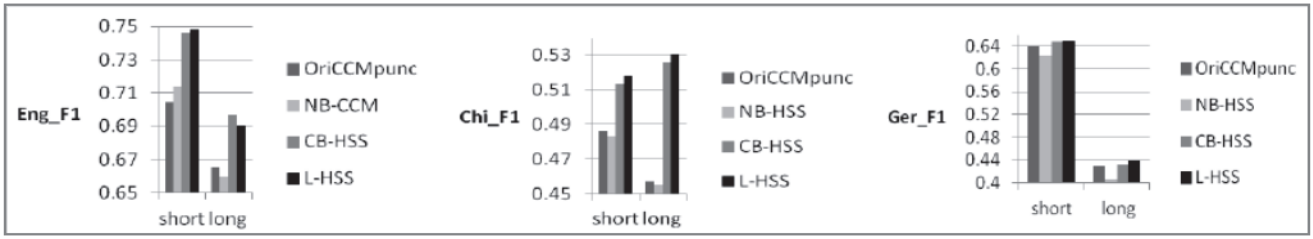


Figure 2. Evaluation (F1) on short and long brackets of all the three languages.

The performance of the baseline, the original CCM (denoted by OriCCM), is reported in the first row of table 3. The suffix punc stands for using the phrasal punctuation constraint. Clearly, the phrasal punctuation constraint is helpful when comparing the first two rows of table 3. Hence, we conducted experiments on all kinds of our proposed HSS, shown in the middle part of table 3, based on using the phrasal punctuation constraint.

The L-HSS attains the best performance among all kinds of HSS. The CB-HSS performs slightly worse than the L-HSS. Both the L-HSS and the CB-HSS significantly improve the performance of the $\text{OriCCM}_{\text{punc}}$, while the NB-HSS is not effective for improving the performance, and harms the performance on Chinese and German data. The effectiveness of the L-HSS and the CB-HSS comes from the constituent smoothing on long constituents. Figure 2 shows F1 scores evaluated on short brackets and long brackets. We set brackets spanning no more than 3 words as short brackets, and set brackets spanning more than 3 words and less than 8 words as long brackets. We omit longer brackets in this evaluation because most of them span whole sentences.

	Eng/WSJ	Chi/CTB	Ger/NEGRA
OriCCM	71.1	52.7	64.6
$\text{OriCCM}_{\text{punc}}$	75.0	57.1	66.3
CB-HSS	78.1	60.7	66.9
L-HSS	**78.4**	**61.4**	**67.8**
NB-HSS	75.4	56.7	63.7
CCL	72.1	48.8	53.0
PRLG	70.5	59.6	63.2
U-DOP*	77.9	42.8	63.8

Table 3. The performances of bracketing induction evaluated by F1 (%) on the test sets consisting of sentences of no more than 10 words.

Figure 2 shows that the L-HSS and the CB-HSS perform significantly better than the $\text{OriCCM}_{\text{punc}}$ on long brackets, which also benefits the performance on short brackets because that, for brackets containing both long brackets and short brackets, correct long brackets also indicate correct short brackets. If long brackets, which is hard to be identified, are wrongly identified, short brackets will be influenced.

Comparison to other bracketing induction systems: The bottom part of table 3 shows the performances of other grammar induction systems. CCL [Seginer, 2007] is an incremental parsing algorithm using common cover links, which is a special representation of syntactic structures. PRLG [Ponvert et al., 2011] performs grammar induction via cascaded induction of chunks, and the induced chunks constitute a bracketing tree. These two methods do not use POS tags. They induce bracketing tree over raw text words. Their best performances are comparable to the original CCM, but when they induce bracketing tree over POS tags, they perform badly, indicating that POS tags are too coarse-grained for them.

Both CCL and PRLG use the phrasal punctuation constraint, while U-DOP* [Bod, 2007] does not. U-DOP* is an efficient all sub-tree method for grammar induction. It decomposes a tree into sub-trees (constrained to be binary branching), which is different to common methods that decompose a tree into context free grammar rules. But U-DOP* is not strictly comparable to our system and other grammar induction systems because the evaluation sets used by U-DOP* is binarized [Bod, 2007; Zhang and Zhao, 2011], while other systems including ours use the original evaluation sets, which contain tree structures with multiple branches. In addition, U-DOP* performance on Chinese is not comparable because U-DOP* used the treebank of CTB3, smaller than our used CTB5.

5 Conclusion

This paper proposes a Bayesian method for constituent smoothing in the CCM, which faces data sparse problem caused by the too straightforward computation of constituents' probabilities. The PYP prior is applied in the proposed Bayesian method. The PYP prior functions as a back-off smoothing method through using a hierarchical smoothing scheme (HSS). Various kinds of HSS are proposed. Experiments show that, two kinds of HSS backing-off to boundaries are effective for smoothing long constituents, resulting in significant improvements over the original CCM. They attain or significantly improve the state-of-the-art performance of the bracketing induction evaluated on treebanks of English, Chinese, and German, while another kind of HSS based on commonly used *n*-gram Markovization on sequences is not effective for improving the performance of the original CCM.

Acknowledgments

This work was supported by the Natural Sciences Foundation of China under grant No. 61273319, and grant No. 61203314. Thanks for the helpful advices of anonymous reviewers.

References

[Blunsom and Cohn,2010]Phil Blunsom and Trevor Cohn. Unsupervised Induction of Tree Substitution Grammars for Dependency Parsing. In *Proc. of the EMNLP*, 2010.

[Blunsom *et al*, 2009] Phil Blunsom, Trevor Cohn, Sharon Goldwater, and Mark Johnson. A Note on the Implementation of Hierarchical Dirichlet Processes. In *Proc. of the ACL-IJCNLP Short Papers*, 2009.

[Bod, 2007] Rens Bod. Is the End of Supervised Parsing in Sight? In *Proc. of ACL*, 2007.

[Borensztajn and Zuidema, 2007] Borensztajn and Zuidema. Bayesian Model Merging for Unsupervised Constituent Labeling and Grammar Induction. *Technical report, ILLC*, 2007.

[Cohen and Smith, 2009] Shay B. Cohen and Noah A. Smith. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proc. of the HLT-NAACL*, 2009.

[Cohn *et al*, 2010] Trevor Cohn, Phil Blunsom, and Sharon Goldwater. Inducing Tree-Substitution Grammars. *Journal of Machine Learning Research*, 2010.

[Duan *et al*, 2007] Xiangyu Duan, Jun Zhao, and Bo Xu. Probabilistic models for action-based Chinese dependency parsing. In *Proc. of ECML/PKDD*, 2007.

[Goldwater *et al*, 2011] Sharon Goldwater, Thomas L. Griffiths and Mark Johnson. Producing Power-Law Distributions and Damping Word Frequencies with Two-Stage Language Models. *Journal of Machine Learning Research*, 2011.

[Hänig, 2010] Christian Hänig. Improvements in unsupervised co-occurence based parsing. In *Proc. of CoNLL*, 2010.

[Haghighi and Klein, 2006] Aria Haghighi and Dan Klein. Prototype-driven grammar induction. In *Proc. of the ACL*, 2006.

[Headden III *et al*, 2009] William P. Headden III, Mark Johnson, and David McClosky. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proc. of the HLT-NAACL*, 2009.

[Johnson *et al*, 2007] Mark Johnson, Thomas Griffiths, and Sharon Goldwater. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proc. of the NAACL-HLT*, 2007.

[Klein and Manning, 2002] Dan Klein and Christopher D. Manning. A generative constituent-context model for improved grammar induction. In *Proc. of the ACL*, 2002.

[Klein and Manning, 2004] Dan Klein and Christopher D. Manning. Corpus-based induction of syntactic structure: Models of dependency and constituent. In *Proc. of the ACL*, 2004.

[Klein 2005] Dan Klein. The Unsupervised Learning of Natural Language Structure. *Ph.D. Thesis, Stanford University*, 2005.

[Krenn *et al*, 1998] B. Krenn, T. Brants, W. Skut, and Hans Uszkoreit. A linguistically interpreted corpus of German newspaper text. In *Proc. of the ESSLLI Workshop on Recent Advances in Corpus Annotation*, 1998.

[Marcus *et al*, 1993] M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Compuational Linguistics*, 1993.

[Palmer *et al*, 2005] M. Palmer, F. D. Chiou, N. Xue, and T. K. Lee. Chinese Treebank 5.0. *LDC*, 2005.

[Pitman and Yor, 1997] Jim Pitman and Marc Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 1997.

[Ponvert *et al*,, 2011] Elias Ponvert, Jason Baldridge and Katrin Erk. Simple Unsupervised Grammar Induction from Raw Text with Cascaded Finite State Models. In *Proc. of the ACL-HLT*, 2011.

[Seginer, 2007] Yoav Seginer. Fast unsupervised incremental parsing. In *Proc. of the ACL*, 2007.

[Shindo *et al*, 2012] Hiroyuki Shindo, Yusuke Miyao, Akinori Fujino and Masaaki Nagata. Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing. In *Proc. of the ACL*, 2012.

[Smith and Eisner, 2004] Noah A. Smith and Jason Eisner. Annealing techniques for unsupervised statistical language learning. In *Proc. of the ACL*, 2004.

[Spitkovsky *et al*, 2010] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Proc. of the NAACL-HLT*, 2010.

[Teh, 2006a] Yee Whye Teh. A Bayesian interpretation of interpolated Kneser-Ney. *Technical Report TRA2/06, School of Computing, National University of Singapore*, 2006.

[Teh, 2006b] Yee Whye Teh. A Hierarchical Bayesian Language Model based on Pitman-Yor Processes. In *Proc. of the ACL*, 2006.

[Zhang and Zhao, 2011] Xiaotian Zhang and Hai Zhao, Unsupervised Chinese Phrase Parsing Based on Tree Pattern Mining. In *Proc. of the 11th Confernece of China Computational Linguistics*, 2011.