「明海日本語」第 18 号　増刊（2013.11）

〈論　　文〉

語形間距離の計算における「重みづけ」

"Weighted Levenshtein Distance" for Calculating the Distance between Words

鑓水兼貴（やりみずかねたか）
国立国語研究所 理論・構造研究系 プロジェクト非常勤研究員
Kanetaka Yarimizu
Adjunct Researcher, Department of Linguistic Theory and Structure,
National Institute for Japanese Language and Linguistics

キーワード：レーベンシュタイン距離 語形間距離 重みづけ
Keywords: Levenshtein distance, distance between words, weighting factor

要旨

本稿は，語形間距離を計算する際に音声間距離を用いる意義を示すものである．様々な語形間の類似度を算出する方法として，本稿ではパターンマッチングの手法の一つであるレーベンシュタイン距離を採用した．しかし語形間の比較処理において，文字（音声）の違いを無視して距離を算出することには問題が残る．そのため，レーベンシュタイン距離を計算する際に，音声間距離を「重みづけ」として利用した．そうした「重みづけ付きのレーベンシュタイン距離」を用いることによって，語形間距離をより正確に計算することができることを示した．

Abstract

This study attempts to show a method to calculate the distance between two words applying the distance between phonemes. As a method to calculate the similarity of various forms, I employ the “Levenshtein distance,” which is one of the methods of pattern matching. However, it is a problem to calculate the distance ignoring the difference in phonemes. In the process of calculating Levenshtein distance, the distance between phonemes is used as weighting factor. By using "weighted Levenshtein distance", it is possible to calculate more exact distance between two words.

1. はじめに

鑓水(2009)は，国立国語研究所編『方言文法全国地図』の回答語形と共通語形と語形間距離を求め，得られたデータに対してクラスター分析を適用することで，日本語方言の計量的区画をおこなった．このほかにも日本の方言研究では，井上・河西(1982)，井上(1983)，熊谷(2002)，沢木(2002)など，方言形どうし，もしくは方言形と共通語形を比較して計量的分類をおこなう研究がなされてきた．これらの語形比較の方法は，どれも完全に一致するか否かという方法であった．

しかし完全一致のみを一致とする場合，１文字（１音）でも異なれば，不一致とみなされるため，一般的な語形の類似性の感覚とは異なる可能性がある．そのため，語形間距離の計算には，レーベンシュタイン距離などの文字単位での比較手法が利用されている．原理は単純だが，通常はコンピュータによって計算するため，日本語学・方言学における研究では，あまり利用されてこなかったと思われる．近年は，検索エンジンの入力欄における類似ワードを示す機能や，大規模コーパスにおける語形処理などで，日本語学・方言学においても文字単位での比較手法が脚光を浴びる可能性がある．

そうした語形間距離の計算において問題となるのは，比較する個々の文字（音声）の間の類似性である．１文字異なっただけで不一致となることが問題となるように，近い音であるにも関わらず不一致とみなされることもまた問題となりうる．この場合，さらに音声間の類似性についても考慮する必要がある．

本稿では，まず語形間距離の計算方法として，レーベンシュタイン距離について解説する．つづいて，レーベンシュタイン距離の計算における「重みづけ」として音声間距離を利用する方法について検討する．なお，本稿では「距離」と「類似度」の使い分けをしていない．異なる部分がある両概念であるが，基本的に正反対の関係であると考え，同じように使用する．

2. 語形間の比較における先行研究

Yarimizu et.al.(2004) では，フランス語方言の計量的区画をおこなう際に，共通語形と方言形との間の類似度の得点を，当該言語の知識を基にして主観的に設定した．このように類似度を決定する場合，個別の判定における信頼性は高くなるが，得点そのものの判断の客観性には疑問が生じる．しかし語形間の類似度を語形の表面的な比較だけで計算すれば，その言語における語構成や，語義の変遷などを無視することになり，言語研究としての疑問が残る．Kawaguchi(2006)は，言語の計量的研究において言語データに対する重みづけが結果に大きく反映するとしている．そのため得点化の方法については，言語的な側面も意識する必要がある．

本節では，語形間の類似性を数量的に測定する方法論について概観する．語形間の類似度の測定には，言語年代学と音声認識という２つの分野が貢献している．

言語年代学は，比較言語学の考え方が源流となっている計量的分析手法であり，アメリカの言語学者 Swadesh らによって提唱された．比較言語学における言語の系統を計量的手法によって探るために，同一の祖語から分裂したとされる２つの言語間の相関関係を測定した．相関が高い場

合には両言語の共通の祖語からの分裂時期は新しく，相関が低い場合には古いと考え，両言語の分裂時期を計算によって推定する．

言語間の距離を求めるためには，語形の比較をする必要があり，さまざまな手法が開発された．代表的な研究としては安本・野崎(1976)がまとめているが，

・ポイヤによる，語頭音の比較をおこなう「語頭音検定法」
・ベンダーによる，「子音－母音－子音」連続の比較をおこなう「CVC 検定法」
・オズワルドによる，類似する子音の比較をおこなう「子音検定法」

などがある．特にオズワルドの方法は，子音の一致を厳密にせず，調音法や調音点を考慮している点で，本稿の趣旨に通じる．現在では下火になっている研究だが，言語における基礎語の選定に貢献した重要な研究といえる．

一方，音声認識は，文字認識や画像認識などパターン認識の一部として情報処理系の分野で発展した実用的な研究である．たとえば，一方に正解となる語形の音声が存在し，もう一方に発声したノイズの多い同じ語形の音声が存在するとき，この２つの音声波形の一致度を計算によって導き出すことで，発声した語の認識が可能となる．

コンピュータの発展にともなって計算手法も発展し複雑化していったが，初期の手法は非常に単純である．２つのパターンの類似性を計算する方法の一つに動的計画法（dynamic programming）がある．動的計画法は，複雑な問題を部分問題に分割することによって効率的に解く手法で，この動的計画法を用いた照合（matching）方法は，一般に「DP マッチング」と呼ばれる．２つの記号列に対して，先頭の記号から順に比較していき，一致する記号が出るまで，双方の記号列に対して，それぞれ挿入・置換・削除といった作業を続ける手法である．

DP マッチングは発想が簡潔であることから広い分野に応用された．音声認識における初期の計算方法に用いられたほか（横山・板橋 1978，中川・義永 1985），生物学における DNA の塩基配列の比較にも用いられている（岸野・浅井 2003）．

DP マッチングの欠点は，比較方向が一方向しかないことである．時系列の比較でしかできないため，片方の記号列がもう一方の記号列の繰り返しの場合（例：abc と abcabc）や，前部要素と後部要素が入れ替っている場合（例：abcdef と defabc）には部分的にしか対応関係を出すことができない．語の内部構造の計算方法については今後の課題とし，本稿では先頭からの文字列比較をおこなう．

音声認識においては，こうした DP マッチングの欠点を補い，記号の並びを出現確率によって考える「隠れマルコフモデル（Hidden Markov Model）」が広く利用さている．しかし，パラメータ設定の複雑化などの欠点もあり，DP マッチングも利用されている．

なお，音声記号からなる文字列を１文字（１音）ずつ比較する場合，１音の継続時間がすべて一定として扱われてしまうことや，強勢や音調といった情報を無視されてしまうことなど，言語

的にすべての情報が正確に扱われているとは限らない点を注意しなければならない.

3. レーベンシュタイン距離

3.1. 考え方

文字列の比較で使用される DP マッチングの方法の一つに「レーベンシュタイン距離」がある. これは Levenshtein(1965)によって考案された文字列間の距離計算方法である. ２つの文字列を一致させるための置き換え作業の回数によって求められる. 実際の編集における校正作業と同じように, 置換, 挿入（φ からの置換), 削除（φ への置換）といった作業のみで一致度を測定するため, 編集距離（edit distance）とも呼ばれている.

たとえば, kaita という文字列を keda という文字列に一致させるとする. この場合, 手順としては, kaita の ai を e に置換し, 次に t を d に置換することが考えられる. ただし ai から e への置換は, １文字単位ではできないため, 最低 2 回の作業（a 削除と i→e 置換, もしくは, a→e 置換と i 削除）が必要である. そのため合計で 3 回の作業で置き換えが完了する. その置き換えに要するコストを, 両文字列間の類似性の指標として用いたものがレーベンシュタイン距離である.

上述の置き換え作業の方法は１通りとは限らない(実際 kaita と keda の場合, ２通り存在する). また, 文字列が複雑になると手作業での比較は困難になる. そのため, 自動的に最も効率のよい, 最小の置き換え回数を求める計算方法が必要となる. その計算方法として動的計画法が用いられる. そのためレーベンシュタイン距離は DP マッチングを利用して求めることができる.

文字列間の類似性を自動的に算出できるレーベンシュタイン距離は, おもに情報工学の分野で多く用いられている. 言語学においても, オランダの計量方言学の研究者である Heeringa(2004)が, オランダ国内の方言区画のための処理にレーベンシュタイン距離を利用している.

3.2. 計算方法

3.2.1. プログラム

計算方法として, Heeringa(2004)で示されたレーベンシュタイン距離を求めるプログラム（疑似コード）を, 以下に引用する.

```
function Levenshtein_distance(S1,S2)
begin
  for i:=0 to m do begin
    for j:=0 to n do begin
      upper=upperleft=left:=maxint;

      if i>0
        then upper:=dist[i-1,j]+weight(S1[i],ø);

      if i>0 and j>0
        then upperleft:=dist[i-1,j-1]+weight(S1[i],S2[j]);

      if j>0
        then left:=dist[i,j-1]+weight(ø,S2[j]);

      dist[i,j]:=min(upper,upperleft,left);
      if dist[i,j]=maxint then dist[i,j]:=0;
    end
  end

  Levenshtein_distance:=dist[m,n];
end
```

プログラム自体は非常に短いが，プログラミング経験者でない場合，理解は容易ではない．そのため，プログラム自体の説明ではなく，プログラムの動作について，２つの文字列の比較例をもとに説明する．文字列には,「書いた」の共通語形である kaita と東北地方にみられる方言形 keda を用いる．

プログラムは，文字列どうしの距離を，互いの先頭から順に比較して算出する．そのため，比較をおこなうための５列４行のマトリクスを用意する．図１は，kaita と keda の比較をおこなうためのマトリクスである．２つの文字列のどちらを基準としてもよいが，ここでは kaita を keda に一致させるための処理，ということで説明していく．

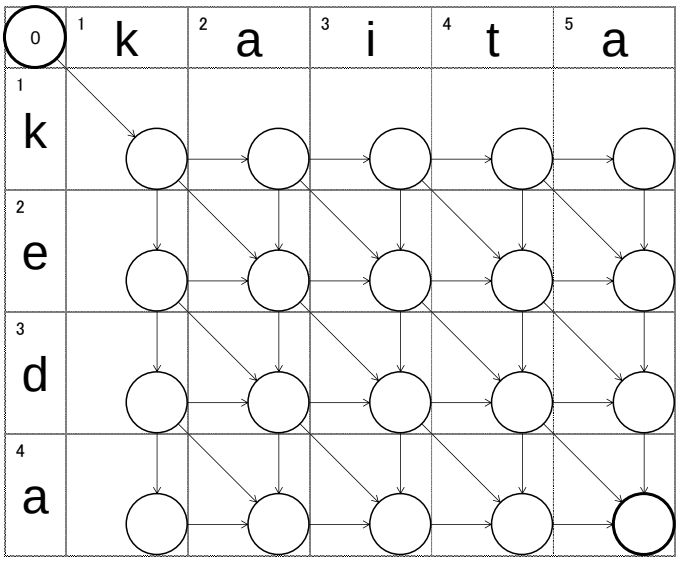
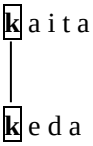


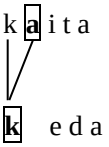
図１　比較作業用のマトリクス

マトリクス内の各セルは，対応する文字の比較をおこなう場所であり，セル内にある円形の部分には両文字列のその位置までの距離が入る．先頭から比較するため，便宜的に互いの文字列に 0 文字目を設け，左上の比較開始位置（0 列 0 行）に距離 0 を設定する．右下（5 列 4 行，以後座標のように(5,4)と記述）が比較終了位置で，最終的な２つの文字列の距離となる．

各円形から右・右下・下の３方向に伸びる矢印線は，上側の文字列からみて，それぞれ削除・置換・挿入をあらわす．たとえば(1,1)の比較は，



のように，先頭の k どうしの比較を意味する．そこから右側(2,1)に移動する場合，



のように，keda は 1 文字目のままで，kaita の 2 文字目の a と比較することになる．対応関係でみると，kaita の 2 文字目の a を削除する（kita となる）ことを意味する．なお逆に左側の keda を基準にすると，右・右下・下の矢印線はそれぞれ挿入・置換・削除となる．上述の右側移動を keda からみれば，a を挿入する（kaeda となる）ことを意味する．

削除・置換・挿入のそれぞれの作業に対しては，コストを定める必要がある．そして総コスト数が，２つの文字列間の距離となる．なお，すべての作業のコストを 1 とする場合，総コスト数は削除・置換・挿入の作業回数に一致する．

初期段階では，文字列のどの部分を挿入・置換・削除するかはわからないため，すべての文字を対応させて距離を計算し，もっとも効率よく一致させる，すなわち，もっとも総コストの低い経路を探索する必要がある．

3.2.2. 先頭２文字までの距離

つづいて実際の比較作業についてみてみる．説明のため，ここでは先頭２文字の文字列間の距離について説明する．図 2 は，図 1 の左上の部分に相当する．円の中には，その位置における２つの文字列の距離を表す.

図 2 において初期状態は(0,0)の距離 0 である．最初の比較は(1,1)における先頭の文字どうしの比較である．両方とも k どうしで一致しており，(1,1)のコストは 0 のままとなる．

図 3 は，(1,1)の次の比較作業であるが，右(2,1)，右下(2,2)，下(1,2)の３方向の比較経路がある．

それぞれの経路で進んだ場合のコストは，それぞれのセルにおける文字の比較によって得られる．k≠a，e≠k，e≠a で，どれもコストは 1 となる．それぞれの方向のコストは斜字体で示す．

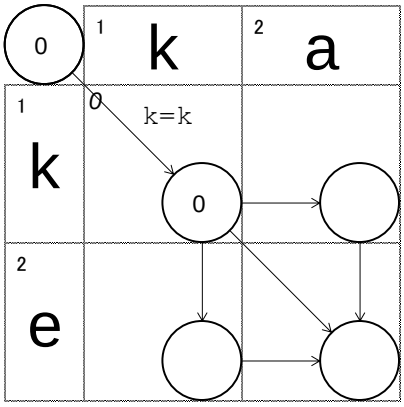


図 2　比較作業(1)

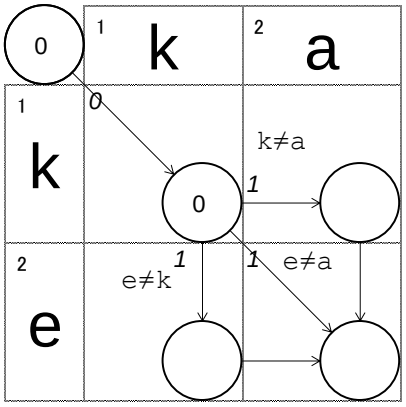


図 3　比較作業(2)

先頭の文字と比較する場合は，それ以前の比較が存在しないため，(2,1)と(1,2)の距離は自動的に決定する．図 4 において(2,1)と(1,2)の円内に 1 が入る．

さらに図 4 では，(2,2)における 2 文字目どうしの比較作業が残っている．すでに左上(1,1)からの斜めの経路はコスト 1 と決定している．上(2,1)からの経路と，左(1,2)からの経路は，ともに e≠a でコスト 1 となる．

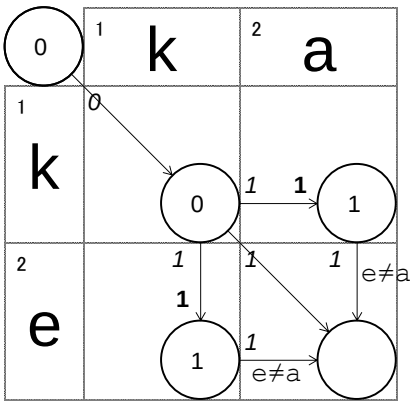


図 4　比較作業(3)

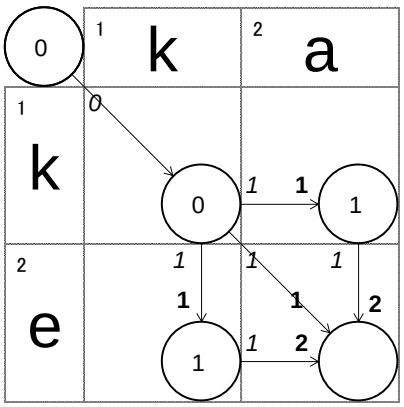


図 5　比較作業(4)

図 5 の(2,2)には，上，左上，左の３方向の経路からの総コストが太字で示されている．たどってきた経路によって総コストが異なる．上からの経路と左から経路による総コストは 2 であるのに対して，左上からの経路だとコストは 1 である．(0,0)から(2,2)までの距離は，その３つの経路の中の最小コストを選択しなければならい．

これを前節における，削除・置換・挿入ということばで説明すると，ka を ke に一致させるためには，

上からの経路　：　a 削除　　＋　e 挿入　コスト 2 (ka→k　→ke)
左上からの経路：　a→e 置換　　　　コスト 1 (ka　→　ke)
左からの経路　：　e 挿入　　＋　a 削除　コスト 2 (ka→kea→ke)

の３つの経路があることになる．図 6 において，(2,2)における距離は，３つの経路のうち総コストがもっとも小さい，左上から経路のコスト 1 が選択される．

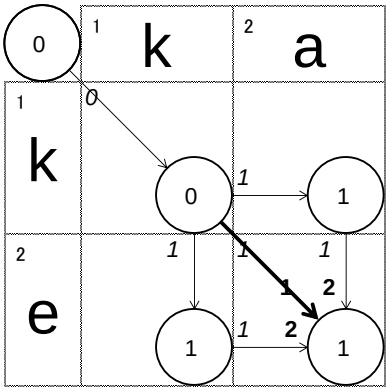


図 6　比較作業(5)

3.2.3. 計算結果

前節の作業を繰り返し，すべてのセルにおける距離を計算したものが図 7 である．そして，比較終了位置である，右下(5,4)のセルの円内の値 3 が，この２つの文字列間の距離ということになる．すなわち，kaita と keda のレーベンシュタイン距離は 3 である．

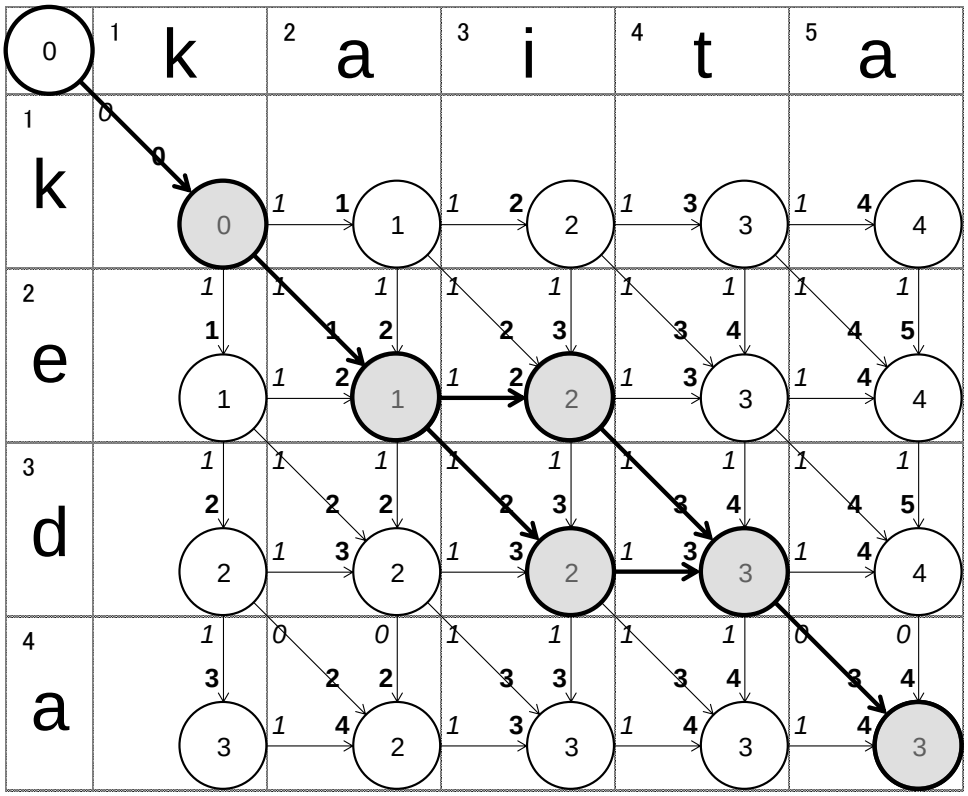


図 7 比較結果

つづいて文字の対応関係をみる．比較終了位置(5,4)から逆に，最小コストの値をたどっていくこと(backtrace)によって，kaita がどのような過程で keda に置き換えられていったかがわかる．たどっていくと，図 7 に示すように総コスト 3 となる経路は２通りあることがわかる．文字の対応関係は,

(1)

k a i t a

k e d a

(2)

k a i t a

k e d a

となり，これを，挿入・置換・削除で表現すると,

(1) kaita → (a→e 置換) keita → (i 削除) keta → (t→d 置換) keda
(2) kaita → (a→e 置換) keita → (i→d 置換) kedta → (t 削除) keda

の２通りとなる．一見すると，(1)の経路のほうが自然のように思える．削除・挿入・置換のコストがすべて１であるため，i の削除コストと，i→d の置換コストを同じなることで複数の経路が選択されたことになる．このように置き換えコストの値が全体の結果に強く影響を及ぼしていることがわかる．

3.3. 置き換えコストの種類

レーベンシュタイン距離で定義されるコストは，「削除」「置換」「挿入」の３種類である．しかし文字列の変換方法はこれだけではない．前後の文字が入れ替わる「転置」や，文字が別の位置が動く「移動」，一部の文字列を繰り返す「複写」などが存在する．

これらは文字列内の位置の入れ替わりを重要視する方法で，形態素レベルで考える場合に意味があると思われる．語義は同じでも，移動によって語の中心的意味をなす要素がさまざまな位置にある場合や，転置によって前部要素と後部要素が入れ替わる場合，複写によって特定要素が繰り返される場合は，レーベンシュタイン距離では，実際の感覚よりも離れた語として扱われてしまう．

しかし，移動や複写には単位の認定が必要であり処理方法が複雑になる．そのため本稿では移動や複写については扱わないことにする．転置については，連続した２文字の入れ替わりであれば計算も比較的簡単である，この手法は Damerau-Levenshtein distance と呼ばれる．転置については後述する「重みづけ」によってある程度解決されると思われる．

4. 音声間類似度

4.1. 「重みづけ」の必要性

レーベンシュタイン距離においては，削除・置換・挿入の置き換えコストは 1 と設定されている．そのため文字（音声）の一致，不一致によるコストの計算では，計算方法に問題がなくても，音声の対応関係が問題となる可能性がある．

図 7 の kaida と keda のレーベンシュタイン距離について，再度検討する．セル(2,2)からは，(1) i を削除する右方向と，(2) i を d に置換する右下方の２つの経路にわかれている．

kaita が keda への変化する場合，音声的には，

・二重母音 ai が融合したのち単母音化して e となる
・無声子音 t が前後の母音の影響で有声化して d となる

というような説明がなされるため，(2)より(1)のほうに説得力がある．

しかし，そもそも 1 文字単位の文字列比較において，前後の音声の影響は考慮されない．(1)の i の削除は，ai の単音化とは異なる現象である．逆に(2)の i から d への置換は，有声化が前後の母音の影響を受けている以上，d の有声化と全く無関係とはいえない．そのため，(1)の経路が自然で，(2)の経路が不自然と簡単に決めることはできない．

レーベンシュタイン距離ではコストはすべて 1 と定義されているため，ai が e に近いことや，t と d が近いことなどは考慮されない．音声的に近い置き換えはコストが低く，音声的に遠い置き換え作業はコストが高いほうが自然な距離計算ができると思われる．

以上から，自動的に最適な対応関係をみつけ，より人間の感覚に近い距離を計算するために，音声の類似性に基づいてコストの「重みづけ」をすることが考えられる．

4.2. 音声の類似性

音声の類似性については，言語間の距離を算出していた Swadesh の方法でも取り入れられていた（安本・野崎 1976）．横山・板橋(1978)は，音声認識において，音声どうしの比較方法として，早田(1973)の生成音韻論における弁別素性を用いている．単語を音声単位に分割し，時系列に各音声の弁別素性が反応しているとみなして比較をおこなっており，比較には動的計画法を用いている．Heeringa(2004)は，音声間の比較にレーベンシュタイン距離を用いている．コストの算出方法として，調音音声学的方法，生成音韻論の弁別素性による方法，音響音声学的方法の 3 種類の方法を用いてオランダ語における音声間の関係を求めている．

計量国語学においては，水谷(1990)が，語形間一致度を算出方法として，レーベンシュタイン距離に類似した計算方法を提案している．動的計画法のように細かく文字を対応させる方法ではないため，計算結果にはやや問題が残るが，単なる文字比較でない点は重要である．日本語に限定しているため仮名文字単位の比較方法となっているが，水谷は仮名文字間の比較において，単純に「一致＝0，不一致＝1」とせずに，「五十音の段のみの一致に 0.5，列のみの一致の場合に 0.25」と中間値を与えることを提案した．これは事実上，母音一致に 0.5，子音一致に 0.25 を与えていることになる．値は水谷による恣意的なものであるが，子音の一致よりも母音の一致を重要視していることがわかる．

4.3. 音声間距離を用いたレーベンシュタイン距離

4.3.1. 音声間距離の計算手順

音声の類似性の指標には，横山・板橋(1978)にならい，生成音韻論における弁別素性を用いた．早田(1973)による日本語の弁別素性の一覧を表 1 に示す．

表 1　日本語の弁別素性表（早田 1973 より）

		p	b	t	d	k	g	s	z	m	n	r	'	w	y	i	e	a	o	u
モーラ性	mora	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(+)	(+)	(+)	(+)	(+)
母音性	vocalic	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(+)	−	−	−	+	+	+	+	+
子音性	consonantal	+	+	+	+	+	+	+	+	+	+	+	−	−	−	−	−	−	−	−
高舌性	high	(−)	(−)	(−)	(−)	(+)	(+)	(−)	(−)	(−)	(−)	(−)	−	+	+	+	−	−	−	+
後方性	back	(−)	(−)	(−)	(−)	(+)	(+)	(−)	(−)	(−)	(−)	(−)	(−)	+	−	−	−	(+)	+	+
低舌性	low	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(+)	(−)	(−)	(−)	−	+	−	(−)
前方性	anterior	+	+	(+)	(+)	+	+	(+)	(+)	(+)	(+)	(+)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)
舌端性	coronal	−	−	+	+	−	−	+	+	−	+	(+)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)
遮音性	obstruent	+	+	+	+	+	+	+	+	−	−	−	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)
有声性	voiced	−	+	−	+	−	+	−	+	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)
連続性	continuant	(−)	(−)	−	−	(−)	(−)	+	+	(−)	(−)	(−)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)
鼻音性	nasal	(−)	(−)	(−)	(−)	(−)	(−)	(−)	(−)	+	+	−	(−)	−	(−)	(−)	(−)	(−)	(−)	(−)

素性の体系は，Chomsky & Halle(1968)にならったものである．括弧で囲まれた部分は，余剰的特徴である. 音素に関する体系であるため, 音声から抽出される要素を表しているとは限らない. しかし，横山・板橋(1978)や，宮城・高良(2007)など，実際の音声に対して適用してもある程度の効果が得られており，音声工学の概説書でも引用されている（板橋 2005）．このため本稿では，この表から音声間距離を計算する.

表 1 のほかにも，母音 i の前の口蓋化子音など，日本語では不可欠な音声がある．本稿においては計算を簡単にするため，この表のみを用いるが，他の音声についても弁別素性を追加するなど，検討する必要がある．また，各弁別素性の機能負担量についても考慮されていないため，この点についても今後の課題としたい.

音声間距離（類似度）の計算には，ピアソンの積率相関係数を採用した．まず表 1 において，余剰的特徴をあえて程度とみなして，－，（−），（＋），＋の 4 段階とし，1〜4 に得点化した．つづいて音声間の相関係数 r を計算し，その値を類似度とみなした.

相関係数はコサイン類似度と同様に扱うことができるため，変換式 $d^2=2(1-r)$ を用いて，相関係数を距離 d に変換した．d は 0〜2 の値をとるため，さらに 2 で割って 0〜1 の値をとる指標に変換した.

音声間の相関係数の一覧を表 2 に，相関係数を距離に変換した結果を表 3 に示す.

表 2　音声間の相関係数

	p	b	t	d	k	g	s	z	m	n	r	`	w	y	i	e	a	o	u
p	1.000	0.678	0.573	0.263	0.935	0.642	0.500	0.198	0.229	-0.113	0.064	-0.344	-0.321	-0.314	-0.397	-0.283	-0.424	-0.342	-0.488
b	0.678	1.000	0.239	0.573	0.579	0.934	0.104	0.470	0.398	-0.007	0.252	0.000	-0.139	-0.054	-0.246	-0.049	-0.294	-0.182	-0.398
t	0.573	0.239	1.000	0.678	0.502	0.179	0.678	0.372	-0.142	0.260	0.221	-0.454	-0.417	-0.412	-0.512	-0.372	-0.545	-0.443	-0.625
d	0.263	0.573	0.678	1.000	0.156	0.482	0.289	0.652	0.029	0.369	0.414	-0.115	-0.241	-0.157	-0.369	-0.142	-0.424	-0.289	-0.545
k	0.935	0.579	0.502	0.156	1.000	0.674	0.390	0.042	0.170	-0.210	0.000	-0.453	-0.040	-0.232	-0.378	-0.462	-0.502	-0.312	-0.255
g	0.642	0.934	0.179	0.482	0.674	1.000	0.000	0.324	0.350	-0.108	0.194	-0.117	0.143	0.024	-0.238	-0.238	-0.388	-0.161	-0.175
s	0.500	0.104	0.678	0.289	0.390	0.000	1.000	0.652	-0.229	0.113	0.127	-0.115	-0.241	-0.157	-0.369	-0.142	-0.424	-0.289	-0.545
z	0.198	0.470	0.372	0.652	0.042	0.324	0.652	1.000	-0.062	0.237	0.343	0.247	-0.065	0.110	-0.237	0.099	-0.319	-0.142	-0.494
m	0.229	0.398	-0.142	0.029	0.170	0.350	-0.229	-0.062	1.000	0.618	0.277	-0.125	-0.350	-0.137	-0.247	-0.124	-0.277	-0.201	-0.344
n	-0.113	-0.007	0.260	0.369	-0.210	-0.108	0.113	0.237	0.618	1.000	0.480	-0.124	-0.411	-0.160	-0.359	-0.145	-0.411	-0.283	-0.525
r	0.064	0.252	0.221	0.414	0.000	0.194	0.127	0.343	0.277	0.480	1.000	-0.277	-0.194	-0.265	-0.069	0.069	-0.102	-0.032	-0.173
`	-0.344	0.000	-0.454	-0.115	-0.453	-0.117	-0.115	0.247	-0.125	-0.124	-0.277	1.000	0.233	0.273	-0.124	0.124	0.492	0.115	-0.125
w	-0.321	-0.139	-0.417	-0.241	-0.040	0.143	-0.241	-0.065	-0.350	-0.411	-0.194	0.233	1.000	0.550	0.151	-0.281	-0.043	0.161	0.612
y	-0.314	-0.054	-0.412	-0.157	-0.232	0.024	-0.157	0.110	-0.137	-0.160	-0.265	0.273	0.550	1.000	0.566	-0.008	-0.252	-0.314	0.239
i	-0.397	-0.246	-0.512	-0.369	-0.378	-0.238	-0.369	-0.237	-0.247	-0.359	-0.069	-0.124	0.151	0.566	1.000	0.603	0.228	0.198	0.618
e	-0.283	-0.049	-0.372	-0.142	-0.462	-0.238	-0.142	0.099	-0.124	-0.145	0.069	0.124	-0.281	-0.008	0.603	1.000	0.502	0.652	0.309
a	-0.424	-0.294	-0.545	-0.424	-0.502	-0.388	-0.424	-0.319	-0.277	-0.411	-0.102	0.492	-0.043	-0.252	0.228	0.502	1.000	0.593	0.369
o	-0.342	-0.182	-0.443	-0.289	-0.312	-0.161	-0.289	-0.142	-0.201	-0.283	-0.032	0.115	0.161	-0.314	0.198	0.652	0.593	1.000	0.631
u	-0.488	-0.398	-0.625	-0.545	-0.255	-0.175	-0.545	-0.494	-0.344	-0.525	-0.173	-0.125	0.612	0.239	0.618	0.309	0.369	0.631	1.000

表 3　音声間の距離（相関係数から算出）

	p	b	t	d	k	g	s	z	m	n	r	`	w	y	i	e	a	o	u
p	0.000	0.402	0.462	0.607	0.180	0.423	0.500	0.633	0.621	0.746	0.684	0.820	0.813	0.810	0.836	0.801	0.844	0.819	0.862
b	0.402	0.000	0.617	0.462	0.459	0.182	0.669	0.515	0.549	0.710	0.612	0.707	0.755	0.726	0.789	0.724	0.804	0.769	0.836
t	0.462	0.617	0.000	0.402	0.499	0.641	0.402	0.560	0.756	0.608	0.624	0.853	0.842	0.840	0.870	0.828	0.879	0.849	0.901
d	0.607	0.462	0.402	0.000	0.650	0.509	0.596	0.417	0.697	0.562	0.541	0.747	0.788	0.761	0.827	0.756	0.844	0.803	0.879
k	0.180	0.459	0.499	0.650	0.000	0.404	0.552	0.692	0.644	0.778	0.707	0.852	0.721	0.785	0.830	0.855	0.867	0.810	0.792
g	0.423	0.182	0.641	0.509	0.404	0.000	0.707	0.581	0.570	0.744	0.635	0.747	0.655	0.699	0.787	0.787	0.833	0.762	0.766
s	0.500	0.669	0.402	0.596	0.552	0.707	0.000	0.417	0.784	0.666	0.661	0.747	0.788	0.761	0.827	0.756	0.844	0.803	0.879
z	0.633	0.515	0.560	0.417	0.692	0.581	0.417	0.000	0.729	0.618	0.573	0.614	0.730	0.667	0.786	0.671	0.812	0.756	0.864
m	0.621	0.549	0.756	0.697	0.644	0.570	0.784	0.729	0.000	0.437	0.601	0.750	0.822	0.754	0.790	0.750	0.799	0.775	0.820
n	0.746	0.710	0.608	0.562	0.778	0.744	0.666	0.618	0.437	0.000	0.510	0.750	0.840	0.762	0.824	0.757	0.840	0.801	0.873
r	0.684	0.612	0.624	0.541	0.707	0.635	0.661	0.573	0.601	0.510	0.000	0.799	0.773	0.795	0.731	0.682	0.742	0.718	0.766
`	0.820	0.707	0.853	0.747	0.852	0.747	0.747	0.614	0.750	0.750	0.799	0.000	0.619	0.603	0.750	0.662	0.504	0.665	0.750
w	0.813	0.755	0.842	0.788	0.721	0.655	0.788	0.730	0.822	0.840	0.773	0.619	0.000	0.474	0.651	0.800	0.722	0.648	0.440
y	0.810	0.726	0.840	0.761	0.785	0.699	0.761	0.667	0.754	0.762	0.795	0.603	0.474	0.000	0.466	0.710	0.791	0.810	0.617
i	0.836	0.789	0.870	0.827	0.830	0.787	0.827	0.786	0.790	0.824	0.731	0.750	0.651	0.466	0.000	0.446	0.621	0.633	0.437
e	0.801	0.724	0.828	0.756	0.855	0.787	0.756	0.671	0.750	0.757	0.682	0.662	0.800	0.710	0.446	0.000	0.499	0.417	0.588
a	0.844	0.804	0.879	0.844	0.867	0.833	0.844	0.812	0.799	0.840	0.742	0.504	0.722	0.791	0.621	0.499	0.000	0.451	0.562
o	0.819	0.769	0.849	0.803	0.810	0.762	0.803	0.756	0.775	0.801	0.718	0.665	0.648	0.810	0.633	0.417	0.451	0.000	0.430
u	0.862	0.836	0.901	0.879	0.792	0.766	0.879	0.864	0.820	0.873	0.766	0.750	0.440	0.617	0.437	0.588	0.562	0.430	0.000

4.3.2. 重みづけのあるレーベンシュタイン距離

音声間距離が算出されたことにより，音声間の違いを，語形間距離に反映させることができる．そのため，この音声間距離をレーベンシュタイン距離における置き換えコストとして利用する．類似度の高い音声どうしの置き換えコストは小さく，類似度の低い音声どうしの置き換えコストは大きい．これにより 2 つの語形を比較する際の，文字の対応経路をより的確にみつけることが可能となると思われる．

ただし，この方法では，同一音声が連続している場合，挿入・削除コストが 0 になってしまう．たとえば，ka と kaaaaaaaaaa は同一文字列とみなされてしまう．これを解決する方法として，一つは，長母音や二重母音も弁別素性表に入れて距離計算をして区別する方法が考えられる．もう一つの方法として，挿入・削除処理が時間の短縮・延長と関係するものであることから，継続時間のコストを設ける方法が考えられる．本稿では計算が簡単な後者を採用する．

継続時間のコストは 0.0707107 とした．この値は，相関係数 r=0.99 における距離であり，恣意

的なものである．このコストを，同一音の連続の場合のみ加算するか，異なる場合にも加算するかが問題となるが，本稿では同一音に限らず，すべての挿入・削除処理において加算することにした．しかし，この点は検討課題である．なお，置換時は，同一音であっても距離は 0 のままである．

こうして音声間距離をコストに用いた，「重みづけのあるレーベンシュタイン距離 （weighted Levenshtein distance）」の計算結果を図 8 に示す．

前述の(2,2)から分岐する 2 経路について図 7 と比較してみる．i と e の比較と，i と d の比較自体は，どちらも不一致であることに変わりはない．しかし距離をみると，図 7 では i の削除も i と d の置換もコスト 1 であるのに対して，図 8 では i の削除コストが 0.516 （i と e の距離 0.446 ＋継続時間コスト 0.071），i と d の置換コストが 0.827 となる．このため，図 8 においては，コストの大きい右下方向の経路は選択されず，コストの小さい右方向の(2,1)への経路が選択されることになる．

こうして図 7 では２つあった経路が，図 8 では１つに絞られた．kaita を keda に置き換えるために，a→e の置換ののち i の削除という処理が選択されたことになる．この処理をコストの面からみると，それぞれ，

（処理）		（コスト）
a→e 置換	⇒	e と a の距離
i 削除	⇒	e と i の距離 ＋ 継続時間

に対応しており，a と i が融合して e になった状態に近いと考えることができる．図 8 において，kai と ked の距離が 1.326，kai と ke の距離が 1.015 であることからも，ai が e と対応していることがわかる．

レーベンシュタイン距離の計算における置き換えコストに音声間距離を適用して語形間距離を求めることにより，語形どうしの類似性をより正確に示すことができると思われる．鑓水(2009)でおこなったような計量的な方言区画や，語形使用のパターン分類などにおいて，こうした語形間距離を活用することができるであろう．このほか，調査における多様な回答語形の一次的な整理にも応用することができると思われる．

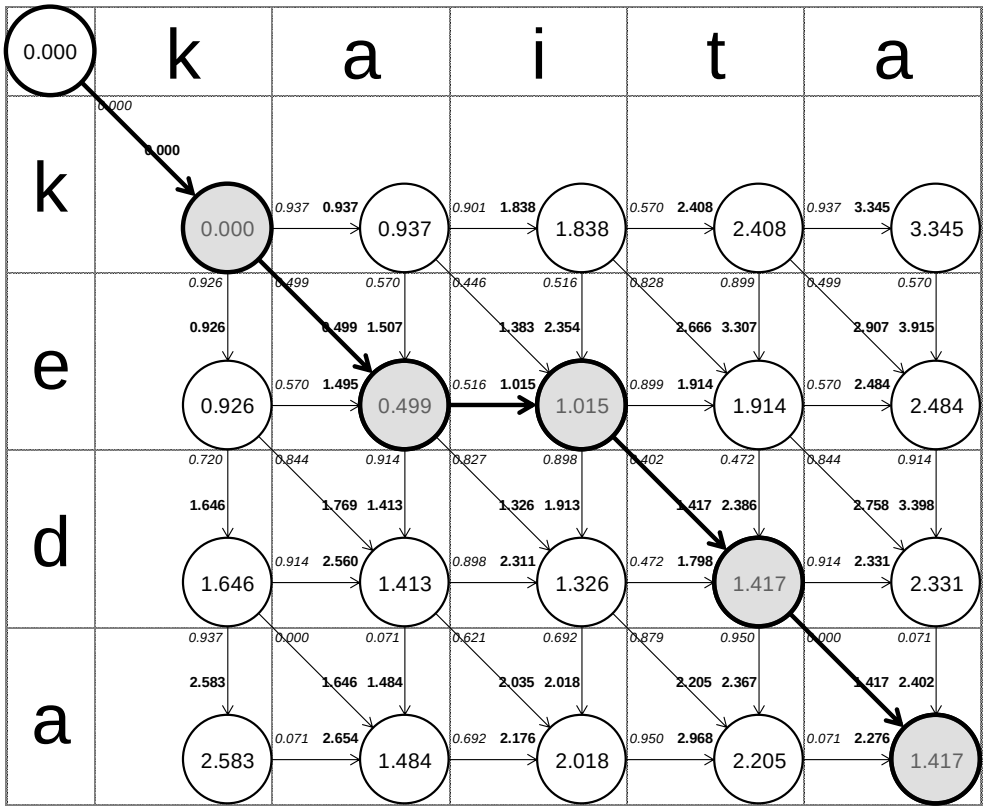


図 8　音声間距離を用いたレーベンシュタイン距離

5. まとめ

以上，本稿では，語形間距離を計算するための方法であるレーベンシュタイン距離について，その原理を説明し，さらに音声間距離をレーベンシュタイン距離の計算における置き換えコストとして使用することの意義について論じた．

鑓水(2009)では，文法項目における方言形と共通語形との語形間距離を求め，計量的な方言区画を試みたが，文法項目のように体系性が関係する場合には，個々の語形間の類似性だけで論じることは難しい場合がある．しかしその一方で，語形の表面的な類似が，類推や異分析による言語変化を引き起こすことにもつながるため，語形間距離からの分類に意味がないとはいえないであろう．

また本稿では，kaita と keda を例にした語形間距離については望ましい結果となったが，表 3 を見る限り，音声間距離には実際の感覚とのずれもみられる．この点は改良すべき課題である．今後も研究を発展させていきたいと考えている．

参考文献

Chomsky Norm and Halle Morris (1968) *The Sound Pattern of English*, Harper & Row.

早田輝洋(1973)「日本語音形論」比企静雄(編)『音声情報処理』東京大学出版会.

Heeringa Wilbert (2004) *Measuring Dialect Pronunciation Differences using Levenshtein Distance*, Rijksuniversiteit Groningen.

井上史雄 (1983)「共通語的文法表現の地理的分布パターン」『国語学』133.

井上史雄・河西秀早子 (1982) 「標準語形による方言区画」『計量国語学』13(6).

板橋秀一(編著) (2005)『音声工学』森北出版.

Kawaguchi Yuji (2006) “Is It Possible to Measure the Distance between Near Language? – A Case Study of French Dialects –“, Near Language Conference, Limerick.

岸野洋久・浅井潔 (2003)『生物配列の統計 核酸・タンパクから情報を読む』 岩波書店.

熊谷康雄 (2002)「方言区画論と方言地理学―計量的方言区画のためのネットワーク法の開発を通して―」馬瀬良雄(監修)『方言地理学の課題』明治書院.

Levenshtein Vladimir Iosifovich (1965) “Binary codes capable of correcting deletions, insertions, and reversals.” *Doklady Akademii Nauk SSSR*. (*Cybernetics and Control Theory* 10 (1966)).

宮城順一・高良富夫(2007) 「外国語・擬音語と弁別素性を用いる音声カナ変換システムの評価」日本音響学会九州支部第 7 回学生のための研究発表会.

水谷静夫 (1990)「距離に基づく語形類似指数」『計量国語学』17(5).

中川聖一・義永洋士 (1985)「誤りを含んだ音素系列からの候補単語の検索」『計量国語学』14(8).

沢木幹栄 (2002)「方言地図データの活用－GAJ のデータによる地点のクラスター分析－」馬瀬良雄(監修)『方言地理学の課題』明治書院.

鑓水兼貴(2009)「共通語化過程の計量的分析―『方言文法全国地図』を中心として―」東京外国語大学博士論文.

Yarimizu Kanetaka, Kawaguchi Yuji and Ichikawa Masanori (2004) "Multivariate Analysis in Dialectology – A Case Study of the Standardization in the Environs of Paris", *Linguistic Informatics* 3, Tokyo University of Foreign Studies.

安本美典・野崎 明弘 (1976)『言語の数理』筑摩書房.

横山晶一・板橋 秀一 (1978)「弁別的特徴と 2 次系モデルに基づいた日本語単語間の距離」『計量国語学』11(4).

本稿は，筆者の博士論文『共通語化過程の計量的分析―『方言文法全国地図』を中心として』第 7 章「『共通語度』を用いた GAJ のクラスター分析」における，レーベンシュタイン距離の説明部分をもとに，全面的に加筆・修正したものである.