ベイズ学習によるカタカナ複合語の分割

村脇 有吾 *‡　　岸本 侑也 †　　黒橋 禎夫 ‡
* 京都大学学術情報メディアセンター　† 京都大学工学部　‡ 京都大学大学院情報学研究科
*‡{murawaki, kuro}@i.kyoto-u.ac.jp　†kishimoto@nlp.ist.i.kyoto-u.ac.jp

1　はじめに

分かち書きしない日本語においては、形態素解析は自然言語処理の第一歩であり、様々な応用処理を実現する上で、高い精度の達成が欠かせない。日本語形態素解析において、現在主流となっている手法は辞書に基づくものであり、テキスト中に出現する形態素があらかじめ辞書登録されていることを前提とする。この手法には、辞書にない形態素 (未知語) の解析を誤りやすいという問題がある。

こうしたテキスト中の未知語を認識することを考える。そのためには、形態素とはどのようなものか、具体的には、形態素内部がどのような構成をとり、また、形態素外部とどのような関係を持つかについての知識が必要となる。まず、内部構成について考えると、形態素を構成する文字列はある程度の規則性を持つものの、基本的には恣意的である。小規模なタグ付きコーパスに出現する語彙から、未知語を直接的に認識可能とする知識を獲得することは期待できない。一方、形態素外部との関係については、日本語が膠着語であることから、比較的少数の形態論的制約を利用して未知語の同定が行えることが知られている [4]。ただし、形態論に基づく手法には、複合名詞の扱いが課題として残っている。複合名詞は、構成形態素が文法的マーカなしに接続して形成されるため、形態論に基づく手法では構成形態素に分割できない。

本稿では、複合名詞に対して、構成形態素の出現頻度に基づく統計的分割を行う。複合名詞の中でも、特にカタカナで表記された複合名詞 (カタカナ複合語) を対象とする。理由は二つあり、一つは、カタカナ複合語は生産性が高く、形態素辞書に登録されていない構成形態素を含むことが少なくないことである [2]。もう一つは、カタカナ複合語は、内部構成がわからないだけで、複合語そのものは字種によって比較的簡単に切り出せるため、様々な発展的手法を試すテストベッドとして使えるからである。

本稿では、ノンパラメットリック・ベイズに基づく生成モデルを用いてカタカナ複合語を分割する。こうした生成モデルの欠点に、人間が持つ分割基準を直接的にモデルに反映させにくいというものがある。そこで、様々な手がかりを追加して、分割精度の向上をはかる。具体的には、様々なゼログラム・モデル、分割済みカウント、カタカナ複合語と共起する用言、カタカナ複合語の言い換え表現を利用し、その効果を検証する。

2　実験設定

分割対象となるカタカナ複合語のリストは、ウェブコーパスの一部から自動構築する。以後これを生コーパスと呼ぶ。カタカナ複合語は、字種の違いによって、複合語外部との境界が明瞭であり、自動解析だけで高精度に抽出できる。ただし、複合語内部の分割は誤りを含むことが少なくない。まず、形態素解析器 JUMAN 7.0[1] を用いて生コーパスの各文を形態素に分割し、構文解析器 KNP 3.0[2]を用いて形態素列を文節にチャンクする。次に、各文節を調べ、付属語列を除いた内容語列がカタカナ名詞列からなる場合に、そのカタカナ名詞列を抽出する。実験では、約 200 万文から、頻度約 89 万、異なり数約 11 万のカタカナ複合語を得た。

得られたカタカナ複合語の一部、967 個に対して、人手で正解分割を付与した。複合語あたりの形態素数は平均 1.77 であった。

分割結果の評価は、形態素単位の適合率、再現率、F 値によって行う。また、複合語単位の分割の完全一致率も評価する。表 1 にベースラインの精度を示す。

必要に応じて京都大学テキストコーパス[3]も用いる。このコーパスは、新聞記事約 4 万文に対して、形態素・構文情報を付与したものである。ただし、分割のアノテーションは主眼ではなく、カタカナ複合語の分割には誤りが少なからず見られたため、カタカナ複合語の分割は人手で修正した。修正済みのコーパスは今

[1] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[2] http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP
[3] http://nlp.ist.i.kyoto-u.ac.jp/index.php?京都大学テキストコーパス

表 1: ベースライン分割

ベースライン	F 値 (適合率 / 再現率)	完全一致
JUMAN	.815 (.856/.777)	.810
分割なし	.296 (.410/.231)	.410

後公開を予定している。以後これを分割済みコーパスと呼ぶ。

分割済みコーパスの各文節からカタカナ複合語を抽出した。その結果、頻度 13,822、異なり数 4,004 のカタカナ複合語を得た。複合語あたりの形態素数は平均 1.15 であり、生コーパスと比べて小さかった。

3 分割モデル

本稿では、複合名詞に対して、構成形態素の出現頻度に基づく統計的分割を試みる。基本的なアイデアは、テキスト中に繰り返し出現する文字列が形態素らしいというものである。通常の単語分割では各文を形態素に分割するのに対し、ここでは各複合語を形態素に分割する。

具体的な分割モデルとして、ノンパラメトリック・ベイズによる生成モデルを採用する。ノンパラメトリック・モデルでは、パラメトリック・モデルと異なり、語彙数を事前に決める必要がなく、データから自動的に学習される。ノンパラメトリック・モデルの中でも、特に Pitman-Yor 過程 [6] を用いる。Pitman-Yor 過程は、Dirichlet 過程にディスカウント係数を追加して一般化したものであり、言語が持つベキ法則、つまり、比較的少数の形態素が頻度の大半を占める一方、低頻度の形態素が数多く存在するという現象を捉えていることが知られている。

まず単純なユニグラム・モデルを考える。ユニグラム・モデルでは、各形態素が前後の文脈と独立に生成される。Pitman-Yor 過程に基づくユニグラム・モデルでは、形態素 w_i は以下のように生成される。

$$\begin{aligned} G|d_U,\theta_U P_0 &\sim \mathrm{PY}(d_U,\theta_U,P_0) \\ w_i|G &\sim G \end{aligned}$$

ここで、PY はディスカウント係数 d_U、強さ係数 θ_U を持つ Pitman-Yor 過程である。$d_U=0$ のとき、Dirichlet 過程 $DP(\theta_U,P_0)$ と一致する。P_0 はゼログラム分布であり、任意のカタカナ文字列に対して適当な確率を与える (4 節参照)。

実際には G を積分消去して、モデルを中華料理店過程により表現する。いま、形態素列 $w_1,\cdots,w_n$ を順に生成するとして、$w_1,\cdots,w_{n-1}$ を観測した後、w_n を生成することを考える。中華料理店過程において、各形態素は客であり、各客は卓に着く。卓にはラベルが貼ってあり、そのラベルが客の形態素ラベルとなる。いま、ある卓 k について、ラベルを l_k、客の数を n_k とする。また、客のいる卓の総数を K とする。次の客 w_n は n_k-d_U に比例する確率で、卓 k に座り、$w_n=l_k$ となる。あるいは、θ_U+d_UK に比例する確率で、新たな卓に座り、そのラベルは P_0 から生成する。

ユニグラム・モデルには、文に関して、高頻度のコロケーションを 1 形態素とみなしがちという傾向が報告されている [1]。そこで、隣接する形態素を考慮するバイグラム・モデルを次に考える。バイグラム・モデルは階層 Pitman-Yor 過程により実現される。

$$\begin{aligned} G|d_U,\theta_U P_0 &\sim \mathrm{PY}(d_U,\theta_U,P_0) \\ H_l|d_B,\theta_B G &\sim \mathrm{PY}(d_B,\theta_B,G) \\ w_i|w_{i-1}=l,H_l &\sim H_l \end{aligned}$$

w_i の生成確率は直前の形態素 w_{i-1} に依存する。ただし、先頭と末尾は特殊であり、$w_1,\cdots,w_n$ に対して、境界 \$ から w_1 の生成、w_n から \$ の生成も行う。バイグラム・モデルでは、コロケーション w_iw_{i+1} が高頻度であっても、w_i と w_{i+1} もそれぞれ高頻度であれば、分割されやすい。

バイグラム・モデルでは、Pitman-Yor 過程が 2 階層となっており、バイグラム分布はユニグラム分布によりスムージングされる。また、バイグラム・モデルも、G や H_l を積分消去した上で、中華料理店過程により表現できる。

こうした通常の前向きバイグラム・モデルに加えて、後ろ向きバイグラム・モデルも試す。後ろ向きモデルでは、w_i は直後の形態素 w_{i+1} から生成される。

推論は、Gibbs サンプリングにより近似的に行う。分割なしのカタカナ複合語のリストを $\boldsymbol{C}$、分割結果を $\boldsymbol{W}$、卓割り当てを $\boldsymbol{Z}$ としたとき、ユニグラム・モデルの場合、$P(\boldsymbol{W},\boldsymbol{Z}|\boldsymbol{C},d_U,\theta_U,P_0)$ からのサンプル $\boldsymbol{W}_i,\boldsymbol{Z}_i$ を Gibbs サンプリングにより得る (バイグラム・モデルについても同様)。手続き的には、まず初期値 $\boldsymbol{W}_0,\boldsymbol{Z}_0$ を適当に決め、その後、局所的かつ確率的に $\boldsymbol{W}_i,\boldsymbol{Z}_i$ を変更して $\boldsymbol{W}_{i+1},\boldsymbol{Z}_{i+1}$ を得るという操作を繰り返す。確率的な $\boldsymbol{W}_i,\boldsymbol{Z}_i$ の変更方法については、文 (ここでは複合語) 単位のブロック・サンプリング [3] が収束の速さで知られており、これを使う。

実際には、ハイパーパラメータもデータから自動推定する [6]。$m\in\{U,B\}$ について $d_m\sim\mathrm{Beta}(a_m,b_m)$、$\theta_m\sim\mathrm{Gamma}(\alpha_m,\beta_m)$ と事前分布を置き、事後分布からのサンプルを得る。

実験では、初期分割としては、分割なし、つまり各

複合語を 1 形態素と見なした状態を用いる (7 節で用いる分割制約を除く)。データ全体の走査を 50 回繰り返し、最後に得られた分割を用いる。これを乱数シードを変えて 10 回行い、精度としては、10 個の分割結果のマイクロ平均を報告する。

4 ゼログラムの効果

ゼログラム分布 P_0 は任意のカタカナ文字列に対して適当な確率を与える。その設計が、間接的ながら、分割精度に影響を与えると予想される。そこで、以下の 4 種類のモデルを試す (括弧内は学習元データ)。

1. 文字ユニグラム (生コーパス)
2. 文字バイグラム (分割済みコーパス)
3. 文字バイグラム (生コーパス)
4. 文字バイグラム (分割済みコーパスと生コーパス)

一番単純な文字ユニグラム・モデル [1] では、形態素を 1 文字ごとに分解して確率を得る。

$$P_0(w = c_1, \cdots, c_k) = p_\$(1 - p_\$)^{k-1} \prod_{i=1}^{k} p(c_i)$$

ここで $p_\$$ は形態素打ち切り確率。分割済みコーパスにおけるカタカナ形態素の平均長がおよそ 4 であることに基づき、$p_\$ = 0.25$ とする。文字ユニグラム・モデルの学習は、カタカナ文字をカウントとするだけで行え、分割に依存しない。

カタカナ形態素の内部構成について、「ン」や「ッ」からはじまることはまずないといった傾向が見られる。こうした知識をモデルに反映させるために、文字バイグラム・モデルを用いる。文字バイグラム・モデルは形態素列と同様に、階層 Pitman-Yor 過程により実現される。このモデルの学習には、形態素に分割されたカタカナ複合語のリストが必要となる。そこで、まず分割済みコーパスから学習するという手法を試す。得られた学習結果はカタカナ分割時に固定で用いる。

次に、モデルを形態素列と文字列の入れ子モデル [3] とみなし、分割と同時にゼログラム・モデルを学習する。入れ子モデルとしては、生コーパスのみから学習する場合と、分割済みコーパスから得られるカウントを生コーパスに混ぜ合わせる場合を試す。

結果を表 2 に示す。文字ユニグラム・モデルよりも文字バイグラム・モデルが良い精度を出す傾向が見られる。また、分割済みコーパスを固定で用いるより、生コーパスから学習する方が良い精度が得られた。推測される原因は、分割済みコーパスの小ささ、および分割済み (新聞記事) と生 (ウェブ) コーパスの分野の違いである。分割済みコーパスに出現しない「ヱヴ」や「ヲタ」が生コーパスには頻出する。一方、分割済みコーパスに頻出する「ゥダ」や「ヌイ」(それぞれ「ドゥダエフ」と「グロズヌイ」に由来) は生コーパスにはそれほど出現しない。また、「ヂ」、「ヰ」、「ヱ」のように規範的な表記ではあまり使われない文字が生コーパスにはある程度出現する。

いずれのモデルも JUMAN に比べて分割精度が大きく下回る。また、いずれのモデルも過小分割傾向にあるが、予想に反して、(形態素) ユニグラム・モデルよりも (形態素) バイグラム・モデルの方が過小分割を起こしている。以降ではゼログラム・モデルとして入れ子モデルを用いる。

5 分割済みカウントの効果

ゼログラムよりもより直接的な形で分割済みコーパスを利用する。分割済みコーパスから得られる形態素のカウントを混ぜ合わせることにより、既知の形態素が再現されることを期待される。

結果を表 3 に示す。分割済みカウントを足しても精度が大きく向上しないどころか、ユニグラム・モデルではかえって悪化した。

6 共起する用言の効果

バイグラム・モデルに対して、カタカナ複合語と共起する用言を用いる。例えば、カタカナ複合語「スパイスライス」が「スパイ」と「スライス」からなるか、「スパイス」と「ライス」からなるかは、頻度だけで決めるのは難しそうだが、「スパイスライス」と共起する用言に「〜を食べる」などがあれば、「ライス」を切り出すのに役立つと期待される。以下では、用言 (食べる) と格助詞 (を) をあわせて用言とよぶ。

共起する用言の抽出には、文法的制約を利用し、構文上の曖昧性がない係り受け関係のみを用いる。ただし、分割済みコーパスについては、正解係り受け関係を用いる。用言が得られたカタカナ複合語について、終端境界 $ を用言で置き換える。生コーパスについて、9.9%のカタカナ複合語に対して用言が得られた。用言の異なり数は約 1.6 万であった。

結果を表 4 に示す。末尾一致は末尾の形態素の正解率を表す。前向きバイグラムにめだった変化はないが、後ろ向きバイグラムでは、用言により有意 ($p < .01$)[4]に精度が改善した。複合語の場合も、依存構造木の生成と同様に、後から前に生成するのが良いのかもしれない。分割済みコーパスはかえって精度を悪化させた。

[4]形態素列を文字 BI ラベルに変換し、ラベルの正誤に対して McNemar 検定を行った。

表 2: ゼログラムの効果

ゼログラム	ユニグラム F 値 (適合率 / 再現率)	ユニグラム 完全一致	前向きバイグラム F 値 (適合率 / 再現率)	前向きバイグラム 完全一致	後ろ向きバイグラム F 値 (適合率 / 再現率)	後ろ向きバイグラム 完全一致
ユニ (生)	.619 (.622/.616)	.594	.568 (.574/.563)	.539	.560 (.566/.554)	.523
バイ (分)	.621 (.622/.620)	.596	.561 (.568/.555)	.534	.573 (.578/.567)	.542
バイ (生)	.635 (.646/.623)	.611	.573 (.585/.560)	.547	.578 (.593/.564)	.553
バイ (分 + 生)	.638 (.649/.628)	.612	.576 (.588/.564)	.544	.575 (.593/.558)	.554

表 3: 分割済みカウントの効果

モデル	ユニグラム F 値 (適合率 / 再現率)	ユニグラム 完全一致	前向きバイグラム F 値 (適合率 / 再現率)	前向きバイグラム 完全一致	後ろ向きバイグラム F 値 (適合率 / 再現率)	後ろ向きバイグラム 完全一致
分割済みあり	.626 (.637/.616)	.602	.576 (.589/.563)	.548	.592 (.606/.578)	.569

表 4: 共起する用言の効果

モデル	前向きバイグラム F 値 (適合率 / 再現率)	前向きバイグラム 完全一致	前向きバイグラム 末尾一致	後ろ向きバイグラム F 値 (適合率 / 再現率)	後ろ向きバイグラム 完全一致	後ろ向きバイグラム 末尾一致
用言なし	.573 (.585/.560)	.547	.602	.578 (.593/.564)	.553	.596
用言	.575 (.586/.564)	.543	.596	.605 (.614/.596)	.578	.626
用言 + 分割済み	.574 (.586/.563)	.545	.604	.601 (.612/.591)	.574	.618

表 5: 言い換え表現の効果

モデル	ユニグラム F 値 (適合率 / 再現率)	ユニグラム 完全一致	前向きバイグラム F 値 (適合率 / 再現率)	前向きバイグラム 完全一致	後ろ向きバイグラム F 値 (適合率 / 再現率)	後ろ向きバイグラム 完全一致
言換	.673 (.677/.670)	.647	.627 (.627/.626)	.599	.631 (.633/.628)	.600
言換 + 分済	.672 (.674/.670)	.645	.632 (.631/.632)	.599	.627 (.625/.629)	.598
言換 + 用言			.642 (.640/.643)	.613	.634 (.623/.646)	.610

7 言い換え表現の利用

言い換え表現 [2] は、カタカナ複合語分割への有効性が報告されている。例えば、テキスト中に「アウトドア・リュック」が出現すれば、カタカナ複合語「アウトドアリュック」を「アウトドア」と「リュック」に分割するのに役立つと期待される。言い換え規則としては従来研究 [2] に従い以下を用いる。

1. n_i ・ n_{i+1}
2. n_i (の | な) n_{i+1}
3. n_i (する | した) n_{i+1}
4. n_i (的 | 的な) n_{i+1}

ここで、n_i はカタカナ複合語を表す。カタカナ複合語のリストの構築時、言い換え規則がマッチしたとき、$n_i||n_{i+1}$ という疑似的なカタカナ複合語を抽出する。ここで || は、該当点でかならず分割するという制約を表し、この制約を満たすようにサンプリングを行う。生コーパスから抽出されたカタカナ複合語のうち、4.6%が言い換え表現だった。分割済みコーパスに対しても同じ処理を適用する。

結果を表 5 に示す。入力データ自体が異なるため、他の実験との比較は厳密ではないことに注意を要する。精度向上が期待されないユニグラム・モデルでも精度が向上しているが、バイグラム・モデルについて、ユニグラム・モデル以上の精度向上が得られた。言い換え表現による改善例を見ると、「ザ・ダムド」を手がかりに「ザダムド」を「ザ」と「ダムド」に分割するなど、過小分割を防ぐ効果が確認された。

8 おわりに

本稿では、ベイズ学習を用いたカタカナ複合語分割において、分割精度の向上をはかるために様々な手がかりを用い、その効果を検証した。言い換え表現は有効だったが、分割済みコーパスに目立った効果が認められず、そもそもいずれもベースライン精度を大きく下回るなど、残された課題は少なくない。

今回用いていない手がかりとしては、外来の複合語に関して、元言語の分かち書きが有効と期待される。多言語の同時分割 [5] が応用できるかもしれない。一方、非外来語のカタカナ表記もウェブテキストでは目立つ。カタカナと一般的な表記を関連付けるようなモデルが必要かもしれない。

参考文献

[1] 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.

[2] Nobuhiro Kaji and Masaru Kitsuregawa. Splitting noun compounds via monolingual and bilingual paraphrasing: A study on Japanese Katakana words. In *Proc. of EMNLP 2011*, pp. 959–969, 2011.

[3] Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proc. of ACL-IJCNLP 2009*, pp. 100–108, 2009.

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

[5] Benjamin Snyder and Regina Barzilay. Unsupervised multilingual learning for morphological segmentation. In *Proc. of ACL 2008*, pp. 737–745, 2008.

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