Inducing Word and Part-of-Speech with Pitman-Yor Hidden Semi-Markov Models

Kei Uchiumi  Hiroshi Tsukahara
Denso IT Laboratory, Inc.
Shibuya Cross Tower 28F
2-15-1 Shibuya, Tokyo, Japan
{kuchiumi,htsukahara}@d-itlab.co.jp

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

Abstract

We propose a nonparametric Bayesian model for joint unsupervised word segmentation and part-of-speech tagging from raw strings. Extending a previous model for word segmentation, our model is called a Pitman-Yor Hidden Semi-Markov Model (PYHSMM) and considered as a method to build a class $n$-gram language model directly from strings, while integrating character and word level information. Experimental results on standard datasets on Japanese, Chinese and Thai revealed it outperforms previous results to yield the state-of-the-art accuracies. This model will also serve to analyze a structure of a language whose words are not identified a priori.

1 Introduction

Morphological analysis is a staple of natural language processing for broad languages. Especially for some East Asian languages such as Japanese, Chinese or Thai, word boundaries are not explicitly written, thus morphological analysis is a crucial first step for further processing. Note that also in Latin and old English, scripts were originally written with no word indications (scripta continua), but people felt no difficulty reading them. Here, morphological analysis means word segmentation and part-of-speech (POS) tagging.

For this purpose, supervised methods have often been employed for training. However, to train such supervised classifiers, we have to prepare a large amount of training data with correct annotations, in this case, word segmentation and POS tags. Creating and maintaining these data is not only costly but also very difficult, because generally there are no clear criteria for either “correct” segmentation or POS tags. In fact, since there are different standards for Chinese word segmentation, widely used SIGHAN Bakeoff dataset (Emerson, 2005) consists of multiple parts employing different annotation schemes.

Lately, this situation has become increasingly important because there are strong demands for processing huge amounts of text in consumer generated media such as Twitter, Weibo or Facebook (Figure 1). They contain a plethora of colloquial expressions and newly coined words, including sentiment expressions such as emoticons that cannot be covered by fixed supervised data.

To automatically recognize such linguistic phenomena beyond small “correct” supervised data, we have to extract linguistic knowledge from the statistics of strings themselves in an unsupervised fashion. Needless to say, such methods will also contribute to analyzing speech transcripts, classic texts, or even unknown languages. From a scientific point of view, it is worth while to find “words” and their part-of-speech purely from a collection of strings without any preconceived assumptions.

To achieve that goal, there have been two kinds of approaches: heuristic methods and statistical generative models. Heuristic methods are based on basic observations such that word boundaries will often occur at the place where predictive entropy of characters is large (i.e. the next character cannot be predicted without assuming

Figure 1: Sample of Japanese Twitter text that is difficult to analyze by ordinary supervised segmentation. It contains a lot of novel words, emoticons, and colloquial expressions.
the next word). By formulating such ideas as search or MDL problems of given coding length\(^1\), word boundaries are found in an algorithmic fashion (Zhikov et al., 2010; Magistry and Sagot, 2013). However, such methods have difficulty incorporating higher-order statistics beyond simple heuristics, such as word transitions, word spelling formation, or word length distribution. Moreover, they usually depend on tuning parameters like thresholds that cannot be learned without human intervention.

In contrast, statistical models are ready to incorporate all such phenomena within a consistent statistical generative model of a string, and often prove to work better than heuristic methods (Goldwater et al., 2006; Mochihashi et al., 2009). In fact, the statistical methods often include the criteria of heuristic methods at least in a conceptual level, which is noted in (Mochihashi et al., 2009) and also explained later in this paper. In a statistical model, each word segmentation \(w\) of a string \(s\) is regarded as a hidden stochastic variable, and the unsupervised learning of word segmentation is formulated as a maximization of a probability of \(w\) given \(s\):

\[
\text{argmax}_w p(w|s).
\]  
(1)

This means that we want the most “natural” segmentation \(w\) that have a high probability in a language model \(p(w|s)\).

Lately, Chen et al. (2014) proposed an intermediate model between heuristic and statistical models as a product of character and word HMMs. However, these two models do not have information shared between the models, which is not the case with generative models.

So far, these approaches only find word segmentation, leaving part-of-speech information behind. These two problems are not actually independent but interrelated, because knowing the part-of-speech of some infrequent or unknown word will give contextual clues to word segmentation, and vice versa. For example, in Japanese

すもももももも

can be segmented into not only すもも/もも/もも/も (plum/too/peach/too), but also into すもも/もも/もも (plum/peach/peach), which is ungrammatical. However, we could exclude the latter case

\[\text{argmax} \ p(w, z | s) \propto p(w, z, s) \tag{2}\]

From the expression above, this amounts to building a generative model of a string \(s\) with words \(w\) and tags \(z\) along with an associated inference procedure. We solve this problem by extending previous generative model of word segmentation. Note that heuristic methods are never able to model the hidden tags, and only statistical generative models can accommodate this objective.

This paper is organized as follows. In Section 2, we briefly introduce NPYLM (Mochihashi et al., 2009) on which our extension is based. Section 3 extends it to include hidden states to yield a hidden semi-Markov models (Murphy, 2002), and we describe its inference procedure in Section 4. We conduct experiments on some East Asian languages in Section 5. Section 6 discusses implications of our model and related work, and Section 7 concludes the paper.

2 Nested Pitman-Yor Language Model

Our joint model of words and states is an extension of the Nested Pitman-Yor Language Model (Mochihashi et al., 2009) of a string, which in turn is an extension of a Bayesian \(n\)-gram language model called Hierarchical Pitman-Yor Language Model (HPYLM) (Teh, 2006).
HPYLM is a nonparametric Bayesian model of n-gram distribution based on the Pitman-Yor process (Pitman and Yor, 1997) that generates a discrete distribution $G$ as $G \sim \text{PY}(G_0, d, \theta)$. Here, $d$ is a discount factor, “parent” distribution $G_0$ is called a base measure and $\theta$ controls how similar $G$ is to $G_0$ in expectation. In HPYLM, n-gram distribution $G_n = \{p(w|u_{t-1} \cdots u_{t-(n-1)})\}$ is assumed to be generated from the Pitman-Yor process

$$G_n \sim \text{PY}(G_{n-1}, d_n, \theta_n), \quad (3)$$

where the base measure $G_{n-1}$ is an $(n-1)$-gram distribution generated recursively in accordance with (3). Note that there are different $G_n$ for each n-gram history $h = u_{t-1} \cdots u_{t-(n-1)}$. When we reach the unigram $G_1$ and need to use a base measure $G_0$, i.e. prior probabilities of words, HPYLM usually uses a uniform distribution over the lexicon.

However, in the case of unsupervised word segmentation, every sequence of characters could be a word, thus the size of the lexicon is unbounded. Moreover, prior probability of forming a word should not be uniform over all sequences of characters: for example, English words rarely begin with ‘gme’ but tend to end with ‘-ent’ like in segment. To model this property, NPYLM assumes that word prior $G_0$ is generated from character HPYLM to model a well-formedness of $w$. In practice, to avoid dependency on $n$ in the character model, we used an infinite VPYLM (Mochihashi and Sumita, 2008) in this research. Finally, NPYLM gives an n-gram probability of word $w$ given a history $h$ recursively by integrating out $G_n$,

$$p(w|h) = \frac{c(w|h) - d \cdot t_{hw}}{\theta + c(h)} + \frac{\theta \cdot d \cdot t_{h}}{\theta + c(h)} p(w|h'), \quad (4)$$

where $h'$ is the shorter history of $(n-1)$-grams. $c(w|h), c(h) = \sum_w c(w|h)$ are n-gram counts of $w$ appearing after $h$, and $t_{hw}, t_h = \sum_w t_{hw}$ are associated latent variables explained below. In case the history $h$ is already empty at the unigram, $p(w|h') = p_0(w)$ is computed from the character $\infty$-grams for the word $w = c_1 \cdots c_k$:

$$p_0(w) = p(c_1 \cdots c_k) = \prod_{i=1}^{k} p(c_i|c_{i-1} \cdots c_1). \quad (5)$$

In practice, we further corrected (6) so that a word length follows a mixture of Poisson distributions. For details, see (Mochihashi et al., 2009).

When we know word segmentation $w$ of the data, the probability above can be computed by adding each n-gram count of $w$ given $h$ to the model, i.e. increment $c(w|h)$ in accordance with a hierarchical Chinese restaurant process associated with HPYLM (Figure 2). When each n-gram count called a customer is inferred to be actually generated from $(n-1)$-grams, we send its proxy customer for smoothing to the parent restaurant and increment $t_{hw}$, and this process will recurse. Notice that if a word $w$ is never seen in $w$, its proxy customer is eventually sent to the parent restaurant of unigrams. In that case, $w$ is decomposed to its character sequence $c_1 \cdots c_k$ and this is added to the character HPYLM in the same way, making it a little “clever” about possible word spellings.

**Inference** Because we do not know word segmentation $w$ beforehand, we begin with a trivial segmentation in which every sentence is a single word. Then, we iteratively refine it by sampling a new word segmentation $w(s)$ of a sentence $s$ in a Markov Chain Monte Carlo (MCMC) framework using a dynamic programming, as is done with PCFG by (Johnson et al., 2007) shown in Figure 3 where we omit MH steps for computational reasons. Further note that every hyperparameter $d_n, \theta_n$ of NPYLM can be sampled from the posterior in a Bayesian fashion, as opposed to heuristic methods that rely on a development set for tuning. For details, see Teh (2006).

3 Pitman-Yor Hidden Semi-Markov Models

NPYLM is a complete generative model of a string, that is, a hierarchical Bayesian n-gram language model. For details, see Teh (2006).

**Input:** a collection of strings $S$

Add initial segmentation $w(s)$ to $\Theta$

for $j = 1 \cdots J$ do

for $s$ in randperm $(S)$ do

Remove customers of $w(s)$ from $\Theta$

Sample $w(s)$ according to $p(w|s, \Theta)$

Add customers of $w(s)$ to $\Theta$

end for

Sample hyperparameters of $\Theta$.

end for

Figure 3: MCMC inference of NPYLM $\Theta$.

---

2To be precise, this occurs whenever $t_{hw}$ is incremented in the unigram restaurant.

3Note that a child first memorizes what his mother says as a single word and gradually learns the lexicon.
The word boundaries are distinguished states for beginning and end of a sentence, respectively. For the transition probability of hidden states, we put a HPY process prior as (Blunsom and Cohn, 2011):

$$p(z_t|z_{t-1}) \sim \text{HPY}(d, \theta)$$

(7)

with the final base measure being a uniform distribution over the states. The word boundaries are known in (Blunsom and Cohn, 2011), but in our case it is also learned from data at the same time. Note that because $$w_t$$ depends on already generated words $$w_1 \cdots w_{t-1}$$, our model is considered as an autoregressive HMM rather than a vanilla HMM, as shown in Figure 4 ($$w_{t-1} \rightarrow w_t$$ dependency).

Since segment models like NPYLM have segment lengths as hidden states, they are called semi-Markov models (Murphy, 2002). In contrast, our model also has hidden part-of-speech, thus we call it a Pitman-Yor Hidden Semi-Markov model (PYHSMM). Notice that this is considered as a generative counterpart of a discriminative model known as a hidden semi-Markov CRF (Sarawagi and Cohen, 2005).

4 Inference

Inference of PYHSMM proceeds in almost the same way as NPYLM in Figure 3: For each sentence, first remove the customers associated with the old segmentation similarly to adding them. After sampling a new segmentation and states, the model is updated by adding new customers in accordance with the new segmentation and hidden states.

4.1 Sampling words and states

To sample words and states (part-of-speech) jointly, we first compute inside probabilities forward from BOS to EOS and sample backwards from EOS according to the Forward filtering-Backward sampling algorithm (Scott, 2002). This

4 Lately, Johnson et al. (2013) proposed a nonparametric Bayesian hidden semi-Markov models for general state spaces. However, it depends on a separate distribution for a state duration, thus is clearly different from ours for a natural language.
can be regarded as a “stochastic Viterbi” algorithm that has the advantage of not being trapped in local minima, since it is a valid move of a Gibbs sampler in a Bayesian model.

For a word bigram case for simplicity, inside variable \( \alpha[t][k][z] \) is a probability that a substring \( c_1 \cdots c_t \) of a string \( s = c_1 \cdots c_N \) is generated with its last \( k \) characters being a word, generated from state \( z \) as shown in Figure 5. From the definition of PYHSMM, this can be computed recursively as follows:

\[
\alpha[t][k][z] = \sum_{j=1}^{L} \sum_{y=1}^{K} p(c_{t-k}^t | c_{t-k-j+1}^t, z) p(z|y) \alpha[t-k][j][y].
\] (8)

Here, \( c_{t}^{k} \) is a substring \( c_a \cdots c_t \) and \( L (\leq t) \) is the maximum length of a word, and \( K \) is the number of hidden states.\(^5\)

In Figure 5, each cell represents \( \alpha[t][k][z] \) and a single path connecting from EOS to BOS corresponds to a word sequence \( w \) and its state sequence \( z \). Note that each cell is not always connected to adjacent cells (we omit the arrows), because the length-\( k \) substring associated with each cell already subsumes that of neighborhood cells.

Once \( w \) and \( z \) are sampled, each \( w_t \) is added to \( z_t \)’th NPYLM to update its statistics.

### 4.2 Efficient computation by the Negative Binomial generalized linear model

Inference algorithm of PYHSMM has a computational complexity of \( O(K^2L^2N) \), where \( N \) is a length of the string to analyze. To reduce computations it is effective to put a small \( L \) of maximum word length, but it might also ignore occasionally long words. Since these long words are often predictable from some character level information including suffixes or character types, in a semi-supervised setting we employ a Negative Binomial generalized linear model (GLM) for setting \( L_t \) adaptively for each character position \( t \) in the corpus.

Specifically, we model the word length \( \ell \) by a Negative Binomial distribution (Cook, 2009):

\[
\ell \sim \text{NB}(\ell | r, p) = \frac{\Gamma(\ell + r)}{\Gamma(\ell) \Gamma(r)} p^\ell (1 - p)^r.
\] (9)

This counts the number of failures of Bernoulli draws with probability \((1 - p)\) before \( r \)’th success. For our model, note that Negative Binomial is obtained from a Poisson distribution Po(\( \lambda \)) whose parameter \( \lambda \) again follows a Gamma distribution \( \text{Ga}(r, b) \) and integrated out:

\[
p(\ell | r, b) = \int \text{Po}(\ell | \lambda) \text{Ga}(\lambda | r, b) d\lambda
\] (10)

\[
= \frac{\Gamma(\ell + r)}{\Gamma(r) \Gamma(1 + b)} \left( \frac{b}{1 + b} \right)^\ell \left( \frac{1}{1 + b} \right)^r.
\] (11)

This construction exactly mirrors the Poisson-Gamma word length distribution in (Mochihashi et al., 2009) with sampled \( \lambda \). Therefore, our Negative Binomial is basically a continuous analogue of the word length distribution in NPYLM.\(^6\)

Since \( r > 0 \) and \( 0 \leq p \leq 1 \), we employ an exponential and sigmoidal linear regression

\[
r = \exp(w_r^T f), \quad p = \sigma(w_p^T f)
\] (12)

where \( \sigma(x) \) is a sigmoid function and \( w_r, w_p \) are weight vectors to learn. \( f \) is a feature vector computed from the substring \( c_1 \cdots c_t \), including \( f_0 = 1 \) for a bias term. Table 1 shows the features we used for this Negative Binomial GLM. Since Negative Binomial GLM is not convex in \( w_r \) and \( w_p \), we endowed a Normal prior \( N(0, \sigma^2 I) \) for them and used a random walk MCMC for inference.

#### Predicting \( L_t \)

Once the model is obtained, we can set \( L_t \) adaptively as the time where the cumulative probability of \( \ell \) exceeds some threshold \( \theta \) (we used \( \theta = 0.99 \)). Table 2 shows the precision of predicting maximum word length learned from 10,000 sentences from each set: it measures whether the correct word boundary in test data is included in the predicted \( L_t \).

Overall it performs very well with high precision, and works better for longer words that cannot be accommodated with a fixed maximum length.

---

\(^5\)For computational reasons, we do not pursue using a Dirichlet process to yield an infinite HMM (Van Gael et al., 2009), but it is straightforward to extend our PYHSMM to the iHMM.

\(^6\)Because NPYLM employs a mixture of Poisson distributions for each character type of a substring, this correspondence is not exact.

<table>
<thead>
<tr>
<th>Type</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_i )</td>
<td>Character at time ( t - i ) (( 0 \leq i \leq 1 ))</td>
</tr>
<tr>
<td>( t_i )</td>
<td>Character type at time ( t - i ) (( 0 \leq i \leq 4 ))</td>
</tr>
<tr>
<td>cont</td>
<td># of the same character types before ( t )</td>
</tr>
<tr>
<td>ch</td>
<td># of times character types changed within 8 characters before ( t )</td>
</tr>
</tbody>
</table>

Table 1: Features used for the Negative Binomial generalized linear model for maximum word length prediction.
Table 3: Datasets used for evaluation. Abbreviations: Ja=Japanese, Zh=Chinese, Th=Thai language.

Figure 6 shows the distribution of predicted maximum lengths for Japanese. Although we used $\theta = 0.99$, it is rather parsimonious but accurate that makes the computation faster.

Because this cumulative Negative Binomial prediction is language independent, we believe it might be beneficial for other natural language processing tasks that require some maximum lengths within which to process the data.

5 Experiments

To validate our model, we conducted experiments on several corpora of East Asian languages with no word boundaries.

Datasets For East Asian languages, we used standard datasets in Japanese, Chinese and Thai as shown in Table 3. The Kyoto corpus is a collection of sentences from Japanese newspaper (Kurohashi and Nagao, 1998) with both word segmentation and part-of-speech annotations. BCCWJ (Balanced Corpus of Contemporary Written Japanese) is a balanced corpus of written Japanese (Maekawa, 2007) from the National Institute of Japanese Language and Linguistics, also with both word segmentation and part-of-speech annotations from slightly different criteria. For experiments on colloquial texts, we used a random subset of “OC” register from this corpus that is comprised of Yahoo!Japan Answers from users. For Chinese, experiments are conducted on standard datasets of SIGHAN Bakeoff 2005 (Emerson, 2005); for comparison we used MSR and PKU datasets for simplified Chinese, and the CITYU dataset for traditional Chinese. SIGHAN datasets have word boundaries only, and we conformed to original training/test splits provided with the data. InterBEST is a dataset in Thai used in the InterBEST 2009 word segmentation contest (Kosawat, 2009). For contrastive purposes, we used a “Novel” subset of it with a random sampling without replacement for training and test data. Accuracies are measured in token $F$-measures computed as follows:

$$F = \frac{2PR}{P+R},$$

$$P = \frac{\text{# of correct words}}{\text{# of words in output}},$$

$$R = \frac{\text{# of correct words}}{\text{# of words in gold standard}}.$$

Unsupervised word segmentation In Table 4, we show the accuracies of unsupervised word segmentation with previous figures. We used bi-gram PYHSMM and set $L = 4$ for Chinese, $L = 5, 8, 10, 21$ for Japanese with different types of contiguous characters, and $L = 6$ for Thai. The number of hidden states are $K = 10$ (Chinese and Thai), $K = 20$ (Kyoto) and $K = 30$ (BCCWJ).

We can see that our PYHSMM outperforms on all the datasets. Huang and Zhao (2007) reports that the maximum possible accuracy in unsupervised Chinese word segmentation is 84.8%, derived through the inconsistency between different segmentation standards of the SIGHAN dataset. Our PYHSMM performs nearer to this best possible accuracy, leveraging both word and character knowledge in a consistent Bayesian fashion. Further note that in Thai, quite high performance is achieved with a very small data compared to previous work.

Unsupervised part-of-speech induction As stated above, Kyoto, BCCWJ and Weibo datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Kyoto</th>
<th>BCCWJ</th>
<th>MSR</th>
<th>CITYU</th>
<th>BEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision (All)</td>
<td>99.9</td>
<td>99.9</td>
<td>99.6</td>
<td>99.9</td>
<td>99.0</td>
</tr>
<tr>
<td>Precision ($\geq 5$)</td>
<td>96.7</td>
<td>98.4</td>
<td>73.6</td>
<td>87.0</td>
<td>91.7</td>
</tr>
<tr>
<td>Maximum length</td>
<td>15</td>
<td>48</td>
<td>23</td>
<td>12</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 2: Precision of maximum word length prediction with a Negative Binomial generalized linear model (in percent). $\geq 5$ are figures for word length $\geq 5$. Final row is the maximum length of a word found in each dataset.
Table 4: Accuracies of unsupervised word segmentation. BE is a Branching Entropy method of Zhikov et al. (2010), and HMM$^2$ is a product of word and character HMMs of Chen et al. (2014). * is the accuracy decoded with $L = 3$: it becomes 81.7 with $L = 4$ as MSR and PKU.

Table 5: Precision of POS tagging on correctly segmented words.

Semi-supervised experiments Because our PYHSMM is a generative model, it is easily amenable to semi-supervised segmentation and tagging. We used random 10,000 sentences from supervised data on Kyoto, BCCWJ, and LWC datasets along with unsupervised datasets in Table 3.

Results are shown in Table 6: segmentation accuracies came close to 90% but do not go beyond. By inspecting the segmentation and POS that PYHSMM has output, we found that this is not necessarily a fault of our model, but it came from the often inconsistent or incorrect tagging of the dataset. In many cases PYHSMM found more “natural” segmentations, but it does not always conform to the gold annotations. On the other hand, it often oversegments emotional expressions (sequence of the same character, for example) and this is one of the major sources of errors.

Finally, we note that our proposed model for unsupervised learning is most effective for the language which we do not know its syntactic behavior but only know raw strings as its data. In Figure 8, we show an excerpt of results to model a Japanese local dialect (Mikawa-ben around Nagoya district) collected from a specific Twitter. Even from the surface appearance of characters, we can see that similar words are assigned to the same state including some emoticons (states 9,29,32), and in fact we can identify a state of postpositions specific to that dialect (state 3). Notice that the words themselves are not trivial before this analysis. There are also some name of local places (state 41) and general Japanese postpositions or nouns (11,18,25,27,31). Because of the sparsity promoting prior (7) over the hidden states, actually used states are sparse and the results can be considered quite satisfactory.

6 Discussion The characteristics of NPYLM is a Baysian integration of character and word level information, which is related to (Blunsom and Cohn, 2011) and the adaptor idea of (Goldwater et al., 2011). This
of different types of characters is an important re-
tute a single word. Therefore, statistical modeling
Hiragana, Katakana, whose mixture would consti-
tions shared from the model structure, and
acter HMMs are just multiplied. There are no in-
model of Chen et al. (2014), where word and char-
is different from (and misunderstood in) a joint
in fact it depends on a BIO-like heuristic tagging
search venue for the future.

NPYLM has already applied and extended to
speech recognition (Neubig et al., 2010), statisti-
cal machine translation (Nguyen et al., 2010), or
even robotics (Nakamura et al., 2014). For all
these research area, we believe PYHSMM would
be beneficial for their extension.

7 Conclusion
In this paper, we proposed a Pitman-Yor Hidden
Semi-Markov model for joint unsupervised word
segmentation and part-of-speech tagging on a raw
sequence of characters. It can also be viewed as
a way to build a class n-gram language model di-
rectly on strings, without any “word” information
a priori.

We applied our PYHSMM on several standard
datasets on Japanese, Chinese and Thai, and it out-
performed previous figures to yield the state-of-
the-art results, as well as automatically induced
word categories. It is especially beneficial for col-
loquial text, local languages or speech transcripts,
where not only words themselves are unknown but
their syntactic behavior is a focus of interest.

In order to adapt to human standards given in
supervised data, it is important to conduct a semi-
supervised learning with discriminative classifiers.
Since semi-supervised learning requires generative
models in advance, our proposed Bayesian
generative model will also lay foundations to such
an extension.

References
Phil Blunsom and Trevor Cohn. 2011. A Hierarchical
Pitman-Yor Process HMM for Unsupervised Part of


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