翻字と言い換えを利用した片仮名複合語の分割

鍛治 伸裕　　喜連川 優
東京大学 生産技術研究所
{kaji,kitsure}@.tkl.iis.u-tokyo.ac.jp

1　はじめに

外国語からの借用 (borrowing) は，日本語における語形成の 1 つとして知られている [7]．特に英語からの借用によって，新造語や専門用語など，多くの言葉が日本語に持ち込まれている．こうした借用語は，主に片仮名を使って表記されることから片仮名語とも呼ばれる．もう 1 つの代表的な語形成として，単語の複合 (compounding) がある [7]．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，日本語の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 (韓国語，ドイツ語，オランダ語など)．そのような言語を対象とした場合，複合語を単語に分割する処理は，統計的機械翻訳 [5]，情報検索 [2]，複合名詞の意味解釈，略語認識などを実現する上で重要な技術となる．例えば，複合語「モンスターペアレント」を「モンスター」と「ペアレント」に分割することができれば，その略語「モンペ」を認識，生成するための手がかりになることが期待できる．

複合語の分割処理を行うためには，言語資源の活用が鍵となる．例えば，単語辞書が複合語分割に有用であることは直感的に明白である [1]．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，過去の研究において指摘されている [3, 5, 6]．そのような対訳資源は，特に英語からの借用語を対象とした場合に有用となる．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，その対応関係から複合語の分割規則を学習することが可能になる．

このような言語資源に依存した複合語分割手法においては，辞書や対訳資源に出現しない未知語の扱いが問題となる．特に日本語を対象とした場合，冒頭で述べたように片仮名語は生産性が高く，その大半は既存の言語資源に出現しないことから，片仮名複合語の扱いが技術的な課題の 1 つとなる [6]．

この問題を解決するために，本論文では，大規模な生テキストを片仮名複合語の分割に活用するための方法を 2 つ提示する．第 1 に，片仮名語の多くが英語を翻字したものであることに着目して，その元となる英語表現をテキストから抽出し，その情報を識別モデル学習のための素性として活用する方法を提案する．第 2 に，英訳表現を利用する既存手法のアナロジーとして，同一言語への翻訳 (=言い換え) の頻度情報を，同様の素性として利用することを提案する．

2　関連研究

これまでにも，生テキストを用いた複合語の分割手法として，複合語の構成語の頻度に基づく手法が提案されている [5, 6]．これに対して本論文では，翻字と言い換えの利用という，従来の頻度に基づく手法とは全く異なるアプローチを提案する．そして，比較実験を通して提案手法の優位性を実証的に示す．

片仮名複合語の分割というタスクは，日本語形態素解析の部分問題であるため，既存の形態素解析手法をそのまま適用することが可能である．しかし，片仮名語には未知語が多いこと，片仮名語の分割には品詞や文字種などの素性が有効ではないこと，などの理由から高い精度での解析を行うことは困難になっている [6]．このことを踏まえると，片仮名複合語の分割とは，既存の形態素解析手法での扱いが困難な部分問題に焦点をあてた試みと見ることができる．

3　分割モデル

本論文では，所与の片仮名列 x を単語列 $\boldsymbol{y} = \{y_i\}$ に分割する問題を考える．ただし，片仮名語の多くは名詞であるため，単語 y_i は全て名詞とし，名詞列に分割できないような x は入力されないと仮定する．

表 1: 単語対応付き翻字対の例．下線に付与された番号は単語の対応関係を表す．

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

我々は，この問題を次のような線形モデルに基づく構造予測問題と捉えて，教師あり学習の手法を用いて解くことを試みる．

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

ここで $\mathcal{Y}(x)$ は x に対する分割候補の集合，$\boldsymbol{w}$ は素性の重みベクトル，$\boldsymbol{\phi}(\boldsymbol{y})$ は分割候補 $\boldsymbol{y}$ の素性ベクトル表現である．$\boldsymbol{w}$ は任意の学習アルゴリズムを用いて最適化可能であるが，計算効率を考えて平均化パーセプトロンを用いた．素性には，単語 n-gram ($n = 1, 2$)，単語の文字数 (1，2，3，4，5 以上)，単語が NAIST-jdic[1]に登録されているか否か，の 3 種類に加えて，次節以降で説明する翻字と言い換えにもとづく素性を用いる．

4 翻字にもとづく素性

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．本論文では，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．

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して，表 1 のような，単語単位での対応関係が付与された翻字対をテキストから自動抽出する．これを**単語対応付き翻字対**と呼ぶ．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 (n=1, 2)．以下本節では，テキストから単語対応付き翻字対を自動抽出する方法について説明する．

4.1 括弧表現の利用

日本語テキストにおいては，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ことがある．

(1) アメリカでジャンクフード (junk food) と言えば...

[1]http://sourceforge.jp/projects/naist-jdic

表 2: 「ジャンクフード」と「junkfood」の部分文字列対応 $\mathcal{A}$ の例 ($|\mathcal{A}| = 4$)．

(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

我々は，このような括弧表現から単語対応付き翻字対の抽出を行う．

このことを実現するため，片仮名語と原語の発音の類似性から得られる部分文字列の対応関係を利用する．例えば，以下のような部分文字列の対応関係が与えられたとする．

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

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

ただし，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．括弧表現においても英語は基本的には空白を使って分かち書きされるため，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 対応するように片仮名列を分かち書きできる．そして，その結果として，単語間の対応関係は自明なものとなる．

(2) の場合では，片仮名列「ジャンク」と「フード」が，それぞれ英単語の「junk」と「food」に対応していることが分かり，表 1 のような単語対応付き翻字対が得られる．この方法は形態素解析処理を必要としないため，未知語に起因する解析誤りの影響を受けずに，片仮名複合語を分割して原語との単語対応を求めることが可能となる．

4.2 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 [4] が提案したものと同様の確率モデルを用いる．f と e をそれぞれ片仮名列とアルファベット列とし，$\mathcal{A}$ 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．このとき，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う．モデルのパラメータは翻字対 (f, e) の集合

表 3: 複合語の言い換え規則．X_1 と X_2 はいずれも名詞列を表す．

$X_1X_2 \rightarrow X_1$ の X_2
$X_1X_2 \rightarrow X_1$ する X_2
$X_1X_2 \rightarrow X_1$ した X_2
$X_1X_2 \rightarrow X_1$ な X_2
$X_1X_2 \rightarrow X_1$ 的 X_2
$X_1X_2 \rightarrow X_1$ 的な X_2

から EM アルゴリズムを用いて推定することができる．詳細は文献 [4] を参照されたい．

この確率モデルを用いて，与えられた翻字対 (f, e) の部分文字列対応を次のように決定する．

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

このとき，$\mathcal{A}^*$ に含まれる e_i が空白をまたいでしまうと，$\mathcal{A}^*$ を使って片仮名列 f を分かち書きできない．そこで，アルファベット列 e が空白を含んでいた場合は，あらかじめ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，e_i がそこをまたがないという制約を加えて argmax の計算を行う．表 2 に「ジャンクフード」と「junkfood」に対する部分文字列対応 $\mathcal{A}$ の具体例を示す．

4.3 単語対応付き翻字対の抽出

上記の確率モデルを用いて単語対応付き翻字対の抽出を行う．その手順は次の通りである．まず，括弧で囲まれたアルファベット列 e と，その直前に出現する片仮名列 f の組を抽出して，これを翻字対の候補 (f, e) とする．ただし，正規化のためアルファベットは全て小文字に変換する．次に，f と e の発音の類似性を表すスコアを次のように定義し，その値が閾値 θ を上回ったものを翻字対とみなす．

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

ただし N は英語側の単語数である．スコア中の $\frac{1}{N}$ という項は，語数の多い候補のスコアが小さくなりすぎるのを防ぐために導入している．こうして得られた翻字対に対して，部分文字列対応 $\mathcal{A}^*$ に基づいて単語対応の情報を付与する．これにより，表 1 のような単語対応付き翻字対を得ることができる．

5 言い換えにもとづく素性

次に，複合語の言い換え表現の頻度情報を識別モデルの素性として使う方法を説明する．

複合語内部の単語境界は，その複合語の言い換え表現から推測することが可能である．例えば，以下のような複合語の言い換え表現を考える．

(3) a. アンチョビソースパスタ
b. アンチョビソースのパスタ

例文 (3a) は複合語であり，その内部の単語境界は曖昧である．一方，(3b) は (3a) に助詞「の」を挿入することによって得られる言い換え表現である．もし (3b) のような言い換え表現がテキスト上に観察されれば，(3a) において，少なくとも「アンチョビソース」と「パスタ」の間に単語境界が存在する可能性が高いと考えることができる．

そこでまず我々は，複合語分割のために有効と思われる言い換え規則を人手で作成した (表 3)．言い換え規則はいずれも $X_1X_2 \rightarrow X_1FX_2$ という形をしている．ここで X_1 と X_2 は名詞列，F は助詞「の」などの機能語である．左辺は複合名詞に対応し，右辺はその言い換え表現にあたる．

これらの言い換え規則を用いて，複合語分割のための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て，テキストに出現する複合語の (潜在的な) 言い換え表現の頻度を事前に求める．

(katakana)+ の (katakana)+
(katakana)+ する (katakana)+
...

ここで (katakana) は片仮名 1 文字にマッチする特殊文字を表す．次に，分割候補 $\boldsymbol{y}$ が与えられると，$X_1 = y_1 \ldots y_i$，$X_2 = y_{i+1} \ldots y_{|\boldsymbol{y}|}$ のように設定して，言い換え表現を生成する．そして，言い換え表現の対数頻度 $\log(c+1)$ を y_i と y_{i+1} の単語境界に対する特徴量とする．ただし c は，表 3 の言い換え規則によって生成された 6 つの言い換え表現の合計頻度である．

6 実験

6.1 実験設定

発音モデルの学習に必要なデータは，固有名が日本語に輸入されるときは翻字されることが多いことに着目して自動構築した．日英固有名辞書 (ENAMDICT)[2] から，片仮名表記の固有名とその原英語の組 (55,737 組) を取り出し，それらを全て訓練事例として用いた．

[2] http://www.csse.monash.edu.au/~jwb/enamdict_doc.html

表 4: 片仮名複合語の分割実験の結果.

システム	P	R	F_1	Acc
GMF	39.9	59.8	47.9	46.2
GMF2	66.9	75.5	70.9	73.2
AP	80.1	82.8	81.4	83.2
AP+GMF2	81.0	83.5	82.2	83.9
PROPOSED	**86.0**	**89.0**	**87.5**	**88.4**
JUMAN	73.2	63.3	67.9	72.9
MECAB	73.4	66.7	69.9	73.9

平均化パーセプトロンの訓練に必要なデータは，日英辞書 (EDICT)[3] をもとにして人手で作成した．EDICT から片仮名表記の見出し語とその英訳表現を取り出し，片仮名語と英語の間で 1 対 1 の単語対応が取れるように，人手で片仮名複合語を分割した．これにより 2094 の片仮名複合語に対してラベルの付与を行った．実験では，このデータを用いて 10 分割交差検定を行った．

評価尺度には，片仮名複合語の構成単語に対する適合率 (P)，再現率 (R)，調和平均 (F_1)，および正しく分割された片仮名複合語の割合 (Acc) を用いた．

翻字と言い換えにもとづく素性の導出には，ウェブから収集した 17 億文の日本語テキストを用いた．このテキストから収集された翻字対の数は 200,992，片仮複合語の言い換え表現数は 9,293,942 であった．

6.2 実験結果

表 4 に，提案手法 (PROPOSED) および比較手法による実験結果を示す．比較に用いた手法は以下の通りである．まず，複合語の構成語の頻度にもとづく手法として，構成単語の頻度の幾何平均が最大となる分割を採用する手法 (GMF)[5] と，GMF に構成語の平均長の情報を加えた手法 (GMF2)[6] を用いた．提案手法と同様の教師あり学習にもとづくものとしては，翻字と言い換えにもとづく素性を用いない平均化パーセプトロン (AP)，GMF2 の結果を AP の素性として加えた手法 (AP+GMF2)[1] を用いた．最後に，形態素解析器 JUMAN[4] と MECAB[5] を用いた．

PROPOSED と AP の比較から，提案した 2 つの素性が分割に有効であることが確認できる．また，提案手法は，頻度にもとづく手法 (GMF と GMF2) やそれを素性に追加した手法 (AP+GMF2) よりも精度が高い．このことから，翻字と言い換えは，構成単語の頻度よりも分割に有用な情報であると考えられる．一方，JUMAN と MECAB の精度は，一般的な単語分割の場合と比較して大きく低下している．このことから，既存の形態素解析器にとって，片仮名複合語の扱いが技術的に難しいことが分かる．提案手法は，そうした弱点を補強するものとして位置付けることができる．

[3] http://www.csse.monash.edu.au/~jwb/edict_doc.html
[4] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html
[5] http://sourceforge.net/projects/mecab

7 おわりに

本論文では，片仮名複合語の分割精度向上のため，翻字と言い換えに関する情報を機械学習の素性として用いる方法を提案し，その有効性を実証的に示した．今後は，情報検索や略語認識などの応用において，片仮名複合語の分割処理を導入する効果の検証を行いたい．また，言い換え表現を素性に用いるという考え方は，片仮名複合語の分割に限らず，単語分割一般に対しても適用できる可能性があるため，その有効性の調査も今後の課題としたい．

参考文献

[1] Enrique Alfonseca, Slaven Bilac, and Stefan Pharies. German decompounding in a difficult corpus. In *Proceedings of CICLing*, pp. 128–139, 2008.

[2] Martin Braschler and Bärbel Ripplinger. How effective is stemming and decompounding for German text retrieval? *Information Retrieval*, Vol. 7, pp. 291–316, 2004.

[3] Ralf D. Brown. Corpus-driven splitting of compound words. In *Proceedings of TMI*, 2002.

[4] Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. Applying many-to-many alignment and hidden Markov models to letter-to-phoneme conversion. In *Proceedings of HLT-NAACL*, pp. 372–379, 2007.

[5] Philip Koehn and Kevin Knight. Empirical methods for compound splitting. In *Proceedings of EACL*, pp. 187–193, 2003.

[6] Toshiaki Nakazawa, Daisuke Kawahara, and Sadao Kurohashi. Automatic acquisition of basic Katakana lexicon from a given corpus. In *Proceedings of IJCNLP*, pp. 682–693, 2005.

[7] Natsuko Tsujimura. *An Introduction to Japanese Linguistics*. Wiley-Blackwell, 2006.