Vol.2012-NL-209 No.8
2012/11/22

リムる・ドヤる・ポジる・パフェる
ー Webを用いたカタカナ動詞の言い換え・語源の獲得ー

鈴木 雄登[1,a)]　笹野 遼平[2,b)]　高村 大也[2,c)]　奥村 学[2,d)]

概要：昨今，Web サービスの発達により気軽に Web 上にテキストを投稿することが可能になった．それに伴い，「パフェる」や「リムる」のような新しいカタカナ動詞も多く使用されるようになった．しかしこうしたカタカナ動詞には一見しただけでは意味が推測できないものが多く存在する．そこで本研究では，カタカナ動詞の入力に対して語源と言い換えの２つを出力として提示すればその意味の理解の助けになるとの考えから，格要素の統計的分布を用いてカタカナ動詞の語源と言い換えを獲得する手法を提案する．

キーワード：カタカナ動詞，言い換え，語源

Rimu-ru, Doya-ru, Poji-ru, Pafe-ru : Acquisition Paraphrases and Etymologies of Katakana Verbs from Web Corpora

Yuto Suzuki[1,a)]　Ryohei Sasano[2,b)]　Hiroya Takamura[2,c)]　Manabu Okumura[2,d)]

Abstract: Due to the development of web services, internet users are given a lot of opportunities to upload text to the internet. As a consequence, we can see many new katakana words such as “パフェる (pafe-ru)” and “リムる (rimu-ru).” However, it is sometimes hard to understand their meanings at a glance. Since paraphrases and etymologies can be a good clue to understand their meanings, we propose a method that acquires paraphrase and etymology of katakana verbs with using statistical distribution of case elements.

Keywords: Katakana verb, Paraphrase, Etymology

1. はじめに

昨今，ブログや Twitter などの Web サービスの発達により，ユーザーが容易にテキストを投稿できるようになった．その結果，日常の延長として口語体でテキストが投稿されることが増えたため，「パフェる」や「リムる」といった新しいカタカナ動詞を含むくだけた表現が Web 上に多く見られるようになった．しかし，これらのくだけた表現の中には，以下の例のように同じコミュニティに属する人は意味を理解することができても，同じ文脈を共有しない異なるコミュニティに属する人には意味を理解することが困難な表現が含まれる．

（1） a, ゲーセンで練習してた曲をパフェった！
　　 b, つまらないツイートばかりだったのでリムった．

この「パフェった」とは「制覇した」という意味であり，語源は「パーフェクト」である．これは主にゲームをやる人たちにのみ使われる表現である．また，「リムった」とは Twitter などのマイクロブログで購読していたユーザーを購読から外すことであり，語源は「リムーブ」である．これは主に Twitter などのマイクロブログを利用している

[1] 東京工業大学大学院 総合理工学研究科
Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology
[2] 東京工業大学 精密工学研究所
Precision and Intelligence Laboratory, Tokyo Institute of Technology
[a)] yuto@lr.pi.titech.ac.jp
[b)] sasano@pi.titech.ac.jp
[c)] takamura@pi.titech.ac.jp
[d)] okumura@pi.titech.ac.jp

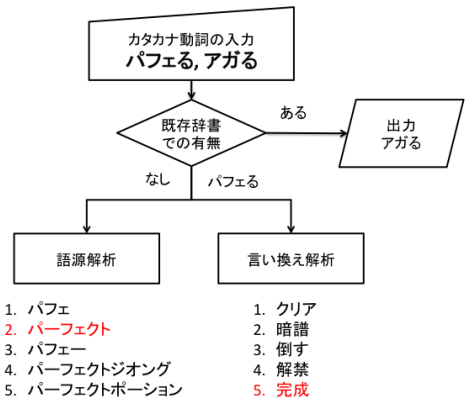

図 1 入力と出力のフロー
Fig. 1 The Flow of Input and Output

ユーザーにのみ使われる．

本研究では，このようなカタカナ動詞の意味を理解する手助けとなるシステムの構築を目指す．カタカナ動詞を理解するためには語源と言い換え両方を提示することが有効であると言える．たとえば，「パフェる」に対して，「完全制覇する」という言い換えのみ提示してもなぜそのような意味で使われるか分からないため，細かい違いなどは分からず，「パーフェクト」という語源を与えられて初めて，その語の意味を深く理解することができると考えられる．そこで本研究では，格要素の統計的分布の類似度を計算することにより語源と言い換えを獲得する手法を提案する．

本研究で想定する入力から出力へのフローを**図 1** に示す．「アガる」のように既存の辞書に存在するカタカナ動詞が入力された場合，辞書を参照することで意味を理解できると考え，処理を終了する．一方，「パフェる」のように既存の辞書に存在しないカタカナ動詞が入力された場合は，語源解析と言い換え解析を行う．それぞれの解析において，スコア順に並べた上位数件を出力する．

2. 関連研究

日本語を対象とした未知語の獲得に関する研究は数多く存在する．Mori and Nagao[4] は，品詞タグ付けされたコーパスを使って品詞ごとの直前直後の文字列の出現確率を求め，候補単語の前後の文字列の出現確率と品詞の出現確率の類似度が閾値以上なら品詞を確定し未知語を獲得する手法を提案している．また，Murawaki and Kurohashi[5] は形態素の語幹に後続し得る付属語列は，品詞ごとに既知の単語の形態素論的制約に従うという考えから，制約充足のチェックにより未知語候補の品詞を絞り込み，未知語の獲得を行っている．鍛治ら [3] はカタカナ用言を Web テキストから自動で獲得する手法を提案している．カタカナ用言には動詞だけでなく，形容詞なども含まれる．鍛治らの手法は，SVM を使ってカタカナの語幹であるか否かを判定する分類器と，形容詞の語幹であるか否かを判定する分類器を学習し，素性にはカタカナ列に後続する文字列を用いてカタカナ用言を獲得するという手法である．ただし，これらの研究はいずれも自動でその単語を獲得するのみであり，言い換えや語源の獲得までは行っていない．

宇野ら [7] は新動詞の形成に注目している．その中でもカタカナが語幹であり，かつ，使用頻度が増える方向に推移している「ファブる」と「モフる」について１０年のスパンでどのようにこの形に至ったかを分析しており，「ファブる」は時系列と共に取る格が変わっていることを指摘している．具体的には，「ファブる」の原形である「ファブリーズする」「ファブリーズをする」の３パターンに着目すると，頻度を計算することによって時間とともに取る格がニ格からヲ格に推移していることを示している．

また，“tmrw” のような辞書に載っていない単語が入力されたとき，その元の形である “tomorrow” を獲得するという研究もある．Bo ら [2] は英語において，マイクロブログ上のテキストの語彙正規化のための辞書構築を提案している．Bo らは，Twitter データから分布類似度を用いて，“tomorrow” と “tmrw” のような既存の単語とその単語が変形した未知語のペアを抽出する手法を提案している．分布類似度からペアを獲得するため “Youtube” と “web” のような誤ったペアが取得されてしまうので，そのようなペアを編集距離で取り除いて出力している．

本研究では，文脈，特に格要素の類似性を利用して言い換えや語源の獲得を行っているが，任意の単語の文脈と言い換えの文脈の類似性を利用して獲得する研究は多い．Bhagat ら [1] は，150GB の巨大コーパスを用いて表層的な共起から任意の二語の類似度を利用し，言い換えの作成を行なっている．Bhagat らは，任意の単語と共起している単語の共起の強さを PMI で重み付けし，その PMI 値を要素としたベクトルを作成し，ベクトル間をコサイン類似度で測るという手法を提案している．Pasca ら [6] はフレーズ単位の n-gram を考え，2 つの n-gram の直前直後が重複している場合，ペアとして保存し，獲得したペアの頻度が閾値以上なら言い換えペアとする手法を提案している．また，重複部分に固有名詞が入っている場合のみ獲得するという制約を加えたほうが良い結果が得られたと報告している．

3. カタカナ動詞の収集と分類

まず，実際にどのようなカタカナ動詞が使用されているのかを調べるため，コーパスからカタカナ動詞を取り出し，それらの分類を行った．

カタカナ動詞の収集には，Twitter のデータを利用した．2011 年 5 月から 2011 年 9 月までの日本語の Twitter デー

モテる，バレる，キレる，イケる，ググる，ウケる，デレる，リムる，パクる，ハゲる，シャワる，ポチる，トゥギャる，
キメる，テンパる，キテる，ボケる，デキる，バテる，ハモる，ツボる，コケる，ズレる，バグる，マミる，ブレる，
メモる，ツッコミる，ツイる，ボコる，

図 2 獲得したカタカナ動詞の一部
Fig. 2 Part of Katakana Verb Acquired

表 1 カタカナ動詞の分類
Table 1 A Classification of Katakana verbs.

タイプ	例	頻度
タイプ 1	アガる，バレる，ウケる	68
タイプ 2	リムる，パフェる，ポジる	118
タイプ 3	タヒる，パシる，ワロる	14
総計		200

タ約 1 億 3 千万ツイートを MeCab[*1] で形態素解析したものを利用した．Twitter を使った理由としては，気軽に投稿できるマイクロブログであり，新しいカタカナ動詞が多く出現しやすいと考えられるためである．カタカナ動詞の抽出法は，Twitter データを形態素解析し，その出力において先行している形態素がカタカナ文字列で，後続している形態素の MeCab 解析結果が「助動詞」もしくは「動詞非自立」であるものを抽出するという方法である．そしてそれらを原形に直したものの頻度を数え，上位 200 個を本研究で使うカタカナ動詞とした．獲得したカタカナ動詞の一部を**図 2** に示す．

カタカナ動詞の語源を元に以下のような 3 つのタイプに分類した．

タイプ 1 「アガる」のように，既存の辞書[*2]に載っている動詞の語幹をカタカナにしたもの．

タイプ 2 語源と別の動詞を用いて言い換えられるもの．二つの種類があり，「リムる」と「リムーブをする」の関係のように，語源と格とサ変動詞に言い換えられるものと「パフェる」と「パーフェクトを達成する」のように語源と格と一般動詞に言い換えられるものがある．

タイプ 3 タイプ 2 の中でも「死」の文字を分解して作成されたカタカナ動詞「タヒる」のように表層から語源の取得が困難なものをタイプ 3 に分類する．[*3]

分類したカタカナ動詞の一例を**表 1** に示す．タイプ 3 については語源を表層から推測することが困難であるため，本研究ではタイプ 2 について注目する．さらに，タイプ 2 を語源の種類によって 3 つに分類することができる．() 内は語源である，

タイプ 2-1 外来語もしくはカタカナ名詞 (84/118)
パフェる (パーフェクト)，リムる (リムーブ)，ポジる（ポジショニング）

タイプ 2-2 オノマトペ (24/118)
イラる (イライラ)，デレる (デレデレ)，ポチる（ポチっと）

タイプ 2-3 語源を一般的に漢字で書くもの (10/118)
キョドる (挙動不審)，コクる (告白)，フロる (風呂)

1 に分類されるものは，語源が「パーフェクト」である「パフェる」のように，語源が外来語，もしくはカタカナ名詞のものである．2 は，語源が「イライラ」である「イラる」のように語源がオノマトペであるものである．最後に 3 は「挙動不審」が語源である「キョドる」のように一般的に漢字で書く語源をもつものである．

4. 提案手法

4.1 言い換えの獲得

言い換えの獲得について説明する．同じ意味の動詞では，同じ格要素を取ることが多いと考えられる．

(2) a, ゲーセンで練習してた曲をパフェった
b, ゲーセンで練習してた曲を制覇した．

上の例を見ると「パフェった」と「制覇した」の意味が類似している二つの動詞が，同じ格要素を取っていることが分かる．その格要素集合をベクトルにして，もしそれらが類似していれば，二つの動詞は類似している意味であると推定できる．したがって，出現する格要素の類似度を計算することによって，二つの動詞の類似度を計算することができる．

そこで，言い換え獲得を格要素ベクトルの作成，格要素ベクトル間の類似度計算によるランキングという手順で行う．獲得方法については**図 3** に示す．3 節で説明した方法でカタカナ動詞を収集し，見つけたカタカナ動詞に係る格要素を見つける．本研究では，格要素としてヲ格，デ格，ニ格のみを使用する．多くの格構造解析ではガ格，ヲ格，ニ格が使われるが，ガ格よりもデ格の方が，動詞の特徴に合わせた要素が含まれやすいと考えたためデ格を利用する．

*1 http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html
*2 本研究では IPA 辞書を用いた
*3 この「タヒる」は「死ぬ」という意味であり，由来は「死」という文字を分解した右側と左側をカタカナと解釈したことからである．

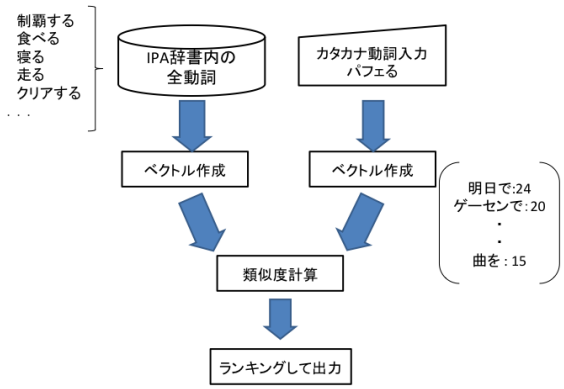

図 3 言い換え獲得のフロー
Fig. 3 Flow of Paraphrase Acquisition

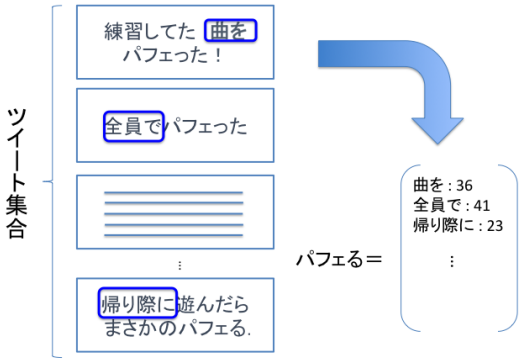

図 4 ベクトル作成の例
Fig. 4 Example of Making Vector

(3) a, 先輩がパフェってた
b, 片手連打でパフェった

上の例を見てみると，ガ格には「先輩」という動作主が入りやすいので，他の動詞との差別化を行いづらいが，デ格である「連打」は，その動詞特有の要素を持ちやすい．以上より，ヲ格，ニ格，デ格の格要素の出現回数を数え，その頻度を要素としたベクトルを作成する．これを格要素ベクトルと呼ぶことにする．

次に格要素ベクトルの類似度計算の方法について述べる．辞書に含まれる動詞全てに対して格要素ベクトルを作成し，入力されたカタカナ動詞の格要素ベクトルと類似度計算を行う．類似度計算には，Jaccard 係数とコサイン類似度を 2 つを用いる．

$$Sim_{\text{jaccard}} = \frac{|A \cap B|}{|A \cup B|} \quad \text{(i)}$$

$$Sim_{\text{cosine}} = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{|\boldsymbol{a}||\boldsymbol{b}|} \quad \text{(ii)}$$

ただし，A は入力されたカタカナ動詞の格要素集合，B は言い換え候補の格要素集合で，$\boldsymbol{a}, \boldsymbol{b}$は入力されたカタカナ動詞と言い換え候補の格要素ベクトルである．

図 4 にベクトル作成の例を示す．

4.2 語源の獲得

次に語源の獲得について考える．

(4) a, ゲーセンで練習してた曲をパフェった．
b, ゲーセンでパーフェクトを取れた．

上の例のように，語源の「パーフェクト」は名詞であることが多く，語源とカタカナ動詞の類似度の比較を行うには名詞と動詞を比較しなければならない．したがって，一般的に同じ品詞を扱う分布類似度をそのまま用いることはできない．しかし，上の例を見ると，語源が含まれる文と，カタカナ動詞が含まれる文は同一文内で同じ格要素が存在

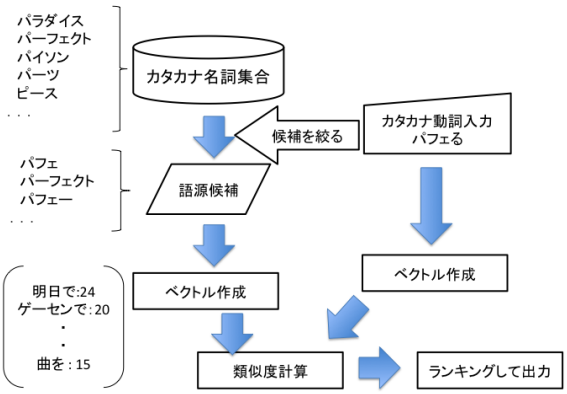

図 5 語源獲得のフロー
Fig. 5 Flow of Etymology Acquisition

することが多い．また Twitter という媒体自体に 140 字という制約があるため，一文が短いものがほとんどである．この制約のおかげで，新聞や書籍などの一文が長い媒体と違って．格要素に関係のない要素が入りづらいため同一文内の格要素を比較することができると考える．

(4)a では，カタカナ動詞である「パフェった」の格要素は「ゲーセンで」と「曲を」であり，(4)b では，語源候補の「パーフェクトを」に係る動詞の格要素が「ゲーセンで」となり，「ゲーセンで」が共通の格要素として存在する．そこで，語源候補に係る動詞の格要素をベクトルの要素とし，語源候補の格要素ベクトルを作成することにする．

語源の獲得方法は以下の 3 つのステップに分かれる．

1 ルールで語源候補を絞る

2 語源候補と入力されたカタカナ動詞に関して格要素ベクトルを作成する．

3 格要素ベクトルを用いて類似度計算を行う．

ほとんどのカタカナ動詞はカタカナ名詞から派生してカ

タカナ動詞となっているので，まず最初に Twitter から収集してきたカタカナ名詞からルールベースで語源候補を絞る．カタカナ動詞を観察した結果よりルールを作成した，具体的には，カタカナ名詞を前から順に見ていき，カタカナ動詞と一致しているかを比較していく．もし，長音，促音以外の文字で入力されたカタカナ動詞の語幹とマッチしていなければ候補から外すというルールを適用した．たとえば「パフェる」と入力された時，「パーフェクト」「パフェ」は候補に含まれるが，「パフューム」は候補対象外となる．また「ゲトる」という入力に対して，「ゲット」や「ゲート」は語源候補になりうるが，「ゲスト」は候補対象外となる．

続いて，絞り込んだカタカナ名詞からベクトルを作成する．言い換え獲得の際もベクトル作成行ったが，語源候補とカタカナ動詞は名詞と動詞の比較になるため作成方法が異なる．入力されたカタカナ動詞を含む文と語源候補が格要素として出現している文では同一の格要素を持ちやすいことから，入力されたカタカナ動詞の格要素と語源候補がヲ格，ニ格，デ格として出現している文の他の格要素をベクトルとして抽出し，比較する．先ほどの例 (4)b を見ると，語源候補「パーフェクト」の他の格要素「ゲーセンで」をベクトルの要素として追加する．このようにして，語源候補のベクトルに関しては，語源候補がヲ格，ニ格，デ格として出現している文脈内の他のヲ格，ニ格，デ格の格要素をベクトルに追加する．入力されたカタカナ動詞の格要素ベクトルに関しては，言い換えの時と同じものを利用する．

最後にベクトル間で類似度計算を行う．計算には同じく Jaccard 係数とコサイン類似度を用いた．計算後，類似度順に出力する．

さらに，出力した結果において，上位に位置した単語で辞書に含まれていないような単語が多く見受けられたため，語源候補から IPA 辞書または Wikipedia の見出し語に載っていない単語を除外した．ただし，商品名や企業名からもカタカナ動詞は多く作られるため，Wikipedia に掲載されている単語を IPA 辞書とあわせて利用した．たとえば，「チャリる」を見てみるとランキングの最初に来ている候補は「チャリチャリ」で Wikipedia にも IPA 辞書にも載っていないが，ランキングの 2 番目にが来ている正解の「チャリンコ」IPA 辞書には載っている．

5. 実験

5.1 使用するデータと評価指標

類似度計算には，3 節で述べた Twitter から取得した形態素解析済みデータを用いる．言い換えの手法の評価として，入力されたカタカナ動詞に対してどれだけ意味が似ている言い換えを提示できるかを評価する．提案手法では，出力は類似度計算によってランキングされているため，正解となる出力がなるべく高い順位に位置していることが望ましい．また，出力された結果には似たような表現が複数あることが多い．そこで指標に情報検索でよく使われている MRR (Mean Reciprocal Rank) を利用する．MRR は正解がどのくらい上位に出て来やすいかをスコア化したものである．

表 2 言い換え獲得結果
Table 2 Result of the acquired Paraphrases

	MRR	top acc@5	top acc@3	top acc@1
cosine	0.491	0.609	0.562	0.375
Jaccard	0.479	0.453	0.406	0.297

※ Accuracy@N のことを top acc@N と表記した．

$$MRR = \frac{1}{|R|}\sum_{i=1}^{R}\frac{1}{Rank} \quad \text{(iii)}$$

ここで R は入力したカタカナ動詞の数である．Rank は最上位に出現した正解の順位であり，Count は出現した正解数である．言い換えの正解は複数あるため，複数語義がある場合でも本研究ではランキングの最上位に来た言い換えを MRR の評価に利用する．もう一つの評価指標として，N 位以内に正解があれば言い換えを獲得できているという指標も行った．ここでは，Accuracy@N と呼ぶことにする．この Accuracy@N は，上位 N 件以内に正解がある確率である．また全ての正解データを作成するのは困難であるため，出力に対し人手で評価した．二人で評価を行い，二人の評価における一致度を測る指標であるカッパ係数は 0.84 であった．

次に語源獲得の評価について述べる．語源の場合は，一意に決まることが多い．例えば「リムる」で語源は「リムーブ」である．そのため，正解データを事前に作成した．MRR の場合は最上位のみ評価するので MRR を用いる．また同様に Accuracy@N も用いる．

5.2 言い換え獲得の結果

言い換え獲得における MRR のスコアを**表 2** に示す．MRR は類似度としてコサイン類似度を用いた場合の方がスコアが高かった．また Accuracy@N においても同様にコサイン類似度を用いた場合の方がスコアが高かった．

言い換え獲得例を**表 3** に示す．太字は正解である．Jaccard 係数は頻度を考慮に入れない類似度計算であるが，コサイン類似度は正規化はしているものの頻度の影響が少なからずある．その影響で，「ポジる」の格要素で多かった「前に」に対し，「ミーティングする」の格要素としても多く，頻度を考慮しない Jaccard 係数の場合，一つの格要素に影響を受けることは少なく正解がランキング上位に出現しているが，コサイン類似度ではランキング上位に出現しなかった．逆に，コサイン類似度は格要素の種類が少ない場合に強く，頻度が高い格要素が一致しているとスコアが高くなるため，「コラボる」の出力の「合作する」は格要素

表 3 言い換え獲得例
Table 3 Example of the acquired Paraphrases

	1	2	3	4	5
cosine					
パフェる	**クリア**	暗譜	倒す	解禁	**完成**
類似度	0.381	0.331	0.314	0.308	0.306
コラボる	**合作**	演奏	練習	**重奏**	レコーディング
類似度	0.637	0.571	0.550	0.549	0.523
ハモる	**輪唱**	合作	**合唱**	熱唱	歌う
類似度	0.527	0.527	0.514	0.514	0.493
リムる	追い詰める	フォロー	泣かす	殺す	発狂
類似度	0.555	0.539	0.483	0.468	0.467
リプる	質問	忠告	メール	愚痴る	電話
類似度	0.499	0.495	0.462	0.459	0.447
ポジる	ミーティング	テスト	予習	凍死	呟く
類似度	0.386	0.379	0.362	0.340	0.339
Jaccard					
パフェる	**クリアー**	平らぐ	連奏	召し上がる	**完走**
類似度	0.124	0.111	0.110	0.104	0.103
コラボる	探す	解禁	鑑賞	制作	消化
類似度	0.145	0.142	0.142	0.139	0.136
ハモる	**合唱**	言い合う	唄う	発声	踏む
類似度	0.122	0.120	0.116	0.114	0.114
リムる	退会	**抹消**	**消去**	凍結	軽蔑
類似度	0.170	0.161	0.156	0.148	0.145
リプる	**リプライ**	メール	呟く	コメント	**返事**
類似度	0.214	0.192	0.191	0.189	0.188
ポジる	きょろきょろ	**陣取る**	捻る	ロール	編み込む
類似度	0.0968	0.0946	0.0892	0.0874	0.0853

表 4 語源獲得結果
Table 4 Result of the Acquired Etymologies

辞書の有無	MRR	TopAcc@5	TopAcc@3	TopAcc@1
なし (cosine)	0.433	0.639	0.583	0.259
なし (Jaccard)	0.677	0.889	0.815	0.519
あり (cosine)	0.462	0.569	0.539	0.363
あり (Jaccard)	0.658	0.725	0.716	0.598

※辞書ありは語源候補を Wikipedia と IPA 辞書で絞った場合である

の種類は少ないのだが，頻度の高い格要素を共に持っているのでランキング上位に正解が出現している．

また，格要素をあまり取らない動詞の獲得が，最も失敗しやすいケースである．本稿の手法では，格要素を類似度計算の要素に使っているので，獲得失敗に至りやすい．Jaccard 係数は頻度が少ない場合に，性能がよく，頻度が多いときはコサイン類似度が良い．今後，頻度に応じて類似度計算手法を選ぶことも考えられる．

5.3 語源獲得の結果

表 4 に語源獲得結果を示す．太字は正解である．まず「辞書なし」はカタカナ動詞の中でも 3 で分類した「語源＋(格) ＋動詞」に言い換えられるもののみで評価した値である．次に「辞書あり」は，語源候補を IPA 辞書と wikipedia にある単語の中に含まれていなかったら，語源候補から外すという制約を加えて実験した結果である．**表 5** に語源獲得例を示す．

実験結果において，現在 Top acc@1 における語源獲得精度が 5 割程度である．誤って出力してしまったものを調査したところ，次のような原因が考えられた．まず，語源の格要素ベクトルを作成する際，候補の単語は (1) 形態素解析で名詞と判定されたもの，(2) 候補の単語の直後に格助詞がくるもの，の二つを満たしているものを使い，そこから同一文内の格要素を取得し，ベクトルを作成している．したがって，候補の単語は名詞であるのだが，一見動詞が獲得されてしまっているように見えるものに「リプ」がある．「リプる」というカタカナ動詞のランキング 1 位は，「リプ」である．これは一見「リプる」の語幹のように見えるが，「リプる」の正解の語源である「リプライ」がカタカナ動詞となって「リプる」になった後，更にその「リプ」が名詞形となって使われてるようになったと考えられる．このような派生した語から更に派生するというものがいくつか見受けられ，これらは本質的には同じものであり類似度が等しいことは当然なのだが，語源を取得するという観点から

表 5　Jaccard 係数を使った語源獲得例
Table 5　Example of the Etymologies Acquired that is ranked by Jaccard Similarity

	1	2	3	4	5
パフェる	*パフェ	***パーフェクト**	*パフェー	*パーフェクトジオング	パーフェクトポーション
類似度	0.0911	0.0767	0.0217	0.0107	0.00282
リムる	**リムーブ**	リムブロ	リム	*リムジン	*リムーバー
類似度	0.214	0.213	0.153	0.0839	0.0378
コラボる	*コラボ	***コラボレーション**	コラボキャンペーン		
類似度	0.504	0.0879	0.00709		
ハモる	ハモリ	*ハモ	***ハーモニー**	*ハーモニカ	*ハモネプ
類似度	0.151	0.147	0.112	0.103	0.0732
リプる	リプ	***リプライ**	*リップ	*リプレイ	*リプトン
類似度	0.382	0.235	0.0978	0.0922	0.0497
チャリる	チャリチャリ	チャリンコ	チャリン	チャリンチャリン	チャリチョコ
類似度	0.0985	0.0817	0.0589	0.0566	0.0563
ポジる	***ポジショニング**	*ポジティブシンキング	*ポジ	*ポージング	*ポジティブ
類似度	0.0778	0.0738	0.0663	0.0640	0.0420

*は wikipedia または IPA 辞書に載っている単語である．

は精度の低下を招いてしまっている．また 3 節で分類した語源が一般的に漢字で書かれるものも，現在カタカナのみで検索しているので，取得できていない．

また，語源獲得の結果の Wikipedia と IPA 辞書を使って候補単語に制約を加えた場合を比較すると，Top acc@1 は制約を加えたほうが良いが，Top acc@3，Top acc@5 の場合は制約を加えないほうが良い結果となっている．これは，制約を加えることで正解の語源が除かれてしまっているためである．IPA 辞書や Wikipedia にはオノマトペがほとんどないため，語源の獲得に失敗しているケースが見られる．この結果からランキング 1 位の場合のみ，IPA 辞書や Wikipedia に載っているかを考慮することも考えられる．

6. おわりに

本研究では，カタカナ動詞の語源と言い換えを獲得する手法を提案することによってカタカナ動詞の意味の理解に貢献した．実験において，1 位のみ表示では約 6 割程度，および 3 位まで表示だと 9 割ほどの精度で語源を獲得することができた．また，言い換え獲得では，1 位に出現する確率は約 4 割程度の精度で獲得できた．

今後の課題として，本稿で対応できない語源が漢字で書かれるものや，辞書に含まれていないオノマトペの場合などにも語源を獲得できるようにすること，言い換えの獲得において格要素ベクトルに制約をつけてさらなる精度向上を目指すことが挙げられる．

参考文献

[1] Rahul Bhagat and Deepak Ravichandran. Large scale acquisition of paraphrases for learning surface patterns. In *Proceedings of ACL-08: HLT*, pages 674–682, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[2] Bo Han, Paul Cook, and Timothy Baldwin. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 421–432, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[3] Nobuhiro KAJI, Ken'ichi FUKUSHIMA, and Masaru KITSUREGAWA. Acquisition of katakana verbs and adjectives from large web text. *The IEICE transactions on information and systems (Japanese edetion)*, 92(3):293–300, 2009.

[4] Shinsuke Mori and Makoto Nagao. Word extraction from corpora and its part-of-speech estimation using distributional analysis. In *Proceedings of the 16th conference on Computational linguistics - Volume 2*, COLING '96, pages 1119–1122, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[5] YUGO MURAWAKI and SADAO KUROHASHI. Online acquisition of japanese unknown morphemes using morphological constraints. *Journal of natural language processing*, 17(1):55–75, 2010.

[6] Marius Paşca and Péter Dienes. Aligning needles in a haystack: paraphrase acquisition across the web. In *Proceedings of the Second international joint conference on Natural Language Processing*, IJCNLP'05, pages 119–130, Berlin, Heidelberg, 2005. Springer-Verlag.

[7] 宇野 良子, 鍛治 伸裕, and 喜連川 優. Exploring from/meaning interaction through the analysis of newly created verbs in japanese. *Proceedings of the Annual Meetings of the Japanese Cognitive Linguistics Association*, 10:377–386, 2010.