混成型別サンプリングを用いた名詞句分割

村脇 有吾　　黒橋 禎夫
京都大学大学院情報学研究科
murawaki@nlp.kuee.kyoto-u.ac.jp kuro@i.kyoto-u.ac.jp

1　はじめに

日本語の形態素解析は実用的な精度を達成しているが、辞書にない形態素 (未知語) の解析を誤りやすいことが知られており、高被覆な辞書の構築が欠かせない。辞書構築にあたっては、分かち書きしない日本語では、形態素の分割基準の設定が必要である。この分割基準の不一致から、人間向けの辞書や事典などの既存の外部語彙資源を形態素解析に利用しにくい。もし、基準の不一致を無視して、外部語彙資源の語彙をそのまま辞書登録すれば、例えば情報検索における再現率の低下など、様々な不都合が生じる。

外部語彙資源の中でも、特に事典は、解析用辞書が網羅していない語彙を数多く含み重要である。事典の語彙は大半が名詞句であり、複数の形態素からなる。

本稿では、形態素解析用辞書への利用を目的として、事典に現れる名詞句を自動分割する。分割の手がかりとして、名詞句に関連するテキストを利用する。本稿における仮定は以下の通りである。ある名詞句がもし複数形態素からなるなら、そうした構成要素がテキスト中に自由に、あるいは他の名詞句の一部として出現するはずである。

分割には頻度に基づく統計的単語分割の手法を採用する。具体的には、階層 Dirichlet 過程に基づくバイグラムモデル [2]、推論手続きとして型別サンプリング (type-based sampling) [3] を用いる。バイグラムモデルは、頻出形態素列を1形態素とみなしやすいというユニグラムモデルの問題を軽減する。型別サンプリングには、局所最適から一気に脱出できるという利点がある。しかし、型別サンプリングはユニグラムモデル向けであり、スパース性と潜在割り当てへの依存が原因で、バイグラムモデルへの適用が難しい。

本稿では、型別サンプリングを拡張し、混成型別サンプリング (hybrid type-based sampling) を提案する。これは、Gibbs サンプリングに Metropolis-Hastings アルゴリズムを組み込む。スパース性の問題は、ユニグラム水準の同時サンプリングにより回避する。また、取りえるすべての組み合わせの同時確率を計算する代わりに、現在と提案状態の確率のみを比較する。これにより、型別サンプリングの効率性を維持しつつ、潜在割り当てへの依存からくる計算の煩雑さを軽減する。実験により、形態素解析器が与える初期分割を提案手法が効率良く修正することが示された。

2　名詞句分割タスク

本稿の目的は、事典に含まれる語彙、特に見出し語を形態素解析に利用することである。そのためには、(1) 各語を基準に従って形態素に分割し、(2) 各形態素に品詞を割り当て、(3) 形態素解析の辞書に組み込む必要がある。3については、形態素解析器 JUMAN は既に連語を扱う機能を持っており、2の品詞割り当ては今後の課題とする。本稿では分割のみに取り組む。

事典の見出し語の大半は名詞句であり、一つ以上の形態素からなる。例えば、見出し語「常山城」の場合、「常山」と「城」への分割が目標となる。各見出し語にはテキストが関連付けられている。こうしたテキストを分割に利用する。本稿では、ある名詞句が複数形態素からなるなら、そうした構成要素がテキスト中に自由に (例えば「常山にあった」)、あるいは他の名詞句の一部として (例えば「常山駅」) 現れると仮定する。なお、構成要素自体が未知語である場合も少なくないため、テキスト中の構成要素は形態素解析結果からは自明には同定できない。

その他に利用できる資源として、形態素解析器、その辞書、およびタグ付きコーパスがある。形態素解析器は未知語を誤解析しやすいが、テキスト全体の解析精度は高い。そこで、形態素解析結果を初期状態とし、これを効率よく修正する手法を探し求める。

3　統計的単語分割

3.1　バイグラム言語モデル

形態素解析器が与える初期分割の修正に、ノンパラメトリック Bayes による統計的単語分割の手法を用いる。一番単純なモデルは、各形態素を独立に生成するユニグラムモデルであり、Dirichlet 過程により実現される。しかし、ユニグラムモデルは大雑把な近似であ

り、高頻度な形態素列を 1 形態素とみなしやすいという欠点が知られている [2]。予備実験により、本稿が扱う名詞句について、この現象の頻発を確認した。

そこで、形態素同士の連接を考慮するバイグラムモデルを用いる。バイグラムモデルは階層 Dirichlet 過程により実現されるが、中華料理店過程により解釈できる [2]。いま、$i-1$ 個の形態素 $\mathbf{w_{-i}} = w_1, \cdots, w_{i-1}$ を観測し、各々にテーブル割り当て $\mathbf{z_{-i}} = z_1, \cdots, z_{i-1}$ が与えられたとき、w_i の生成確率は以下で与えられる。

$$P_2(w_i|\mathbf{h_{-i}}) = \frac{n^{\mathbf{h_{-i}}}_{(w_{i-1},w_i)} + \alpha_1 P_1(w_i|\mathbf{h_{-i}})}{n^{\mathbf{h_{-i}}}_{(w_{i-1},*)} + \alpha_1} \quad (1)$$

$$P_1(w_i|\mathbf{h_{-i}}) = \frac{t^{\mathbf{h_{-i}}}_{w_i} + \alpha_0 P_0(w_i)}{t^{\mathbf{h_{-i}}}_{*} + \alpha_0}$$

ここで、$\mathbf{h_{-i}} = (\mathbf{w_{-i}}, \mathbf{z_{-i}})$、$t^{\mathbf{h_{-i}}}_{w_i}$ は w_i の形態素が着席するテーブル数、$t^{\mathbf{h_{-i}}}_{*}$ はテーブルの総数、α_0 と α_1 は Dirichlet 過程のハイパーパラメータ、P_0 は任意の文字列に対して適当な確率を返すゼログラム確率である。

生成確率の計算には $\mathbf{h_{-i}}$ が必要となる。交換可能性を用いると、テーブル割り当て $\mathbf{z_{-i}}$ を直接管理する必要はないが、各 w_i について、テーブルごとの客の数 (**ヒストグラム**) を管理する必要がある [1]。

3.2 型別サンプリング

型別サンプリング [3] は、Gibbs サンプリングの一手法である。崩壊 Gibbs サンプリングが、一つの変数をサンプリングするのに対して、型別サンプリングは同じ型に属す複数の変数を同時にサンプリングする。

統計的単語分割の場合、文字間の各点に隠れ変数を設定する [2]。変数は 2 値であり、その点が境界か否かを表す。ユニグラムモデルの場合、注目する点の値は、テキスト中の局所区間が 1 形態素 w_1 (境界でない) か 2 形態素 $w_2 w_3$ ($w_1 = w_2.w_3$) (境界) かを決める。崩壊サンプリングでは、各点をサンプリングする。すなわち、テキストのその他の部分の現在の状態を所与として、境界か否かの 2 通りの条件付き確率を求め、その確率に従って変数の新たな値を決める。

型別サンプリングにおける型は、ユニグラムモデルの場合次の通りである。ある二つの点は、対応する局所区間が w_1 あるいは $w_2 w_3$ からなる場合に、同じ型に属す。型別サンプリングは、同じ型に属す複数の点を、交換可能性を利用して効率良く同時にサンプリングする。すなわち、同じ型に属す n 個の点について、まず境界の数 m $(0 \leq m \leq n)$ をサンプリングし、次に m 個の境界を n 個の点に無作為に配置する。

本稿が着目する型別サンプリングの特性は、局所最適を一気に脱出し得ることである。形態素解析は所与のパラメータを用いた決定的解析であり、未知語に関する誤りも含めて、一貫した分割を返す。形態素解析結果は、直感的には、単語分割において、大域解からは遠くないものの、局所解である。したがって、崩壊サンプリングではなかなか初期状態を脱出できない。これに対し、型別サンプリングは、同じ型を同時にサンプリングするため、局所解から効率的に脱出し得る。

3.3 バイグラムモデルへの適用時の問題

型別サンプリングはユニグラムモデル向けであり、以下の二つの理由から、バイグラムへの適用が難しい。第一の問題はスパース性である。バイグラムモデルのサンプリングでは、隣接形態素も考慮しなければならない。つまり、型は、$w_l w_1 w_r$ あるいは $w_l w_2 w_3 w_r$ という 3、4 形態素の連続からなり、型あたりの点の数がユニグラムモデルに比べて極端に少ない。また、隣接形態素が異なる $w_{l'} w_1 w_{r'}$ や $w_{l'} w_2 w_3 w_{r'}$ は、密接な依存関係にあるにも関わらず同時サンプリングの対象とならない。したがって、推論の効率が悪い。

第二の問題はヒストグラム管理の煩雑さである。ユニグラムモデルの同時分布は解析形 [3] が知られているが、バイグラムモデルでは、Dirichlet 過程の基底測度 P_1 が変化するため、解析形が明らかでない。そのため、同時分布の計算はシミュレーションにより行う。すなわち、新たな形態素を観測するたびに、モデルカウントを更新して (1) を求めると、その積が同時分布となる。n 個の点に対して、$n+1$ 通りの割り当てをシミュレートする必要があり、非常に煩雑である。

3.4 混成型別サンプリング

上記の問題を解決するために、本稿では混成型別サンプリングを提案する。これは Metropolis-Hastings アルゴリズムを組み込む。Metropolis-Hastings は確率分布 P からのサンプルをマルコフ連鎖を用いて得る手法である。現在の状態を h としたとき、まずある提案分布 $Q(h'; h)$ に従って、次の状態 h' を提案する。次に、この提案を以下の確率で採択する。

$$\min\left\{\frac{P(h')Q(h;h')}{P(h)Q(h';h)}, 1\right\} \quad (2)$$

採択された場合は、提案状態を次の状態とし、棄却された場合は、現在の状態を次の状態とする。Metropolis-Hastings は P から直接サンプリングするのが難しい場合に有用である。

混成型別サンプリングは、型別サンプリングの中で Metropolis-Hastings を用いる。すなわち、$n+1$ 通りの確率を求める代わりに、現在と提案状態の 2 通りの確率のみを比較する。また、バイグラムではなく、ユ

ニグラム水準の型を同時にサンプリングする。つまり、同じ型に属す点は、局所区間が w_1 あるいは w_2w_3 からなり、隣接形態素を考慮しない。ユニグラム水準で同じ型に属す点は、バイグラムモデルでは交換可能でない。したがって、具体的に境界を点に配置した割り当てを提案状態として用いる。

提案状態は次の 2 段階で決める。同じ型に属す点の数を n、そのうち境界の数を m とすると、まず提案状態の境界数 m' をある確率分布 $f_n(m';m)$ に従って決める。次に、無作為に m' 個の境界を配置する。これにより確率質量は ${}_nC_{m'}$ 通りの割り当てに均等に分割される。したがって、提案分布は以下の通りである。

$$Q(h';h) = \frac{f_n(m';m)}{{}_nC_{m'}} \tag{3}$$

$f_n(m';m)$ は、図 1 のように、ベータ分布 ($\alpha = \beta < 1$) と m を平均とする正規分布を混合、離散化して作る。前者は極端な値を、後者は近傍を好む。

無作為な境界配置は、理想的な割り当てが一貫せず、隣接形態素に依存する場合は、不都合かもしれない。現在のバイグラム確率に応じた割り当て確率を設計することにより、この問題は回避できるかもしれない。

まとめると、各型のサンプリングは以下の手順で行う。

1. ユニグラム水準の型を共有する n 個の点を集める。
2. (3) に従い提案状態を決める。以下では、境界が入れ替わる点のみを扱う。現在と提案状態で変化がない点は両者の尤度比に影響しないからである。
3. 現在の状態の確率を求める。この確率は、N グラムを一つ一つ取り除き、モデルカウントを更新しながら、(1) を繰り返し計算すると得られる。
4. 同様にして、N グラムを一つ一つ加えながら、提案状態の確率を求める。
5. (2) に従い、提案を採択するか決める。採択されれば割り当てを確定させ、棄却されれば現在の割り当てに戻す。

なお、本稿ではスキップ近似 [3] を実装し、一つの型は 1 反復で 1 回サンプリングする。多数の点の同時サンプリングは計算コストが高い一方で、受理割合が非常に低いからである。

4 実験

4.1 設定

評価実験にウィキペディア日本語版を用いた。見出し語を分割対象の名詞句とし、見出し語と記事本文を対象に単語分割を行った。見出し語は、曖昧さ回避用の括弧を除いて正規化した。分割の初期状態は形態素

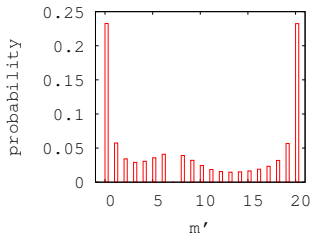

図 1: $f_{20}(m';7)$ の場合の提案境界数の分布

解析器 JUMAN の解析結果とした。ただし、ウィキのマークアップを確からしい境界制約として利用した。

記事のうち、(1) 見出し語が 2 文字以上で日本語を含み、(2) 本文文字数が 1,000 以上で、(3) 見出し語が本文に 5 回以上出現するもののみを対象とした。約 14%の記事がこれらの基準を満たした。うち 500 記事の見出し語を人手で分割して正解データとした。2 人の作業者の一致率は $\kappa = 0.95$ (文字単位) であった。

言語モデルとしてユニグラムモデルとバイグラムモデルを用いた。推論手法として、崩壊サンプリング (**CL**)、型別サンプリング (**TB**) および混成型別サンプリング (**HTB**) を比較した。また、京都テキストコーパスから得られるカウントを加えたモデル (**REF**)、初期分割を無作為に行った場合 (**RAND**) も比較に用いた。ベースラインは JUMAN の解析結果とした。

Dirichlet 過程のハイパーパラメータ α_0 と α_1 として以下の値の組み合わせを試した。

α_0: 10, 50, 100, 500, 1,000, 5,000, 10,000 and 50,000

α_1: 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100 and 500

ゼログラム確率 P_0 は文字バイグラムと文字長 Poisson 分布から構成する。各パラメータはあらかじめ京都テキストコーパスから学習した。各モデル設定について、バーンイン 10 反復の後、10 反復からサンプルを集めた。10 サンプルの**平均**精度、および最頻出分割の精度 (**多数決**) を報告する。ハイパーパラメータ設定は、F 値に基づき**最善**の結果と**中央値**を報告する。

4.2 結果

表 1 に見出し語の分割精度を示す。ベースラインが新聞について報告された精度に比べて著しく低いことに気づく。ウィキペディアの見出し語は高頻度で未知語を含んでいるからである。最善の設定では、混成型別サンプリングがベースラインを有意に上回り、最高の F 値を達成している。ただし、α_1 の値に敏感であり、中央値の設定では振るわない。

タグ付きコーパスのカウントを加えると型別サンプリングで精度が大幅に悪化した。記事本文よりもコーパスがオーダレベルで大きいことが原因と考えられ

表 1: 見出し語分割結果 (F 値 (適合率/再現率))

モデル	最善: 多数決		最善: 平均		中央値: 多数決	
unigram + CL	80.94	(77.41/84.80)	80.59	(77.03/84.50)	80.12	(75.87/84.89)
unigram + TB	68.84	(77.61/61.85)	68.30	(77.02/61.35)	68.14	(76.87/61.19)
bigram + CL	80.23	(75.76/85.27)	80.04	(75.78/84.80)	79.96	(75.51/84.65)
bigram + HTB	**83.81**	(82.39/85.27)**	**83.37**	(82.15/84.64)	72.03	(66.77/78.19)
unigram + CL + REF	82.51	(79.51/**85.74**)**	82.35	(79.37/**85.57**)	**81.38**	(**77.68**/**85.46**)**
unigram + TB + REF	43.00	(55.18/35.22)	42.73	(54.92/34.97)	42.26	(54.38/34.56)
bigram + CL + REF	80.25	(75.86/85.17)	80.12	(75.72/85.06)	79.75	(75.27/84.80)
bigram + HTB + REF	69.41	(65.41/73.94)	69.30	(65.42/73.66)	68.13	(63.66/73.28)
unigram + TB + RAND	69.25	(77.61/62.51)	68.92	(77.24/62.21)	68.16	(76.05/61.76)
bigram + HTB + RAND	82.61	(**86.63**/78.94)	82.63	(**86.49**/79.09)	67.41	(61.81/74.13)
ベースライン	80.09	(75.80/84.89)				

** 統計的に有意な改善 ($p < 0.01$)。

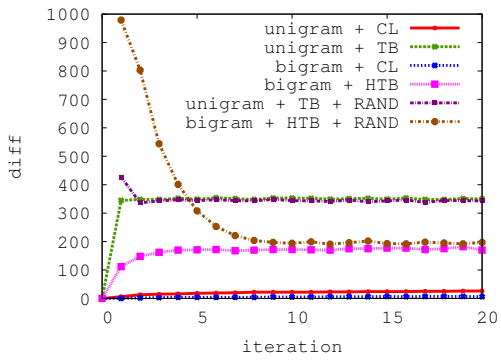

図 2: 推論過程でのベースラインからの差分

る。こうした状態では、カウントを直接混ぜるのではなく、モデルを二つに分けて線形補完を行う方が良いかもしれない。意外なことに崩壊サンプリングで精度が向上した。ただし、型別サンプリングの結果は、反復をさらに進めると精度が悪化することを示唆する。

適合率と再現率の比較から、ベースラインが過分割傾向にあることがわかる。ユニグラムモデルは逆に分割不足である。これらに比べてバイグラムモデルは適度な粒度で分割を行えている。

図 2 に推論過程での各モデルの初期状態 (ベースライン) からの差分を示す。ここで、差分とは、初期状態とモデル出力の文字単位の異なり数である。崩壊サンプリングでは初期状態からほとんど動けていない。型別サンプリングでは大きく動いたが、ユニグラムモデルでは望まない方向に向かう。混成型別サンプリングは、型別サンプリングよりも遅いが、それでも数反復で収束している。また、形態素解析結果を初期状態とした方が、無作為な初期化の場合より収束が速い。

4.3 議論

混成型別サンプリングによる改善例には「宇部 + 興 + 産 ⇒ 宇部+ 興産」、「こな + みる + く ⇒ こなみるく」、「ランゲンブレッタハ ⇒ ランゲン + ブレッタハ」(ドイツの複合地名) などがある。「こなみるく」は「コナミ」と「粉ミルク」を掛けていると推測されるが、テキストから語源が分析できないため、1 形態素を正解とした。こうしたひらがな語を正しく認識することは、応用を考えると非常に重要である。誤分割すると機能語やその他の基本語と誤認識され、例えば、構文解析に深刻な悪影響を及ぼすからである。

誤りの主な原因として、与えられたテキストで本稿の仮定が成り立っていないことが挙げられる。各構成要素が独立に振る舞う様子がテキスト中で十分に観測できなければ、正しく分割できない。特に人名に対して体系的なバイアスが確認される。氏名が一度紹介された後は、氏または名の一方のみが参照されがちであり、参照されない要素の学習に失敗しやすい。

5 結論

本稿では、統計的単語分割におけるバイグラムモデルの効率的な推論手続きとして、混成型別サンプリングを提案した。また、提案手法を事典の見出し語に適用し、名詞句の自動分割を行った。

提案手法の他の応用先として、テキストからの未知語の自動獲得 [4] が考えられる。このタスクでは、複合名詞の分割が課題となっている。日本語の複合名詞は構成要素がマーカなしに接続して形成されるため、形態論的振る舞いに基づく同定手法が機能しないからである。今後、獲得された複合名詞候補に対して、獲得に用いたテキストに提案手法を適用することにより、形態素への分割を行いたい。

参考文献

[1] Phil Blunsom, Trevor Cohn, Sharon Goldwater, and Mark Johnson. A note on the implementation of hierarchical Dirichlet processes. In *Proc. of ACL-IJCNLP 2009: Short Papers*, pp. 337–340, 2009.

[2] Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, Vol. 112, No. 1, pp. 21–54, 2009.

[3] Percy Liang, Michael I. Jordan, and Dan Klein. Type-based MCMC. In *Proc. of NAACL 2010*, pp. 573–581, 2010.

[4] Yugo Murawaki and Sadao Kurohashi. Online acquisition of Japanese unknown morphemes using morphological constraints. In *Proc. of EMNLP 2008*, pp. 429–437, 2008.