TR-SLT-0059

コーパス中の異表記統一手法

A method to detect variant notations in a large corpus

関口 洋一　　大竹 清敬
Youichi SEKIGUCHI　　Kiyonori OHTAKE
坂本 仁
Masashi SAKAMOTO

2004.2.20

コーパス中に存在する送り仮名，混ぜ書きならびに，カタカナ語のゆれを解消するための手法を提案する．カタカナ語のゆれを解消するためにカタカナ語に特化した編集距離，ならびに文脈類似度と呼ぶ該当カタカナ語の周辺情報を用いた類似度を定義し，これら2つを組み合わせ，総合的に表記ゆれを判定する手法を提案する．旅行会話基本表現集を用いた実験の結果，通常の編集距離を用いる方法より良い精度を得ることができた．

(株) 国際電気通信基礎技術研究所
音声言語コミュニケーション研究所
〒619–0288 京都府相楽郡精華町光台二丁目2番地2 TEL: 0774-95-1301

Advanced Telecommunication Research Institute International
Spoken Language Translation Research Laboratories
2-2-2 Hikaridai Seika-cho Soraku-gun Kyoto 619-0288, Japan
Telephone: +81-774-95-1301
Fax: +81-774-95-1301

目 次

1 はじめに

近年，コーパスを用いた自然言語処理の有効性は広く認知され，実用化されるに至っている．それと同時に実用的な自然言語処理を実現するためには，より大規模なコーパスが求められる．

一方で，日本語には使用する文字種の多さ，その表音文字体系などから表記のゆれが多く存在する．この表記のゆれは検索における不一致をはじめとして，自然言語処理の様々な点で問題を引き起こすため，コーパスにおける表記は統一されていることが望まれる．

また，大規模なコーパスを構築するために，一つの情報源のみを用いることは制約が大きく，達成可能なコーパスの規模も限定される．そこで，大規模なコーパスを実現するために，複数の情報源を用いることになる．しかしながら，複数の情報源を用いることは，表記の統一に対して複数の基準を含むことになり，表記の統一は保たれない．WWW に代表される不特定多数の人間によって記述されたコーパスはそのような複数の基準を含む典型例である．さらに，表記の統一の基準以前に，誤りが混入している可能性がある場合には，表記のゆれを統一することが困難になる．

表記のゆれの種類には，送りがな (行う-行なう)，文字種 (猫-ねこ-ネコ)，カタカナ (メード-メイド) が存在する．なかでもカタカナは，西洋からの外来語やオノマトペ，植物ならびに動物の名前，外国の人名や地名，強調，俗語などを表すために用いられる．このことから，カタカナ語は増加する傾向にあると言える．外来語に対するカタカナ語は，基本的にその外来語の原語の発音の近似として表記される．そこに，ある程度の規則性はあるものの，すべてを網羅することはできず，表記ゆれが生じる．

本研究では，カタカナ語の表記ゆれに焦点をあて，あるカタカナ語に対してコーパス中でその表記ゆれと考えることができる文字列を特定する方法を提案する．このようなカタカナ語の表記ゆれの問題は以前から広く認識され，いくつかの研究も行われてきた [Shi94, Kub94]．しかしながら，それらの研究が対象としていたのは，辞書などの比較的よく整備された言語資源であり，さまざな表記ゆれや誤りの混入も考えられる大規模なコーパスを想定したものではない．また，これらの研究は基本的に規則に基づく方法を用いており，あらかじめ予想される表記ゆれを特定するための規則による手法だけでは，誤りによってゆれている表記を特定することは困難である．

そこで，本報告では，そのような誤りに対しても頑健な表記ゆれ特定方法として，編集距離に基づいた類似度と，文中の共起単語ならびに，文の依存構造を利用した文脈ベクトル間の類似度を用いて表記ゆれを特定する方法を提案する．また，送りがな，文字種のゆれを特定する方法について検討した結果も報告する．カタカナ語の表記揺れ特定手法に関しては，旅行会話基本表現集 (BTEC) を用いて実験を行ったので，それについても報告する．

2 表記ゆれ

表記ゆれの種類はいくつか存在する．コーパス中でよく見られるものとしては，次のものがある．

- 送り仮名
- 文字種
- 誤字・脱字
- カタカナ
- 省略語

これらの例を挙げながら分類し，提案する手法で対象とする表記ゆれを定義する．

2.1 種類と分類

(2.1.1) 送り仮名

送り仮名においてゆれやすいタイプは，１文字で読みが長いものや複合語である．たとえば，読みの長い単語でゆれているものには次のようなものがある．

- 「快い」 $\longleftrightarrow$ 「快ろよい」
- 「捕らえる」 $\longleftrightarrow$ 「捕える」

また，複合語では次のようなものがある．

- 「取り扱い」 $\longleftrightarrow$ 「取扱い」
- 「当たり外れ」 $\longleftrightarrow$ 「当り外れ」

(2.1.2) 文字種

日本語で文章を書くときには，さまざまな文字を用いる．漢字，ひらがな，カタカナ，アルファベット，数字などである．文章の中で強調したりするときに，通常は漢字で書くところをカタカナで書くことも多い．文字種による表記ゆれの例は以下のようなものである．

- 「大勢」 $\longleftrightarrow$ 「おおぜい」
- 「皮膚」 $\longleftrightarrow$ 「皮フ」
- 「ワイシャツ」 $\longleftrightarrow$ 「Ｙシャツ」

(2.1.3) 誤字・脱字

誤字には文を入力する際に誤って変換してしまったものや，入力者の誤った知識によるものがある．例を以下に示す．

- 「栄養」「影響」
- 「調査値」「調査地」

また，脱字は本来何らかの原因によって単語の一部の文字が欠けてしまうことで，以下のような例がある．

- 「オーストラリア」≠「オーストリア」
- 「下さい」≠「下い」

(2.1.4) カタカナ

外来語をカタカナで表記する場合は，その原語における発音を近似してカタカナ表記する．近似に際してある程度の規則性は存在するが，頻繁にゆれる．一方で，発音の近似と文字列に対する近似の2つの表記があり，ゆれる場合がある．たとえば，英語の 'report' の発音の近似としてリポートがあり，文字列表記の近似の結果としてレポート (repo^to) がある．

- シベリア ⟷ シベリヤ
- ロサンジェルス ⟷ ロスアンゼルス
- ミネラルウォーター ⟷ ミネラルウオータ

また，近年英語を訳さずそのまま日本語に定着してしまう例が増えている．「リストラする」といったものである．したがって，外来語の原語の発音の近似表現であるカタカナ語が日本語における新語として増加する．新語が一般に広く用いられるようになるまでは，ゆれる傾向がある．たとえば，英語の形容詞 'mobile' に対してモービルとモバイルの両方が用いられる（たとえば，モバイルパソコン/モービルパソコン）が，今では主にモバイルが用いられる．

(2.1.5) 省略語

省略語は，もとの単語の部分文字列から構成され，もとの単語よりも短い単語と定義される．省略語の例として以下のものをあげる．

- コンビニエンスストア ⟶ コンビニ
- 電子計算機 ⟶ 電算機
- 携帯電話 ⟶ 携帯
- シングルベット ⟶ シングル

ここで，省略語は完全に文脈に依存し，意味が決定されるものが少なくない．たとえば，上の例で「シングル」とだけ言ったときに，それが「シングルベット」をさすものなのか，酒の割合である「シングル」をさすものなのかはわからない．

2.2 対象

既に述べたように表記ゆれにはいくつかの種類が存在し，それらは複合的に発生する．たとえば，省略語のゆれを解消する際に，それと同時に起こり得る表記ゆれにも対処しておかなければ，その処理は煩雑になる．具体的には，

コミュニティーセンター ⟷ コミセン

のような省略語を解消する場合,「コミセン」に「コミュニティセンター」が一対一の関係で成り立っていれば簡単に解消できる．しかし，実際のコーパスでは「コミュニティ」を「コミニュティ」と書く人もいる．そのためこのような小さなゆれを解消し，それから省略語のように大きなゆれに対処する必要がある．

本研究では，文字種，送り仮名，カタカナに付随する表記ゆれについて，ある単語に対する上記の表記ゆれをコーパスから特定する手法を提案する．

3 文字種と送りがな表記ゆれの特定手法の検討

この節では，本研究で提案する文字種と送りがな表記ゆれの特定方法について説明する．対象とする表記ゆれをコーパスから特定するために，入力として与えられた単語に対してその表記ゆれ文字列を生成し，その文字列をコーパスから検索する生成的アプローチをとる．

ひらがな単語(ひらがなのみによって構成される単語)のゆれとして，それに対応するかな漢字文字列を特定することは，かな漢字変換と同等の問題であり，単語のみを入力とする状況では，非常に困難な問題となる．本研究では，入力として与える単語は1文字以上の漢字を含む単語で，なおかつその読みが正確に与えられているものとする．

3.1 表記ゆれ特定方法

まず，読みと漢字の対応をとる．そして，入力が複数の漢字を含む場合は，表記ゆれがそれぞれの組みあわせとして起こりうるので，可能性のある組み合わせはすべて検討する．また，入力が動詞などの活用させることができる単語の場合は，活用させた候補を生成する．以下に詳細な手順を示す．

(3.1.1) 表記ゆれ候補生成

入力として与えられた単語に対して，表記ゆれ候補を生成する．候補を生成する際には，入力単語を分割し(複合動詞/複合名詞の場合)，単動詞/ 単名詞とする．その後送りがなを変更した候補をそれぞれ作成し，組み合わせる．また入力単語を活用させることができる場合は，活用させた候補を作成する．

複合動詞/名詞はその間に送り仮名のゆれが含まれる場合がある．その例を以下に示す．また，例2のような複合名詞では混ぜ書きの問題も含まれる．

(例1) 取り扱う ⟷ 取扱う

(例2) 踏み切り ⟷ 踏みきり ⟷ 踏切

(例3) 空き缶 ⟷ 空缶 ⟷ あきかん ⟷ アキカン

従って，生成する候補は，漢字のみ，漢字とひらがな混在，漢字とカタカナ，ひらがなのみ，カタカナのみによって構成される5種類となる．

(3.1.2) 分割と組み合わせ

図1に示した送り仮名のゆれは，複合動詞であるために「取り」の部分でゆれが生じる．そこで複合動詞の場合には入力単語の読みを用いて分割をする．入力単語にひらがなを含んでいる場合には，そのひらがなを手がかりにして読みを分割することができる．一方で「取扱う」の様に区切る部分にひらがなが無い場合には，後ろから単語を調べ，その読みが一致し，漢字を含む最短の部分で区切る．

次に分割した単語とその読みを用いて候補を生成する．すなわち「取り」に対して「取」と「とり」を,「扱う」に対しては「扱」,「扱う」,「扱かう」,「あつかう」の候補をそれぞれ

生成する．この後，すべての組合せを生成し，表記ゆれ候補とする．さらに入力単語が活用させることができる場合は，活用させ，候補を生成する．

以上の処理によって得られた単語を候補群とし，コーパス中から探し出す．

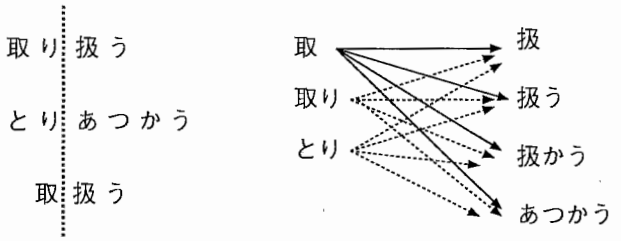

図 1:「取り扱う」の分割・組み合わせ例

3.2 問題点

候補群に含まれる単語がコーパス中に存在するかを調べるが，文字列が存在したとしてもそこから切り出して良いのか判断する必要がある．この解決手法としては，形態素解析器によりその単語を含む文を解析する．そしてその単語部分が分割されていれば，辞書に登録されていないために誤った解析をしている可能性があるので候補にする．一方で正確に解析された場合であっても解析範囲された範囲と合致しているようであれば，候補とすることができる．

また，対象となる漢字に複数の読みがあるとき，どの候補を用いるか判断することが必要になる．

4 カタカナ語の表記ゆれ抽出

本研究で提案するカタカナ語の表記ゆれ抽出法は，以下の 3 つから成り立つ．

- カタカナに特化した編集距離 (kED)
- ベクトル空間法を用いた文脈類似度
- 文字長を用いたヒューリスティック

図 2 に本手法で提案するカタカナ語の表記ゆれ抽出の概要を示す．

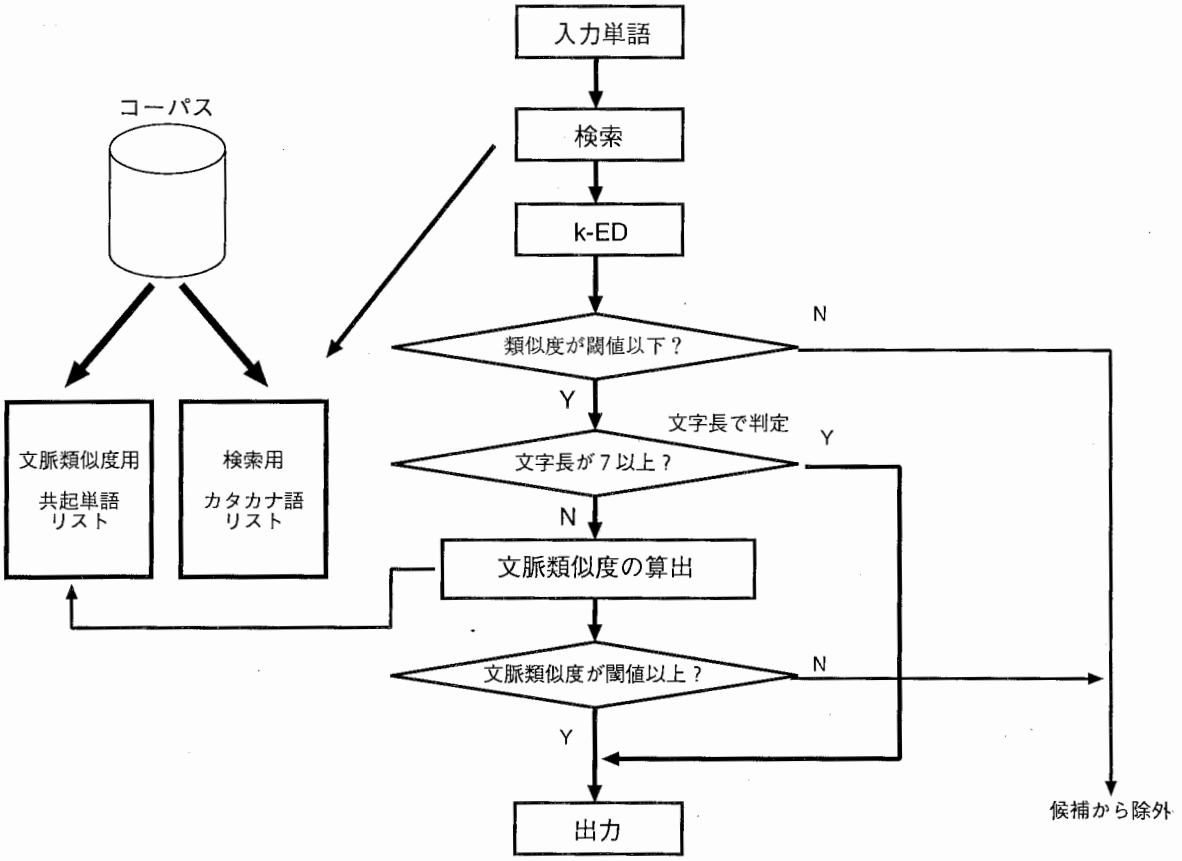

図 2: 手法の全体構成図

ゆれを調べたい単語を入力し，出力はその単語と置き換えられる可能性のある単語のリストとなる．処理は入力単語から検索文字列を生成し，コーパス中のカタカナ語からその候補を検索する．その候補を絞り込むために，カタカナに特化した編集距離 (kED) を用いる．さらに文脈を用いた類似度による判定も行うが，これらを入力単語の文字長により組み合わせる．

4.1 検索

(4.1.1) カタカナ語リストの作成

カタカナ語リストとは，表記ゆれを含んでいるコーパスから作ったカタカナ語のリストである．リストはカタカナ文字列の連続を抜き出して作成した．

(4.1.2) 検索文字列の生成

入力単語 (カタカナ語) から検索文字列を生成する．まず，入力のカタカナ文字列にゆれやすい文字を含んでいる場合は，その前後で分割する．ここでゆれやすい文字とは，拗音のすべてと促音，長音で10種類とした．

［ァ イ ゥ エ ォ ヤ ユ ヨ ッ ー］

以下に「ミネラルウォーター」の例を示しているが，この場合検索文字列となるのは，{ミネラルウ, タ}となる．

また，ゆれやすい文字列を含んでいない単語の場合は，それらを1文字ずつに分解し検索文字列とする．入力単語をを「バイオリン」とすると検索文字列は，{バ, イ, オ, リ, ン}となる．

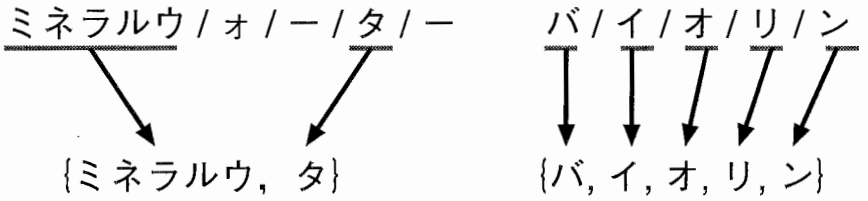

図 3: 検索文字列生成例

(4.1.3) 表記ゆれ候補検索

(4.1.2) 節で生成した検索文字列を SUFARY [1]⟨1⟩ を用いて，(4.1.1) 節のカタカナリスト中から検索した．

ここで得られた候補すべてに類似度を測定したのでは，効率が悪いため文字の並びを考慮し，簡単な選別を行った．以下に入力単語「ウィスキー」での例を示す.「ウィスキー」の検索文字列は，{ウ, スキ}となる．この「ウ」と「スキ」の順番が同じものを候補として抽出する．すなわち，検索例にある「スキーウェア」は，検索文字列を含んではいるが，その現れる順番が入力単語と異なるため，選別され候補とならない．

4.2 カタカナに特化した編集距離: kED

通常の編集距離を基にカタカナに特化した編集距離 (以下，kED) を定義した．これを単語対に適用しkEDを求め，その結果から類似度を算出する．このとき，2つの単語 A, B に対するkEDによる類似度 sim_{kED} は，

$$sim_{kED} = \frac{2kED(rom(A), rom(B))}{|rom(A)| + |rom(B)|} \tag{1}$$

によって定義される．ここで，$rom(A)$ は A をローマ字表記した文字列を示す．sim_{kED} は，小さければ小さいほど，2つの文字列は類似していると判断される．逆に大きい場合は，それらの文字列は異なることになる．

[1]SUFARY とは suffix array というデータ構造を用いて高速な文字列検索を行うためのライブラリを中心としたパッケージで，奈良先端大学院大学松本研究室が開発

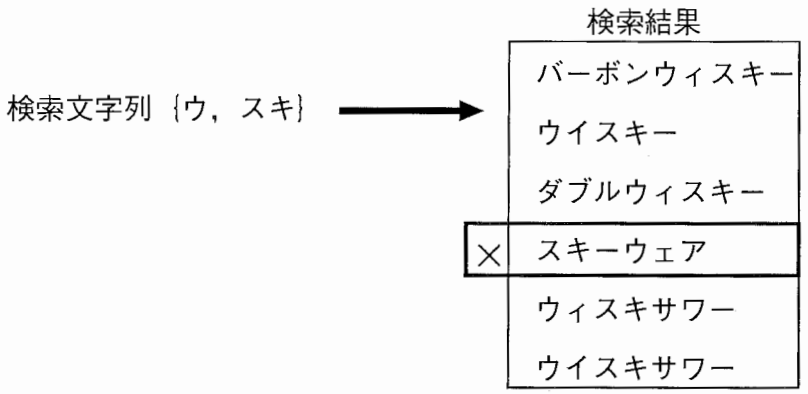

図 4: 検索例

(4.2.1) kED と通常の編集距離の違い

kED と通常の編集距離との比較を行う．編集距離とは，文字列 1 と文字列 2 を比較し，文字列 1 を文字列 2 へ（あるいはその逆）編集する場合の編集操作の数として定義される．用いられる編集操作は，挿入，削除，そして置換である．しかしながら，置換操作は，挿入ならびに削除操作によって実現可能であるため使用しない場合もある．通常の編集距離では，比較する文字列を 1 文字ずつ走査し，DP マッチングによって最小の編集距離を求める．したがって，比較における文字が異なれば，どのような文字かに依存せず，そこで編集操作が発生し，その編集操作に応じた編集距離が求められる．

しかし，表記ゆれを効率よく特定するためには，ゆれやすい文字列を考慮した編集距離を定義すべきである．そこで，kED では，該当比較文字の前後 2 文字を考慮し，編集操作に対して重みを定義する．その点から，kED は重み付き編集距離 [Gus97] の一種といえる．また通常の編集距離では，比較対象の文字は制限されずどのような文字列でも比較できるが，kED では，カタカナの元の発音を近似するために，ヘボン式ローマ字を導入し，ローマ字文字列に対して重みを定義しているため，ローマ字文字列の比較しかできない．さらに，通常の編集距離では，各編集操作に対して 1 という固定された編集距離を定義するのに対して，kED により算出される距離は，比較文字ごとに値を決定しているため，可変値をとる．比較文字列が異なる場合に，ゆれやすい文字列には小さい値を，ゆれとは考えにくい文字列には大きな値を定義することによって精度が上がると考える．

	kED	通常の編集距離
比較対象	ローマ字	制約なし
比較範囲	前後 2 文字	該当文字
距離	可変	固定 (0/1)

(4.2.2) kED の適用例

kED を用いて「ミネラルウォーター」と「ミネラルウオータ」の編集距離を算出する．まずカタカナ文字列のまま通常の編集距離を算出すると,「オ」と拗音の「ォ」の違いから置換操作が起こる．ここでは置換の編集操作を定義せず，削除と挿入操作による編集距離 2 とす

る．また，最後の長音記号が無いために，削除操作が起こり編集距離はさらに1大きくなる．よってカタカナ文字列そのままで編集距離を算出した場合は，3となる．

次に入力単語と比較単語文字列をローマ字に変換し，編集距離を算出する場合を考える．ここでは，各単語をヘボン式のローマ字文字列に変換し，長音記号はハット(^)で表現した．ローマ字表記を用いることによって「オ」と拗音の「ォ」との区別はなくなるため，この違いによる編集操作は発生しない．そのため発生する編集操作は長音記号の削除のみとなり，最終的な編集距離は1となる．

一方で，これらの文字列に対するkEDは，0.7となる．これは末尾の長音記号がないことが許容されている((4.2.3)節，規則1)ため，距離が1よりも短くなる．

ミネラルウォーター　ミネラルウオータ
mineraruuo^ta^　　mineraruuo^ta

(4.2.3)　kEDの距離算出規則

kEDでは，比較文字列が異なる場合に，ゆれやすい文字列に対しては，小さい編集距離を，ゆれにくい文字列に対しては，大きい編集距離となるよう規則を作成している．その規則について例を挙げて説明する．この場合，“小さい/大きい”編集距離とは，通常の編集距離を求める際に1つの編集操作に付与される距離: 1よりも小さい/大きいということである．

距離を小さくする場合

1. 末尾の長音記号がない

   比較する単語対の片方に長音記号が無い場合は，距離を小さくする．

   (例1) ミネラルウォーター (mineraruuo^ta^) ⟷ ミネラルウオータ (mineraruuo^ta)

   (例2) ラジエーター (rajie^ta^) ⟷ ラジエータ (rajie^ta)

2. 母音の重複と長音記号の違い

   母音の重複と長音記号の違いならば，距離を小さくする．

   (例1) ウェイトレス (ueitoresu) ⟷ ウエートレス (ue^tores)

   (例2) イタリア (itaria) ⟷ イタリー (itari^)

3. 促音の処理

   促音のあとが「タ行」の場合は，促音が落ちても同じ意味になる可能性が高いので距離を小さくした．

   (例1) カフェラッテ (kaferatte) ⟷ カフェラテ (kaferate)

   (例2) スパゲッティ (supagettei) ⟷ スパゲティ (supagetei)

(例 3) ココナッツ (kokonattu) ⟷ ココナツ (kokonatu)

4. 文字 [b] と [v]

比較文字が [b] と [v] の違いならば，距離を小さくする．

(例 1) バイオリン (baiorin) ⟷ ヴァイオリン (vaiorin)

(例 2) バカンス (bakansu) ⟷ ヴァカンス (vakansu)

5. 文字列 [tei] と [chi]

[tei] と [chi] の違いならば，距離を小さくする．

(例 1) プラスティック (purasuteikku) ⟷ プラスチック (purasuchikku)

(例 2) イニシアティブ (inisiateibu) ⟷ イニシアチブ (inisiachibu)

6. 拗音で [i]，[e] の音を含むとき

拗音で [i] や [e] の音を含む場合，距離を小さくする．

(例 1) サンディエゴ (sandeiego) ⟷ サンジエゴ (sanjiego)

(例 2) ロサンジェルス (rosanjierusu) ⟷ ロサンゼルス (rosanzerusu)

7. 拗音で [y] の音を含むとき

拗音に [y] の音を含むときは，距離を小さくする．

(例 1) マニキュア (manikyua) ⟷ マニュキュア (manyukyua)

(例 2) ピニャコラーダ (pinyakora^da) ⟷ ピナコラーダ (pinakora^da)

(例 3) プラネタリュウム (puranetaryuumu) ⟷ プラネタリウム (puranetariumu)

8. 濁音などの許容

子音同士を比較し，清音と濁音もしくは清音と半濁音の違いならば寛容し距離を小さくする．

- k ⟷ g　[例：バック (bakku) ↔ バッグ (baggu)]
- s ⟷ z　[例：ウィリアムス (uiriamusu) ↔ ウィリアムズ (uiriamuzu)]
- t ⟷ d　[例：グット (gutto) ↔ グッド (guddo)]
- h ⟷ b
- h ⟷ p

距離を大きくする場合

1. 単語のはじめの子音が異なる

   比較するカタカナ単語対の第1文字を構成する子音が異なる場合，距離を大きくする．

   (例 1) ロジャース (roja^su) ⟷ ドジャース (doja^su)

   (例 2) カップ (kappu) ⟷ タップ (tappu)

2. 先頭文字が子音と母音

   比較する単語対の先頭文字が，母音と子音で異なった場合，距離を大きくする．

   (例 1) イアホン (iahon) ⟷ ドアホン (doahon)

   (例 2) インダクター (indakuta^) ⟷ コンダクター (kondakuta^)

3. 長音記号の前の子音が異なる

   長音記号の前の子音が違うときは，距離を大きくする．

   (例 1) デパート (depa^to) ⟷ デザート (deza^to)

   (例 2) レーダー (re^da^) ⟷ レーザー (re^za^)

4. 末尾の文字が異なる

   比較するカタカナ単語の末尾の文字が異なる，つまり末尾の文字を構成するローマ字文字列の子音が異なる場合，距離を大きくする．

   (例 1) トランス (toransu) ⟷ トランプ (toranpu)

   (例 2) ボール (bo^ru) ⟷ ボート (bo^to)

5. 子音と撥音

   子音と撥音との間で表記がゆれることは，ほぼ無いと考えられるので，距離を大きくする．

   (例 1) トランク (toranku) ⟷ トラック (torakku)

   (例 2) ウエストバンク (uesutobanku) ⟷ ウエストバック (uesutobakku)

4.3 ベクトル空間法の適用 (文脈類似度)

文字列の比較のみよる表記ゆれの特定には限界がある．たとえば，レザー (*leather*) とレーザー (*laser*) は通常の編集距離においても 1 違うだけで，文字列上は非常に類似している．しかしながら，これらの文字列は，その用いられ方が大きく異なるため，それを比較することによって表記ゆれか否かをより正確に判断できると考える．

そこで，それらの文字列の周辺の情報を用いる．具体的には，該当するカタカナ語を含む文を文脈と考え，カタカナ語と共起する名詞や，カタカナ語が係る述語表現を要素とする文脈ベクトルをカタカナ語に対応させる．そして，対象となるカタカナ語に対応する文脈ベクトルのコサイン角を求めることによって 2 つのカタカナ語の類似度を求める．この類似度を文脈類似度と呼び，2 つのカタカナ単語 A, B の文脈類似度 $sim_v(A, B)$ を次の式により定義する．

$$sim_v(A, B) = cos(vec(A), vec(B)) = \frac{\sum A_i B_i}{\sqrt{\sum A_i^2}\sqrt{\sum B_i^2}} \tag{2}$$

ここで，$vec(A)$ は単語 A に対応する文脈ベクトルであり，A_i は文脈ベクトル $vec(A)$ の要素を示している．

(4.3.1) 文脈ベクトルに用いる素性の種類

文脈ベクトルに用いる素性の種類は次の 3 つである．

- カタカナ語の現れている行に含まれる名詞:N
- カタカナ語の係り先の述語:V
- カタカナ語の係り先の述語とその直前の助詞の組[2]:PV

以上の素性を CaboCha(2)[3]で解析したコーパスから抽出し，出現頻度をその値として集計し，最終的に各文脈ベクトルは，単位ベクトルとした．

(4.3.2) 文脈類似度の適用例

文脈類似度の適用例を示す．ここでは，例として「スニーカー」と「スピーカー」を示す．まず,「スニーカー」の文脈ベクトルは,

$$\begin{aligned} vec(\text{スニーカー}) = \{N &: \text{ジーンズ} = 0.4, \text{これ} = 0.1, ..., \\ V &: \text{履く} = 0.1, \text{行ける} = 0.4, ..., \\ PV &: \text{で行ける} = 0.4, \text{を探す} = 0.1, ...\} \end{aligned}$$

次に,「スピーカー」の文脈ベクトルは,

$$\begin{aligned} vec(\text{スピーカー}) = \{N &: \text{音} = 0.333, \text{声援} = 0.333, ... \\ V &: \text{くださる} = 0.333, NULL = 0.667, \\ PV &: \text{をくださる} = 0.333, NULL = 0.667\} \end{aligned}$$

[2]抽出対象となるカタカナ語の直後の助詞ではなく，述語の直前の助詞．これが本当に効果的か，という実験は行っていない．

[3]CaboCha は，奈良先端大学院大学の松本研究室が開発した日本語係り受け解析器である．

と表現できる．これらから文脈類似度を求めると，

$$sim_v(\text{スピーカー}, \text{スニーカー}) = cos(vec\,\text{スピーカー}, vec\,\text{スニーカー}) = 0 \qquad (3)$$

となる．「スニーカー」と「スピーカー」と共起する単語に一致がみられなかったという結果である．

5 実験

予備実験として，kEDのパラメータ決定，文脈類似度における重み付け実験を行い，提案手法に用いるパラメータを決定した．さらに提案する手法の有効性を確認するために，いくつかの手法を用いて比較実験を行った．また，本来の目的である特定のコーパスに対して表記ゆれの抽出を行った結果を示す．

5.1 予備実験

(5.1.1) 文脈類似度における述語の重み決定

はじめに，文脈ベクトルを構成する素性として抽出してある，名詞，述語，述語とその直前の助詞の組のどれが効果的に機能するかを旅行会話基本コーパスを用いて調査した．その結果を表1に示す．このとき重みとして該当素性の値を1.5倍している．述語に重みを付けて文脈類似度を算出することで，他の素性に重みを付けるよりも高いF値が得られた．これはカタカナ語の用いられ方をある程度特定できたためだと考える．

表1: 素性別に重みをつけた場合の結果

	候補数	正解数	精度	再現率	F値
kEDのみ	199	100	0.50	1.00	0.67
名詞	132	79	0.60	0.79	0.68
述語	124	87	0.70	0.87	0.78
述語と助詞	199	98	0.49	0.98	0.66

(5.1.2) パラメータ決定

次に述語の重み（倍率），文脈類似度の閾値を決定するための実験を行った．表(5.1.2)にkEDの閾値が4以下を対象に2つのパラメータを変化させた．このときF値の最高は0.80である．これ以降の実験では，文脈類似度の閾値を0.2，述語の倍率を1.4倍とする．

(5.1.3) 文字長によるヒューリスティックの効果

パラメータ決定のための予備実験の結果，すべての候補対を文脈類似度で判定してしまうと再現率が急激に低下した．その原因として，比較する単語対の頻度が大きく異なった．もしくは比較する単語対のいずれも頻度が小さかった．表記ゆれは標準的に用いられる表記に比較して頻度が小さいことは明確であり，そのようなゆれこそ抽出すべきものである．そこで文脈類似度を回避させる条件を文字長によって作成した．

kEDのみを用いて表記ゆれを判断した場合の結果を用いて，文字数と誤り率の関係を描いたグラフを図5に示す．文字数が7文字以上であれば誤り率はほぼ20%以下に収まる．しかし，12文字の点では誤り率が50%となっている．これは2候補のうち1候補に誤りを含んでいたためである．以上から文字数が大きい，長い単語であれば間違いにくいという結果が得

表 2: kED 閾値が 4 の結果

文脈類似度	述語の倍率	出力候補数	正解候補数	精度	再現率	F 値
0.1	1.0	129	103	0.80	0.77	0.78
0.1	1.2	131	104	0.79	0.78	0.78
0.1	1.4	133	105	0.79	0.78	0.79
0.1	1.6	134	106	0.79	0.79	0.79
0.1	1.8	137	106	0.77	0.79	0.78
0.1	2.0	141	107	0.76	0.80	0.78
0.2	1.0	112	97	0.87	0.72	0.79
0.2	1.2	113	98	0.87	0.73	0.79
0.2	1.4	116	100	0.86	0.75	0.80
0.2	1.6	120	100	0.83	0.75	0.79
0.2	1.8	120	100	0.83	0.75	0.79
0.2	2.0	121	101	0.83	0.75	0.79
0.3	1.0	105	92	0.88	0.69	0.77
0.3	1.2	107	94	0.88	0.70	0.78
0.3	1.4	107	94	0.88	0.70	0.78
0.3	1.6	109	95	0.87	0.71	0.78
0.3	1.8	113	98	0.87	0.73	0.79
0.3	2.0	113	98	0.87	0.73	0.79
0.4	1.0	101	89	0.88	0.66	0.76
0.4	1.2	102	90	0.88	0.67	0.76
0.4	1.4	103	91	0.88	0.68	0.77
0.4	1.6	105	92	0.88	0.69	0.77
0.4	1.8	105	92	0.88	0.69	0.77
0.4	2.0	107	94	0.88	0.70	0.78

られた．そこで文字数が 7 以上のものは，間違いにくいと判断し文脈類似度を回避させる．反対に 6 文字以下の単語の場合は，誤る可能性が高いため文脈類似度を用いて比較する．

5.2 各手法の比較

入力に対して，カタカナ語のゆれをコーパスからどれくらい抽出できるか実験を行った．ゆれているカタカナ語候補を検索するコーパスは，BTEC1(learning) を用いる．また，入力とするカタカナ語は，BTEC1(test) から抽出した 1020 語 (異なり) とした．

ゆれているカタカナ語を抽出するための手法は次の 5 つを用いた．

手法 1 カタカナ文字列のまま編集距離を算出

手法 2 カタカナ文字列をローマ字に変換して編集距離を算出

手法 3 sim_{kED} を算出

手法 4 sim_{kED} を算出し，さらに文脈類似度でふるいわけ

手法 5 sim_{kED} を算出し，単語の長さにより文脈類似度を使用

各手法を用いて得られた結果を表 3 に示す．コーパス中のカタカナ語のゆれをすべて見つけだすことは困難である．そこで閾値をゆるめた上で手法 1 を適用し，候補となる対を抽出した．このカタカナ語対に対して人手で正解判定を行い，正解と判定された 143 対から再現率を求めた．

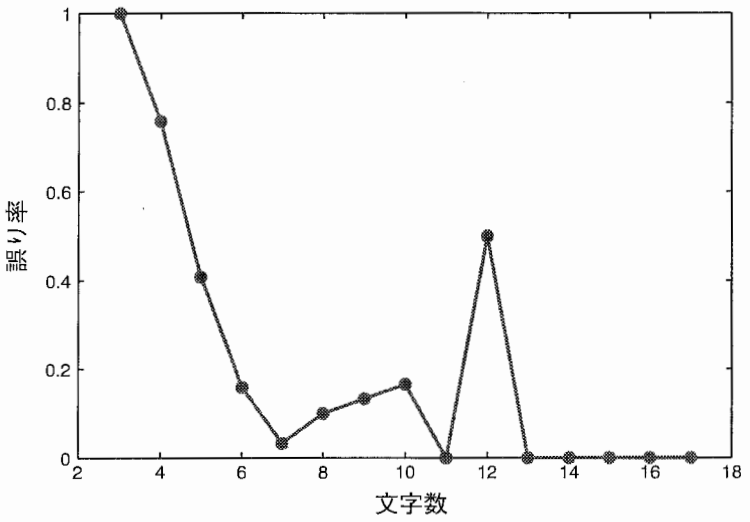

図 5: 文字数と誤り率のグラフ

表 3: 手法別の抽出結果

	手法 1	手法 2	手法 3	手法 4	手法 5
出力候補数	994	244	178	51	119
正解数	113	119	120	47	105
精度	11%	48%	79%	92%	88%
再現率	79%	83%	84%	33%	83%
F 値	0.18	0.64	0.77	0.49	0.85

5.3 同一コーパス中における表記ゆれ抽出

同一コーパス中で表記ゆれをどの程度抽出できるかについて実験した．対象とするコーパスは，BTEC(travel1, learning) 約 16 万文である．5.2 節で最良の結果を得た手法 5 を用いて実験を行った．その結果，462 対を候補として出力し，うち正解は 438 対であった．このときの精度は 94.8%であり，再現率は人間が測定できないので算出していない．

6 考察

表3は，カタカナに特化した編集距離に基づく類似度 sim_{kED}，文脈類似度 sim_v，および文字長による処理を組み合わせた手法の結果である．類似度 sim_{kED} のみを用いただけでは，精度が低い．そして，それに文脈類似度を適用すると今度は絞り込みすぎてしまう．しかし，入力単語の文字長を考慮して，入力単語が長い場合には文脈類似度の適用を回避する．これによって手法5では，再現率を落とすことなく精度の改善に成功している．

一方で，文脈類似度を用いると失敗する場合がある．以下にその例を示す．この例の中で[ ]内の数字は，コーパス中での頻度を表している．

(例1) セイフティボックス [15] ⟷ セフティボックス [2]

(例2) シャンペン [14]⟷ シャンパン [4]

実験の結果から比較対象の単語対に大きな頻度差がある場合に，文脈類似度の判定誤りが多いことがわかった．また，頻度が低いために失敗する場合も多数ある．

類似度 sim_{kED}，sim_v および文字長を考慮したとしても，判定を間違う場合がある．たとえば，

(例3) ブリーチーズ ⟷ ブルーチーズ

このような単語対である．

例3では第2音の母音，第3音の母音が異なるだけであり，単語自体の長さも長い．そのため文字長で正規化している類似度 sim_{kED} では，値が小さくなり類似していると判断する．また，この例に文脈類似度を適用させたとしても，上位概念は同じ「チーズ」となり，単語の使われ方が類似していると想像できる．従ってそのまま文脈類似度を用いたとしても判別は難しい．

このような問題を解決するためには次のような手法が考えられる．単語1と単語2が次のような場合，共通している文字列を除いた部分で，類似度 sim_{kED} を取るというものである．単語対の上位概念が等しく，それ自体が単語として用いられる共通の部分文字列を含む場合（たとえば，バニラ**アイスクリーム**とバナナ**アイスクリーム**）に有効な手段であると考える．

単語1: A B C a b c d
単語2: A D E a b c d

さらに上記の手法と文脈類似度を用いることで，文脈類似度が高いとき上位概念語を効率的に見つけられることができるのであれば，シソーラスの構築をすることも可能である．

次にみつけることのできた候補について考察する．表記ゆれを探すための入力には，1020語(異なり)と6219語(異なり)のカタカナ語を用いた．その結果，1020語に対しては121語を抽出することができた．6219語では462語である．これらの結果のうち70%以上が辞書(IPAL名詞辞書[4])に無い候補であった．詳細を表4に示す．この結果より提案した手法が新語や造語に対しても効果を発揮すること，もしくは誤字・脱字の様なカタカナ語にも対応することがわかる．誤字・脱字を含むカタカナ語を正確に抽出できた例として，例4をあげる．

[4]情報処理振興事業協会が公開している辞書

(例 4) ロサンジェルス ⟷ ロサンゼルス
⟷ ロザンジェルス
⟷ ロセンゼルス
⟷ ロスアンゼルス

この例ではロサンジェルスを入力とし，表記ゆれの候補を抽出したものである．例の中で一般的に使われるものは，入力の「ロサンジェルス」もしくは「ロスアンゼルス」のどちらかであろう．しかし実際にはこのような表記ゆれも存在するため，誤字のようなものでも抽出できることが確認できた．

表 4: 抽出できた候補数の詳細

入力	全候補	辞書に未掲載	コーパス中の延べ数
1020	121	95	1205
6219	462	332	5205

7 おわりに

本研究では，コーパス中の表記ゆれを統一する方法をカタカナ語を中心に行った．カタカナ語の表記ゆれ解消法には，カタカナ語の表記ゆれに特化した類似度を提案した．また共起単語などを考慮した文脈類似度や文字長を用いることで，カタカナに特化した類似度をそのまま用いるよりも効果的であることを確認した．その結果，BTECの一部およそ16万文の中で延べ5205語の表記ゆれを抽出し有用性を確かめた．

使用したツール

⟨1⟩ Sufary [http://nais.to/ yto/tools/sufary/]

⟨2⟩ CaboCha [http://cl.aist-nara.ac.jp/ taku-ku/software/cabocha/]

⟨3⟩ IPAL 辞書 [http://www.ipa.go.jp/STC/NIHONGO/IPAL/ipal.html]

参考文献

[Gus97] GUSFIELD, D.: *Algorithms on Strings, Trees, and Sequences*, CAMBRIDGE University Press (1997).

[Kub94] 久保田淳市, 庄田幸恵, 河合眞宏, 玉川博文, 杉村領一：カタカナ表記の統一–予備分類とグラフ比較によるカタカナ表記–, 情報処理学会論文誌, Vol. 35, No. 12, pp. 2745–2751 (1994).

[Shi94] 獅子堀正幹, 津田和彦, 青江順一：片仮名異表記の生成および統一手法, 電子情報通信学会論文誌, Vol. J-77-D-II, No. 2, pp. 380–387 (1994).