

カタカナ表記の統一方式

—予備分類とグラフ比較によるカタカナ表記のゆらぎ検出法—

久保田 淳市[†] 庄田 幸恵[†] 河合 眞宏[†]
玉川 博文^{††} 杉村 領一[†]

文章中のカタカナ表現を統一する効率的なカタカナ表記不統一検出方式を開発した。従来、辞書の見出しと文章中のカタカナ表記をゆらぎを許容して比較し、不統一箇所を検出する方式が多く提案されたが、カタカナ語は外来語を中心に新語が多く、予め辞書を整備しておくことは困難である。本方式では、辞書を使わず、カタカナ文字列同士のゆらぎ規則を文章中のカタカナ語に適用し、カタカナ表記のゆらぎを検出する。事前分類したカタカナ表記を有向グラフ形式の中間形式に変換し、効率良く比較検出を行う。試作と実験を行った結果、再現率 97.4%、適合率 86.7% という結果を得た。

A Method of Detecting KATAKANA Variants
in a DocumentJUN'ICH KUBOTA,[†] YUKIE SHODA,[†] MASAHIRO KAWAI,[†] HIROFUMI TAMAGAWA^{††}
and RYOICHI SUGIMURA[†]

A method to detect KATAKANA variants (equivalents for approximate KATAKANA expressions) from a Japanese document is proposed. As a Japanese phonetic symbol system (KATAKANA) cannot express precise pronunciation of loan words, not a few loan words can have many KATAKANA expressions. Increasing newly imported words also make it difficult to maintain a dictionary with all KATAKANA expression items. To detect KATAKANA variants in a document without a dictionary, this algorithm transforms KATAKANA strings to directed graphs based on rewrite rules, then checks whether they have the same labeled path or not. This method can recall KATAKANA variants with an accuracy of 97.4%.

1. はじめに

文章の校正や推敲を計算機で支援する試みを目的別に大別すると、同音語ミスや文法ミスを検出する「誤り検出機能」、単語の語義や類義表現を参考情報として提供する「表現展開機能」、文体、送りがな、カタカナ表記等を統一する「表現統一機能」の三つがある。

このうち「表現統一」の重要課題にカタカナ表記の統一がある。通常カタカナ語は文章中に 5~25% 含まれる¹⁾が、主要なカタカナ語である外来語は発音表現が不安定なため原語で同じ単語（例えば show）が異

なるカタカナ表現（ショーやショウ）に表される「表記のゆらぎ」という現象が発生する。特に新語は、各人が元の発音を異なる日本語のカタカナにあてはめてしまうことが多く、表記のゆらぎが生じやすい。

すべての表記を辞書の見出しとして登録し異表記同士をグループ化すれば良いが、外来語は新語も多くこれらをすべて辞書の見出しとして登録することは事実上不可能である。

従来もこの問題に着目した研究は行われてきたが、主として未登録語として扱われるカタカナ異表記を減らす目的で、辞書のカタカナ見出しと「ゆらぎを許容したマッチング」を行うもの^{2),3)}であった。また、ソートしたカタカナ語を KWIC 形式で表示する機能も提案されている⁴⁾が、ゆらぎと無関係な候補も出るため、特に規模が大きい文章を対象とする場合の確認作業負担が大きくなる。

文章表現を統一するため文章内のカタカナ表記のゆ

[†] 松下電器産業(株)情報通信研究所
Information and Communications Technology
Laboratory, Matsushita Electric Industrial Co.,
Ltd.

^{††} 松下電器産業(株)ワープロ事業部
Word Processor Division, Matsushita Electric
Industrial Co., Ltd.

らぎを検出するものでは、カタカナ列同士の距離を計算するもの⁵⁾や、カタカナ読み列を書き換え規則を用いて「標準読み列」へ変換した上で相互に比較しカタカナ語の同一性を判定する方式⁶⁾がある。これらの手法で、基本的なカタカナ表記のゆらぎを検出することは可能だが、平均的な類似度を基にする「距離方式」では、処理が単純化されるという利点がある反面、単一の類似度と識別値との大小関係をもとにゆれの有無を判断するためゆらぎ特性に応じた最適化が行いにくく、適合率*や再現率**等の精度を上げにくい。

また、「標準読み列方式」では、ルールの適用条件を文字列の出現位置、前後のコンテキスト等で調整したりルール間の相互関係、優先順位の関係を調整する必要がありルールのチューニングが難しい。

本稿では、文章内のカタカナ表記のゆらぎを解消するため、辞書を使わず、標準読み列を設定せず独立のルールを複数個同時に適用し、カタカナ列をグラフ型の中間形式に変換し、それら同士の類似性に基づいてゆらぎの検出を行うカタカナ表記統一方式を紹介する。

本方式により、カタカナ語を延べ6.8万語含む技術文書を対象に評価を行い、再現率97.4%、適合率86.7%という結果を得た。また、30%のカタカナ含有量の900字の文書を処理した時レスポンス時間は0.8秒(Sun3ワークステーション)であった。また独立ルール数は58である。

2. ゆらぎの実態

カタカナ表記のゆらぎは種類が多く、規則も不安定である。国語審議会による1991年2月の「外来語の表記」に関する答申⁷⁾では、外来語の表記に用いるカナ符号表1と2を定め、原則留意事項1(6項目)と細則留意事項2(18項目)を併記した。基本的には従来『バと表記するのが望ましい』としていた『ヴェ』なども表記として認め、規則を緩やかにする方針である。一方、国立国語研究所刊「現代表記のゆれ⁸⁾」では、新聞などに現れた「表記のゆらぎ」の実

表1 計算機マニュアル中のカタカナ表記の不統一実態
Table 1 Statistics of Katakana variants in a computer manual.

ゆらぎの分類	種類	例
中点の有無	145	ウィンドウシステム/ウィンドウ・システム
長音の有無	13	インターフェース/インターフェス
その他	6	フロピィ/フロッピー
延べカタカナ語数	24,000	異なりカタカナ語数 1,171

表2 技術文書中のカタカナ表記の不統一実態
Table 2 Statistics of Katakana variants in technical reports.

ゆらぎの分類	種類	例
中点の有無	221	メモリ・アクセス/メモリアクセス
長音の有無	134	パートナ/パートナー
その他	141	シベリア/シベリヤ
延べカタカナ語数	68,447	異なりカタカナ語数 7,605

例を挙げているが、「カーテンウォール/カーテンオール」「キャー/キャアー」のように必ずしも答申に現れていないものがある。また、カタカナ表記のゆらぎは、必ずしも外来語の発音表記のゆらぎのために生じているだけでなく、「ドッジボール/ドッヂボール」のような日本語固有の問題であるかな遣いのゆれによるものも現実には出現していることを指摘している。

これらに対して実態を把握するため、二つの実文例中における文章中のカタカナ表記のゆらぎを調査すると「・」(中点)と「ー」(長音)の有無にかかわるものが多いことが分かった。

第一に、数名で作成した計算機マニュアル原稿中のカタカナ語の抽出を行ったところ、24,000語(1,171種)のカタカナ語から164種類のカタカナ表記のゆれが発見された(表1参照)が、その大半(およそ96%)が中点と長音の有無によるものであった。予め用語統一を図って執筆するマニュアルでも中点、長音のゆらぎは多く見落とされている。

また、第二の例として、多数の人が独立して作成した技術文書の集合を対象として調査した結果を表2に示す。延べカタカナ語数68,447語(異なり数7,605種)に対し、496種のゆらぎがあり、そのうち、中点の有無221件、長音の有無134件であり、この二種で全体の72%を占める。表1と比べると予め用語を統制していないので、中点と長音以外のカタカナ表記のゆらぎが増えている。

* 適合率 = $\frac{\text{正しく検出したゆらぎの数}}{\text{検出したゆらぎ候補の数}}$

** 再現率 = $\frac{\text{正しく検出したゆらぎの数}}{\text{検出すべきゆらぎの数}}$

以上の結果から、中点、長音を十分にカバーした上で、多様なゆらぎを扱え、漸増するルールを簡単に整理できる拡張性の高い検出方式が必要であることが分かる。

なお、「ゆらぎ」の定義は「現代表記のゆれ」⁸⁾では「スペシャル/スペシエル」のような表記のゆれと「クロス/クロース」のような語形のゆれに分けてあるが、本稿では基本的には類似性の高いものは語形のゆれも含めて対象とした。また、実文例に生じている「ー(長音)/ー(マイナス)」などのような誤用も検出対象に含めた。ただし、以下のように発音上の差異が大きい、もしくは国語辞典などで別見出しが立っているという理由で、語形が明らかに異なると判断したものは対象外とした。

「プリン/プディング」

「ヘボン/ヘップバーン」

「エルゴノミクス/アーゴノミクス」

3. ルール記法と中間形式

従来のカタカナ表記ゆらぎ検出方式で用いられている規則は、次のようにカタカナ元読み列を標準読み列に変換する書き換え規則であった。規則の適用を完了したカタカナ列が同一の標準読み列である場合、元のカタカナ列同士は表記のゆらぎの関係にあると判断する。

元読み列 → 標準読み列 (適用条件)

【例】クェ→クエ (語頭)

この表現では、複数のルールを記述する場合適用規則の衝突の問題が生じる。例えば、以下のように左辺に同一の記号が現れるルールが複数あると、正しく規則を選択できずに、誤ったカタカナ列に書き換えてしまう可能性があり、合流性も保証できない。例えば、「ハンカチ」と「ハンケチ」を比較する場合は規則Aを適用すべきであるが、「コンマ」と「カンマ」の場合は規則Bを適用する必要がある。また、「カルテット」と「クアルテット」との場合は規則Cを適用する必要がある。

【規則例1】 A: カ→ケ

B: カ→コ

C: カ→クァ

【適用例】 A: スカッチ→スケッチ

B: スカッチ→スコッチ

C: スカッチ→スクァッチ

仮に、入力カタカナ列として「スカッチ」と「スコッチ」が与えられた時、上記のA、B、Cの順の優先順で規則が適用されると、「スカッチ」は適用例のように「スケッチ」に書き換えられ、「スコッチ」との同一性が認識できずゆれは検出できない。これを解決するために、上記規則をチューニングすると以下のようになる。

【規則例2】 A: ケ→カ

B: コ→カ

C: クァ→カ

今度は、「スコッチ」が「スカッチ」に書き換えられて、ゆれが認識されるようになる。しかし、さらに「コッチェジとコテージ」のゆれに対応して、「コッコ」というルールが追加されると、検出できなくなるため更に何らかのチューニングが必要になる。

このように、前後の文脈や、出現位置等を条件に適用条件、適用順序、さらに優先度を考慮して、規則の適用対象や順序を制御してゆかなければならない。このため許容が変化しつつある外来語表記のゆらぎのルールを副作用を抑制しながら増加・調整することは簡単ではない。

そこで、本方式では標準読み列を設定せず、コンパクトに複数の解釈を表現できるグラフ型の中間形式を定義する。このグラフは始点一つ、終点一つを有するループのない有向グラフである。従来は、各規則を適用する段階で他の可能性を棄却していたのが、この中間形式を用いることにより、すべての可能性を留保することが可能になった。

また、規則は、部分カタカナ列同士の書き換え規則ではなく、以下のように同一音を表すカタカナ列同士を組にしてゆらぎグループとして記載するため、現実の現象を表現しやすい。また、各ルールは独立であり左辺に同一文字列が現れても構わない。以下に、規則例1と対応するゆらぎを表現した規則を示す。

部分列 1, 部分列 2, …部分列 N

→ゆらぎグループ識別子

【例】 A: カ, ケ → a

B: カ, コ → b

C: クァ, カ → c

つまり、本方式では規則には優先順位がなく、例えば「カ, ケ→a」, 「カ, コ→b」, 「クァ, カ→c」という独立した規則が可能な限り適用され、両方の規則の適用可能性を留保したグラフ型の中間形式が生成され、最終段階で照合する時まで保存される。

この中間形式は、入力読み列に各規則が適用されると、各辺がカタカナ文字ないし、ゆらぎグループ識別子に対応する有向グラフで表現される。例えば入力カタカナ列「スカッチ」には以上の規則が適用され図 1 の上に示すような三つのパスを持つ中間形式を得る。グラフは各頂点（ノード）がカタカナ列の解釈位置、各辺（エッジ）が解釈記号（ゆらぎグループ識別子もしくはカタカナ文字）に対応している。スタートノード(S)から終端ノード(E)に達する三つのパスは各々 A, B, Cの規則を適用されたものに対応している。同様に、「スコッチ」というカタカナ語からは図 1 の下に示す中間形式が得られる。

二つのカタカナ列に対応したグラフ形式の中間形式 G_1 と G_2 が得られると、これらのグラフの S から E に至るパスの中で共通のパスがあるかどうかで、これらの二つのグラフが同一のカタカナ語の異表記であるかどうかを判定する。つまり、従来と異なり中間結果がグラフであるため、中間結果の同一性は問わずパスの共通性を判定基準とする。図 1 の例では「ス→b→ッ→チ」というパスが両方のグラフに含まれているので、カタカナ表記のゆれであると認識される。

この状態でたとえ、新しく、「 коттеージとコテージ」のゆれに対応して「コッ, コ→d」という規則が追加されても、グラフのパスが増加するだけで認識不能になることはない。

グラフの比較は縦型優先に行う。すなわち、S から E までの共通パスが最初に見つかった時点で計算を打ち切り、また、共通の探索パスがなくなった時点で「不一致」と判定し計算を打ち切ることで、平均の探索時間を削減するが、明らかに、最悪の場合はすべてのパスを比較することもありえる。つまり、グラフ形の中間形式を用いることでルールの保守性、一貫性は向上するが、処理時間は増加するおそれがある。そこで、本方式では次章で述べるように処理時間を短縮するための予備分類、予備比較という処理を行う。

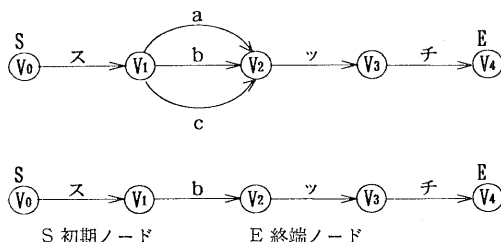


図 1 グラフ型中間表現

Fig. 1 An example of directed graph form.

4. カタカナ表記のゆらぎ検出手順

基本的には、文章中のカタカナ表記を抽出し中間形式に書き換えた後、中間形式同士の共通パスを探索する。しかし、単純にこれを実施するとカタカナ表記の数を n とすると比較回数は $n \cdot C_2$ となり n が増すと、組合せの数が増加する ($n=10$ のとき 45, $n=20$ のとき 190, $n=30$ のとき 435)。そこで、比較回数を低減するために、予めカタカナ語を予備分類して、 n の数を減らす手段を設けた。図 2 に予備分類を含めた検出手順を示す。C と D が予備分類に相当する。

A. カタカナ列抽出

テキスト中からカタカナ文字列を抽出する。ただし、「長音 (ー) とハイフン (ー) のゆらぎ」なども存在することから、中点や等号記号などの一部の記号文字を含めて抽出する。

B. カタカナ列索引生成

文章中から抽出したカタカナ列をソートして異なり表記ごとに文章中の格納位置情報を記憶する。表 2 のように大規模な文書の場合は、カタカナ異なり数は延べ語数の 10 分の 1 程度に低減する。

C. 安定カタカナ文字による分類

ゆらぎ規則の左辺に出現しないカタカナ文字を安定カタカナ文字と呼ぶ。カタカナ文字の中にはほとんどゆれが生じないものがある。少なくともゆらぎ規則に

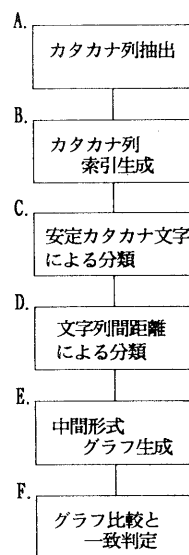


図 2 検出手順

Fig. 2 Detection procedure.

出現しないカタカナ文字は他の記号に変わらないので、この安定カタカナ文字をインデックスとして異なりカタカナ語を分類する。例えば、「ス」「ル」「セ」がゆらぎ規則に現れていないとすると、「テキストチャ、テクスチャ、ペンシルバニア、アクセント、ペンシルヴァニア、テスト、パセリ、ベルト」の8種類のカタカナ列は以下のように三つに分類される。

「ス」…テキスチャ、テクスチャ、テスト
 「ル」…ペンシルバニア、ベルト、ペンシルヴァニア
 「セ」…アクセント、パセリ

D. 距離による分類

Cで分類されたカタカナ列同士の類似度を距離関数で定義し、距離がある閾値以下のもの同士だけを比較対象とする。距離関数は、中点等の一文字の異なりを許容するように、以下のように定義する。

二つの文字列の距離

不一致の場合は次文字を比較するという条件で、先頭から順に二つの文字列の一致文字を数え、これを k とする。

その一致文字の間に他の文字が存在するとき、 k から (間の文字数-1) × 0.5 をペナルティーとして除いた値を x とする。

この x を短い方の文字列長さで割った値の百分率を二つの文字列の距離とする。

距離関数 D 、一致得点を計算する関数 $match$ は以下のように定義する、なお、 s_2 や t_3 等のサフィックスがついた英小文字は各々カタカナ 1 文字を表す。

文字列 $S = s_1, s_2, s_3 \dots s_n$

文字列 $T = t_1, t_2, t_3 \dots t_m$

$$n \geq m$$

とすると、

$$D(S, T) = match(S, T) / m$$

$$m = \min\{|S|, |T|\}$$

s_i, t_j 以降の一致得点を計算する $match()$ は以下のように定義する。

$$match(s_i, t_j) =$$

{

$$s_i = t_j \text{ の場合 } 1 + match(s_{i+1}, t_{j+1})$$

その他の場合

$$1 + match(s_{i+y+1}, t_{j+x+1})$$

$$- 0.5 * (x - 1) - 0.5 * (y - 1)$$

この際

$$s_{i+y} = t_{j+x} \text{ かつ } \min(x) \text{ かつ } \min(y)$$

}

E. 中間形式生成

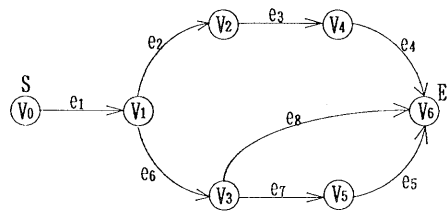
中間形式同士の比較はステップDまでの処理で限定された同一分類中の二つのカタカナ列を対象に行う。まず、二つのカタカナ列を各々グラフ形中間形式 G_1, G_2 に展開する。

3章で説明したように、グラフは各頂点がカタカナ列の解釈位置に、各辺が解釈記号(ゆらぎグループ識別子もしくは安定カタカナ文字)に対応している。まず、カタカナ列は先頭からゆらぎ規則の左辺と比較され、合致しない場合は、安定カタカナ文字を対応させた辺を生成する。合致する場合は、合致する部分をゆらぎグループ識別子記号に書き換え、識別子と対応させた辺を生成する。各辺の先の頂点はカタカナ列の解釈位置に対応しているため、各頂点から同様に辺を伸ばす。この際、解釈位置が共通な場合は、頂点を共有する。

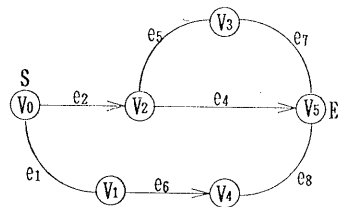
図3に中間形式例を示すが、図で V_n (n は 0 以上の整数) が頂点、 e_n が辺を表す。また、スタート位置の頂点は S 、最終位置の頂点は E と対応する。

F. グラフ比較

二つのグラフの頂点 S から頂点 E に達するパスの上にある辺の解釈記号(安定カタカナ文字もしくはゆらぎ識別子)の列同士の中に、一つでも等しいものがあれば、二つのグラフで表現される中間形式は、「相互



(a) 中間形式グラフ例 G_2
 (a) Example of intermediate form G_2



(b) 中間形式グラフ例 G_1
 (b) Example of intermediate form G_1
 S 初期ノード E 終端ノード

図3 グラフパスの比較例
 Fig. 3 An example of the graph comparison.

にゆらぎ関係にある」と判定する。逆に、共通のパスがないことが判明すると「相互にゆらぎの関係がない」と判定する。

基本的なグラフ比較手順は以下のような縦方向優先の探索である。

頂点 u から v へ記号 e で繋がっているとき、これを $v = \text{next}(u, e)$ と表現する。また、頂点 v から出ている辺の記号の集合を $P = \text{out}(v)$ と表現する。すると、 G_1 上のある頂点 x と、 G_2 上の頂点 y が対応しているとき、次に探索すべき有効辺上の記号の集合 P_e は

$$P_e = \text{out}(x) \cap \text{out}(y) \text{ である。}$$

このとき、 x と y 以降が一致しているかどうかを判定する関数を $\text{yure}(x, y)$ とすると

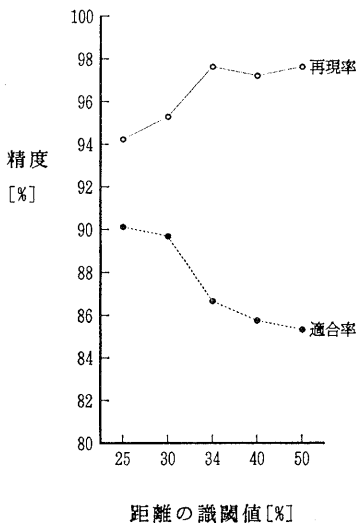
$$\text{yure}(x, y) = \begin{cases} x = y = T & \text{の場合 一致} \\ x = E \quad y \neq E & \text{の場合 不一致} \\ x \neq E \quad y = E & \text{の場合 不一致} \\ x \neq E \quad y \neq E & \text{の場合} \\ \{ \\ P_e = \text{out}(x) \cap \text{out}(y) \\ P_e = \phi & \text{の場合 不一致} \\ \text{その他の場合} \\ \text{repeat } (P_e \text{ の各要素 } e \text{ について}) \\ \{ \\ \text{yure}(\text{next}(x, e), \text{next}(y, e)) \text{ の値が} \end{cases}$$


図 4 距離の識閾値に対する精度の変化
Fig. 4 Precision and recall rates for the distance threshold.

一致なら一致

}
不一致

}
}

例えば、図 3 に二つのグラフの例を示すが、 G_1 におけるパス $v_0-v_1-v_3-v_6$ と G_2 におけるパス、 $v_0-v_1-v_4-v_5$ はその辺に対応する記号が e_1, e_6, e_8 と共通なので、これら二つのグラフで表現されるカタカナ列は表記のゆらぎの関係にあると判定する。

5. 評価／結果

表 2 のデータ (複数著者) を評価対象として検出実験を行った結果を図 4 に示す。予備分類段階で用いる距離の閾値をパラメータとして再現率と適合率の変化をグラフ化した。その結果、評価指標である再現率と適合率は片方が高いときに片方が低くなるという傾向が現れた。これは、閾値が大きいと候補数が増加するため、再現率が上がり、適合率が下がったものと解釈できる。

再現率が急に下がる直前の距離閾値 34% という条件で、再現率 97.4%、適合率 86.7% であり、少数の規則で良好な結果を得た。なお、ゆらぎ規則数は 58 である。

また、典型例として 30% のカタカナ含有量の 900 字の文書进行处理した時レスポンス時間は 0.8 秒 (Sun 3 ワークステーション) であった。

6. むすび

本方式では、基本的には辞書を使わず、カタカナ文字列同士のゆらぎ規則を文章中のカタカナ語に適用し、カタカナ表記のゆらぎを検出する。事前分類したカタカナ表記を有向グラフ形式の中間形式に変換し、効率良く比較検出を行う。試作と実験を行った結果、再現率 97.4%、適合率 86.7% という結果を得た。

曖昧性を許容して一致を判定する方式としては、スペルチェックにおける「編集距離」⁹⁾ (Edit Distance) を用いるものが良く知られているが、これは基本的に一文字誤りの検出を対象とするものであり、カタカナ表記のゆらぎ検出のように数箇所以上が異なる不一致の度合いが高い場合には、粗い基準で予備分類した後、グラフ比較等で最終判定する本方式が再現率を上げるのには適していると考えられる。

今後の課題としては、高速化および「マンマシンイ

「インターフェイス」と「インタフェイス」のようなカタカナ複合語中の表記が他のカタカナ語とゆらぐケースの対策が残っている。

また、本方式を拡張し、辞書見出しを予め予備分類しておけば、入力読みに対して、曖昧性を許容した辞書検索を行うこともできる。今後検討する応用先として、校正推敲支援以外にテキスト検索や音声認識や文字認識の後に行う認識誤り補正等も考えてゆきたい。

参 考 文 献

- 1) 黒田ほか：日本語文におけるカタカナ英語の研究，情報処理学会自然言語処理研究会，68-3 (1988)。
- 2) 島津ほか：カタカナ異表記・誤記修正機能の開発・評価，第44回情報処理学会全国大会論文集，3Q-4，pp. 3-249-250 (1992)。
- 3) 青江：カタカナ異表記の生成および統一手法，情報処理学会自然言語処理研究会，94-5，pp. 33-40 (1993)。
- 4) 牛島ほか：日本語文章推敲支援ツール『推敲』のパソコン上での実用化，コンピュータソフトウェア，Vol. 6, No. 4，pp. 55-67 (1989)。
- 5) 奥村ほか：日本語校正支援システム「FleCS」，情報処理学会自然言語処理研究会，87-11，pp. 83-90 (1992)。
- 6) 福島ほか：日本語文章作成支援システム COM-ET，信学技報，OS 86-21，pp. 15-22 (1986)。
- 7) 国語審議会：外来語の表記，答申 (1991.2)。
- 8) 国立国語研究所：現代表記のゆれ，秀英出版 (1983)。
- 9) Hall, P. A. V. and Dowling, G. R.: Approximate String Matching, *Comput. Surv.*, Vol. 12, No. 4, pp. 381-402 (1980)。

(平成6年1月10日受付)
(平成6年9月6日採録)



久保田淳市 (正会員)

1957年生。1980年早稲田大学理工学部電気工学科卒業。1982年同大学院博士前期課程修了。同年松下電器産業(株)入社。以来同社中央研究所，情報システム研究所，情報通信研究所にて日本語入力，日本語文書処理の研究に従事。1988年情報処理学会研究賞受賞。電子情報通信学会会員。



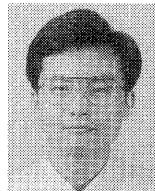
庄田 幸恵

1963年生。1986年大阪大学理学部数学科卒業。同年松下電器産業(株)に入社。現在，同社情報通信研究所にて日本語処理技術，文書処理技術の研究開発に従事。



河合 眞宏 (正会員)

1960年奈良で生をうける。1985年大阪大学基礎工学研究科情報工学専攻修士課程修了。同年松下電器産業(株)に入社。現在，同社情報通信研究所にて日本語処理技術，文書処理技術の研究開発に従事。



玉川 博文 (正会員)

1964年生。1987年広島大学工学部第2類(電気系)卒業。1989年同大学院博士課程前期修了。同年松下電器産業(株)入社。現在，日本語ワープロの開発に従事。



杉村 領一 (正会員)

昭和55年京都大学工学部情報工学科卒業。同年松下電器産業(株)入社。かな漢字変換アルゴリズム，定形文書作成システム等の開発に従事。昭和60年より新世代コンピュータ技術開発機構(ICOT)へ出向。談話理解実験システムDUALS，汎用日本語処理系LTB，構文解析システムSAX，形態素解析システムLAX等の研究開発に従事。平成元年より松下電器産業(株)にて機械翻訳システムの研究開発に従事。昭和63年人工知能学会論文賞受賞。電子情報通信学会，人工知能学会，ソフトウェア科学会，言語処理学会，ACM各会員。