言い換えと逆翻字を用いた片仮名複合名詞の分割

鍛治　伸裕[†]・喜連川　優[†]

日本語を含めた多くの言語において，複合名詞内部の単語境界は空白で分かち書きされない．こうした複合名詞を構成語列へと分割する処理は，多くの自然言語処理の応用において重要な基礎技術となる．日本語の場合，片仮名語は生産性が高く未知語が多いことから，特に片仮名複合名詞の扱いが技術的な問題となる．この問題の解決を図るため，本論文は片仮名複合名詞の言い換えと逆翻字を分割処理に利用する方法を提案する．実験では，言い換えと逆翻字をラベルなしテキストから抽出し，その情報を利用することによって，分割精度が統計的に有意に向上することを確認した．

キーワード：言い換え，逆翻字，片仮名語，複合名詞分割，単語分割

Splitting Katakana Noun Compounds by Paraphrasing and Back-transliteration

Nobuhiro Kaji[†] and Masaru Kitsuregawa[†]

Word boundaries within noun compounds are not marked by white spaces in a number of languages including Japanese, and it is beneficial for various NLP applications to split such noun compounds. In the case of Japanese, noun compounds made up of katakana words are particularly difficult to split, because katakana words are highly productive and are often out-of-vocabulary. To overcome this difficulty, we propose using paraphrases and back-transliteration of katakana noun compounds for splitting them. Experiments demonstrated that splitting accuracy is improved with a statistical significance by extracting both paraphrases and back-transliterations from unlabeled textual data, and then using that information for constructing splitting models.

Key Words: *paraphrasing, back-transliteration, katakana words, noun compound splitting, word segmentation*

1　はじめに

1.1　片仮名語と複合名詞分割

外国語からの借用 (borrowing) は，日本語における代表的な語形成の 1 つとして知られている (Tsujimura 2006)．特に英語からの借用によって，新造語や専門用語など，多くの言葉が日々日本語に取り込まれている．そうした借用語は，主に片仮名を使って表記されることから片仮名

[†] 東京大学生産技術研究所, IIS, the University of Tokyo

語とも呼ばれる．日本語におけるもう 1 つの代表的な語形成として，単語の複合 (compounding) を挙げることができる (Tsujimura 2006)．日本語は複合語が豊富な言語として知られており，とりわけ複合名詞にその数が多い．これら 2 つの語形成は，日本語における片仮名複合語を非常に生産性の高いものとしている．

日本語を含めたアジアおよびヨーロッパ系言語においては，複合語を分かち書きせずに表記するものが多数存在する（ドイツ語，オランダ語，韓国語など）．そのような言語で記述されたテキストを処理対象とする場合，複合語を単語に分割する処理は，統計的機械翻訳，情報検索，略語認識などを実現する上で重要な基礎技術となる．例えば，統計的機械翻訳システムにおいては，複合語が構成語に分割されていれば，その複合語自体が翻訳表に登録されていなかったとしても，逐語的に翻訳を生成することが可能となる (Koehn and Knight 2003)．情報検索においては，複合語を適切に分割することによって検索精度が向上することが Braschler らの実験によって示されている (Braschler and Ripplinger 2004)．また，複合語内部の単語境界の情報は，その複合語の省略表現を生成または認識するための手がかりとして広く用いられている (Schwartz and Hearst 2003; Okazaki, Ananiadou, and Tsujii 2008)．

高い精度での複合語分割処理を実現するためには，言語資源を有効的に活用することが重要となる．例えば，Alfonseca ら (2008) は単語辞書を学習器の素性として利用しているが，これが分割精度の向上に寄与することは直感的に明白である．これに加えて，対訳コーパスや対訳辞書といった対訳資源の有用性も，これまでの研究において指摘されている (Brown 2002; Koehn and Knight 2003; Nakazawa, Kawahara, and Kurohashi 2005)．英語表記において複合語は分かち書きされるため，複合語に対応する英訳表現を対訳資源から発見することができれば，その対応関係に基づいて複合語の分割規則を学習することが可能になる．

複合語分割処理の精度低下を引き起こす大きな要因は，言語資源に登録されていない未知語の存在である．特に日本語の場合においては，片仮名語が未知語の中の大きな割合を占めていることが，これまでにも多くの研究者によって指摘されている (Brill, Kacmarcik, and Brockett 2001; Nakazawa et al. 2005; Breen 2009)．冒頭でも述べたように，片仮名語は生産性が非常に高いため，既存の言語資源に登録されていないものが多い．例えば Breen (2009) らによると，新聞記事から抽出した片仮名語のうち，およそ 20%は既存の言語資源に登録されていなかったことが報告されている．こうした片仮名語から構成される複合名詞は，分割処理を行うことがとりわけ困難となっている (Nakazawa et al. 2005)．

分割が難しい片仮名複合名詞として，例えば「モンスターペアレント」がある．この複合名詞を「モンスター」と「ペアレント」に分割することは一見容易なタスクに見えるが，一般的な形態素解析辞書[1]には「ペアレント」が登録されていないことから，既存の形態素解析器にとっ

[1] ここでは JUMAN 辞書 ver. 6.0 と NAIST-jdic ver. 0.6.0 を調べた．

ては困難な処理となっている．実際に，MeCab ver. 0.98 を用いて解析を行ったところ（解析辞書は NAIST-jdic ver. 0.6.0 を用いた），正しく分割することはできなかった．

1.2　言い換えと逆翻字の利用

こうした未知語の問題に対処するため，本論文では，大規模なラベルなしテキストを用いることによって，片仮名複合名詞の分割精度を向上させる方法を提案する．近年では特にウェブの発達によって，極めて大量のラベルなしテキストが容易に入手可能となっている．そうしたラベルなしテキストを有効活用することが可能になれば，辞書や対訳コーパスなどの高価で小規模な言語資源に依存した手法と比べ，未知語の問題が大幅に緩和されることが期待できる．これまでにも，ラベルなしテキストを複合名詞分割のために利用する方法はいくつか提案されているが，いずれも十分な精度は実現されていない．こうした関連研究については 2 節において改めて議論を行う．

提案手法の基本的な考え方は，片仮名複合名詞の言い換えを利用するというものである．一般的に，複合名詞は様々な形態・統語構造へと言い換えることが可能であるが，それらの中には，元の複合名詞内の単語境界の場所を強く示唆するものが存在する．そのため，そうした言い換え表現をラベルなしテキストから抽出し，その情報を機械学習の素性として利用することによって，分割精度の向上が可能となる．これと同様のことは，片仮名語から英語への言い換え，すなわち逆翻字に対しても言うことができる．基本的に片仮名語は英語を翻字したものであるため，単語境界が自明な元の英語表現を復元することができれば，その情報を分割処理に利用することが可能となる．

提案手法の有効性を検証するための実験を行ったところ，言い換えと逆翻字のいずれを用いた場合においても，それらを用いなかった場合と比較して，F 値において統計的に有意な改善が見られた．また，これまでに提案されている複合語分割手法との比較を行ったところ，提案手法の精度はそれらを大幅に上回っていることも確認することができた．これらの実験結果から，片仮名複合名詞の分割処理における，言い換えと逆翻字の有効性を実証的に確認することができた．

本論文の構成は以下の通りである．まず 2 節において，複合名詞分割に関する従来研究，およびその周辺分野における研究状況を概観する．次に 3 節では，教師あり学習を用いて片仮名複合名詞の分割処理を行う枠組みを説明する．続いて 4 節と 5 節においては，言い換えと逆翻字を学習素性として使う手法について説明する．6 節では分割実験の結果を報告し，それに関する議論を行う．最後に 7 節においてまとめを行う．

2 関連研究

2.1 複合語分割

これまでにも，ラベルなしテキストを用いた複合語分割手法はいくつか提案されている．それらはいずれも，複合語の構成語の頻度をラベルなしテキストから推定し，その頻度情報に基づいて分割候補を順位付けするものとなっている (Koehn and Knight 2003; Ando and Lee 2003; Schiller 2005; Nakazawa et al. 2005; Holz and Biemann 2008)．とりわけ本研究と関連が深いのは (Nakazawa et al. 2005) であり，彼らもまた片仮名複合名詞を対象としている．しかし，こうした単語頻度に基づく手法は，対訳資源を用いた手法と比較して，十分な分割精度が得られないという問題が指摘されている (Koehn and Knight 2003; Nakazawa et al. 2005)．実際，我々の実験においても，これら単語頻度に基づく手法と提案手法との比較を行ったが，提案手法の方が大幅に高い分割精度を実現可能であることを確認した．

一方，Alfonseca ら (2008) は，ラベルなしテキストではなくクエリログを複合語分割に利用することを提案している[2]．しかし彼らの実験報告によると，クエリログを用いなかった場合の精度が 90.45%であるのに対して，クエリログを用いた場合の精度は 90.55%であり，その改善幅は極めて小さい．一方，本研究の実験（6 節）では，提案手法の導入によって精度は 83.4%から 87.6%へと大きく向上し，なおかつ，その差は統計的に有意であることが確認された．また，クエリログは一部の組織以外では入手が困難であるのに対し，提案手法に必要なラベルなしテキストは容易に入手することが可能である．

Holz と Biemann (2008) は独語の複合語に対する分割手法と言い換え手法を提案しており，本研究との関連性が高い．しかし，彼らが提案しているアルゴリズムは，複合語の分割と言い換えをパイプライン的に行うものであるため，提案手法とは異なり，言い換えに関する情報は分割時に用いられない．

2.2 その他の関連研究

片仮名複合名詞の分割処理は単語分割の部分問題であると考えることができる．そのため，既存の単語分割器を用いて片仮名複合名詞の分割処理を行うことも可能であるが，実際問題として，それでは十分な分割精度を得ることは難しい（6 節の実験結果を参照）．この原因として，既存の単語分割器は辞書に強く依存した設計となっており，未知語が多い片仮名語の解析に失敗しやすいことが挙げられる．これに関する議論は (Nakazawa et al. 2005) が詳しい．単語分割の視点から見た本研究は，片仮名複合名詞という特に解析が困難な言語表現に焦点をあてた試みであると言える．

[2] 彼らはウェブテキストのアンカーテキストを用いることも提案しているが，精度の向上は実現されていない．

5 節において我々は，片仮名複合名詞の分割のために逆翻字を利用する手法を提案する．提案手法は，技術的な観点から見ると，ウェブから片仮名語の逆翻字を自動抽出する既存手法 (Brill et al. 2001; Cao, Gao, and Nie 2007; Oh and Isahara 2008; Wu, Okazaki, and Tsujii 2009) と関連が深い．しかしながら，そうした関連研究は翻字辞書や翻字生成システムを構築することを目的としており，自動抽出した逆翻字を複合語の分割処理に利用する試みは本研究が初めてである．

3　教師あり学習に基づく手法

本論文では, 片仮名複合名詞 x が入力として与えられたとき, それを構成語列 $\boldsymbol{y} = (y_1, y_2 \dots y_{|\boldsymbol{y}|})$ へと分割する問題を取り扱う．ここでは，出力 $\boldsymbol{y}$ が 1 語（すなわち $|\boldsymbol{y}| = 1$）である場合もありうることに注意をされたい．

1 節においても議論したように，片仮名名詞は英語の翻字が多く，提案する素性の 1 つもその性質を利用したものとなっているため，以下では入力される片仮名語は英語の翻字であると仮定する．この仮定が実テキストにおいてどの程度成立しているのかを検証することは難しいが，例えばウェブ検索エンジンのクエリにおいては，片仮名のクエリの約 87%は翻字であることが報告されている (Brill et al. 2001)．このデータから上記の仮定にはある程度の妥当性があることが推測され，実テキストを処理する際にも提案手法の効果を期待することができる．

我々は片仮名複合名詞の分割処理を「片仮名複合名詞 x に対する構成語列 $\boldsymbol{y}$ を予測する構造予測問題」と捉えて，これを以下のような線形モデルを用いて解く．

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}(x)} \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{y})$$

ここで $\mathcal{Y}(x)$ は入力 x に対する全分割候補の集合を表す．$\boldsymbol{\phi}(\boldsymbol{y})$ は分割候補 $\boldsymbol{y}$ の素性ベクトル表現，$\boldsymbol{w}$ は訓練事例から推定される重みベクトルである．

表 1 に我々が実験で用いた素性テンプレートを示す．テンプレート 1 からは，ある構成語

表 1　実験で使用した素性テンプレート

ID	テンプレート	説明
1	y_i	構成語 1-gram
2	$y_{i-1}y_i$	構成語 2-gram
3	Len (y_i)	y_i の文字数 (1, 2, 3, 4, or $\geq$5)
4	Dict (y_i)	y_i が辞書に登録されているか否か
5	Para (y_{i-1}, y_i)	構成語 2-gram の言い換え頻度
6	BackTrans (y_i)	y_i が英単語 1-gram に逆翻字可能か否か
7	BackTrans $(y_{i-1}y_i)$	$y_{i-1}y_i$ が英単語 2-gram に逆翻字可能か否か

1-gram が出現したか否かを示す 2 値素性が，訓練事例に出現した全ての構成語 1-gram について生成される．テンプレート 2 は同様の 2-gram 素性である．テンプレート 3 からは，構成語の文字数 (1, 2, 3, 4, ≥5) を示す 2 値素性が 5 種類生成される．テンプレート 4 は構成語 y が外部辞書[3]に登録されているか否かを表す 2 値素性であり，構成語 y が外部辞書に登録されていれば 1 を返す 2 値素性が 1 つ生成される．テンプレート 5 から 7 は，片仮名複合名詞の言い換えと逆翻字を用いたものであり，4 節と 5 節において詳しく説明する．以下の議論では，テンプレート 1 から 4 によって生成される素性を基本素性，テンプレート 5 から生成される素性を言い換え素性，テンプレート 6 と 7 から生成される素性を逆翻字素性と呼んで互いに区別をする．

重みベクトル $\boldsymbol{w}$ は任意の学習アルゴリズムを用いて最適化することが可能であるが，ここでは計算効率を考慮して平均化パーセプトロン (Freund and Schapire 1999) を用いた．平均化パーセプトロンはオンライン学習アルゴリズムの一種であり，高速に学習を行うことができると同時に，多くのタスクにおいて SVM などのバッチ学習アルゴリズムと比較しても遜色のない精度を達成できることが知られている．パーセプトロンの訓練時およびテスト時には $\boldsymbol{y}^*$ を求める操作が必要となるが，セミマルコフモデルにおいて用いられるのと同様の動的計画法によって効率的に実行可能である．

4 言い換え素性

本節では，片仮名複合名詞の言い換え表現を，教師あり学習の素性として使う方法について述べる（表 1 におけるテンプレート 5 に対応する）．

4.1 複合名詞の言い換え

一般的に，複合名詞は様々な形へと言い換えることが可能であるが，そうした言い換え表現の中には，元の複合名詞の単語境界を認識する手がかりとなるものが存在する．以下に具体例を示す．

(1) a. アンチョビパスタ
b. アンチョビ・パスタ
c. アンチョビのパスタ

(1b) は，複合名詞 (1a) の構成語間に中黒を挿入することによって生成された言い換え表現である．同様に (1c) は助詞「の」を挿入することによって生成された言い換え表現である．もしラベルなしテキストにおいて (1b) や (1c) のような言い換え表現を観測することができれば，この

[3] 外部辞書としては NAIST-jdic ver. 0.6.0 を用いた．

ことは複合名詞 (1a) を「アンチョビ」と「パスタ」に正しく分割するための手がかりとなることが考えられる．

4.2 言い換え規則

このような言い換えを利用して片仮名複合名詞の分割処理を行うため，複合名詞の言い換え規則を 7 つ作成した（表 2）．言い換え規則の作成にあたっては，Kageura ら (2004) の研究を参考にしながら，分割処理に有用と思われるものを人手で選定した．作成した言い換え規則は全て $X_1X_2 \rightarrow X_1MX_2$ という形式をしており（X_1 と X_2 は名詞，M は助詞などの機能語），左辺が言い換え前の複合名詞，右辺が言い換え後の表現に対応している．

4.3 言い換え頻度に基づく素性

これらの規則を用いて，次のように新しい素性を定義する．まず前処理として，以下のような正規表現を用いることにより，片仮名複合名詞の言い換え表現の出現頻度をラベルなしテキストから求める．

(katakana)+ ・ (katakana)+
(katakana)+ の (katakana)+
(katakana)+ する (katakana)+
(katakana)+ した (katakana)+
(katakana)+ な (katakana)+
(katakana)+ 的 (katakana)+
(katakana)+ 的な (katakana)+

ただし (katakana) は片仮名 1 文字にマッチする特殊文字である．また + は文字の繰り返しを表す量指定子であり，最長一致が適用されるものとする．

このような正規表現を用いることによって，単語分割処理を行わずに言い換え表現を抽出することができるのは，表 2 のような片仮名複合語の言い換え表現に対象を限定しているためで

表 2　作成した言い換え規則の一覧とその適用例．X_1 と X_2 は名詞を表す

タイプ	規則	適用例
中黒	$X_1X_2 \rightarrow X_1$・X_2	アンチョビパスタ → アンチョビ・パスタ
助詞「の」	$X_1X_2 \rightarrow X_1$ の X_2	アンチョビパスタ → アンチョビのパスタ
補助動詞	$X_1X_2 \rightarrow X_1$ する X_2	ダウンロードファイル → ダウンロードしたファイル
	$X_1X_2 \rightarrow X_1$ した X_2	
形容詞性接尾辞	$X_1X_2 \rightarrow X_1$ な X_2	サプライズギフト → サプライズなギフト
	$X_1X_2 \rightarrow X_1$ 的 X_2	
	$X_1X_2 \rightarrow X_1$ 的な X_2	

ある．上記の正規表現にマッチするテキストは，必ず前後が片仮名以外の文字（漢字や平仮名）に囲まれていることになる．そのような文字種の変わり目には，単語境界が存在する場合が多いため，このような単純な文字列処理であっても言い換え表現を抽出することが可能になっている．

分割処理時に分割候補 $\boldsymbol{y}$ が提示された際には，構成語 2-gram に対する言い換え素性 PARA (y_{i-1}, y_i) の値を次のように定義する．まず $X_1 = y_{i-1}$，$X_2 = y_i$ と代入することにより，表 2 の規則から言い換え表現を生成する．そして，生成された 7 つの言い換え表現の頻度の和を F としたとき，その対数 $\log(F+1)$ を素性値として用いる．

ここでは素性値の計算に非常に単純な方法を用いているため，X_1 や X_2 に名詞ではなく，名詞連続が代入された場合であっても，素性が発火してしまうということがある．また逆に，正解となる構成語よりも小さな単位の文字列が代入された場合であっても，同様に素性が発火してしまうことがあり，精度に悪影響を及ぼす可能性がある．しかし，このような手法であっても実験において分割精度の向上を十分確認することができたため，シンプルさを重視して現在のような手法とした．

素性値として頻度ではなく対数頻度を用いているのはスケーリングのためである．予備的な実験においては，頻度をそのまま素性値として用いることも行ったが，対数頻度を用いた場合の方が高い精度が得られた．なお，$\log F$ ではなく $\log(F+1)$ としているのは，$F = 1$ であった場合に素性値が 0 となるのを防ぐためである．

5 逆翻字素性

片仮名語の多くは英語を翻字したものであり，元となる英語表現が存在する．以下では，そのような英語表現のことを**原語**と呼び，片仮名語と原語の対のことを**翻字対**と呼ぶこととする．我々は，片仮名語が原語の発音情報をおおよそ保持しているという特性を利用することによって，単語単位での対応関係が付与された翻字対（**単語対応付き翻字対**）をラベルなしテキストから自動抽出する（表 3）．そして，得られた単語対応付き翻字対に基づいて，分割結果 $\boldsymbol{y}$ に出現する単語 n-gram が，英単語 n-gram と対応付け可能であるかを示す 2 値素性を用いる（表 1 におけるテンプレート 6 と 7 に対応する）．以下本節では，テキストから単語対応付き翻字対を

表 3 単語対応付き翻字対の例．下線部に付与された数字は単語の対応を表す

片仮名語	原語
ジャンク$_1$フード$_2$	junk$_1$ food$_2$
スパム$_3$	spam$_3$

自動抽出する方法について説明する．

5.1 括弧表現

日本語においては，括弧表現を使って片仮名語の原語がテキスト中に挿入される場合がある．

(2) a. アメリカでジャンクフード (junk food) と言えば...

b. トラックバックスパム (spam) を撃退するため...

いずれの例文においても，下線を引いた片仮名語に対して，その原語が括弧を使って併記されている．我々はこのような括弧表現を利用することにより，単語対応付き翻字対の自動抽出を行う．

こうした括弧表現から単語対応付き翻字対の抽出を行うためには，少なくとも以下の 3 つのことが技術的な問題となる

問題 A 片仮名語の直後に出現する括弧表現が必ずしもその原語であるとは限らないため，原語が記述されている括弧表現とそうでない括弧表現を区別する必要がある．

問題 B 翻字対の関係にある片仮名語の開始位置を決定しなくてはならない．例えば (2b) においては，原語「spam」の翻字は「トラックバックスパム」ではなく「スパム」である．

問題 C 片仮名語と原語の単語対応を求めるためには，片仮名語を分かち書きしなくてはならない．例えば (2a) から表 3 のような単語対応付き翻字対を獲得するためには，片仮名列「ジャンクフード」を「ジャンク」と「フード」に分割することが必要である．

5.2 発音の類似性の利用

これまでにも，前述のような括弧表現から翻字対を自動抽出する研究は数多く存在するが，問題 C に対する本質的な解決策はいまだ提案されていない．これまでの研究においては，基本的に既存の単語分割器を用いることによって片仮名語の分割が行われている (Cao et al. 2007; Wu et al. 2009)．しかし，2 節において議論を行ったように，片仮名語の分かち書きを行うことは現在のところ技術的に困難であり，このようなアプローチは望ましくない．

我々は上記の 3 つの問題を解決するため，片仮名語と原語の発音の類似性を利用することを提案する．以下の議論では，説明のために，まず問題 C だけを議論の対象とする．具体例として，片仮名語「ジャンクフード」と原語「junk food」に対して，それらの発音の類似性に基づき以下のような部分文字列の対応関係が得られたとする．

(3) a. [ジャン]$_1$[ク]$_2$[フー]$_3$[ド]$_4$

b. [jun]$_1$[k]$_2$ [foo]$_3$[d]$_4$

ここでは，括弧で囲まれて同じ番号を添えられている部分文字列が，互いに対応関係にあるも

のとする．括弧表現内の英語は空白を使って分かち書きされているため，上記のような部分文字列の対応関係を利用すれば，片仮名語と英単語が 1 対 1 に対応するように片仮名列を分かち書きすることができる．また，その過程において，単語間の対応関係も明らかにすることができる．

残る問題 A および問題 B に対しても，発音の類似性に基づいて同様に解決を図ることが可能である．以下の例において，下線が引かれた片仮名語と括弧内の英語表現が翻字対であるか否かを判定することを考える．

(4) a. 検索エンジン (Google) を使って...

b. トラックバックスパム (spam) を撃退する...

このように，括弧内に原語ではない表現が出現したり，片仮名語の開始位置が正しく認識されなかった場合には，片仮名列とアルファベット列の発音の類似度が低くなることが期待できるため，フィルタリングできると考えられる．単語対応付き翻字対の具体的な抽出手順については，5.4 節において説明を行う．

5.3 発音モデル

片仮名語と原語における部分文字列の対応関係の発見には，Jiampojamarn ら (2007) が提案した生成モデルを用いる．f と e をそれぞれ片仮名列とアルファベット列とし，これらの間の対応関係を見つけることを考える．ただし，原語には空白が存在する可能性があるが，空白に対応する片仮名文字列は存在しないことから，部分文字列の対応を求めるときにはアルファベット列から空白を取り除いておく．例えば「ジャンクフード」と「junk food」の部分文字列対応を求める場合には「$f =$ ジャンクフード」「$e =$ junkfood」とする．ここで，$\mathcal{A}$ をそれらの間の部分文字列の対応とする．具体的には，$\mathcal{A}$ は対応付けられている部分文字列の組 (f_i, e_i) の集合であり，$f = f_1 f_2 \dots f_{|\mathcal{A}|}$ および $e = e_1 e_2 \dots e_{|\mathcal{A}|}$ となる．この部分文字列対応 $\mathcal{A}$ の確率を以下のように定義する．

$$\log p(f, e, \mathcal{A}) = \sum_{(f_i, e_i) \in \mathcal{A}} \log p(f_i, e_i)$$

一般に $\mathcal{A}$ は観測することができないため隠れ変数として扱い，モデルのパラメータは翻字対 (f, e) の集合から EM アルゴリズムを用いて推定する．詳細は文献 (Jiampojamarn et al. 2007) を参照されたい．表 4 に「ジャンクフード」と「junkfood」に対する部分文字列対応 $\mathcal{A}$ の具体例，および実験において計算された確率値を示す．

この確率モデルを用いて，与えられた (f, e) に対する部分文字列の対応を次のように決定する．

$$\mathcal{A}^* = \operatorname*{argmax}_{\mathcal{A}} \log p(f, e, \mathcal{A})$$

表 4　片仮名列「f = ジャンクフード」とアルファベット列「e = junkfood」に対する部分文字列対応 $\mathcal{A}$ の具体例 ($|\mathcal{A}| = 4$)

(f_i, e_i)	$\log p(f_i, e_i)$
(ジャン，jun)	−10.767
(ク，k)	−5.319
(フー，foo)	−11.755
(ド，d)	−5.178

このとき $\mathcal{A}^*$ の中の部分文字列 e_i が空白をまたいでしまうと（ジャンクフードの例であれば e_i = kfoo などとなった場合），$\mathcal{A}^*$ を使って片仮名列 f を分かち書きすることができなくなってしまう．そこで，アルファベット列 e が空白を含んでいた場合は，前述のとおり空白を取り除いて確率値の計算を行うが，空白の存在した箇所は記憶しておき，部分文字列 e_i が空白をまたがないという制約を加えて argmax の計算を行う．

5.4　単語対応付き翻字対の抽出

この発音モデルを用いて，以下のような手順で単語対応付き翻字対の抽出を行う．

手順 1　括弧内に出現するアルファベット列 e と，その直前に出現する片仮名列 f を抽出し，それらの組 (f, e) を翻字対の候補とする．ただしアルファベット列は全て小文字に正規化する．

手順 2　翻字対候補 (f, e) に対するスコアを以下のように定義し，それが閾値 θ を越えたものを正しい翻字対と判定する．

$$\frac{1}{N} \log p(f, e, \mathcal{A}^*)$$

式中の N は e に含まれる単語数であり，$\frac{1}{N}$ という項は単語数が多い場合にスコアが過剰に小さくなるのを防ぐために導入している．ここでスコアが閾値を下回っていた場合には，片仮名語の開始位置を正しく判定できていない可能性がある．そこで，片仮名列 f の最左文字を 1 文字ずつ削除していき，閾値を上回るものが見つかればそれを翻字対と判定し，次の翻字対候補の処理に移る．

手順 3　得られた翻字対 (f, e) に対して，部分文字列対応 $\mathcal{A}^*$ に基づいて片仮名列 f を分かち書きし，単語の対応関係を求める．これにより，単語対応付き翻字対のリストを得ることができる．

ただし，手順 2 においては，表記揺れやタイポなどの要因により，1 つの片仮名列に対して複数の逆翻字が見つかる可能性がある．その場合は，各片仮名列 f に対して，最もスコアの高い翻字対 (f, e) のみを保持して，それ以外のものは使用しない．

6 実験と議論

本節では，提案する 2 つの素性（言い換え素性と翻字素性）が片仮名複合名詞の分割処理の精度に与える効果について報告を行う．

6.1 実験設定

発音モデルのパラメータ推定に必要な翻字対のデータは，外国人の名前を日本語で表記するときにはほぼ常に翻字が行われることに着目し，Wikipedia[4]を用いて自動的に構築した．構築手順としては，まず「存命人物」のカテゴリに所属する Wikipeida 記事のタイトルを抽出することにより，片仮名表記の人名リストを作成した．そして次に，Wikipedia の言語間リンクを利

表 5 表 2 の規則をもとに抽出された言い換え表現（の候補）．括弧内の数字は頻度を表す

カリフォルニアのギャング (5)	キャラ・リラ (1)
チエミのファルセット (1)	ラフトレードのニューショップ (1)
イヤ〜なオヤジ (1)	ヒストリカル・イフ (5)
マサルのチャレンジレース (1)	ザキル・ハッサン (1)
コムタローのブーム (1)	イケメン・アスラ (1)
ユー・ハッピーアム (1)	ジュンノのサイン (1)
ルカのガイドブック (1)	レーンのウェア (1)
コスナーのフィールドオブドリームズ (1)	ルーカスアーツのヘイデン (1)
トリオ・ボン (5)	マーメイドのレインダンス (1)
ノニュースのインタビュー (1)	トップページ・コンテンツページ (1)
エガゴロのペン (1)	スティーブン・ローンチ (1)
キュンキュンするラジオドラマ (1)	ショートのライダース (1)
トチ・バスミルク (1)	ランナー・ラヴテーマ (1)
アートスペースバレーナのギャラリースペース (3)	フィリエルのパパ
ゴージャスなワーグナー (1)	ザビーネ・マイアー (4)
デザイン・グラフィックコース (1)	チャパツのマッシュルームヘア (1)
ライティングのテンプレート (1)	チャンスオーフレッシュのイイ (1)
ファミリー・フード (1)	ポスターのキレイ (7)
マインドマップのユーザ (2)	コースのコアジサイ (10)
ホットのストレートティー (10)	グスタベルクのミミズ (1)
ペッパー・ポーク (1)	ピースフルなラウンド (1)
アニメメジャー・ロードオブメジャー (1)	アンチエイジング・アトピー (35)
クローゼット・シック (1)	ポルチーニのミルクシチュー (1)
スタイリッシュなキャリーバック (1)	ビスケットのサル (1)
ワンクリックアンケートのプログラム〜〜 (1)	ドアガラスのガイドレール (8)

[4] http://ja.wikipedia.org/

表 6 単語対応付き翻字対の例．スラッシュは抽出時に検出された片仮名語の単語境界を示す．単語間の対応関係は自明なので省略する

ウィキペディア wikipedia	スパ spa
ランキング ranking	マングローブ manglobe
フライド/コーク fried coke	ンス ds
オフ off	ヤマサキ/オサム yamasaki osamu
カウパレード cowparade	レブロス revros
バズコックス buzzcocks	ハリウッド/リポーター hollywood reporter
ハーグ hague	クリーム/カーキ cream khaki
バリー/ゴルフ bally golf	グリーン/ウッド green wood
ズライ/エナオ zulay henao	オートラント otranto
サンディ sandy	ビッグ/ショー big show
テケリ tekeli	ンダー/カラー under color
ーアシックス asics	フルクサス fluxus
シックス/フラッグス six flags	クイックスキャット quickscat
ポール/エリュアール paul eluard	アンヘル/サンチェス angel sanchez
ピマン pimento	ベンジャミン/カーティス benjamin curtis
オウン/ゴール own goal	ドリメル dremel
アントニオ/ヴィヴァルディ antonio vivaldi	スパイス/トワイス spice twice
ジョン/フォン/ノイマン john von neumann	アウエアネス awareness
アレクサンドル/チャバン alexander chaban	マーク/パドモア mark padmore
アントルコート entrecote	ー/コレ u colle
エベン/モグレン eben moglen	ジェイムズ/デバージ james debarge
ヨハネス/シュタルク johannes stark	タン/ドゥン tan dun
シャーリーロジャーズ shirleyrogers	タップ/ウォーター/プロジェクト tap water project
サンダー/ベイ thunder bay	ノースウッド northwood
レトロ/キッチン retro kitchen	オマリ/ハードウィック omari hardwick

用し，各人名に対する原語を抽出した．これにより 17,509 の翻字対を収集することができた．このように自動収集したデータの中には翻字対として不適切なものも含まれている可能性があるが，大量のデータを手軽に用意できるという利点を重視して，この方法を採用している．実際，このようにパラメータ推定のためのデータを大量に生成するアプローチは，翻字生成において有効であることが報告されている (Cherry and Suzuki 2009)．パラメータ推定時には，EM アルゴリズムの初期値を無作為に 10 回変化させ，尤度が最大となったモデルを以降の実験で用いた．

平均化パープトロンの学習に必要なラベル付きデータは，日英対訳辞書 EDICT[5]を利用して手作業で作成した．具体的には，まず，EDICT の見出し語から，翻字である片仮名（複合）名

[5] http://www.csse.monash.edu.au/~jwb/edict_doc.html

詞を無作為に抽出した．そして，EDICT に記載されている英訳に基づき，単語境界のラベルを付与した．この結果，5286 の片仮名語データを得た．このデータにおける構成語数の分布を調べたところ，構成語が 1 語のものが 3041，2 語のものが 2081，3 語以上のものが 164 となっていた ($3041 + 2081 + 164 = 5286$)．また，複合名詞 1 つあたりの平均文字数および平均構成語数は 6.60 および 1.46 であった．以下本節において報告する実験結果は，このラベル付きデータを用いて 2 分割交差検定を行ったものである．

言い換え及び逆翻字を抽出するためのテキストには，ウェブから収集した 17 億文のブログ記事を用いた．このテキストを用いることによって 14,966,205 の言い換え表現と，116,027 の単語対応付き翻字対を抽出することができた．表 5 と 6 に，実際に抽出された言い換え表現（の候補）と単語対応付き翻字対の具体例を示す．

単語対応付き翻字対の抽出を行う際には閾値 θ を設定する必要がある．θ は確率の対数に対する閾値であるため，0 より小さな任意の値を設定することが可能であるが，ここでは $\{-10, -20, \ldots -150\}$ の範囲で値を変化させ，実験において最も高い F 値が得られた値 ($\theta = -80$) を採用した．

6.2 ベースライン手法

実験では，3 つの教師なし学習手法 (Unigram, GMF, GMF2)，2 つの教師あり学習手法 (AP, AP + GMF2)，3 つの単語分割器 (JUMAN, MeCab, KyTea) との比較を行った．以下ではこれらベースライン手法について簡単に説明を行う．

教師なし学習

Unigram 分割結果 $\boldsymbol{y}$ に対する 1-gram 言語モデルの尤度 $p(\boldsymbol{y})$ が最も大きくなる分割を選択する手法 (Schiller 2005; Alfonseca et al. 2008):

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}(x)} p(\boldsymbol{y}) = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}(x)} \prod_i p(y_i)$$

ここで $p(y_i)$ は構成語 y_i の出現確率であり，6.1 節で述べたウェブテキストから推定をした値を用いた．

GMF 構成語 y_i の頻度の幾何平均 $\mathrm{GMF}(\boldsymbol{y})$ が最大となる分割 $\boldsymbol{y}$ を選択する手法 (Koehn and Knight 2003):

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}(x)} \mathrm{GMF}(\boldsymbol{y}) = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}(x)} \Big\{ \prod_i f(y_i) \Big\}^{1/|\boldsymbol{y}|}$$

ここで $f(y_i)$ は構成語 y_i の出現頻度であり，$p(y_i)$ と同様にウェブテキストから推定した値を用いた．

GMF2 頻度の幾何平均に構成語の長さに基づく補正を導入したスコアを用いる手法 (Nakazawa et al. 2005):

$$\mathrm{GMF2}(\boldsymbol{y}) = \begin{cases} \mathrm{GMF}(\boldsymbol{y}) & (|\boldsymbol{y}| = 1) \\ \frac{\mathrm{GMF}(\boldsymbol{y})}{\frac{C}{N^l}+\alpha} & (|\boldsymbol{y}| \geq 2), \end{cases}$$

ここで C，N，α は超パラメータ，l は構成語の平均文字数を表す．本実験では Nakazawa ら (2005) と同じく $C = 2500$，$N = 4$，$\alpha = 0.7$ とした．

教師あり学習

AP　基本素性（3 節参照）のみを用いた平均化パーセプトロン．

AP + GMF2　基本素性に加えて GMF2 の処理結果を素性として用いた平均化パーセプトロン．Alfonseca ら (2008) に従って，(i) GMF2($\boldsymbol{y}$) の値が全分割候補中で最大であるか否かを表す 2 値素性，(ii) 分割を行わない候補（i.e., $|\boldsymbol{y}| = 1$ となる候補）よりも GMF2 の値が大きくなるか否かを表す 2 値素性を追加した．

単語分割器

JUMAN　ルールベースの単語分割器[6] JUMAN ver. 6.0 (Kurohashi and Nagao 1994).

MeCab　対数線形モデルに基づく単語分割器 MeCab ver. 0.98 (Kudo, Yamamoto, and Matsumoto 2004)．解析辞書には NAIST-jdic ver. 0.6.0 を用いた．

KyTea　点推定モデルに基づく単語分割器 KyTea ver. 0.3.1 (Neubig, Nakata, and Mori 2011).

6.3　ベースライン手法との比較

表 7 に提案手法 (Proposed) とベースライン手法との比較結果を示す．この表の結果から以下のようなことが分かる．

表 7　ベースライン手法との比較．表中の P，R，F_1 は，認識された単語境界の適合率，再現率，F 値を示す．また Acc は分割精度，すなわち正しく分割された片仮名複合名詞の割合を示す

タイプ	システム	P	R	F_1	Acc
教師なし学習	Unigram	64.2	49.7	56.0	63.0
	GMF	42.9	62.0	50.7	47.5
	GMF2	67.4	76.0	71.5	72.5
教師あり学習	AP	81.9	82.5	82.2	83.4
	AP + GMF2	83.0	83.9	83.4	84.2
	Proposed	**86.4**	**87.4**	**87.1**	**87.6**
単語分割器	JUMAN	71.4	60.1	65.3	69.8
	MeCab	72.4	73.7	67.8	71.6
	KyTea	79.8	84.3	82.0	82.6

[6] 正確には形態素解析器であるが，本実験では品詞タグ付与の議論は行わないのでこう呼ぶ．

まず，Proposed と AP の結果の比較から，言い換え素性と逆翻字素性を導入することにより，分割精度が大きく向上したことが分かる．マクネマー検定を行ったところ，この精度変化は統計的に有意なものであることが確認された ($p < 0.01$)．この結果は，提案する 2 つの素性の有効性を示すものである．

次に，提案手法の精度は，全ての教師なし学習ベースライン (Unigram, GMF, GMF2) 及び AP + GMF2 の精度を上回っていることが確認できる．これらの結果は，複合名詞の言い換えや逆翻字の情報が，構成語の頻度情報よりも効果的であることを示唆している．なお，マクネマー検定を行ったところ，これらの精度向上も全て統計的に有意であることが確認できた ($p < 0.01$)．

単語分割器 (JUMAN, MeCab, KyTea) の結果は，これまでに単語分割タスクにおいて報告されている精度 (Kudo et al. 2004; Neubig et al. 2011) を大きく下回っている．このことから，一般的な単語分割と比較して，片仮名複合語の分割処理が困難なタスクであることが分かる．さらに，提案手法の精度は，単語分割器の精度を大きく上回っており，提案手法が既存の単語分割器の弱点補強に有用であることが示唆されている．例えば，既存の単語分割器によって「片仮名表記の名詞の連続」と解析された部分を，提案手法を用いて再分割することにより，解析結果の改善を期待することができる．

表 8 に，MeCab では分割に失敗したが，Proposed では正しく分割することができた例を示す．まず最初の例では，片仮名語「ディクショナリー」が NAIST-jdic に登録されていなかったため，MeCab は分割に失敗している．一方，Proposed においては，以下のような単語対応付き翻字対が学習されており，これに基づいて発火した逆翻字素性 (1-gram) が有効に働いた結果，正しく分割することに成功している．

$\underline{\text{オックスフォード}}_1\underline{\text{ディクショナリー}}_2$　$\underline{\text{oxford}}_1\ \underline{\text{dictionary}}_2$

次の例では「メイン」と「タイトル」が両方とも NAIST-jdic に登録されているにも関わらず，MeCab は分割に失敗している．これは，MeCab の未知語処理に起因する誤りであると考えられる．その一方で Proposed が分割に成功しているのは，例えば「メインのタイトル」といった言い換え表現に基づく素性など，分割を示唆する素性がより多く発火しているためだと推測できる．最後の例では，NAIST-jdic に人名「トミー」が登録されているため，MeCab は過分割を行ってしまっているが，Proposed では「アナトミー」に対する逆翻字素性が適切に発火してお

表 8 MeCab と Proposed の出力比較．スラッシュはシステムに認識された単語境界を表す

MeCab	Proposed
オンラインディクショナリー	オンライン/ディクショナリー
メインタイトル	メイン/タイトル
アナ/トミー	アナトミー

り，過分割を防ぐことに成功している．

本論文の趣旨からは外れるが，3 つの単語分割器のなかでは KyTea の精度が他の 2 つを大きく引き離している点は非常に興味深い．これは，JUMAN や MeCab の解析アルゴリズムが，辞書引きによる候補選択に強く依存しているのに対して，KyTea はそのような候補選択を行っていないことが要因と考えられる．

6.4 未知語に関する考察

実験に使用した 5286 の片仮名複合名詞のうち，2542 は少なくとも 1 つの未知語を構成語に含んでいた．ただし，ここで言う未知語とは，訓練データに出現せず，なおかつ外部辞書 NAIST-jdic にも登録されていない単語のことを指す．未知語が分割精度に与える影響について考察するため，提案手法を含む 3 つの教師あり学習手法 (AP, AP + GMF2, Proposed) と単語分割器 MeCab の分類結果を，1 つ以上の未知語を含む 2542 の片仮名複合名詞と残る 2744 の片仮名複合名詞に分けて集計した（表 9）．以下では，前者のサブセットを w/ OOV データ，後者を w/o OOV データと呼ぶ．

この表から，3 つの教師あり学習手法については，w/o OOV データに対しては 90%を越える高い精度が達成されているのに対して，w/ OOV データの精度は大きく低下していることが確認できる．同様の傾向は MeCab の結果においても見られる．MeCab は汎用的な単語分割器であるため，複合名詞分割というタスクに特化して学習された提案手法 (Proposed) やその他の教師あり学習手法（AP や AP + GMF2）と比べると，精度自体はどちらのデータにおいても大きく低下している．しかし，w/ OOV データよりも w/o OOV データのほうが精度が高くなるという傾向は，依然として確認することができる．これらの結果は，片仮名複合名詞の分割処理を困難にしている要因は未知語であるという我々の主張を支持するものである．

3 つの教師あり学習手法は，w/o OOV データについてはほぼ同じ精度を達成していることが分かる．これは，既知語に対しては，基本素性だけを使ってすでに高い分類精度が達成されているため，これ以上の精度向上が困難であるからだと考えられる．一方，精度向上の余地が残

表 9 未知語を含む片仮名複合名詞 (w/ OOV) とそれ以外の片仮名複合名詞 (w/o OOV) に対する分割結果の比較

システム	w/ OOV				w/o OOV			
	P	R	F_1	Acc	P	R	F_1	Acc
MeCab	53.6	47.8	50.5	61.0	86.4	75.3	80.5	81.5
AP	65.9	69.7	67.8	72.8	94.6	91.9	93.2	93.1
AP + GMF2	68.7	73.6	71.1	75.1	94.5	91.3	93.0	92.7
Proposed	**76.2**	**79.4**	**77.8**	**81.0**	**94.7**	**93.2**	**93.9**	**93.7**

されている w/ OOV データについては，3 つのシステムの間に大きな精度の差を見てとることができる．そのため，表 7 の結果よりも，言い換え素性と翻字素性を導入する効果をより直接的に確認することができる．

6.5 言い換え素性と翻字素性の効果

言い換え素性と翻字素性の有効性について詳細に検証するため，異なる 4 つの素性集合を用いたときの平均化パーセプトロンの分割結果の比較を行った（表 10）．表の 1 行目は使用した素性集合を表す．Basic は基本素性，Para と Trans はそれぞれ言い換え素性と翻字素性，All は全ての素性集合を表す．この表より，言い換え素性と翻字素性の両方ともが分割精度向上に大きく貢献していることを確認することができた．いずれの場合においても，基本素性だけを使った場合と比較して，精度の向上は統計的に有意であった（$p < 0.01$，マクネマー検定）．

次に，各素性の発火率について調査を行った．実験で用いたラベル付きデータには 7709 の構成語が含まれており，そのうち 64.0% (4937/7709) は外部辞書に登録されていた．これに対して，単語対応付き翻字対に出現していた構成語の割合は 64.0% (4935/7709)，外部辞書か単語対応付き翻字対のいずれかに出現していたものの割合は 77.1% (5941/7709) であった．これにより，翻字素性を導入することによって，未知語の数が大幅に減少していることが確認された．

表 10 言い換え素性と逆翻字素性の効果．表中の Basic, Para, BackTrans は，それぞれ基本素性，言い換え素性，逆翻字素性を示す．また All はそれら全ての素性を示す

素性集合	P	R	F_1	Acc
Basic	81.9	82.5	82.2	83.4
Basic + Para	85.1	85.3	85.2	85.9
Basic + BackTrans	85.1	86.3	85.7	86.5
All	**86.4**	**87.4**	**87.1**	**87.6**

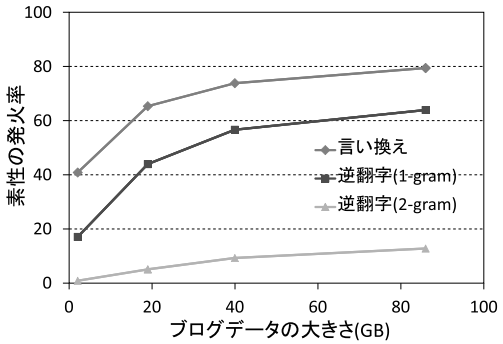

図 1 言い換えと単語対応付き翻字対の抽出に用いたブログデータのサイズ（横軸）と各素性の発火率（縦軸）の関係

一方，ラベル付きデータに含まれる構成語 2-gram の数は 2423 であったが，それらに対して発火していた言い換え素性と翻字素性の割合は，それぞれ 79.5% (1926/2423) と 12.8% (331/2423) であった．これらの結果から，提案素性はいずれも精度向上に寄与しているものの，カバレージにはまだ改善の余地があることが分かった．

続いて，素性の発火率と収集元であるブログデータの大きさの関係を調査した（図 1）．ここではブログデータの大きさとして，収集したブログ記事（UTF8 エンコーディング）を gzip で圧縮したデータのサイズをギガバイト単位で表示している．この図から，大量のブログデータを使うことによって，高い発火率を実現できていることが確認できる．しかし，その一方で，データが増えるにつれて，発火率の向上の度合いは鈍りつつある．このことから，データを単純に増加させるだけでは，ここからの大幅な発火率の改善を期待することは難しく，言い換え規則の拡張などの方法も併せて検討していくことが今後重要になると考えられる．

6.6　パラメータ θ

最後に，パラメータ θ の値を変化させたときの影響について調査を行った（図 2–4）．図 2 と 3 は，様々な値の θ に対する，単語対応付き翻字対の抽出数および逆翻字素性の発火した割合（6.5 節において議論したもの）を示している．これらの図から，θ の値をある程度小さく設定すれば，十分な数の翻字対が抽出され，その結果として多くの事例において素性が発火するようになることが分かる．図 4 は θ と F 値の関係を示している．さきほどの 2 つの図との比較すると，翻字対の抽出数と素性の発火数の増加が，F 値の向上に直接結びついていることが分かる．

パラメータの値が極端に大きい場合 (e.g., –20) においては，F 値が低下する傾向が見られたものの，パラメータによらず F 値はおおよそ一定であった．この結果から，提案手法の精度はパラメータ設定に敏感ではなく，パラメータ調整は難しい作業ではないことが示唆される．また，少なくとも実験において調べた範囲では，提案手法はパラメータ値によらず，基本素性のみを用いた場合よりも高い F 値を達成することができた．そのため，パラメータの微調整が提

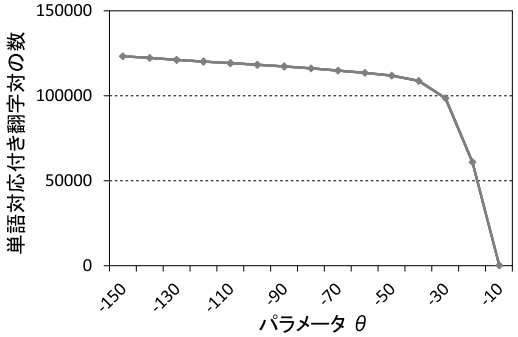

図 2　パラメータ θ（横軸）と抽出された単語対応付き翻字対の数（縦軸）

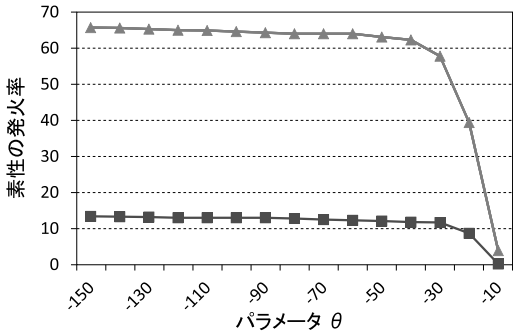

図 3　パラメータ θ（横軸）と逆翻字素性の発火率（縦軸）．グラフ中の三角と四角は，それぞれ構成語 1-gram と 2-gram に対する発火率を表す

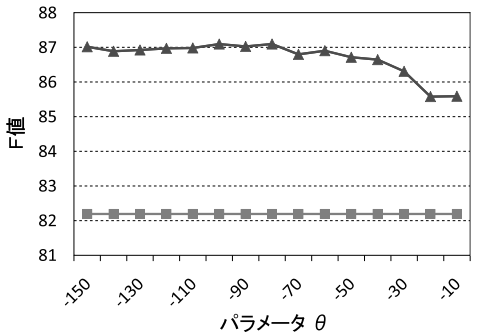

図 4　パラメータ θ（横軸）と F 値（縦軸）の関係．グラフ中の三角と四角は，それぞれ全素性を使った場合と，基本素性のみを使った場合に相当する

案手法の性能に与える影響は小さいと言うことができる．

6.7　誤りの分析

提案手法が分割を誤った事例を調べたところ，「アップロード」を「アップ」と「ロード」，「トランスフォーマー」を「トランス」と「フォーマー」に分割するなど，単語を過分割している事例が見られた．ここでの「アップ」や「トランス」は接頭辞であると考えられるため，これらの分割結果は形態論的分割 (morphological segmentation) としては正しいものであるかもしれないが，単語分割としては不適切であると考えられる．

こうした過分割が発生する要因として，接辞と単語の曖昧性を指摘することができる．例えば「アップ」は，確かに接頭辞の 1 つであるが，文脈によっては「給料がアップする」のように独立した名詞として使われる場合もある．同じく「トランス」に対しても「トランス状態」のような名詞用法を考えることができる．このような曖昧性によって引き起こされる最も顕著な問題は，辞書素性（表 1 におけるテンプレート ID4）が過剰に発火することである．前述の過

表 11　過分割結果に影響を与えたと思われる単語対応付き翻字対の一部

片仮名語	原語
アップロード$_1$	upload$_1$
アップ$_2$ローダー$_3$	up$_2$ loader$_3$
トランスフォーマー$_4$	transformer$_4$
トランス$_5$フォーム$_6$	trans$_5$ form$_6$

分割の事例においては，NAIST-jdic に「アップ」と「トランス」がともに名詞として登録されていたため，本来不適切な分割結果であるにも関わらず辞書素性が発火していた．

これと同様の問題は，辞書素性だけでなく，逆翻字素性においても発生しうる．5 節で説明した単語対応付き翻字対の抽出手法は，原語が正しく分かち書きされていることを前提としていた．しかしながら，実際には接頭辞や接尾辞の前後に空白区切りを挿入しているテキストも存在するため，不適切な対応関係が学習されてしまう場合がある．

表 11 は上記の過分割結果に影響を与えたと思われる単語対応付き翻字対の一部である．この表から，「アップロード」と「トランスフォーマー」については，それぞれ原語との対応関係が適切に学習されていることが確認できる．しかしながら「アップローダー」と「トランスフォーム」については，原語が接頭辞の直後で分かち書きされていたため，不適切な単語対応が学習されていることが分かる．こうした対応付け結果から導出された逆翻字素性（この例では特に 1-gram）は分割に悪影響を与えている可能性がある．翻字抽出の手法を改善することにより，こうした誤りを減少させることは，今後の課題の一つであると考えている．

過分割が多くみられた別要因としてデータの偏りを考えることもできる．今回使用したデータの半数以上は構成語数が 1 つであったため，そもそも過分割が発生しやすい設定の実験になっていた可能性がある（6.1 節を参照）．現在のところ，当該タスクに対する別のデータセットを用意することは難しいため，この点をすぐに調査することはできないが，今後の研究の中で議論を深めていくべきであると考えられる．

7　おわりに

本論文では，言い換えと逆翻字を用いて，片仮名複合語の分割処理の精度を向上させる方法を提案した．提案手法により，大規模なラベルなしテキストを分割処理に利用することが可能となり，分割精度の低下の要因となる未知語の影響を軽減させることが可能となる．実験においては，8 つのベースライン手法との比較を通じて，提案手法の有効性を実証的に示した．

今後の課題としては，提案手法と既存の単語分割手法を融合した解析モデルの構築に取り組

みたい．6.3 節においては，提案手法を後処理に利用可能であることについて言及したが，そうしたアドホックな方法は，学術的立場からは必ずしも満足のいくものではないと考えている．提案手法と既存の単語分割を組み合わせる方法としては，今回提案した素性を統計的な単語分割器に追加することなどが考えられるが，現時点ではその有効性について十分な検証を行うことができておらず，今後調査すべき課題であろう．また，近年では，教師なし学習による単語分割手法も盛んに研究されているが，そうした手法に言い換えや翻字の情報を取り入れることも興味深い問題である (Mochihashi, Yamada, Naonori, and Ueda 2009)．

これに加えて，本論文の中で提案したアイデアを一般化していくことも，今後重要な研究課題になると考えている．本論文では議論の対象を英語由来の片仮名複合名詞に限定していたが，同様の手法は，その他の片仮名語に対しても有効である可能性が高い．例えば，翻字素性は，英語以外の言語からの借用語に対しても有効に働くことが期待できる．また，言い換え素性は，和語や漢語の片仮名表記に対しても有効である可能性が高い（例えば「トンコツラーメン」に対する「トンコツのラーメン」などの言い換え）．さらに，言い換えを単語境界の認識に利用するという考え方は，複合名詞に限らず，単語分割処理一般に対しても適用できる可能性がある．今後はこうした方向についても研究を進めていきたい．

参考文献

Alfonseca, E., Bilac, S., and Pharies, S. (2008). “German Decompounding in a Difficult Corpus.” In *Proceedings of CICLing*, pp. 128–139.

Ando, R. K. and Lee, L. (2003). “Mostly-unsupervised statistical segmentation of Japanese kanji sequences.” *Natural Language Engineering*, **9** (2), pp. 127–149.

Braschler, M. and Ripplinger, B. (2004). “How effective is stemming and decompounding for German text retrieval?” *Information Retrieval*, **7**, pp. 291–316.

Breen, J. (2009). “Identification of Neologisms in Japanese by Corpus Analysis.” In *Proceedings of eLexicography in the 21st centry conference*, pp. 13–22.

Brill, E., Kacmarcik, G., and Brockett, C. (2001). “Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs.” In *Proceedings of NLPRS*, pp. 393–399.

Brown, R. D. (2002). “Corpus-Driven Splitting of Compound Words.” In *Proceedings of TMI*.

Cao, G., Gao, J., and Nie, J.-Y. (2007). “A System to Mine Large-Scale Bilingual Dictionaries from Monolingual Web Pages.” In *Proceedings of MT Summit*, pp. 57–64.

Cherry, C. and Suzuki, H. (2009). “Discriminative Substring Decoding for Transliteration.” In *Proceedings of EMNLP*, pp. 1066–1075.

Freund, Y. and Schapire, R. E. (1999). “Large margin classification using the perceptron algo-

rithm.” *Machine Learning*, **37** (3), pp. 277–296.

Holz, F. and Biemann, C. (2008). “Unsupervised and Knowledge-Free Learning of Compound Splits and Periphrases.” In *Proceedings of CICLing*, pp. 117–127.

Jiampojamarn, S., Kondrak, G., and Sherif, T. (2007). “Applying Many-to-many Alignment and Hidden Markov Models to Letter-to-phoneme Conversion.” In *HLT-NAACL*, pp. 372–379.

Kageura, K., Yoshikane, F., and Nozawa, T. (2004). “Parallel Bilingual Paraphrase Rules for Noun Compounds: Concepts and Rules for Exploring Web Language Resources.” In *Proceedings of Workshop on Asian Language Resources*, pp. 54–61.

Koehn, P. and Knight, K. (2003). “Empirical Methods for Compound Splitting.” In *Proceedings of EACL*, pp. 187–193.

Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). “Applying Conditional Random Fields to Japanese Morphological Analysis.” In *Proceedings of EMNLP*, pp. 230–237.

Kurohashi, S. and Nagao, M. (1994). “Improvements of Japanese Morphological Analyzer JUMAN.” In *Proceedings of the International Workshop on Sharable Natural Language Resources*, pp. 22–38.

Mochihashi, D., Yamada, T., Naonori, and Ueda (2009). “Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling.” In *Proceedings of ACL*, pp. 100–108.

Nakazawa, T., Kawahara, D., and Kurohashi, S. (2005). “Automatic Acquisition of Basic Katakana Lexicon from a Given Corpus.” In *Proceedings of IJCNLP*, pp. 682–693.

Neubig, G., Nakata, Y., and Mori, S. (2011). “Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis.” In *Proceedings of ACL*, pp. 529–533.

Oh, J.-H. and Isahara, H. (2008). “Hypothesis Selection in Machine Transliteration: A Web Mining Approach.” In *Proceedings of IJCNLP*, pp. 233–240.

Okazaki, N., Ananiadou, S., and Tsujii, J. (2008). “A Discriminative Alignment Model for Abbreviation Recognition.” In *Proceedings of COLING*, pp. 657–664.

Schiller, A. (2005). “German Compound Analysis with wfsc.” In *Proceedings of Finite State Methods and Natural Language Processing*, pp. 239–246.

Schwartz, A. S. and Hearst, M. A. (2003). “A simple algorithm for identifying abbreviation definitions in biomedical text.” In *Proceedings of PSB*, pp. 451–462.

Tsujimura, N. (2006). *An Introduction to Japanese Linguistics*. Wiley-Blackwell.

Wu, X., Okazaki, N., and Tsujii, J. (2009). “Semi-supervised Lexicon Mining from Parenthetical Expressions in Monolingual Web Pages.” In *Proceedings of NAACL*, pp. 424–432.

略歴

鍛治　伸裕：2005 年東京大学大学院情報理工学系研究科博士課程修了．情報理工学博士．東京大学生産技術研究所産学官連携研究員および特任助教を経て，現在，同特任准教授．CGM テキストの解析を中心とした自然言語処理の研究に興味を持つ．

喜連川　優：1983 年東京大学工学系研究科情報工学専攻博士課程修了．工学博士．東京大学生産技術研究所講師．助教授を経て，現在，同教授．東京大学地球観測データ統融合連携研究機構長．東京大学生産技術研究所戦略情報融合国際研究センター長．文部科学官．2005 年から 2010 年まで文部科学省「情報爆発」特定研究領域代表，2007 年から 2009 年まで経済産業省「情報大航海プロジェクト」戦略会議委員長，情報処理学会フェロー，2008 年から 2009 年まで副会長，データベース工学の研究に従事．

（2011 年 6 月 17 日　受付）
（2011 年 10 月 10 日　再受付）
（2011 年 12 月 27 日　再々受付）
（2012 年 1 月 11 日　採録）