ベイズ階層言語モデルによる教師なし形態素解析

持橋 大地　　山田 武士　　上田 修功
NTT コミュニケーション科学基礎研究所
〒 619-0237 京都府相楽郡精華町「けいはんな学研都市」光台 2–4
daichi@cslab.kecl.ntt.co.jp {yamada,ueda}@cslab.kecl.ntt.co.jp

概要

本論文では, 教師データや辞書を必要とせず, あらゆる言語に適用できる教師なし形態素解析器および言語モデルを提案する. 観測された文字列を, 文字 nグラム-単語 nグラムをノンパラメトリックベイズ法の枠組で統合した確率モデルからの出力とみなし, MCMC 法と動的計画法を用いて, 繰り返し隠れた「単語」を推定する. 提案法は, あらゆる言語の生文字列から直接, 全く知識なしに Kneser-Ney と同等に高精度にスムージングされ, 未知語のない nグラム言語モデルを構築する方法とみなすこともできる.
話し言葉や古文を含む日本語, および中国語単語分割の標準的なデータセットでの実験により, 提案法の有効性および効率性を確認した.

キーワード: 形態素解析, 単語分割, 言語モデル, ノンパラメトリックベイズ法, MCMC

Bayesian Unsupervised Word Segmentation with Hierarchical Language Modeling

Daichi Mochihashi　Takeshi Yamada　Naonori Ueda
NTT Communication Science Laboratories
Hikaridai 2-4, Keihanna Science City, Kyoto Japan 619-0237
daichi@cslab.kecl.ntt.co.jp {yamada,ueda}@cslab.kecl.ntt.co.jp

Abstract

This paper proposes a novel unsupervised morphological analyzer of arbitrary language that does not need any supervised segmentation nor dictionary. Assuming a string as the output from a nonparametric Bayesian hierarchical n-gram language model of words and characters, "words" are iteratively estimated during inference by a combination of MCMC and an efficient dynamic programming. This model can also be considered as a method to learn an accurate n-gram language model directly from characters without any "word" information.

Keywords: Word segmentation, Language Modeling, Nonparametric Bayes, MCMC

1　はじめに

日本語の形態素解析は現在 99%以上の性能を持っていると言われるが [1], はたして本当だろうか.

現在の高精度な形態素解析器はすべて, 人手で作成した教師データをもとに機械学習またはルールによって構築されており, その際の教師データは新聞記事がほとんどである. 話し言葉や, ブログ等でみられる口語体の日本語には次々に新語や新しい表現が生まれ, また単語分割の基準が曖昧なため, 形態素解析を高精度に行うことは困難である. 教師データを人手で作成する場合でも, その構築やメンテナンスには莫大なコストがかかり, それが何らかの意味で「正解」であるという保証もない.[1]

さらに, 古文や未知の言語などにはそもそも教師データがなく, これまで形態素解析は不可能であった. 図 1 に, 『源氏物語』の冒頭の一部を MeCab [2] で形態素解析した例を示す. 「た | ぐひなしと」「なほ | に | ほ | は | し | さ」などの解析結果を見るとわかるように, 現代文の教師あり学習に基づく形態素解析器では, こうした文を適切に分割することができない.

形態素解析された結果は, かな漢字変換や統計的機械翻訳, 音声認識など多くの場合, そこで用いられる nグラムなどの言語モデルへの入力として使われる. 人手による教師データを基本とした従来の形態素解析には, 適用の際のこうした性能を最適化していないという問題もあった. また理学的あるいは計算言語学的にみると, たとえ未知の言語であったとしても, 言語データに隠れた統計的性質を用いて, 「単語」のような基礎的な単位については導出できることが望ましい.

こうした考えに基づき, 本論文では任意の言語について, 観測された文字列のみから辞書や教師データを全く使わずに「単語」を推定することのできる, ノンパラメトリックベイズ法に基づいた教師なし形態素解析器および言語モデルを提案する. 提案法は任意

[1] 新聞記事の場合でも問題は同様であり, 「正解」データは本質的に一意ではない. よって, 複数の品詞体系やタグ付け基準があり, 教師あり学習はそうした恣意性から逃れることができない.

世に | た | ぐひなしと | 見 | たてまつり | た | まひ | 、 | 名高う | お | はする | 宮 | の | 御 | 容貌 | に | も | 、 | なほ | に | ほ | は | し | さ | はたと | へ | む | 方 | なく | 、 | うつくし | げ | なる | を | 、 | 世 | の | 人 | 光る | 君 | と | 聞こ | ゆ | 。 | 藤 | 壺 | なら | びたまひて | 、 | 御 | おぼえ | も | とりどり | なれ | ば | 、 | かか | やく | 日 | の | 宮 | と | 聞こ | ゆ | 。 · · ·

図 1: 『源氏物語』の MeCab による解析.

の言語の文字列から直接言語モデルを学習する方法とも見なすことができ, 推論の際に効率的な MCMC 法を用いて繰り返し単語分割を改良していくことで学習を行う. 最終的に学習データの最適な単語分割と言語モデルが得られ, 言語モデルを用いてビタビアルゴリズムで解析することにより未知データの形態素解析も行うことができる.

教師なし学習のため, 提案法は学習データを原理的にいくらでも増やすことができ, 「未知語」が存在せず, ドメイン適応も容易である. また, 教師ありデータを事前知識として組み込むこともできる.

以下ではまず, 2 章で教師なし形態素解析の定式化と, これまでの関連研究について説明する. 3 章では階層ベイズ法による n グラムモデルを文字–単語とさらに階層化して得られる言語モデルを示し, 4 章で MCMC 法と動的計画法を組み合わせた学習法について述べる. 5 章で新聞記事・話し言葉・古文の日本語, および中国語, 英語の単語分割の実験を行って有効性を示し, 6 章で考察を行って全体をまとめる.

2 教師なし形態素解析とは

自然言語の文字列 $s = c_1 c_2 \cdots c_N$ が与えられたとき, 教師なし形態素解析とは, s を分割して得られる単語列 $\mathbf{w} = w_1 w_2 \cdots w_M$ の確率 $p(\mathbf{w}|s)$ を最大にする単語列 $\hat{\mathbf{w}}$ を求める問題と考えることができる. [2]

$$\hat{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} p(\mathbf{w}|s) \tag{1}$$

これは, 「言語として最も自然な単語分割」を求めたいということと等しい. 「形態素解析」というと $\mathbf{w}$ の品詞タグ付けも含むことが多いが, 品詞の決定には本来, 構文解析を必要とすると考えられること, また n グラムや統計的機械翻訳など多くのタスクにおいて単語分割のみが必要とされることから, 本論文では「形態素解析」とは最も基本的な, 単語分割を指すこととする.[3]

(1) 式の確率 $p(\mathbf{w}|s)$ は言語モデルによって計算することができ, これを最大化する $\hat{\mathbf{w}}$ は, 単語辞書および言語モデルが存在すれば, 可能な単語の組み合わせについてビタビアルゴリズムを適用することで得ることができる.

しかし, 教師なし形態素解析においてはそもそも単語が未知である. [3][5] ではこの制約をやや緩め, 未知語の単語らしさを文字 n グラムで与えたり, 単語リストを与えた下で, (1) 式による分割と言語モデルを交互に最適化する方法を示したが, 依然として単語分割済みコーパスや, 単語リストを必要としていた. これらは未知の言語については原理的に準備不可能であり, また既知の言語についても, 単語分割の「正解」は一意ではなく [6], たとえば話し言葉や口語体については何を「単語」とすべきか同定することも非常に難しい. さらに, 単語の種類は有限ではなく, テキストには既存の単語リストでカバーできない大量の「未知語」が含まれており, こうした未知語の取り扱いが形態素解析の重要な問題となってきている [7].

$s =$	彼	女	の	言	っ	た	言	葉	は	· · ·
$\mathbf{z} =$	0	1	1	1	0	1	0	1	1	· · ·
$\mathbf{w} =$	彼女		の	言	っ	た	言	葉	は	· · ·

図 2: 単語分割と潜在変数ベクトル $\mathbf{z}$.

純粋に統計的機械学習の問題としてみると, (1) 式は s の各文字 c_i にその直後が単語境界のとき 1, そうでないとき 0 をとる潜在変数 z_i があると考えれば, $\mathbf{w}$ は潜在変数ベクトル $\mathbf{z} = z_1 z_2 \cdots z_N$ と同一視できるから,

$$\hat{\mathbf{z}} = \operatorname*{argmax}_{\mathbf{z}} p(\mathbf{z}|s) \tag{2}$$

を最大化する $\hat{\mathbf{z}}$ を求める学習問題と考えることができる. これは $\mathbf{z}$ を隠れ状態とする, semi-Markov モデルまたは分割モデル [8] と呼ばれる HMM の変種であり, 各文 s について可能な $\mathbf{z}$ は指数的に存在するため, 効率的な学習が必要となる.

簡単な方法として最近のものに, MDL を基準に文字のチャンキングを繰り返す方法 [9] があり, またよりベイズ的な方法として, [10] は階層ディリクレ過程 (HDP) による単語バイグラムモデルを用いて, z_i をギブスサンプラにより一文字ずつ更新する方法を示した.

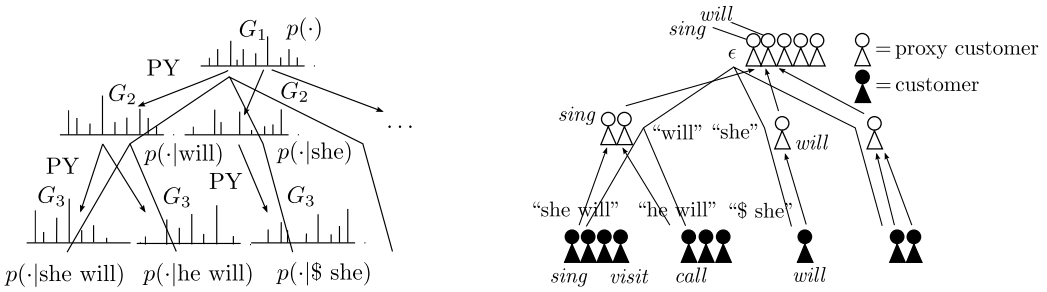
しかし, これらの方法は単語分割を一箇所ずつ変えるために, 膨大な計算量を必要とする. さらに, 単語分割では異なる z_i の間に高い相関があるために収束がきわめて遅く, 非常に少量のコーパスについてしか適用できなかった. また, この方法では単語のバイグラムまでしか考慮することができず, モデルも単語分割のために補助的に導入されたもので, 何が「単語らしい」かの基準を持っていないという問題がある.

これに対し本論文では, 文字–単語の階層 n グラム言語モデルの性能と, それに基づく単語分割を直接最適化する方法を示し, このために動的計画法と MCMC を組み合わせた効率的な学習法を提案する.

提案法は n グラム言語モデルのベイズモデルである HPYLM を基にしているため, 次にまず HPYLM について説明し, 続いてそれを文字–単語と階層化することで, あらゆる言語および未知語に対応し形態素解析を行うことのできる言語モデルを示す.

[2] この定式化は一般化すると統計翻訳とみることができ, s がひらがな列のとき, かな漢字変換と等価となる.

[3] 簡単な教師なし品詞推定には, 提案法によって単語分割を行った後, HMM を走らせる方法 [4] がある.



(a) Pitman-Yor 過程による, nグラム分布 G_n の階層的な生成.

(b) 等価な CRP を用いた表現. 学習データの各単語を「客」とみて, 対応する文脈ノードに一つずつ追加していく.

図 3: n グラム言語モデルのベイズ学習.

3 HPYLM から NPYLM へ

3.1 HPYLM: ベイズ n グラム言語モデル

言語モデルを用いて形態素解析を行うためには, 可能なあらゆる単語分割について確率を与える方法が必要となる. 従来これには, 未知語を表す特別なトークン UNK を導入して確率を求めるなど, ヒューリスティックな方法が使用されてきたが [3], ディリクレ過程およびその一般化である Pitman-Yor 過程による nグラムモデルを用いることで, 理論的に見通しよく, 精密なモデル化が可能になる. これについて簡単に説明する.

Pitman-Yor(PY) 過程は, 基底測度とよばれるある確率分布 G_0 に似たランダムな離散確率分布 G を生成する確率過程であり, 下のように書かれる.

$$G \sim \mathrm{PY}(G_0, d, \theta)\,. \tag{3}$$

d はディスカウント係数, θ は G が平均的にどのくらい G_0 と似ているかを制御する, PY 過程のパラメータである. $d=0$ のとき, $\mathrm{PY}(G_0, 0, \theta)$ はディリクレ過程 $\mathrm{DP}(\theta)$ と一致する.

いまユニグラム分布 $G_1 = \{\, p(\cdot)\,\}$ があるとすると, 単語 v を文脈としたバイグラム分布 $G_2 = \{\, p(\cdot|v)\,\}$ は G_1 とは異なるが, 高頻度語などについて G_1 を反映していると考えられるから, G_1 を基底測度とした PY 過程により $G_2 \sim \mathrm{PY}(G_1, d, \theta)$ と生成されたと仮定することができる. 同様にトライグラム分布 $G_3 = \{\, p(\cdot|v'v)\,\}$ はバイグラム分布を基底測度として $G_3 \sim \mathrm{PY}(G_2, d, \theta)$ と生成でき, G_1, G_2, G_3 は図 3(a) のような木構造をなすことになる.

実際には G は積分消去することができ, このとき, 階層 Pitman-Yor 過程に基づく nグラム言語モデル (HPYLM) は図 3(b) のように, 階層的な CRP(中華料理店過程) で表現することができる. この CRP では, 学習データの各単語を「客」と呼び, nグラム文脈に対応する木の葉に一つずつ追加していく. 例えば, トライグラムの学習データに「彼 は 行く」という文があったとき, 4 人の客 "彼" "は" "行く" "$" を, それぞれ直前の 2 単語 "$ $" "$ 彼" "彼 は" "は 行く" の文脈に対応する葉に追加する. "$" は言語モデルで必要な文境界を表す, 長さ 0 の単語である.

単語 w の客をノード h に追加することは, 対応する nグラムカウント $c(w|h)$ を 1 増やすことを意味する. ただし, バックオフと同じ意味でこれは本当は, 親ノードでの 1 つ短い文脈 h' を用いた $(n-1)$ グラムから生成された可能性がある.[4] この時, 客 w のコピーを「代理客」として親 h' にも同様に追加する. この客の追加は再帰的に行うため, すべての種類の単語は必ず, 対応する客をユニグラムすなわち根ノードに 1 つ以上持つことになる (図 3(b)).

こうして, カウント $c(w|h)$ のうち, 親ノードから生成されたと推定された回数を t_{hw} とおくと, HPYLM での nグラム確率 $p(w|h)$ は (n–1) グラム確率 $p(w|h')$ を使って, 次のように階層的に表すことができる.

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

ここで, $t_{h\cdot} = \sum_w t_{hw}$, $c(h) = \sum_w c(w|h)$ とした.

一般には t_{hw} は $c(w|h)$ の対数のオーダーの数になるが [11], t_{hw} を常に 1 にすると (4) は Kneser-Ney スムージング [12] と一致し, HPYLM は Kneser-Ney nグラムの, より精密なベイズモデルであることがわかる. 学習の際には MCMC 法を用い, 客をランダムに選んで削除し, また追加することを繰り返すことで t_{hw} を最適化していく. d, θ の推定など詳しくは, [11] を参照されたい.

3.2 HPYLM の階層化

(4) 式は単語ユニグラムの場合は, $p(w|h')$ が単語の事前確率を表すゼログラムとなるが, これはどのように与えたらよいだろうか.

語彙が有限ならば $1/|V|$ (V は語彙集合) とすればよいが, 形態素解析においては語彙は無限であり, あらゆる部分文字列が単語となる可能性がある.

ただし, 言語において単語となるべき綴りは決してランダムではない. そこで, 本研究では [3] と同様に, 単語の事前確率をその綴りの文字 nグラムによって与え,

$$G_0(w) = p(c_1 c_2 \cdots c_k) \tag{5}$$

[4] もともと $c(w|h)=0$ だったとき, 確率 0 の事象からカウントが生成されたことになってしまうから, 最初は必ず親から生成されたものである. しかし, 2 回目以降はそうとは限らない.

と事前確率を計算することにする．$c_1 \cdots c_k$ は, 単語 w の文字列としての表記である．$p(c_1 \cdots c_k)$ は文字 HPYLM によって同様に計算される.[5] 文字 n グラムオーダー n に対する依存性を避けるため, 本研究では文字モデルには可変長の ∞-グラム言語モデル [13] を用いた．このとき, 単語ユニグラム分布 G_1 は (5) 式で与えられる単語事前確率 G_0 を基底測度として, $G_1 \sim \mathrm{PY}(G_0, d_0, \theta_0)$ のように同様に PY 過程から生成されることになる．

これは図 4 のように, 単語 HPYLM の基底測度にまた文字 HPYLM が埋め込まれた, 階層 n グラムモデルであり, 以下 Nested Pitman-Yor Language Model (NPYLM) と呼ぶ.[6] このモデルでは, まず文字 n グラムによって単語が無限に生成され, それを単語 n グラムによって組み合わせることで文字列が生成される．われわれの目標は, 観測値であるこの文字列のみから, 隠れた「単語」を推定し, 単語モデルと文字モデルを同時に求めることである．

(5) 式はあらゆる綴りに確率を与えるため, G_0, およびそこから生成される $G_1, G_2, \cdots$ はすべて可算無限次元となることに注意されたい．その場合でも CRP に基づき, (4) 式および (5) 式を素直に適用することで n グラム確率が求まる．こうした構成から, NPYLM での単語 n グラム確率にはつねに, 文字 n グラムで計算される単語の表記確率が反映されており, 両者を見通しよく統合する言語モデルとなっている．

実際には, (5) 式だけでは長い単語の確率が小さくなりすぎるため, 本研究では単語長がポアソン分布に従うようにさらに補正を行った．これについては 4.3 節で詳しく述べる．

CRP 表現 NPYLM では単語モデルと文字モデルは独立ではなく, CRP を介して繋がっている．単語 HPYLM のユニグラムに単語 w が新しく現れたり, 対応する変数 $t_{\epsilon w}$ が 1 増えたとき, これは w がユニグラムの基底測度, すなわち文字 HPYLM から生成されたことを意味するので, w を文字列 $c_1 \cdots c_k$ に分解して得られた “文” を文字 HPYLM にデータとして追加する．逆にユニグラムから w が消えたり, $t_{\epsilon w}$ が 1 減った場合, 対応するデータが無効となったことを意味するので, 文字 HPYLM からそのデータを削除する．

これらはすべて, 通常の HPYLM と同様に MCMC の中で単語の削除と再追加をランダムに繰り返すときに起こるが, いま単語は未知であるから, まず文を単語に分解する必要がある．本研究ではこれを動的計画法によって効率的に行い, MCMC と組み合わせてモデル全体を学習していく．これについて次に説明する．

[5] 文字 HPYLM での最終的な基底測度 G_0 には, 対象とする言語の可能な文字集合 (JIS X0208 ならば 6879 個) について等確率の事前分布を用いる．

[6] 厳密には, これは Nested Dirichlet Process [14] の意味で「ネスト」しているわけではないが, 直観的な名称を用いた．

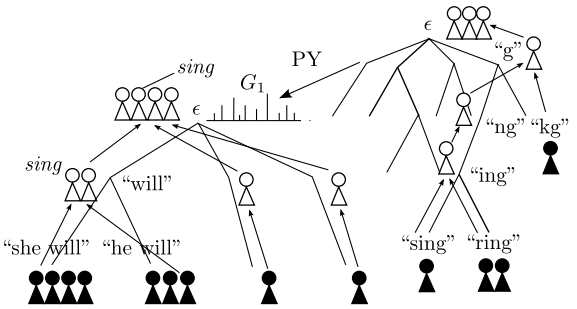


図 4: NPYLM の階層 CRP 表現.

4 学習

各文の単語分割 $\mathbf{w}$, すなわち $\mathbf{z}$ を求める最も簡単な方法は, $\mathbf{z}_1, \ldots, \mathbf{z}_D$ の中から 1 つの文字に対応する z_i をランダムに選び, それが 1 か 0 かを言語モデルから得られる確率を用いてサンプリングし, その結果によって言語モデルを更新する, というギブスサンプリングを繰り返す方法である．充分サンプリングを繰り返せば, $\mathbf{z}$ は真の分布である (2) 式からのサンプルに収束する．[15]

しかし, この方法は学習データのすべての文字毎にサンプリングを繰り返すため, 2 章で述べたように特に単語分割の場合はきわめて非効率的であり[7], アニーリングを行わない限り収束も難しい [10]．また, 隣同士の単語の関係のみを見ているため, バイグラムまでしか考慮できないという問題もある．

4.1 Blocked Gibbs Sampler

これに代わり, 本研究では文ごとの単語分割 $\mathbf{w}$ を, 動的計画法により効率的にサンプリングする．$\mathbf{w}$ すなわち $\mathbf{z}$ をまとめてサンプリングするため, これはブロック化ギブスサンプラ [15] と呼ばれるものとなり, 図 5 に示したアルゴリズムとなる．

最初は単語が未知のため, 文字列 s 全体が一つの「単語」となりそのまま文字モデルに渡されるが, 2 回目以降は古い単語分割によるデータを言語モデルから削除した後, s の新しい単語分割 $\mathbf{w}(s)$ を $p(\mathbf{w}|s)$ からサンプルし, 言語モデルを更新する．この操作をすべての文についてランダムな順番で繰り返し行い,

```
for j = 1 ··· J do
  for s in randperm (s_1, ··· , s_D) do
    if j > 1 then
      Remove customers of w(s) from Θ
    end if
    Draw w(s) according to p(w|s, Θ)
    Add customers of w(s) to Θ
  end for
  Sample hyperparameters of Θ
end for
```

図 5: NPYLM Θ のブロック化ギブスサンプラ．

[7] [16] では, この方法は “Direct Gibbs” と呼ばれている．

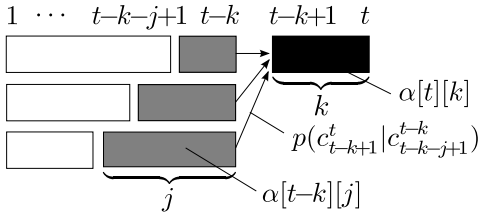


図 6: 可能な単語分割 j の周辺化による前向き確率 $\alpha[t][k]$ の計算.

for $t = 1$ to N **do**
  for $k = \max(1, t-L)$ to t **do**
    Compute $\alpha[t][k]$ according to (6).
  end for
end for
Initialize $t \leftarrow N$, $i \leftarrow 0$, $w_0 \leftarrow \$$
while $t > 0$ **do**
  Draw $k \propto p(w_i | c_{t-k+1}^t, \Theta) \cdot \alpha[t][k]$
  Set $w_i \leftarrow c_{t-k+1}^t$
  Set $t \leftarrow t-k$, $i \leftarrow i+1$
end while
Return $\mathbf{w} = w_i, w_{i-1}, \cdots, w_1$.

図 7: 単語分割 $\mathbf{w}$ の Forward-Backward サンプリング (バイグラムの場合).

単語分割とそれに基づく言語モデルを交互に最適化していく. 「京都大学」のように複数の分割がありうる場合, 「京都大学」と「京都 大学」の両方を確率的に考慮することで, 局所解に陥ることを避け, よりよいモデルを得ることができる. 図 8 に, 京大コーパスにおいて Gibbs の繰り返し毎に単語分割 $\mathbf{w}(s)$ が確率的に改良されていく様子を示した.

4.2 Forward filtering-Backward sampling

それでは, 具体的に $\mathbf{w}(s)$ をサンプリングするにはどうすればいいのだろうか. HMM のベイズ学習で知られている Forward filtering-Backward sampling 法 [16] を応用すると, これは PCFG の構文木の MCMC によるサンプリング [17] と本質的に同じ方法で行うことができることがわかる.

Forward filtering このために, バイグラムの場合は前向き確率 $\alpha[t][k]$ を導入する. $\alpha[t][k]$ は s の部分文字列 $c_1 \cdots c_t$ が, 最後の k 文字を単語として生成された確率であり (図 6), 次の再帰式により, それ以前の可能な分割すべてについて周辺化されている.

$$\alpha[t][k] = \sum_{j=1}^{t-k} p(c_{t-k+1}^t | c_{t-k-j+1}^{t-k}) \cdot \alpha[t-k][j] \quad (6)$$

ただし $\alpha[0][0]=1$ であり, $c_n \cdots c_m = c_n^m$ と書いた.

この関係が成り立つ理由は以下である. 二値変数列 $z_1 \cdots z_N$ を保持することは, 各時刻 t において左側の最も近い単語境界への距離 q_t を保持することと等価であるから,

$$\alpha[t][k] = p(c_1^t, q_t = k) \quad (7)$$

```
1 神戸では異人館 街の 二十棟 が破損した 。
2 神戸 では 異人館 街の 二十棟 が破損した 。
10 神戸 では 異人館 街の 二十棟 が破損した 。
50 神戸 で は異人 館 街 の 二 十 棟 が 破損 し た 。
100 神戸 で は 異 人館 街 の 二 十 棟 が 破損 し た 。
200 神戸 で は 異人館 街 の 二 十 棟 が 破損 し た 。
```

図 8: ギブスサンプリングの繰り返しと単語分割 $\mathbf{w}(s)$ の改良. $\mathbf{w}(s)$ は最尤解とは限らず, 確率的である.

$$= \sum_j p(c_{t-k+1}^t, c_1^{t-k}, q_t = k, q_{t-k} = j) \quad (8)$$

$$= \sum_j p(c_{t-k+1}^t | c_1^{t-k}, q_{t-k} = j) p(c_1^{t-k}, q_{t-k} = j) \quad (9)$$

$$= \sum_j p(c_{t-k+1}^t | c_{t-k-j+1}^{t-k}) \alpha[t-k][j] \quad (10)$$

が成り立っている. ここで, (9) 式で q_t と q_{t-k} の条件つき独立性を用いた.

Backward sampling 前向き確率テーブル $\alpha[t][k]$ が求まると, 文末から後向きに可能な単語分割をサンプリングすることができる. $\alpha[N][k]$ は文字列 c_1^N のうち最後の k 文字が単語である確率であり, 文末には必ず特別な単語\$が存在するから, $p(\$ | c_{N-k}^N) \cdot \alpha[N][k]$ に比例する確率で k をサンプルし, 最後の単語を決めることができる. その前の単語も今決めた単語に前接するように同様にサンプルでき, これを文字列の先頭に達するまで繰り返す. (図 7)

トライグラム 上では簡単のためバイグラムの場合を説明したが, トライグラムの場合は, 前向き確率に $\alpha[t][k][j]$ を用いる.[8] これは文字列 c_1^t が, 最後の k 文字, およびさらにその前の j 文字を単語として生成された確率である. Forward-Backward アルゴリズムは複雑になるため省略するが, 2 次の HMM のビタビアルゴリズム[19] と同様にして導出することができる.

計算量 このアルゴリズムの計算量は文字列長を N として, 文ごとにバイグラムの場合は $O(NL^2)$, トライグラムは $O(NL^3)$ である. ただし, L は単語の可能な最大長 ($\leq N$) とした.

4.3 単語モデルとポアソン補正

このモデルはベイズ的な階層 n グラムモデルとして自然なものであるが, 実際には式 (5) だけでは, カタカナ語など, 綴りの長い単語の確率が小さくなりすぎるという問題が生じる [3]. 単語長は大まかにポアソン分布に従うから, これを補正するために, (5) 式を

$$p(c_1 \cdots c_k) = p(c_1 \cdots c_k, k | \Theta) \quad (11)$$

$$= \frac{p(c_1 \cdots c_k, k | \Theta)}{p(k | \Theta)} \mathrm{Po}(k | \lambda) \quad (12)$$

と変形する. $p(k|\Theta)$ は文字 n グラムモデル Θ から

[8] 理論的には 4 グラムやそれ以上も可能であるが, あまりに複雑になる一方で, 差はそれほど大きくないと考えられる. むしろこのような場合は Particle MCMC 法 [18] が有望だと思われるが, 予備実験では動的計画法ほど効率的ではなかった.

モデル	P	R	F	LP	LR	LF
NPYLM	**74.8**	**75.2**	**75.0**	47.8	**59.7**	53.1
HDP	61.9	47.6	53.8	57.0	57.5	57.2

表 1: 英語音素列データでの性能比較. NPYLM が提案法を示す. “HDP” の結果は [10] から引用した.

モデル	計算時間	iteration
NPYLM	17 分	200
HDP	10 時間 55 分	20000

表 2: 表 1 の結果に要した計算量. NPYLM は実際には 50 回, 4 分ほどでほぼ収束した.

長さ k の単語が出現する確率であり, [3] などでは $p(k|\Theta) = (1-p(\$))^{k-1}p(\$)$ と計算しているが, これはユニグラムの場合以外は正しくない. 本研究では, モンテカルロサンプリングを用いて Θ から単語をランダムに生成し, 正確な値を推定した.[9]

λ の推定　本研究では (12) のポアソン分布 $\mathrm{Po}(k|\lambda)$ のパラメータ λ も定数ではなく, ガンマ事前分布

$$p(\lambda) = \mathrm{Ga}(a,b) = \frac{b^a}{\Gamma(a)}\lambda^{a-1}e^{-b\lambda} \qquad (13)$$

を与えて, データから自動的に推定する. a, b は $p(\lambda)$ がほぼ一様分布となるハイパーパラメータである.

単語分割で得られた語彙集合を W とすると, λ の事後分布は $|\cdot|$ を単語の長さを返す関数として,

$$\begin{aligned} p(\lambda|W) &\propto p(W|\lambda)p(\lambda) \\ &= \prod_{w\in W}\left(e^{-\lambda}\frac{\lambda^{|w|}}{|w|!}\right)^{t(w)}\cdot\frac{b^a}{\Gamma(a)}\lambda^{a-1}e^{-b\lambda} \\ &= \mathrm{Ga}\left(a+\sum_{w\in W}t(w)|w|,\ b+\sum_{w\in W}t(w)\right) \qquad (14) \end{aligned}$$

となる. ここで, $t(w)$ は同じ単語 w が文字 HPYLM から生成されたと推定された回数, すなわち単語ユニグラムでの $t_{\epsilon w}$ である. カタカナ語や漢字など, 単語種毎に長さの分布は異なるため [3], 各単語種[10]毎に異なる λ を用い, Gibbs の繰り返し毎に λ を (14) からサンプリングした.

5 実験

5.1 英語音素列データ

直接の先行研究である [10] と比較するため, 最初に [10] で使われている英語の音素列データを用いて実験を行った. このデータは CHILDES データベースを基に作成された, 9,790 個の音素列書き起こしデータである.[11] 一文の平均は 9.79 文字とかなり短いため, 実験では $L=4$ とした.

[9] この計算は, 現在の計算機では数秒で終了する.

[10] 単語種としては, 英字, 数字, 記号, ひらがな, カタカナ, 漢字, 漢字+ひらがな混合, 漢字+カタカナ混合, それ以外の計 9 種を用いた. 実装は Unicode で行って文字種判定には ICU [20] を使用しているため, 言語には依存しない.

[11] このデータは実装および評価用プログラムとともに, `http://homepages.inf.ed.ac.uk/sgwater/` から入手できる.

モデル	MSR	CITYU	京大
NPY(2)	0.802 (51.9)	**0.824** (**126.5**)	0.621 (23.1)
NPY(3)	**0.807** (**48.8**)	0.817 (128.3)	**0.666** (**20.6**)
NPY(+)	0.804 (**38.8**)	0.823 (**126.0**)	**0.682** (**19.1**)
ZK08	0.667 (—)	0.692 (—)	—

表 3: 正解との一致率 (F 値) および, 文字あたりパープレキシティ. NPY(2), NPY(3) は単語バイグラムおよびトライグラムの NPYLM, NPY(+) は NPY(3) の学習データを 2 倍にした場合. ZK08 は [21] での最高値を示す. 文字モデルには ∞ グラムを用いた.

	MSR	CITYU	京大
Semi	0.893 (**48.8**)	0.895 (**124.6**)	0.914 (**20.3**)
Sup	**0.945** (81.5)	**0.941** (194.8)	**0.971** (21.3)

表 4: 半教師ありおよび教師あり学習の精度. 半教師あり学習では, 10000 文の教師データを用いた.

表 1 に, 200 回の Gibbs iteration 後の結果を示す. 精度 (P), 再現率 (R), F 値 (F) とも [10] に比べて大幅に上昇しており, 提案法の性能の高さを示している. 一方, 単語分割で得られた語彙に対する同様の値 (LP,LR,LF) は必ずしも上昇しているわけではない. 表 2 に, 表 1 の結果を得るために必要とした計算時間を示す. [10] の繰り返し回数は, 論文に書かれているものを使用した. MCMC の収束は一意ではないが, 推定も非常に効率的になっていることがわかる.

5.2 日本語および中国語コーパス

次に, 実際の標準的なコーパスとして, 公開データセットである中国語の SIGHAN Bakeoff 2005 [22] の単語分割データセットおよび京大コーパスを使って実験を行った.

中国語　教師なしでの最新の結果である [21] (Bakeoff 2006 のクローズドデータを使用) と比較するため, 二者で共通なものとして簡体中国語用に Microsoft Research Asia (MSR) のセット, 繁体中国語用に City University of Hong Kong (CITYU) のセットを使用した. それぞれ 50,000 文をランダムに選んで学習データとし, 評価データは同梱のものを用いた.

日本語　京大コーパスバージョン 4 のうち, ランダムに選んだ 1,000 文を評価データ, 残りの 37,400 文を学習データとして用いた.

いずれも学習データは空白をすべて取り除いた生文字列であり, 中国語では $L=4$, 日本語では $L=8$ とした.

なお, 上記の元データは京大コーパス約 3.7 万文, MSR 8.6 万文, CITYU 5.3 万文であるが, 提案法は教師なし学習のため, 学習データを原理的にいくらでも増やすことができる. この効果を検証するため, さらに同量の学習データを京大コーパスは毎日新聞 1996 年度[12]から, MSR は未使用の部分および PKU

[12] 京大コーパス (1995 年度毎日新聞) と近い年度を用いた.

九 日 付 の 英 有力 紙 タイムズ は 、 同 国 南部 の ウェイマス に近い ポートランド の 海軍基地 を 欧州 向け 物資 の 陸揚げ 基地 として 日本 企業 ない し 企業 連合 に そっくり 売却 する 構想 が 浮上 し て い る と 報じ た 。
五輪 五 位 の 清水宏保 は インカレ も 2 種目 を 制 し て おり 、 堀井 に どこ まで 迫 る か 。
第百十二回 芥川 ・ 直木賞 の 選考 委員 会 は 、 十 二 日 夜 、 東京 ・ 築地 の 「 新 喜楽 」 で 行われ 、 芥川 賞 、 直木賞 と も 該当 作 なし と 決ま っ た 。

図 9: 京大コーパスの形態素解析 (NPY(3+)).

セットから, CITYU は Sinica セットから追加した実験も同時に行った.

結果　400 回の Gibbs iteration 後の京大コーパスのテストデータの形態素解析例を図 9 に, 数値結果を表 3 に示す.[13] 京大コーパスの F 値が直感ほど高くないのは, “正解コーパス” と活用語尾の扱いが異なることや, 「に近い」のような慣用句, 「海軍基地」「清水宏保」といった固有名詞が提案法では適切に結合されていることにあると考えられる. 一方で低頻度語はデータが少ないため助詞と結合する場合があり, 予め文字モデルを学習したり, さらにデータを増やす必要がある.

中国語ではいずれのセットについても, ヒューリスティックな [21] での最高値を大きく上回っており, 精密な確率モデルに基づく提案法の有効性を示している. 中国語についてはバイグラムとトライグラムの結果に大きな違いはないが, 日本語ではトライグラムの方が性能がかなり上昇している. 実際に表 3 には表れていないが, 単語あたりパープレキシティは 336.1(バイグラム) から 154.0(トライグラム) へと大きく減少している. これはトライグラムが日本語の単語間の複雑な関係をとらえ, 高精度な予測とより短い単語分割を生んでいる (学習データの平均単語長 2.02→1.80) ことを意味する.

半教師あり学習　提案法は完全な生成モデルであるが, 教師なし学習だけではなく, 半教師あり学習や教師あり学習も行うことができる. これには図 5 のアルゴリズムにおいて, 単語分割 $\mathbf{w}(s)$ を教師ありのものに固定すればよい. 表 4 に, 通常の学習データのうちそれぞれ 1 万文を教師ありとした場合, およびすべて教師ありとした場合の精度を示す. 教師ありの場合, 日本語で 97%, 中国語で 94%程度, 半教師ありの場合も, 1/5 程度の教師ありデータで日本語・中国語とも 90%程度の性能を達成する.

ただし, 教師なし学習にとって人手による分割との一致率が高いことが「正解」とは限らないことに注意されたい. 実際にテストデータの文字あたりパープレキシティは, 教師なし, 半教師ありの方が正解コーパスの単語分割を用いた場合よりずっと高い性能を持っており, 人手で与えた単語分割が言語モデルとして最適とは限らないことを示している.

[13] 日本語は $L=8$ と探索範囲が広いため, 組み合わせも考慮すると, 中国語より問題がかなり難しい.

いづれ の 御 時 に か 、 女御 更衣 あまた さぶらひ たまひ け る 中に 、 いと やむごと なき 際 には あら ぬ が 、 すぐれて 時めき たまふ あり けり 。 はじめより 我 は と 思ひ あが り たまへ る 御方々 、 めざまし き ものに おとしめ そねみ たまふ 。 同じ ほど 、 それ より 下 臈 の 更衣 たち は 、 まして やす から ず 。 朝夕 の 宮仕 につけて も 、 人の 心を のみ 動かし 、 恨みを 負ふ つも り にや あり け ん 、 いと あつし く な り ゆき 、 もの心 細 げ に 里 がち なる を 、 いよいよ あかず あはれ なる ものに 思ほし て 、 ···

図 10: 『源氏物語』の教師なし形態素解析.

5.3 話し言葉コーパス

提案法は話し言葉やブログ等にみられる口語など, 単語の基準が曖昧な場合に特に効果的だと考えられる. これを調べるため, [9] と同様に, 日本語話し言葉コーパス [23] (CSJ) の「対話」部分を用いて実験を行った. [9] では文という単位が存在しないなど前処理が異なるが, 学習および評価に用いた書き起こしデータは同一である. このデータは学習 6405 文, テスト 322 文とかなり少ないため, さらに「対話」部分以外から 5 万文を学習データとして追加した実験も行った.

図 11 に単語分割の例を, 表 5 に文字あたりパープレキシティの比較を示す. 「っていうの」のような会話文特有の表現やフィラーが教師なしで認識されており, 文字あたりパープレキシティでは CSJ の短単位を用いた場合よりも優れた性能を持っている. [14]

NPY(2)	NPY(2+)	NPY(3)	NPY(3+)	短単位 (+)
16.8	**13.9**	21.2	18.1	14.9

表 5: CSJ の文字当たりテストセットパープレキシティ. +は学習データを増やした場合を表す.

5.4 古文および西欧語

提案法は教師データを必要とせず, すべてのパラメータをデータから学習するため, あらゆる言語に適用することができる. 特に, 古文や文語文の形態素解析は, 本手法により初めて完全に可能になった. 図 10 に, 『源氏物語』の冒頭を形態素解析した例を示す.

現代文の場合と同様に, 低頻度語と助詞が結合することがあるが, 古典文法や教師データを一切与えていないにもかかわらず, 多くの場合にきわめて適切な単

口 が 口 が 動い て ます よ ね 口 の 形 が はい 口 っていうの は 唇 も含める んだ けど あー はい はい から 喉 も含める んだ けど も そう いった 運動 のことを 調音運動 って 言う 訳 うん うん
あ ー その 言葉の 発声 する 時 の そう そうそう 運動 言葉 を 発声 する 為に 為に 行なう その 一 舌 だとか 唇 だとか はい あるいは 喉頭 だとか ふーん そう いった もの が こう みんな 協力して こう 協調 し て 非常に こう 素早く 動く 訳 です よ ね はい そう いった もの を それ を 調音運動 って 言う んです ね ほー お

図 11: 日本語話し言葉コーパスの形態素解析.

[14] バイグラムの性能が高い理由は, 比較の必要からフィラーを残したため, データが少ない場合はトライグラムが情報源としてふさわしくないためだと考えられる.

lastly,shepicturedtoherselfhowthissamelittlesisterofhe rswould,intheafter-time,beherselfagrownwoman;andh owshewouldkeep,throughallherriperyears,thesimplean dlovingheartofherchildhood:andhowshewouldgathera boutherotherlittlechildren,andmaketheireyesbrightan deagerwithmanyastrangetale,perhapsevenwiththedre amofwonderlandoflongago:andhowshewouldfeelwitha lltheirsimplesorrows,andfindapleasureinalltheirsimple joys,rememberingherownchild-life,andthehappysumm erdays.

(a) 学習データ (部分).

last ly , she pictured to herself how this same little sister of her s would , inthe after - time , be herself agrown woman ; and how she would keep , through allher ripery ears , the simple and loving heart of her child hood : and how she would gather about her other little children ,and make theireyes bright and eager with many a strange tale , perhaps even with the dream of wonderland of longago : and how she would feel with all their simple sorrow s , and find a pleasure in all their simple joys , remember ing her own child - life , and thehappy summerday s .

(b) 単語分割結果. 辞書は一切使用していない.

図 12: "*Alice in Wonderland*" の単語分割.

語分割が得られていることがわかる. 低頻度語についても, 古文の見出し語を文字モデルに事前に与えておくことによって, さらに改善されると期待できる.

最後に, 提案法は東洋語だけでなく, 西欧語やアラビア語にもそのまま適用することができる. 図 12 に, 空白をすべて削除した "*Alice in Wonderland*" の学習テキストと, そこから推定した単語分割を示す. この学習テキストは 1,431 文, 115,961 文字と非常に小さいにもかかわらず, 教師なしで驚くほど正確な単語分割が得られている. また, last-ly, her-s など接尾辞が自動的に分離されていることに注意されたい. こうした結果は屈折や複合語の多いドイツ語, フィンランド語等の解析に特に有用だと考えられる.

6 考察およびまとめ

本研究では, 階層 Pitman-Yor 過程によるベイズ n グラム言語モデルを文字–単語とさらに階層化した言語モデルを用い, MCMC 法と動的計画法により, あらゆる言語に隠れた「単語」を文字列から発見する言語モデルおよび形態素解析器を提案した.

提案法は識別モデルにおける CRF のような前向き–後ろ向きアルゴリズムの教師なし学習版ともみることができ, CRF+HMM による半教師あり品詞タグ付け [24] のように, 識別学習との融合の基盤を与えると考えられる. 一方で, より高度な単語モデルや隠れ状態を用いるなど, 言語モデル自体の高度化による高精度化も行っていきたい.

謝辞

本研究を行う動機付けとなった Vikash Mansinghka 氏 (MIT), 実装に関して有益なアドバイスをいただいた高林哲氏 (Google), 実験データの詳細を教えていただいた松原勇介氏 (東大) に感謝します.

参考文献

[1] 工藤拓, 山本薫, 松本裕治. Conditional Random Fields を用いた日本語形態素解析. **情報処理学会研究報告** *NL-161*, pages 89–96, 2004.
[2] Taku Kudo. MeCab: Yet Another Part-of-Speech and Morphological Analyzer. http://mecab.sourceforge.net/.
[3] 永田昌明. 単語出現頻度の期待値に基づくテキストからの語彙獲得. **情報処理学会論文誌**, 40(9):3373–3386, 1999.
[4] Sharon Goldwater and Tom Griffiths. A Fully Bayesian Approach to Unsupervised Part-of-Speech Tagging. In *Proceedings of ACL 2007*, pages 744–751, 2007.
[5] 山本博文, 菊井玄一郎. 教師なし学習による文の分割. In **言語処理学会第** *8* **回年次大会発表論文集** *(NLP2002)*, pages 579–582, 2002.
[6] 工藤拓. 形態素周辺確率を用いた分かち書きの一般化とその応用. In **言語処理学会全国大会論文集** *NLP-2005*, 2005.
[7] 中川哲治, 松本裕治. 単語レベルと文字レベルの情報を用いた中国語・日本語単語分割. **情報処理学会論文誌**, 46(11):2714–2727, 2005.
[8] Kevin Murphy. Hidden semi-Markov models (segment models), 2002. http://www.cs.ubc.ca/˜murphyk/Papers/segment.pdf.
[9] 松原勇介, 秋葉友良, 辻井潤一. 最小記述長原理に基づいた日本語話し言葉の単語分割. In **言語処理学会第** *13* **回年次大会発表論文集** *(NLP2007)*, 2007.
[10] Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. Contextual Dependencies in Unsupervised Word Segmentation. In *Proceedings of ACL/COLING 2006*, pages 673–680, 2006.
[11] Yee Whye Teh. A Bayesian Interpretation of Interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, NUS, 2006.
[12] Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proceedings of ICASSP*, volume 1, pages 181–184, 1995.
[13] 持橋大地, 隅田英一郎. Pitman-Yor 過程に基づく可変長 n-gram 言語モデル. **情報処理学会研究報告** *2007-NL-178*, pages 63–70, 2007.
[14] Abel Rodriguez, David Dunson, and Alan Gelfand. The Nested Dirichlet Process. *Journal of the American Statistical Association*, 103:1131–1154, 2008.
[15] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall / CRC, 1996.
[16] Steven L. Scott. Bayesian Methods for Hidden Markov Models. *Journal of the American Statistical Association*, 97:337–351, 2002.
[17] Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Bayesian Inference for PCFGs via Markov Chain Monte Carlo. In *Proceedings of HLT/NAACL 2007*, pages 139–146, 2007.
[18] Arnaud Doucet, Christophe Andrieu, and Roman Holenstein. Particle Markov Chain Monte Carlo. *in submission*, 2009.
[19] Yang He. Extended Viterbi algorithm for second order hidden Markov process. In *Proceedings of ICPR 1988*, pages 718–720, 1988.
[20] ICU: International Components for Unicode. http://site.icu-project.org/.
[21] Hai Zhao and Chunyu Kit. An Empirical Comparison of Goodness Measures for Unsupervised Chinese Word Segmentation with a Unified Framework. In *Proceedings of IJCNLP 2008*, 2008.
[22] Tom Emerson. SIGHAN Bakeoff 2005, 2005. http://www.sighan.org/bakeoff2005/.
[23] 国立国語研究所. 日本語話し言葉コーパス, 2008. http://www.kokken.go.jp/katsudo/seika/corpus/.
[24] Jun Suzuki, Akinori Fujino, and Hideki Isozaki. Semi-Supervised Structured Output Learning Based on a Hybrid Generative and Discriminative Approach. In *Proceedings of EMNLP-CoNLL 2007*, pages 791–800, 2007.