Nonparametric Bayesian Semi-supervised Word Segmentation

Ryo Fujii **Ryo Domoto**
Hakuhodo Inc. R&D Division
5-3-1 Akasaka, Minato-ku, Tokyo
{ryo.b.fujii,ryo.domoto}@hakuhodo.co.jp

Daichi Mochihashi
The Institute of Statistical Mathematics
10-3 Midori-cho, Tachikawa city, Tokyo
daichi@ism.ac.jp

Abstract

This paper presents a novel hybrid generative/discriminative model of word segmentation based on nonparametric Bayesian methods. Unlike ordinary discriminative word segmentation which relies only on labeled data, our semi-supervised model also leverages a huge amounts of unlabeled text to automatically learn new "words", and further constrains them by using a labeled data to segment non-standard texts such as those found in social networking services.

Specifically, our hybrid model combines a discriminative classifier (CRF; Lafferty et al. (2001) and unsupervised word segmentation (NPYLM; Mochihashi et al. (2009)), with a transparent exchange of information between these two model structures within the semi-supervised framework (JESS-CM; Suzuki and Isozaki (2008)). We confirmed that it can appropriately segment non-standard texts like those in Twitter and Weibo and has nearly state-of-the-art accuracy on standard datasets in Japanese, Chinese, and Thai.

1 Introduction

For any unsegmented language, especially East Asian languages such as Chinese, Japanese and Thai, word segmentation is almost an inevitable first step in natural language processing. In fact, it is becoming increasingly important lately because of the growing interest in processing user-generated media, such as Twitter and blogs. Texts in such media are often written in a colloquial style that contains many new words and expressions that are not present in any existing dictionaries. Since such words are theoretically infinite in number, we need to leverage unsupervised learning to automatically identify them in corpora.

For this purpose, ordinary supervised learning is clearly unsatisfactory; even hand-crafted dictionaries will not suffice because functional expressions more complex than simple nouns need to be recognized through their relationship with other words in text, which also might be unknown in advance. Previous studies of this issue used character and word information in the framework of supervised learning (Kruengkrai et al., 2009; Sun et al., 2009; Sun and Xu, 2011). However, they

(1) did not explicitly model new words, or
(2) did not give a seamless combination with discriminative classifiers (e.g., they just used a threshold to discriminate between known and unknown words).

In contrast, unsupervised word segmentation methods (Goldwater et al., 2006; Mochihashi et al., 2009) use nonparametric Bayesian generative models for word generation to infer the "words" only from observations of raw input strings. These methods work quite well and have been used not only for tokenization but also for machine translation (Nguyen et al., 2010), speech recognition (Lee and Glass, 2012; Heymann et al., 2014), and even robotics (Nakamura et al., 2014).

However, from a practical point of view, such purely unsupervised approaches do not suffice. Since they only aim to maximize the probability of the language model on the observed set of strings, they sometimes yield word segmentations that are

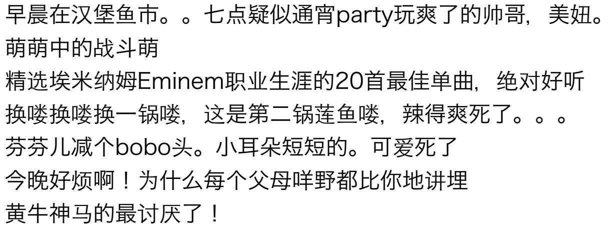

早晨在汉堡鱼市。。七点疑似通宵party玩爽了的帅哥，美妞。
萌萌中的战斗萌
精选埃米纳姆Eminem职业生涯的20首最佳单曲，绝对好听
换喽换喽换一锅喽，这是第二锅莲鱼喽，辣得爽死了。。。
芬芬儿减个bobo头。小耳朵短短的。可爱死了
今晚好烦啊！为什么每个父母咩野都比你地讲埋
黄牛神马的最讨厌了！

Figure 1: Excerpt of Weibo tweets. It contains many "unknown" words such as novel proper nouns, terms from local dialects, etc., that cannot be covered by ordinary labeled data or dictionaries.

Transactions of the Association for Computational Linguistics, vol. 5, pp. 179–189, 2017. Action Editor: Masaaki Nagata.
Submission batch: 10/2016; Revision batch: 12/2016; Published 6/2017.

different from human standards on low frequency words.

To solve this problem, this paper describes a novel combination of a nonparametric Bayesian generative model (NPYLM; Mochihashi et al. (2009)) and a discriminative classifier (CRF; Lafferty et al. (2001)). This combination is based on a semi-supervised framework called JESS-CM (Suzuki and Isozaki, 2008), and it requires a nontrivial exchange of information between these two models. In this approach, the generative and discriminative models will "teach each other" and yield a novel log-linear model for word segmentation.

Experiments on standard datasets of Chinese, Japanese, and Thai indicate that this hybrid model achieves nearly state-of-the-art accuracy on standard corpora, and, thanks to our nonparametric Bayesian model of infinite vocabulary it can accurately segment non-standard texts like those in Twitter and Weibo (the Chinese equivalent of Twitter) without any human intervention.

This paper is organized as follows. Section 2 introduces NPYLM which will be leveraged in the framework of JESS-CM, described in Section 3. Section 4 introduces our model, NPYCRF, and the necessary exchange of information, while Section 5 is devoted to experiments on datasets in Chinese, Japanese, and Thai. We analyze the results and discuss future directions of research on semi-supervised learning in Section 6 and conclude in Section 7.

2 Unsupervised Word Segmentation

To acquire new words from an observation consisting of raw strings, a generative model of words can be extremely useful for word segmentation. Goldwater et al. (2006) showed that a bigram hierarchical Dirichlet process (HDP) model based on Gibbs sampling can effectively find "words" in small corpora. In extending this work, Mochihashi et al. (2009) proposed a nested Pitman-Yor language model (NPYLM), a hierarchical Bayesian language model, where character n-grams (actually, ∞-grams (Mochihashi and Sumita, 2008)) are embedded in word n-grams, and an efficient dynamic programming algorithm for inference exists. Conceptually, NPYLM posits that an infinite number of spellings,

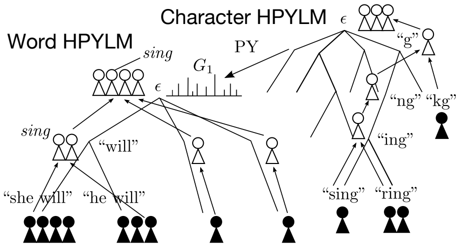


Figure 2: The structure of NPYLM by a Chinese Restaurant Process representation (replicated from Mochihashi et al. (2009)). The word and character HPYLM are drawn as suffix trees; the character HPYLM is a base measure for the word HPYLM, and the two are learned as a single model. Each black customer is a count in HPYLM, and a white customer is a latent proxy customer initiated from each black customer: see Teh (2006) for details.

i.e., "words", are probabilistically generated from character n-grams, and a word unigram is drawn using the character n-grams as the base measure. Then bigram and trigram distributions are hierarchically generated and the final string is yielded from the "word" n-grams, as shown in Figure 2.

Practically, NPYLM can be considered as a hierarchical smoothing of the Bayesian n-gram language model, HPYLM (Teh, 2006). In HPYLM, the predictive distribution of a word $w = w_t$ given a history $h = w_{t-(n-1)} \cdots w_{t-1}$ is expressed as

$$p(w|h) = \frac{c(w|h) - d \cdot t_{hw}}{\theta + c(h)} + \frac{\theta + d \cdot t_{h\cdot}}{\theta + c(h)} \cdot p(w|h') \quad (1)$$

where $c(w|h)$ denotes the observed counts, θ and d are model parameters, and t_{hw} and $t_{h\cdot} = \sum_w t_{hw}$ are latent variables estimated in the model.

The probability of w given h is recursively interpolated using a shorter history $h' = w_{t-(n-2)} \cdots w_{t-1}$. If h is already empty at the unigram level, NPYLM employs a back-off distribution using character n-grams for $p(w|h')$:

$$\begin{aligned} p_0(w) &= p(c_1 \cdots c_k) & (2) \\ &= \textstyle\prod_{i=1}^{k} p(c_i | c_1 \cdots c_{i-1}) \,. & (3) \end{aligned}$$

In this way, NPYLM can assign appropriate probabilities to every possible sequence of segmentation and learn the word and character n-grams at the same time by using a single generative model (Mochihashi et al., 2009).

Semi-Markov view of NPYLM NPYLM formulates unsupervised word segmentation as learning with a semi-Markov model (Figure 3). Here, each node corresponds to an inside probability $\alpha[t][k]$[1] that equals the probability of a substring $c_1^t = c_1 \cdots c_t$ with the last k characters c_{t-k+1}^t being a word. This inside probability can be computed recursively as follows:

$$\alpha[t][k] = \sum_{j=1}^{L} p(c_{t-k+1}^{t} | c_{t-k-j+1}^{t-k}) \cdot \alpha[t-k][j] \quad (4)$$

Here, $1 \leq L \leq t-k$ is the maximum allowed length of a word. With these inside probabilities, we can make use of Markov Chain Monte Carlo (MCMC) method with an efficient forward filtering-backward sampling algorithm (Scott, 2002), namely a "stochastic Viterbi" algorithm to iteratively sample "words" from raw strings in a completely unsupervised fashion, while avoiding local minima.

Problems and Beyond Unsupervised word segmentation with NPYLM works surprisingly well for many languages (Mochihashi et al., 2009); however, it has certain issues. First, since it optimizes the performance of the language model, its segmentation does not always conform to human standards and depends on subtle modeling decisions. For example, NPYLM often separates inflectional suffixes in Japanese like "る" in "見–る" from the rest of the verb, when it is actually a part of the verb itself. Second, it can produce deficient segmentations for low-frequency words and the beginning or ending of a string where the available information comes from only one direction. These issues can be alleviated by using naïve semi-supervised learning method (Mochihashi et al., 2009) that simply adds n-gram counts from supervised segmentations in advance. However, this solution is not perfect because these supervised counts will eventually be overwhelmed by the unsupervised counts, because the overall objective function remains unsupervised.

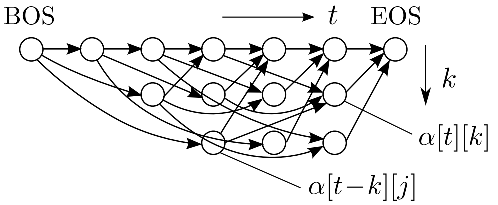


Figure 3: Semi-Markov model representation of NPYLM (simplest case of segment length ≤ 3). Each node corresponds to a substring ending at time t, and its length k is indexed by each row.

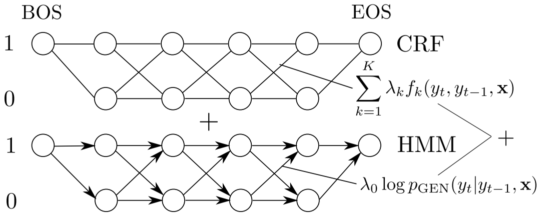

BOS
EOS
1
0
CRF
$\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x})$
+
1
0
HMM
+
$\lambda_0 \log p_{\text{GEN}}(y_t|y_{t-1}, \mathbf{x})$

Figure 4: Semi-supervised learning of the same model structure (HMM and CRF) with JESS-CM. Discriminative and generative potentials are given relative weights $1 : \lambda_0$, and added together in the log probability domain.

To resolve this issue, we must resort to an explicit semi-supervised learning framework that combines both discriminative and generative models. We used JESS-CM (Suzuki and Isozaki, 2008), currently the best such framework for this purpose, which we will briefly introduce below.

3 Integration with a Discriminative Model

JESS-CM (Joint probability model Embedding style Semi-Supervised Conditional Model) is a semi-supervised learning framework that outperforms other generative and log-linear models (Druck and McCallum, 2010). In JESS-CM, the probability of a label sequence $\mathbf{y}$ given an input sequence $\mathbf{x}$ is written as follows:

$$p(\mathbf{y}|\mathbf{x}) \propto p_{\text{DISC}}(\mathbf{y}|\mathbf{x};\Lambda)\, p_{\text{GEN}}(\mathbf{y},\mathbf{x};\Theta)^{\lambda_0} \quad (5)$$

where p_{DISC} and p_{GEN} are respectively the discriminative and generative models, and Λ and Θ are their corresponding parameters. Equation (5) is the product of the experts, where each expert works as a "constraint" to the other with a relative geometrical interpolation weight $1 : \lambda_0$. If we take p_{DISC} to be a log-linear model like CRF (Lafferty et al., 2001):

$$p_{\text{DISC}}(\mathbf{y}|\mathbf{x}) \propto \exp\left(\textstyle\sum_{k=1}^{K} \lambda_k f_k(\mathbf{y},\mathbf{x})\right) , \quad (6)$$

Equation (5) can be also expressed as a log-linear model with a new "feature function" $\log p_{\text{GEN}}(\mathbf{y},\mathbf{x})$:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &\propto \exp\left(\lambda_0 \log p_{\text{GEN}}(\mathbf{y},\mathbf{x}) + \textstyle\sum_{k=1}^{K} \lambda_k f_k(\mathbf{y},\mathbf{x})\right) \\ &= \exp\left(\Lambda \cdot F(\mathbf{y},\mathbf{x})\right) . \quad (7) \end{aligned}$$

[1] While we consider only bigrams in this paper for simplicity, the theory can be naturally extended to higher-order n-grams. However, it requires quite a complicated implementation, and the expected gain in performance will not be large, even if we use trigrams (Mochihashi et al., 2009).

Here, the parameter $\Lambda = (\lambda_0, \lambda_1, \cdots, \lambda_K)$ includes the interpolation weight λ_0 and $F(\mathbf{y}, \mathbf{x}) = (\log p_{\mathrm{GEN}}(\mathbf{y}, \mathbf{x}), f_1(\mathbf{y}, \mathbf{x}), \cdots, f_K(\mathbf{y}, \mathbf{x}))$.

JESS-CM interleaves the optimization of Λ and Θ to maximize the objective function

$$p(Y_l, X_u | X_l; \Lambda, \Theta) = p(Y_l | X_l; \Lambda) \cdot p(X_u; \Theta) \quad (8)$$

where $\langle X_l, Y_l \rangle$ is the labeled dataset and X_u is the unlabeled dataset.

Suzuki and Isozaki (2008) conducted semi-supervised learning on a combination of a CRF and an HMM, as shown in Figure 4. Since CRF and HMM have the same Markov model structure, they interpolate two weights

$$\textstyle\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}) \quad \text{and} \quad (9)$$

$$\lambda_0 \log p_{\mathrm{GEN}}(y_t | y_{t-1}, \mathbf{x}) \quad (10)$$

on the corresponding path, altenately

- fixing Θ and optimizing Λ of CRF on $\langle X_l, Y_l \rangle$, and
- fixing Λ and optimizing Θ of HMM on X_u

until convergence, and thereby iteratively maximizing the two terms in (8).

Through this optimization, p_{DISC} and p_{GEN} will "teach each other" to make the feature $\log p_{\mathrm{GEN}}$ more accurate, and further rectified by p_{DISC} with respect to the labeled data. Note that the interpolation weight λ_0 is automatically computed through this process.

4 Connecting Two Worlds: NPYCRF

We wish to integrate NPYLM and CRF, applying semi-supervised learning via JESS-CM. Note that Suzuki and Isozaki (2008) implicitly assumed that the discriminative and generative models have the same structure as shown in Figure 4. Since NPYLM is a semi-Markov model as described in Section 2, a naïve approach would be to combine it with a semi-Markov CRF (Sarawagi and Cohen, 2005) as the discriminative model.

However, this strategy does not work well for two reasons: First, since a semi-Markov CRF is a model for transitions between segments, it cannot deal with character-level transitions and thus performs suboptimally on its own. In fact, our preliminary supervised word segmentation experiments showed a F_1 measure of around 95%, whereas a character-wise Markov CRF achieves >99%. Second, the semi-Markov CRF was originally designed to chunk at most a few words (Sarawagi and Cohen, 2005). However, in word segmentation of Japanese, for example, we often encounter long proper nouns or Katakana sequences that are more than ten characters, requiring a huge amount of memory even for a small dataset.

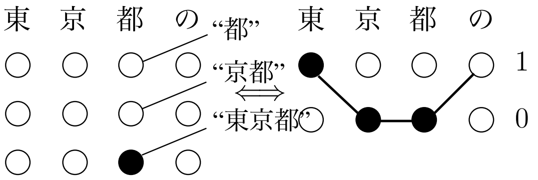


Figure 5: Equivalence of semi-Markov (left) and Markov (right) potentials. The potential of substring "東京都" (Tokyo prefecture) being a word on the left is equivalent to the sum of potentials along the U-shaped path on the right.

In this paper we instead transparently exchange information between the Markov model (CRF) on characters and the semi-Markov model (NPYLM) on words to perform a semi-supervised learning on different model structures. Called NPYCRF, this unified statistical model makes good use of the discriminative model (CRF) from the labeled data and the generative model (NPYLM) from the unlabeled data.

4.1 CRF→NPYLM

To convert from a CRF to NPYLM, we can easily translate Markov potentials into semi-Markov potentials as shown in Andrew (2006) for the supervised learning case.

Consider the situation depicted in Figure 5. Here we can see that the potential of the substring "東京都" (Tokyo prefecture) in the semi-Markov model (left) corresponds to the sum of the potentials in the Markov model (right) along the path shown in bold. Here, we introduce binary hidden states in the Markov model for each character, similarly to the BI tags used in supervised learning, where state 1 represents the beginning of a word and state 0 represents a continuation of the word.

Mathematically, we define $\gamma[a, b)$ as the sum of the potentials along a U-shaped path over an interval $[a, b)$ $(a < b)$ as shown in Figure 5, which begins with state 1 and ends with (but does not include) 1.

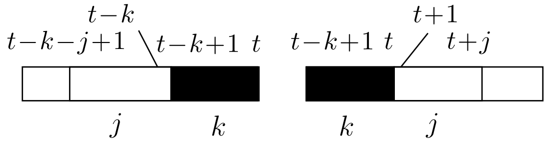


Figure 6: Substring transitions for marginalization.

Using this notation, the potential that corresponds to $\alpha[t][k]$ is $\gamma[t-k+1,t+1)$ covering $c_{t-k+1}\cdots c_t$, and thus the forward recursion of the inside probability $\alpha[t][k]$ that incorporates the information from the CRF can be written as follows, instead of (4):

$$\alpha[t][k] = \sum_{j=1}^{L} \exp\Big[\lambda_0 \log p(c_{t-k+1}^{t} | c_{t-k-j+1}^{t-k}) + \gamma[t-k+1,t+1)\Big] \cdot \alpha[t-k][j]. \quad (11)$$

Backward sampling can be performed in a similar fashion. In this way, we can incorporate information from the character-wise discriminative model (CRF) into the language model segmentation of NPYLM.

4.2 NPYLM→CRF

On the other hand, translating the information from the semi-Markov to Markov model, i.e., translating a potential from the word-based language model into the character-wise discriminative classifier, is not trivial. However, as we describe below, it is actually possible to do so by extending the technique proposed in Andrew (2006).

Note that for the inference of CRF, from the standard theory of log-linear models we only have to compute its gradient with respect to the expectation of each feature in the current model. This reduces the problem to a computation of the marginal probability of each path, which can be derived within the framework of semi-Markov models as follows:

Semi-Markov feature λ_0**.** Following the line of argument presented in the Section 4.1, the potential with respect to the semi-Markov feature weight λ_0 that is associated with the word transition $c_{t-k-j+1}^{t-k} \rightarrow c_{t-k+1}^{t}$, shown in Figure 6, can be expressed as an expectation using the standard forward-backward formula:

$$p(c_{t-k+1}^{t}, c_{t-k-j+1}^{t-k}|\mathbf{s}) = \alpha[t-k][j]\,\beta[t][k] \cdot \exp\Big[\lambda_0 \log p(c_{t-k+1}^{t}|c_{t-k-j+1}^{t-k}) + \gamma[t-k+1,t+1)\Big] / Z(\mathbf{s}) \quad (12)$$

Here, $Z(\mathbf{s})$ is a normalizing constant associated with each input string $\mathbf{s}$, and $\beta[t][k]$ is a backward probability similar to (11) computed by

$$\beta[t][k] = \sum_{j=1}^{L} \exp\Big[\lambda_0 \log p(c_{t+1}^{t+j}|c_{t-k+1}^{t}) \; \gamma[t+1,t+j+2)\Big] \cdot \beta[t+j][j]\,. \quad (13)$$

Markov features $\lambda_1,\cdots,\lambda_K$**.** Note that the features associated with label bigrams in our binary CRF can be divided into four types: 1-1,1-0,0-1, and 0-0, as shown in Figure 7.

Case 1-1: As shown in Figure 8(a), this case means that a word of length 1 begins at time t, which is equivalent to the probability of substring c_t^t being a word:

$$p(z_t\!=\!1, z_{t+1}\!=\!1|\mathbf{s}) = p(c_t^t|\mathbf{s}). \quad (14)$$

Here, $p(c_\ell^k|\mathbf{s})$ is the marginal probability of a substring $c_\ell\cdots c_k$ being a word, which can be derived from equation (12):

$$\begin{aligned} p(c_\ell^k|\mathbf{s}) &= \sum_j p(c_\ell^k, c_{\ell-j}^{\ell-1}|\mathbf{s}) \\ &= \sum_j \alpha[\ell-1][j] \cdot \beta[k][k-\ell+1] \cdot \\ &\quad \exp\Big[\lambda_0 \log p(c_\ell^k|c_{\ell-j}^{\ell-1}) + \gamma[\ell,k+1)\Big] / Z(\mathbf{s}) \\ &= \frac{\beta[k][k-\ell+1]}{Z(\mathbf{s})} \cdot \sum_j \exp\big[\lambda_0 \log p(c_\ell^k|c_{\ell-j}^{\ell-1}) \\ &\quad + \gamma[\ell,k+1)\big]\,\alpha[\ell-1][j] \\ &= \frac{\alpha[k][k-\ell+1] \cdot \beta[k][k-\ell+1]}{Z(\mathbf{s})} \end{aligned} \quad (15)$$

Case 1-0: As shown in Figure 8(b), this case means that a word begins at time t and has length at least 2. Since we do not know the endpoint of this word, we can obtain the probability $p(z_t = 1, z_{t+1}\!=\!0)$ by marginalizing over the endpoint j ($\cdots$ means values all 0):

$$\begin{aligned} & p(z_t\!=\!1, z_{t+1}\!=\!0|\mathbf{s}) \\ &= \sum_{j=2} p(z_t\!=\!1, z_{t+1}\!=\!0, \cdots, z_{t+j}\!=\!1|\mathbf{s}) \\ &= \sum_{j=2} p(c_t^{t+j-1}|\mathbf{s}) \end{aligned} \quad (16)$$

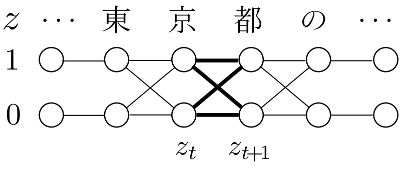
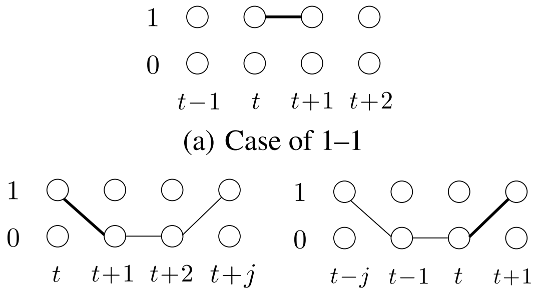


Figure 7: Four types of label transitions in Markov CRF.



(a) Case of 1–1

(b) Case of 1–0

(c) Case of 0–1

Figure 8: Label bigram potentials for marginalization. The probability of each label bigram (bold) of the Markov model can be obtained by marginalizing the probability of the U-shaped path including it, which is computed in the semi-Markov model.

where $p(c_t^{t+j-1}|\mathbf{s})$ is obtained from (15).

Case 0-1: Similarly, as shown in Figure 8(c) this case means that a word of length at least 2 begins before time t and ends at time t. Therefore, we can marginalize over the start point of a possible word to obtain the marginal probability:

$$\begin{aligned} & p(z_t\!=\!0, z_{t+1}\!=\!1|\mathbf{s}) \\ & = \sum_{j=1} p(z_{t-j}\!=\!1, \cdots, z_t\!=\!0, z_{t+1}\!=\!1|\mathbf{s}) \quad (17) \\ & = \sum_{j=1} p(c_{t-j}^{t}|\mathbf{s})\,. \quad (18) \end{aligned}$$

Case 0-0: In principle, this means that a word begins before time t and ends later than (and including) time $t+1$. Therefore, we can marginalize over both the start and end time of a possible word spanning $[t, t\!+\!1]$ to obtain:

$$p(z_t\!=\!0, z_{t+1}\!=\!0|\mathbf{s}) = \sum_{j=1}\sum_{k=1} p(c_{t-j}^{t+k}|\mathbf{s})\,. \quad (19)$$

However, in fact we can avoid this nested computation because the probability of $p(z_t, z_{t+1})$ over the possible values of z_t and z_{t+1} must sum to 1. We can therefore simply calculate it as follows (Andrew, 2006):

$$p(z_t\!=\!0, z_{t+1}\!=\!0|\mathbf{s}) = 1-p(1,1)-p(1,0)-p(0,1) \quad (20)$$

where $p(x,y)$ means $p(z_t\!=\!x, z_{t+1}\!=\!y|\mathbf{s})$.

4.3 Inference

Finally, we obtain the inference algorithm for NPYCRF as a variant of the MCMC-EM algorithm (Wei and Tanner, 1990) shown in Figure 9.[2] In learning of a NPYLM, we add the CRF potentials as described in Section 4.1, and sample a possible segmentation from the posterior through Forward filtering-Backward sampling to update the model parameters. On the basis of this improved language model, the CRF weights are then optimized by incorporating language model features as explained in Section 4.2. We iterate this process until convergence.

Note that we first have to learn an unsupervised segmentation in Step 2 before training the CRF. Since our inference algorithm includes an optimization of CRF and thus is not a true MCMC, the learning of word segmentation *after* the supervised information will be severely constrained and likely to get stuck in local minima.

In practice, we found that the EM-style batch learning of CRF described above often fails because our objective function is non-convex. Therefore, we switched to ADF below (Sun et al., 2014), an adaptive stochastic gradient descent that yields state-of-the-art accuracies for natural language processing problems including word segmentation. In this case, Λ in Figure 9 was optimized with each minibatch through the labeled data $\langle X_l, Y_l\rangle$, while incorporating information from the unlabeled data X_u by the language model.

```
1:  Add $\langle Y_l, X_l\rangle$ to NPYLM.
2:  Optimize $\Lambda$ on $\langle Y_l, X_l\rangle$. (pure CRF)
3:  for $j = 1 \cdots M$ do
4:      for $i = \mathrm{randperm}(1 \cdots N)$ do
5:          if $j > 1$ then
6:              Remove customers of $X_u^{(i)}$ from NPYLM $\Theta$
7:          end if
8:          Draw segmentations of $X_u^{(i)}$ from NPYCRF
9:          Add customers of $X_u^{(i)}$ to NPYLM $\Theta$
10:     end for
11:     Optimize $\Lambda$ of NPYCRF on $\langle Y_l, X_l\rangle$.
12: end for
```

Figure 9: Basic learning algorithm for NPYCRF. $X_u^{(i)}$ denotes the i-th sentence in the unlabeled data X_u. We can also iterate steps 4 to 10 several times until Θ approximately converges, before updating Λ.

Because of its heavy computational demands,

[2] It is possible to fix NPYLM and just use this as a feature to CRF: this amounts to running only the first iteration ($j = 1$) of the EM algorithm. However, it still requires NPYLM→CRF conversion in Section 4.2, and we found that the performance is not optimal while slightly better than plain CRF.

Language	Dataset	Labeled	Unlabeled	Test
Chinese	MSR	86,924	865,679	3,985
	Weibo	10K-40K	880,920[3]	30,000
Japanese	Twitter	59,931	600,000	444
Thai	InterBEST	10,000	30,133	10,000

Table 1: Statistics of the datasets for the experiments.

we parallelized the NPYLM sampling over several processors and because of the possible correlation of segmentations within the samples, used the Metropolis-Hastings algorithm to correct them. The acceptance rate in our experiments was over 99%.

For decoding, we can simply find a Viterbi path in the integrated semi-Markov model while fixing all the sampled segmentations on the unlabeled data.

5 Experiments

We conducted experiments on several corpora of unsegmented languages: Japanese, Chinese, and Thai. The corpora included standard corpora as well as text from Twitter and its equivalent, Weibo, in Chinese.

5.1 Data

Chinese For Chinese, we first used a standard dataset from the SIGHAN Bakeoff 2005 (Emerson, 2005) for the labeled and test data, and Chinese gigaword version 2 (LDC2009T14) for the unlabeled data. We chose the MSR subset of SIGHAN Bakeoff written in simplified Chinese together with the provided training and test splits, which contain about 87K/40K sentences, respectively. For the unlabeled data, i.e., a collection of raw strings, we used a random subset of 880K sentences from Chinese gigaword with all spaces removed. We chose this size to be about 10 times larger than the labeled data, considering current computational requirements. We used the part from the Xinhua news agency 2004 and split the data into sentences at the end-of-sentence character “。”.

Because the MSR and Xinhua datasets were compiled from newspapers, to meet our objective on informal text we conducted further experiments using the Leiden Weibo corpus[4] from Weibo, a Twitter equivalent in China. From this dataset, we used the sentences that have exact correspondence between the provided segmented-unsegmented pair, yielding about 880K sentences. Since we did not know how much supervision would be necessary for a decent performance, we conducted experiments with different amounts of labeled data: 10K, 20K, 40K and 880K(all). Note that the final case amounts to complete supervision, an ideal situation that is not likely in practice.

Name	Labels	F	Filtered
Sun+ (2009)	2	N/A	0.973
Sun+ (2014)	3	N/A	0.975
Chen+ (2015)	4 (Neural)	0.976	–
Zhang+ (2016)	2 (Neural)	**0.977**	–
NPYCRF	2	0.970	0.973
NPYCRF	3	0.973	**0.976**

Table 2: Accuracies of Bakeoff MSR dataset in Chinese. “Filtered” are the results with a simple post-hoc filter described in Sun et al. (2009).

Data	Label	Unlabel	IV	OOV	F
Topline	880K	–	*0.981*	*0.699*	*0.977*
Sup 10K	10K	–	0.949	0.690	0.928
Sup 20K	20K	–	0.957	0.683	0.941
Sup 40K	40K	–	0.963	0.682	0.951
Semi 10K	10K	870K	0.954	0.698	0.933
Semi 20K	20K	860K	0.961	0.690	0.945
Semi 40K	40K	840K	0.970	0.648	**0.955**

Table 3: Accuracies on Leiden Weibo corpus in Chinese. ‘Label’ and ‘Unlabel’ are the amounts of labeled and unlabeled data, respectively. “Topline” is an ideal situation of complete supervision, and $K=10^3$ sentences.

Japanese Word segmentation accuracies around 99% have already been reported for newspaper domains in Japanese (Kudo et al., 2004). Therefore, we only conducted experiments on segmenting Twitter text. In addition to our random Twitter crawl in April 2014, we used a corpus of Japanese Twitter text compiled by the Tokyo Metropolitan University[5]. This corpus is actually very small, 944 sentences. It mainly targets transfer learning and is segmented according to BCCWJ (Basic Corpus of Contemporary

[3] This is the total number of sentences in the experiment: the actual number of unsupervised sentences is this set minus the different number of supervised sentences.

[4] http://www.leidenweibocorpus.nl/openaccess.php

[5] https://github.com/tmu-nlp/TwitterCorpus

Written Japanese) standards from the National Institute of Japanese Language (Maekawa, 2007). Therefore, for the labeled data we used the "core" subset of BCCWJ consisting of about 59K sentences plus 500 random sentences from the Twitter dataset. We used the remaining 444 sentences for testing. For the unlabeled data, we used a random crawl of 600K Japanese sentences collected from Twitter in March-April, 2014.

Thai Unsegmented languages, such as Thai, Lao, Myanmar, and Kumer, are also prevalent in South East Asia and are becoming increasingly important targets of natural language processing. Thus we also conducted an experiment on Thai, using the standard InterBEST 2009 dataset (Kosawat, 2009). Since it is reported that the "novel" subset of InterBEST has relatively low precision, we used this part with a random split of 10K sentences for supervised learning, 30K sentences for unsupervised learning, and a further 10K sentences for testing.

5.2 Training Settings

Because Sun et al. (2012) report increased accuracy with three tags, $\{B,I,E\}$[6], we also tried these tags in place of the binary tags described in Section 4.2. This modification resulted in 6 possible transitions out of $3^2 = 9$ transitions, whose computation follows from the binary case in Section 4.2. We used normal priors of truncated $N(1, \sigma^2)$ and $N(0, \sigma^2)$ for λ_0 and $\lambda_1 \cdots \lambda_K$, respectively, and fixed the CRF regularization parameter C to 1.0, and σ to 1.0 by preliminary experiments on the same data.

For the feature templates, we followed Sun et al. (2012). In addition to those templates, we used character type bigrams, where the 'character type' was defined by Unicode blocks (like Hiragana or CJK Unified Ideographs for Chinese and Japanese) or Unicode character categories (Thai).

To reduce computations by restricting the search space appropriately, we employed a Negative Binomial generalized linear model on string features (Uchiumi et al., 2015) to predict the maximum length of a possible word for each character position in the training data. Therefore, the upper limit of L in (11) and (13) was L_t for each position t, obtained from this statistical model trained on labeled segmentations. We observed that this prediction made the computation several times faster than, for example, using a fixed threshold in Japanese where quite long words are occasionally encountered.

东软集团	19
游景玉	17
任尧森	17
南昆铁路	16
东方红三号"卫星	13
刘积仁	13
ｉｎｔｅｒｎｅｔ	11
东宝	11
张肇群	10
彭云	10
玲英	10
抚州	10
亚仿	10
南丁格尔	9
中远香港集团	7
海尔-波普彗星	7
第九届全国人民代表大会	6
巨型机	6

(a) MSR (Simplified Chinese)

にゃん	6
セフレ	6
フォロワー	5
https	4
December	4
トナカイ	3
アオサギ	3
フォロバ	3
じゅん	3
環奈	3
リプ	3
トッキュウジャー	2
リフォロー	2
酔っ払い	2
ツイート	2
クシャミ	2
エタフォ	2
まじかよ	2
ググれ	2
ふりふり	2

(b) Twitter (Japanese)

Figure 10: New words acquired by NPYCRF. For each figure, the left column is the words that did not appear in the provided labeled data, and the right column is the frequencies NPYCRF recognized in the test data. In Chinese, we found many proper names including company and person name, and in Japanese, we found many novel slang words and proper names.

5.3 Experimental results

Chinese Tables 2 and 3 show IV (in-vocabulary) and OOV (out-of-vocabulary) precision and F-measure, computed against segmented tokens. The results for standard newspaper text indicate that NPYCRF is basically comparable in performance to state-of-the-art supervised neural networks (Chen et al., 2015; Zhang et al., 2016) that require hand tuning of hyperparameters or model architectures. Figure 10 shows some of the learned words in the test-set of the Bakeoff MSR corpus. As shown in Table 3, NPYCRF also yields higher precision than supervised learning on non-standard text like Weibo, which is the main objective for this study. Contrary to ordinary supervised learning, we can see that NPYCRF effectively learns many "new words" from the large amount of unlabeled data thanks to the generative model, while observing human standards of segmentation by the discriminative model. Note that in Weibo segmentation, complete supervision is not

[6] The B, I, and E tags mean the beginning, internal part, and end of a word, respectively.

CRF	NPYLM	NPYCRF	Gold
有些	有些	有些	有些
大学生	大学生	大学生	大学生
眼	眼高手低	眼	眼高手低
高手		高手	
低		低	
，	，	，	，
不屑	不屑于	不屑于	不屑于
于			
做	做	做	做
小	小	小	小
事情	事情	事情	事情
。	。	。	。
王思斌	王	王思斌	王思斌
	思		
	斌		
，	，	，	，
男	男	男	男
，	，	，	，
１９４９年１０月	１９４９年	１９４９年１０月	１９４９年１０月
	１０月		
生	生	生	生
。	。	。	。

Figure 11: Example of segmentation of the SIGHAN Bakeoff MSR dataset made with supervised (CRF), unsupervised (NPYLM), and semi-supervised (NPYCRF) models in comparison with gold segmentations (Gold). "眼高手低" is a proverb and "王思斌" is a full name of a person.

available in practice. In fact, we realized that the Weibo segmentations were given automatically by an existing classifier, and contain many inappropriate segmentations, while NPYCRF finds much "better" segmentations.

Figure 11 compares the results of CRF, NPYLM, and NPYCRF with the gold segmentation. While proverbs like "眼高手低" (wide vision without action) are correctly captured from the unlabeled data by NPYLM, it is sometimes broken by CRF through integration. In another case, the name of a person is properly connected because of the information provided by the CRF. This comparison shows that there is still room for improvement in NPYCRF. Section 6 discusses future research directions for improvements.

Japanese and Thai Figure 12 shows an example of the analysis of Japanese Twitter text. Shaded words are those that are not contained in labeled data (BCCWJ core) but were found by NPYCRF. Many segmentations, including new words, are correct. We expect NPYCRF would perform better with more unlabeled data that are easily obtained.

Tables 4 and 5 show the segmentation accuracies of the Twitter data in Japanese and novel data in Thai. While there are no publicly available results for these data (the InterBEST testset is closed during competition), NPYCRF achieved better accuracies than vanilla supervised segmentation based on CRF. Considering that many new words were found in Figure 12, for example, we believe NPYCRF is quite competitive thanks to its ability to learn the infinite vocabulary, which it inherits from NPYLM.

いや 他 でも 普通 に する よ シシルス サーチ 用 に
ボーナス 持ちピン と か も ある し
誰 だ スズカ エルフォトン に ぶっこん だ の…
電車 で 座っ て ん だ けど 、 目 の 前 が 酔っ払い が
相華 で フラフラ し て て ハラハラ する … 手遅れ に な
ら ない うち に 離脱 する か …
ひこ にゃん 音頭 だ にゃん ♪ ひこ にゃん ひこ
にゃん ひこ にゃん にゃん っ
ほんと です よ ほんと 嬉しい くっそ 倍率 やば そう 、
、 、 ほんと それ ！ ！ みほりん と 参戦 し たい まぢ で
みほりんまま に も 会い たい
初め まし て ♪ 私 は 絢瀬 絵里 役 の 鈴々 蝶 です ♪ 似
て ない です が 、 応援 し て くれる と 嬉しい です ♪
ちょくちょく 絡みだす なら な w w w これ から も ちょ
くちょく 絡む から よろしく (´ω`)

Figure 12: Samples of NPYCRF segmentation of Twitter text in Japanese that are difficult to analyze by ordinary supervised segmentation. It contains a lot of novel words, emoticons, and colloquial expressions that are not contained in the BCCWJ core text (shaded).

6 Analysis

As shown in Figure 11, NPYCRF makes good use of NPYLM but sometimes ignores its prediction by falling back to CRF, yielding suboptimal performance. This is mainly because the geometric interpolation weight λ_0 is always constant and does not vary according to the input. For example, even if the substring to segment is very rare in the labeled data, NPYCRF trusts the supervised classifier (CRF) with a constant rate of $1/(1+\lambda_0)$ in the log probability domain. To alleviate this problem,

Model	IV	OOV	F
CRF	0.939	0.706	0.916
NPYCRF	0.947	0.708	**0.921**

Table 4: Accuracies for Twitter text in Japanese.

Model	IV	OOV	F
CRF	0.961	0.409	0.948
NPYCRF	0.959	0.362	**0.954**

Table 5: Accuracies for InterBEST novel dataset in Thai.

it is necessary to change λ_0 depending on the input string in a log-linear framework.[7] While this might be achieved through Density Ratio estimation framework (Sugiyama et al., 2012; Tsuboi et al., 2009), we believe it is a general problem of semi-supervised learning and is beyond the scope of this paper.

This issue also affects the estimation of λ_0 as a scalar: that is, we found that λ_0 often fluctuates during training because Λ (which includes λ_0) is estimated using only limited $\langle X_l, Y_l \rangle$. In practice, we terminated the EM algorithm in Figure 9 early after a few iterations. Therefore, with a more adaptive semi-supervised learning framework, we expect that NPYCRF will achieve higher accuracy than the current performance.

7 Conclusion

In this paper, we presented a hybrid generative/discriminative model of word segmentation, leveraging a nonparametric Bayesian model for unsupervised segmentation. By combining CRF and NPYLM within the semi-supervised framework of JESS-CM, our NPYCRF not only works as well as the state-of-the-art neural segmentation without hand tuning of hyperparameters on standard corpora, but also appropriately segments non-standard texts found in Twitter and Weibo, for example, by automatically finding "new words" thanks to a nonparametric model of infinite vocabulary.

We believe that our model lays the foundation for developing a methodology of combining nonparametric Bayesian models and discriminative classifiers, as well as providing an example of semi-supervised learning on different model structures, i.e. Markov and semi-Markov models for word segmentation.

Acknowledgments

We are deeply grateful to Jun Suzuki (NTT CS Labs) for important discussions leading to this research, Xu Sun (Peking University) for details of his experiments in Chinese. We would also like to thank anonymous reviewers and the action editor, especially the editors-in-chief for the thorough comments for the final manuscript.

References

Galen Andrew. 2006. A Hybrid Markov/Semi-Markov Conditional Random Field for Sequence Segmentation. In *EMNLP 2006*, pages 465–472.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015. Gated Recursive Neural Network for Chinese Word Segmentation. In *ACL 2015*, pages 1744–1753.

Gregory Druck and Andrew McCallum. 2010. High-Performance Semi-Supervised Learning using Discriminatively Constrained Generative Models. In *ICML 2010*, pages 319–326.

Tom Emerson. 2005. The Second International Chinese Word Segmentation Bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual Dependencies in Unsupervised Word Segmentation. In *Proceedings of ACL/COLING 2006*, pages 673–680.

Joshua T. Goodman. 2001. A Bit of Progress in Language Modeling, Extended Version. Technical Report MSR–TR–2001–72, Microsoft Research.

Jahn Heymann, Oliver Walter, Reinhold Häb-Umbach, and Bhiksha Raj. 2014. Iterative Bayesian Word Segmentation for Unsupervised Vocabulary Discovery from Phoneme Lattices. In *ICASSP 2014*, pages 4057–4061.

Krit Kosawat. 2009. InterBEST 2009: Thai Word Segmentation Workshop. In *Proceedings of 2009 Eighth International Symposium on Natural Language Processing (SNLP2009)*, Thailand.

Canasai Kruengkrai, Kiyotaka Uchimoto, Junichi Kazama, Kentaro Torisawa, Hiroshi Isahara, and Chuleerat Jaruskulchai. 2009. A word and character-cluster hybrid model for Thai word segmentation. In *Eighth International Symposium on Natural Lanugage Processing*.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. In *EMNLP 2004*, pages 230–237.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML 2001*, pages 282–289.

Chia-ying Lee and James Glass. 2012. A Nonparametric Bayesian Approach to Acoustic Model Discovery. In *ACL 2012*, pages 40–49.

[7] This is reminiscent of context-dependent Bayesian smoothing of MacKay (1994) in the probability domain, as opposed to the fixed Jelinek-Mercer smoothing (Goodman, 2001).

David J. C. MacKay and L. Peto. 1994. A Hierarchical Dirichlet Language Model. *Natural Language Engineering*, 1(3):1–19.

Kikuo Maekawa. 2007. Kotonoha and BCCWJ: Development of a Balanced Corpus of Contemporary Written Japanese. In *Corpora and Language Research: Proceedings of the First International Conference on Korean Language, Literature, and Culture*, pages 158–177.

Daichi Mochihashi and Eiichiro Sumita. 2008. The Infinite Markov Model. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pages 1017–1024.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling. In *Proceedings of ACL-IJCNLP 2009*, pages 100–108.

Tomoaki Nakamura, Takayuki Nagai, Kotaro Funakoshi, Shogo Nagasaka, Tadahiro Taniguchi, and Naoto Iwahashi. 2014. Mutual Learning of an Object Concept and Language Model Based on MLDA and NPYLM. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'14)*, pages 600–607.

ThuyLinh Nguyen, Stephan Vogel, and Noah A. Smith. 2010. Nonparametric Word Segmentation for Machine Translation. In *COLING 2010*, pages 815–823.

Sunita Sarawagi and William W. Cohen. 2005. Semi-Markov Conditional Random Fields for Information Extraction. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, pages 1185–1192.

Steven L. Scott. 2002. Bayesian Methods for Hidden Markov Models. *Journal of the American Statistical Association*, 97:337–351.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. 2012. *Density Ratio Estimation in Machine Learning*. Cambridge University Press.

Weiwei Sun and Jia Xu. 2011. Enhancing Chinese Word Segmentation using Unlabeled Data. In *EMNLP 2011*, pages 970–979.

Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2009. A Discriminative Latent Variable Chinese Segmenter with Hybrid Word/Character Information. In *NAACL 2009*, pages 56–64.

Xu Sun, Houfeng Wang, and Wenjie Li. 2012. Fast Online Training with Frequency-Adaptive Learning Rates for Chinese Word Segmentation and New Word Detection. In *ACL 2012*, pages 253–262.

Xu Sun, Wenjie Li, Houfeng Wang, and Qin Lu. 2014. Feature-Frequency-Adaptive Online Training for Fast and Accurate Natural Language Processing. *Computational Linguistics*, 40(3):563–586.

Jun Suzuki and Hideki Isozaki. 2008. Semi-Supervised Sequential Labeling and Segmentation Using Giga-Word Scale Unlabeled Data. In *ACL:HLT 2008*, pages 665–673.

Yee Whye Teh. 2006. A Hierarchical Bayesian Language Model based on Pitman-Yor Processes. In *Proceedings of ACL/COLING 2006*, pages 985–992.

Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel, and Masashi Sugiyama. 2009. Direct Density Ratio Estimation for Large-scale Covariate Shift Adaptation. *Information and Media Technologies*, 4(2):529–546.

Kei Uchiumi, Hiroshi Tsukahara, and Daichi Mochihashi. 2015. Inducing Word and Part-of-speech with Pitman-Yor Hidden Semi-Markov Models. In *ACL-IJCNLP 2015*, pages 1774–1782.

Greg C.G. Wei and Martin A. Tanner. 1990. A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association*, 85(411):699–704.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Transition-Based Neural Word Segmentation. In *ACL 2016*.