Vol.2010-MPS-77 No.21
2010/3/5

木構造と文字列類似度に基づく言語の同一性判定

呉　　靭[†1]　松野　浩嗣[†1]

Yamamoto-Data と SilGIS-Data は異なる学者によって編成された世界諸言語に関するデータである．言語の別名の存在や表記ゆれなどのため，両データに含まれる言語の同一性を判定する必要がある．本研究は Yamamoto-Data と SilGIS-Data に含まれる同一言語を見つけ出すことを目的とし，言語名に加えて言語系統分類も取り入れた手法を提案する．本論文では，まず，系統分類がなす木構造に基づき，言語系統木を定義する．次に，言語名と言語系統分類の曖昧な性質に対し，文字列類似度に基づく言語名と系統分類の類似度を導入し，これらの類似度を用いた言語同一性判定の手法を述べる．さらに，Yamamoto-Data と SilGIS-Data について処理した結果を提示し，提案した類似度が有用かつ効果的であることを示す．

Identifying Same Languages Based on Tree Structure and String Similarity

Ren Wu[†1] and Hiroshi Matsuno[†1]

Yamamoto-Data and SilGIS-Data are world's languages data individually provided by different language researchers. Because of the existence of alternative names of languages as well as their ambiguities, some same languages are expressed by different writings in Yamamoto-Data and SilGIS-Data. Therefore, it is important to identify if two writings express a same language. In order to cope with this problem, we introduce a new method to absorb these ambiguities by applying string alignment technique. Our experimental result for the two language data shows that our proposed method is usefull and effective.

†1 山口大学大学院理工学研究科，山口県
Graduate School of Science and Engineering, Yamaguchi University, Yoshida 1677–1, Yamaguchi City, Yamaguchi Prefecture, 753–8512 Japan

1. はじめに

近年，GIS（地理情報システム）が，そのもつ情報の多角的な分析機能が注目され，言語類型論的研究分野における有効かつ強力な手段として期待されている[1),2)]．GIS を用いた言語類型論的研究を行う際，世界諸言語の言語特徴を集めた属性データと言語の話されている地域の地理情報などを含めた空間データが必要であるが，世界諸言語に関する GIS データは，一般的に利用可能なものは少なく，入手困難な状況にある．そこで，我々は，Yamamoto-Data と SilGIS-Data （詳細は後述する）という 2 つの言語属性データを処理することによって，必要な GIS データを生成する手法を提案した[3)]．その手法では，Yamamoto-Data と SilGIS-Data のそれぞれに含まれている言語の同一性判定，すなわち両データに含まれている言語の対応づけが必要である．

SilGIS-Data では言語の一意的識別子である言語コードがあるが，Yamamoto-Data では言語の識別子として言語の名前が使われている[3)–6)]．両データに含まれている多くの言語は言語の名前が一致していて，一見して言語の名前の一致によるマッチングが可能なように見えるが，実際には，一方のデータに同じ名前の言語が複数含まれていることや，両方のデータに異なる名前をもつ同一言語が存在していることなど，言語名が一致するか否かのみによって言語の同一性を結論づけることができない．故に，言語の名前のほかに，さらなる判定尺度が必要不可欠である．そこで，我々は，この問題を解決するため，世界諸言語が系統的に分類されていて，系統分類情報が木構造をなしていることに注目し，言語名に加えて系統分類の情報も言語同定に取り入れることを着想した．

言語名も系統分類も文字列で表されており，いずれも曖昧な性質をもつ情報である．文字列類似性の評価には編集距離[7),8)] による手法がよく用いられている[9),10)]．それに基づき，我々は言語名の類似度と系統分類の類似度という概念を導入し，それらの類似度に基づく判定ルールを定め，Yamamoto-Data と SilGIS-Data に含まれている同一言語を見つけ出す手法を提案する．

以下，**2.** では Yamamoto-Data と SilGIS-Data の言語同一性判定の必要性について述べる．**3.** では系統分類の角度からの言語データ構造である言語系統木について定義を行い，完全一致言語の検出法について述べる．**4.** では言語名の類似度と系統分類の類似度について定義を行い，Yamamoto-Data の言語に対し，SilGIS-Data から同一言語を見つけ出す手法について述べる．**5.** では，4. の手法の実装を行い，実際処理した結果を提示し，本研究で提案した手法の妥当性および有用性などを考察する．最後の **6.** で本稿をまとめる。

表 1 Yamamoto-Data と SilGIS-Data
Table 1 Yamamoto-Data and SilGIS-Data

No.	第一言語名	属性
212	BAI	···
213	BAI	···
485	CHINANTECO, LALANA	···
1015	JAPANESE	···
1855	NHARON	···
1959	OTOMI, STATE OF MEXICO	···

(A) Yamamoto-Data

No.	第一言語名	言語コード	(複数の)別名	属性
733	Bai	bdj	Bari	···
1565	Chinantec, Lalana	cnl	Chinanteco de San Juan Lalana	···
3295	Japanese	jpn		···
5763	Naro	nkr	Nharo, Nharon, Nhauru, ···	···
6262	Otomi, Estado de Mexico	ots	Hnatho, Otomi del Estado de Mexico, ···, State of Mexico Otomi	···

(B) SilGIS-Data

2. 言語同一性判定の必要性

2.1 2 つの言語データの概要

表 1(A) は文献[6] に掲載されている「言語別語順データ」を指し，2,932 言語の語順に関する言語特徴がまとめられている。下位の方言を言語として編入しているところがあるため（言語と方言の定義が元々曖昧である），実質言語数は 2,870 である．一方，表 1(B) は *Ethnologue* 第 15 版 Web サイト[4),5)] から世界諸言語の属性情報を取得し，表形式にしたデータ[11)] を指し，言語数は 7,229 である．表 1 の (A) と (B) をそれぞれ Yamamoto-Data と SilGIS-Data で表す．

表 1(A) と (B) のいずれも，各行のレコードは 1 言語を表す．表 1(A) の 3 つのフィールドは表 1(B) にもあり，共通項目となっている．**第一言語名**[4)] は言語の名前の 1 つで，**属性**は複数フィールドを含む場合があり，語順や話者人口や言語使用状況等の言語に関する属性情報である．*No* は各々のデータのレコード番号である．なお，アルファベット表記は特に大文字と小文字を区別しない．

一方，**言語コード**[4)] と**（複数の）別名**[4)] は表 1(B) にのみ用いられている．言語コードは国際標準化機構によって定められた ISO639_3 言語コード[4)] で，アルファベット 3 文字から構成され（例えば，日本語は jpn），言語の一意的識別子となる．一方，(複数の）別名は複数の**別名**[4)] を合成した文字列である．

1 つの言語に複数の名前が付けられていること（例えば,「日本語」を例にとれば，英語読みでは Japanese，日本語読みでは/nippon-go/と/nihon-go/，などの名前がある）がよくある．*Ethnologue* 第 15 版[4),5)] では，その研究・調査の結果がデータベースにまとめられ，公開されている．複数の名前の中の 1 つは第一言語名，その他は別名とされている．以降では，**言語名**は第一言語名または別名を指す．

第一言語名と別名の指定は学者独自に行われているため，同じ言語が違う言語名（第一言語名）になっているケースがよくある．我々は Yamamoto-Data と SilGIS-Data のそれぞれに含まれる言語の対応づけが必要であるが，両データの言語数がいずれも千単位にのぼるため，自動処理によって，なるべく多くの同一言語を発見することが我々の狙いである．

次節において，この処理における問題点を述べる．

2.2 言語名による言語の同一性判定の問題点

第一言語名には，(i) アルファベットからなる Japanese のような文字列（**語**とよぶ），(ii) 2 つ以上の語をカンマ (,)，空白 (Space) またはハイフン (-)（**区切り記号**とよぶ）でつないだ ‘‘Otomi, Estado de Mexico’’ のような語のリスト，という 2 つのケースがある．(複数の）別名は複数の別名をカンマでつないだ文字列になっており，カンマを検出すれば，複数の別名に分割できる．以下では，表 1 のサンプルデータを例に，言語名による言語の同一性判定の問題点について説明する．

(a) 第一言語名による判定 (A)*No*=1015 と (B)*No*=3295 は，第一言語名がそれぞれ JAPANESE と Japanese で，一致しているため，同一言語と判定できる．

(b) 第一言語名と別名による判定 (A)*No*=1855 と (B)*No*=5763 は，第一言語名がそれぞれ NHARON と Naro で，上記 (a) の方法では判定できないが，(B)（複数の）別名 ‘‘Nharo, Nharon, Nhauru, ···’’ の下線部分の語との一致が認められるため，別名による処理で判定が可能であろう．

(c) 第一言語名と別名による判定その 2 (A)*No*=1959 と (B)*No*=6262 は上記 (a) と (b) の方法では同一性判定ができないが，(A) 第一言語名 ‘‘OTOMI, STATE OF MEXICO’’ と (B)（複数の）別名 ‘‘Hnatho, Otomi del Estado de Mexico, ···, State of Mexico Otomi’’ を比較すると，下線部分の語リストが何らかの方法で一致が認められそうなため，こちらも別名による処理で判定が可能であろう．

上記 (a) 〜 (c) のケースからわかるように，第一言語名および別名は言語の同一性を判定する上で重要な情報であることがいえる．しかし，それだけでは情報不足で，**判定不能**のケースもあり，次の (d) と (e) に示す．

(d) 言語名の重複出現 (A)*No*=212 と *No*=213 はともに第一言語名が BAI で，(B)*No*=733 も第一言語名が Bai である．(A) では同一性判定の情報として第一言語名しか含まれていないため，(B)*No*=733 が同じ言語名をもつ (A) の *No*=212 と *No*=213 のどちらに対応しているかが，判定不能である．あるいはどちらにも対応していないことも否

定できない．

(e) 言語名の類似 (A)No=485 と (B)No=1565 は，第一言語名がそれぞれ ‘‘CHINANTECO, LALANA’’ と ‘‘Chinantec, Lalana’’ で，下線部分の CO と c は直感的に表記上の違いなどによる変化で，本来は同じ言語名なのではないか，との推測がつくと思われるが，第一言語名または別名が一致するかどうかによっては判定できない．一般的に，言語名に表記ゆれが含まれることが少なくない．表記ゆれは，例えば「バイオリン」と「ヴァイオリン」や,「サーバー」と「サーバ」など，多くのケースがあり，外来語の日本語表記で特に多く現れる．世界諸言語データの言語名はデータ編成者にとっていわばそのような「外来語」ならぬ外国語ばかりであるため，表記ゆれが含まれている可能性は大きい．

また，(d) のケースは第一言語名または別名以外のさらなる情報がなければ判定不能であるが，(a) 〜 (c)，または (e) のケースについても，2 つの言語が同一である可能性は高いが，これを別の角度からも示すことができれば，その同一性判定はより正確なものになる．

3. 言語系統木を用いた完全一致言語の検出

3.1 系統分類を考慮した言語の同一性判定

世界諸言語は系統分類されている．系統分類のもつ木構造の性質から，我々は言語の同一性判定処理に言語の系統分類も考慮することを提案する．そうすれば，**2.2(d)** で述べたような，異なる言語が同じ言語名になっているという判定不能の問題は解消される．

また，**2.2(e)** で述べたような言語名が類似しているケース（系統分類は一致）は，言語名の類似度という概念を導入すれば，同一性判定が可能になる．

一方，言語の系統分類は学者によって異なることがある．ゆえに，本来同じ言語であっても，各々の言語データでの系統分類は必ずしも一致しない．このような異なる学者の異なる知見による相違のほかに，系統分類の表現には言語名表記がともなうため，言語名のゆれによる相違も存在すると思われる．

このようなことを踏まえて，我々は言語名の類似度に加えて，系統分類の類似度についても定量化を行う．また，系統分類は言語名につぐ有益な情報とするのが妥当であると考える．つまり，言語名の一致または類似を確認した上で，さらに系統分類も一致または類似しているならば言語の同一性を肯定する，という 2 つの角度から評価を行う．これによって，同定できる言語の数および正確性を向上させる．

3.2 言語系統木

世界諸言語は多くの語族に分類され，1 つの語族は 1 つの系統樹を構成する[12]．語族は

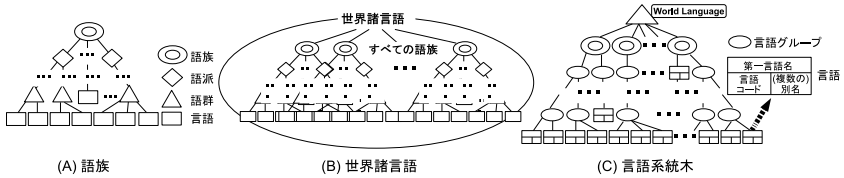

図 1 言語系統分類と言語系統木
Fig. 1 Language classification and World Language Tree

系統樹の最大の分類で，語派と語群は同じ語族の中での中分類と小分類で，最下位にあるのが言語である．図 1(A) は語族のイメージを示している．

本研究では，図 1(B) に示すように，語族の森を世界諸言語の下にまとめ，1 本の**木**として扱うことにする．さらに，(i) 語派と語群をまとめて言語グループとし，(ii) 言語名（言語グループ名を含む）は木構造のノードのラベルとして，また（複数の）別名などは最下位にあるリーフノード（言語）属性情報として，それぞれもたせることにする．その構造を**言語系統木**とよび，図 1(C) に示す．以下，木と言語系統木を定義する．

[定義 1] T をラベル付き順序木とする．T のルートを $r(T)$，T のノード集合を $V(T)$，辺集合を $E(T)$ で表す．ノード $x{\in}V(T)$ のラベルを $L(x)$ で表す．

(1) $x, z{\in}V(T)$，$(x,z){\in}E(T)$ ならば，x (z) は z (x) の**親** (**子**) という．同じ親をもつノードを**兄弟**とよび，子をもたないノードを**リーフ**とよぶ．T のリーフノード集合を $V_{leaf}(T)$ で表す．

(2) $x_0{=}r(T)$ から x までのパス $x_0x_1{\cdots}x_{k-1}x_k = x$ を $p(x_0,x)$ で表し，k を x の**レベル**とよぶ．特に，$p(x_0,x)$ の部分パス $x_1x_2{\cdots}x_{k-1}$ を $p(x)$ で表す．

(3) リーフでない兄弟 x, z に対し，$L(x){\preceq}L(z)$ なら，x を z の左に位置する． □

定義 1(3) の $\preceq$ は，$L(x)$ と $L(z)$ の順序関係を示しており，その定義はラベル関数が具体的に与えられたときに，示すことができる．本研究で用いる順序 $\preceq$ は，次の定義 2 で与えている．

[定義 2] 次の条件を満たす T を**言語系統木**とよぶ．

(1) ノード $x{\in}V_{leaf}(T)$ の**ノードラベル** $L(x)$ は $L(x){=}(\mathcal{L}_x, A_x, C_x)$ で表す．ただし，(i) $\mathcal{L}_x$ は集合 $\mathcal{L}_x{=}\{w_1, w_2, \cdots\}$ （$w_i{\in}\mathcal{L}_x$ は語）で，第一言語名を表す．(ii) C_x はアルファベット 3 文字からなる文字列で，言語コードを表す．(iii) A_x は集合 $A_x{=}\{\mathcal{A}_1^x, \mathcal{A}_2^x, \cdots\}$ で,（複数の）別名を表す．ここで，$\mathcal{A}_i^x = \{w_1, w_2, \cdots\}$ （$w_j{\in}\mathcal{A}_i^x$ は語）は別名を表す．

(2) ノード $x{\notin}V_{leaf}(T)$ の**ノードラベル**は $L(x) = \mathcal{L}_x = \{w_1, w_2, \cdots\}$（$w_i{\in}\mathcal{L}_x$ は (1)(i) と同様）で，言語グループ名を表す．また，ルート $r(T)$ のラベルは

$\mathcal{L}_{r(T)}$={World, Language} である．

(3) ノードラベルで表した $p(x)=x_1x_2\cdots x_{k-1}$ に対応するパスを $\mathcal{P}(x)=\mathcal{L}_{x_1}\mathcal{L}_{x_2}\cdots\mathcal{L}_{x_{k-1}}$ で表す．

(4) リーフでない兄弟 x, z のラベルを $\mathcal{L}_x=\{w_1^x, w_2^x, \cdots\}$ $(w_1^x \leq w_2^x \leq \cdots)$, $\mathcal{L}_z=\{w_1^z, w_2^z, \cdots\}$ $(w_1^z \leq w_2^z \leq \cdots)$ とする．$w_1^x=w_1^z$，$w_2^x=w_2^z, \cdots$，$w_{i-1}^x=w_{i-1}^z$，$w_i^x \leq w_i^z$ $(i \geq 1)$ ならば，$L(x) \preceq L(z)$ である．ここで，$w_i^x \leq w_i^z$ は w_i^x と w_i^z の辞書式順序を表す． □

3.3 2 つの言語系統木 T_Y と T_S

Yamamoto-Data と SilGIS-Data に関連する言語系統木は次の 2 つである．

(i) Yamamoto-Data のデータソースの文献にある「系統別語順分布表」は Yamamoto-Data の言語を系統分類の角度から整理した語順データである．このデータを言語系統木の定義にしたがって XML 形式に変換したデータを T_Y とする．言語数は 2,870 で，117 語族を構成している．

(ii) SilGIS-Data のデータソースである *Ethnologue* 第 15 版 Web サイトには世界諸言語の系統分類の情報も掲載されている．それを取得し，言語系統木の定義にしたがって XML 形式に変換したデータを T_S とする．言語数は 7,229 で, 108 語族を構成している．

T_Y と T_S を図 2 に示す．この 2 つの言語系統木の生成処理についての詳細は文献[11] を参照されたい．

定義 2 で述べたように，リーフノード $x \in (V_{leaf}(T_Y) \cup V_{leaf}(T_S))$ には言語コード C_x と (複数の) 別名 A_x の属性が付与されている．ただし，(i) T_Y のどのリーフノード $y \in V_{leaf}(T_Y)$ においても，C_y=Null，A_y=Null（Null は空値を表わす．以降も同様）．(ii) T_S のリーフノード $s \in V_{leaf}(T_S)$ については $C_s \neq$Null，$A_s \neq$Null または A_s=Null．つまり，T_S ではリーフノードの属性として，言語コードは必ず存在するが，(複数の) 別名は存在しない場合もある．

このように，T_S には言語を一意的に識別できる言語コードが付与されており，いわば基準となる言語系統木である．T_Y は何らかの処理で T_S の言語との対応関係を明らかにする必要がある言語系統木である．

本研究では，2 つの言語系統木 T_Y と T_S を対象に，言語 $y \in V_{leaf}(T_Y)$ に対し，T_S から y の同一言語である言語 $s \in V_{leaf}(T_S)$ を見つけ出すことを目的とする．

3.4 言語系統木を用いた完全一致言語の検出

T_Y と T_S のそれぞれに含まれる 2 つの言語に対し，系統分類が一致し，かつ言語名が一致するならば，この 2 つの言語は同一言語と判定してよい，と考える．言語系統木 T の言語はリーフノードにあたり，系統分類はパス，言語名はノードラベルなどで表せる．

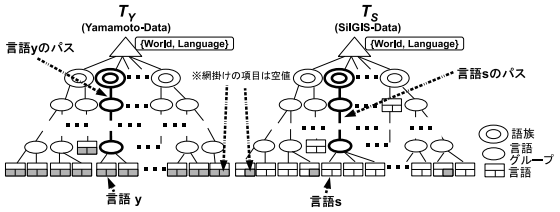

図 2 2 つの言語系統木 T_Y と T_S
Fig. 2 Two World Language Trees T_Y and T_S

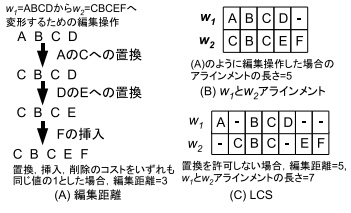

図 3 編集距離と LCS
Fig. 3 Edit distance and LCS

[定義 3] T_Y と T_S は 2 つの異なる言語系統木であり，y と s はそれぞれ T_Y と T_S のリーフ（$y \in V_{leaf}(T_Y)$，$s \in V_{leaf}(T_S)$）である．

(1) T_Y のパス $\mathcal{P}(y)=\mathcal{L}_{y_1}\mathcal{L}_{y_2}\cdots\mathcal{L}_{y_{k-1}}$ と T_S のパス $\mathcal{P}(s)=\mathcal{L}_{s_1}\mathcal{L}_{s_2}\cdots\mathcal{L}_{s_{m-1}}$ について，(i) $k=m$，(ii) $\mathcal{L}_{x_i}=\mathcal{L}_{y_i}$ $(i=1, 2, \cdots, \mathrm{k})$ が成立つとき y と s は**系統分類一致**といい，$\mathcal{P}(y)=\mathcal{P}(s)$ で表わす．

(2) $\mathcal{L}_y=\mathcal{L}_s$ または $\mathcal{L}_y \in \mathcal{A}_s$ が成立つとき，すなわち y と s のノードラベルが一致し，または y のノードラベルが s の（複数の）別名に含まれるとき，y と s は**言語名一致**という．

(3) y と s が系統分類一致で，かつ言語名一致であるならば，y と s を同一言語と判定し，(y, s) を**完全一致言語**とよぶ． □

$y \in V_{leaf}(T_Y)$ に対し，$\mathcal{P}(s)$ と系統分類一致の $\mathcal{P}(s)$ をもつ $s \in V_{leaf}(T_S)$ は複数存在しうる．そのため，完全一致言語の検出は (i) T_S において $\mathcal{P}(y)=\mathcal{P}(s)$ を満たすパス $\mathcal{P}(s)$ を検索し，(ii) (i) で得られた $\mathcal{P}(s)$ をもつ複数のリーフノードの中から，$\mathcal{L}_y=\mathcal{L}_s$ または $\mathcal{L}_y \in \mathcal{A}_s$ を満たす s を見つければよい．T_Y と T_S が順序木（定義 1(3)，定義 2(4)）であるため，(i) の処理を効率よく行うことが可能である．

この完全一致言語の検出処理では，T_Y と T_S のそれぞれにある言語が言語名または系統分類が一致ではない場合は検出されない．次節ではこの問題を解決する方法について述べる．

4. 類似度を用いたゆれのある言語の検出

本節では，言語名の類似度と言語系統分類の類似度の概念を導入し，それぞれについて定量化を行う．また，言語名と言語系統分類の比較はいれずも文字列の比較を基本とするため，以下では，まず編集距離に基づく文字列類似性の一般的な評価手法について説明し，次に文字列類似度に基づく言語名の類似度と系統分類の類似度の計算方法について述べる．

4.1 文字列の類似性評価

2つの文字列の類似性または非類似性（距離）を計る尺度として，編集距離 (Edit distance) がよく用いられる[7),8)]．ここでは，w_1=ABCD, 長さ n_1=4 と w_2=CBCEF, 長さ n_2=5 という2つの文字列を例にとり，説明していく．

(1) 編集距離

w_1 と w_2 の編集距離は，1文字の挿入，1文字の削除，または1文字から別の1文字への置換という3つの編集操作のもとで，w_1 から w_2 に変形するための最少の編集操作の回数として定義されている．$ed(w_1, w_2)$ で表す．挿入，削除，置換の3つの編集操作のコストをいずれも同じ1とした場合，図3(A) に示すように，編集距離 $ed(w_1, w_2)$ は3である．

編集距離の計算は，動的計画法に基づいている．編集距離は編集操作のコストを定めた前提での，**編集グラフ**[7)] とよばれるグリッドを通る最短距離となる．

一方，w_1 と w_2 を揃えた様子を図3(B) に示す．(ABCD−, CBCEF) のような2行の文字列を w_1 と w_2 の**アラインメント**[7)] という．アラインメントは編集グラフを走査することで得られる．走査ルートは複数通り可能なため，w_1 と w_2 のアラインメントは複数通り存在することがある．それらの長さ（以降，$l_A(w_1, w_2)$ で表す）はいずれも同じである．

アラインメントの2行の文字列について，同じ列の2つの文字が一致する個数 $ss(w_1, w_2)$ はアラインメントの長さ $l_A(w_1, w_2)$ から編集距離 $ed(w_1, w_2)$ を引いた値になる．つまり，$ss(w_1, w_2)$ は次の式を満たす．

$$ss(w_1, w_2) = l_A(w_1, w_2) - ed(w_1, w_2) \tag{1}$$

(2) LCS

編集距離を求めるには編集操作のコストを定めることが前提となっているが，置換を排除し，挿入と削除の2つの編集操作のみを許し，それらのコストをいずれも1とするならば，w_1 と w_2 のアラインメントは図3(C) のようになり，編集距離 $ed(w_1, w_2)$ は5になる．

このような置換を考慮しない編集操作の場合は，w_1 と w_2 の2つの文字列に含まれる最長共通部分列 **LCS**(The Longest Common Subsequence)[7)] を求めることができる．ここでは下線部分の BC である．もっとも，部分列は連続している必要はない．w_1 と w_2 の LCS の長さを $l_{LCS}(w_1, w_2)$ で表す．

4.2 言語名の類似度

言語名は1つ以上の語からなる集合として定義されている．言語名間の類似度の計算は次の2つのステップに分けて考える．(i) $\mathcal{L}_1^{T_Y}$ に含まれる語と $\mathcal{L}_1^{T_S}$ に含まれる語との間の類似度を計算し，(ii) (i) で計算された語類似度に基づき，言語名の類似度を計算する．

(1) 語類似度

言語名に含まれる表記ゆれには (i) $\mathcal{L}_1^{T_Y}$={CHINANTECO, LALANA} と $\mathcal{L}_1^{T_S}$={Chinantec, Lalana}，(ii) $\mathcal{L}_2^{T_Y}$={CHINESE, MEI PEI} と $\mathcal{L}_2^{T_S}$={Chinese, Mei Bei}，のようなケースがある．(i) の違い（下線部分）は文字の挿入または削除によるものであり，(ii) の違いは文字の置換によるものである．

この両者の表記ゆれによる言語名の変化の度合いは同等である．すなわち，CHINANTECO と Chinantec および PEI と Bei の編集距離はどちらも同じ値の1と考えるのが妥当だ，ということである．v と w をそれぞれ2つの語とし，v と w の語類似度を次のように定義する．

[定義 4] 次の式を満たす $sd\_w(v, w)$ は2つの語 v と w の語類似度である．

$$sd\_w(v, w) = \frac{l_A(v, w) - ed(v, w)}{l_A(v, w)} \tag{2}$$

ここで，$ed(v, w)$ と $l_A(v, w)$ はそれぞれ置換，挿入，削除の3つの編集操作を許可し，コストをいずれも1としたときの編集距離とアラインメントの長さである． □

(2) 言語名の類似度

$\mathcal{L}_1^{T_Y}$={CHINANTECO, LALANA}，$\mathcal{L}_1^{T_S}$={Chinantec, Lalana} を例にとり，説明していく．$\mathcal{L}_1^{T_Y}$ には2つの語，$\mathcal{L}_1^{T_S}$ にも2つの語が含まれている．$\mathcal{L}_1^{T_Y}$ の1つ目の語 CHINANTECO に対しては，(CHINANTECO, Chinantec), (CHINANTECO, Lalana) の2通り，$\mathcal{L}_1^{T_Y}$ の2つ目の語 LALANA に対しては，(LALANA, Chinantec), (LALANA, Lalana) の2通り，の計4通りの組合せがある．式 (2) にしたがって，前者の2通りの組合せの語類似度を計算すると，0.88 と 0.2 が得られる．この中で，(CHINANTECO, Chinantec) の語の組合せの語類似度が最大となる．このような組合せを**語ペア**とよぶことにする．

すべての語ペアを求めるには，次の操作を行えばよい．(i) すべての組合せの語類似度を計算し，最大語類似度をもつ語の組合せを見つけ，語ペアとする．(ii) 語ペアに含まれる語を含む組合せを削除する．(iii) 残りの組合せの中から，最大語類似度をもつ語の組合せを見つけ，語ペアとする．(iv) 残りの組合せがなくなるまで，(ii) と (iii) を繰り返す．

$\mathcal{L}_1^{T_Y}$ と $\mathcal{L}_1^{T_S}$ の語ペアは全部で2つで，(LALANA, Lalana) と (CHINANTECO, Chinantec) が得られ，それぞれの語ペアの類似度が1と 0.88 である．

言語名 $\mathcal{L}_1$ と $\mathcal{L}_2$ の類似度を $sd\_ln(\mathcal{L}_1, \mathcal{L}_2)$ で表し，次のように定義する．

[定義 5] $\mathcal{L}_1$={$v_1, v_2, \cdots, v_m$} と $\mathcal{L}_2$={$w_1, w_2, \cdots, w_n$} ($m \geq n$) は言語名であり，$v_i \in \mathcal{L}_1$ に対応する語ペアは (v_i, w_i') である．ただし，$w_i' \in \mathcal{L}_2$ で，v_i の語ペアが存在しない場合は

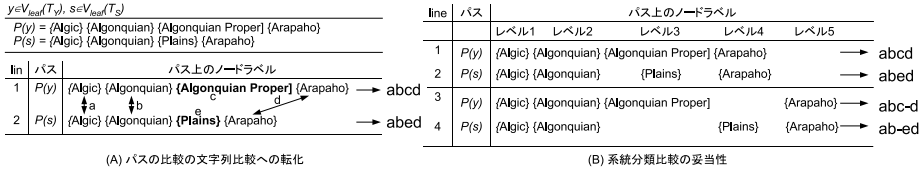

図 4 系統分類の比較
Fig. 4 Comparison of language classification

w'_i=NULL である．次の式を満たす $sd\_ln(\mathcal{L}_1, \mathcal{L}_2)$ は言語名 $\mathcal{L}_1$ と $\mathcal{L}_2$ の類似度である．

$$sd\_ln(\mathcal{L}_1, \mathcal{L}_2) = \frac{\sum_{i=1}^{m} sd\_w(v_i, w'_i)}{m} \quad (3)$$

□

4.3 言語系統分類の類似度

言語の系統分類は言語系統木におけるパスで表すことができる．以下では，パスの比較を文字列の比較に転化させ，文字列類似度に基づく系統分類の類似度について述べる．

(1) 言語系統分類の比較

言語 $y \in V_{leaf}(T_Y)$ と $s \in V_{leaf}(T_S)$ のパスをそれぞれ $\mathcal{P}(y)$ と $\mathcal{P}(s)$ とする．パスは $\mathcal{P}(y)=\mathcal{L}_{y_1}\mathcal{L}_{y_2}\cdots\mathcal{L}_{y_{k-1}}$ のように，言語系統木のルートの子からリーフノードである言語の親までのノードラベルのリストとして定めている（**3.2** 参照）．

言語 y は ‘‘Algic’’ という語族（レベル 1）の言語で，レベル 2, レベル 3, レベル 4 の言語グループ名はそれぞれ ‘‘Algonquian’’, ‘‘Algonquian Proper’’, ‘‘Arapaho’’ である．図 4 に示すように，言語 y のパス $\mathcal{P}(y)$={Algic}{Algonquian}{Algonquian, Proper}{Arapaho}. 言語 s のパスは $\mathcal{P}(s)$={Algic}{Algonquian}{Plains}{Arapaho}.

$\mathcal{P}(y)$ と $\mathcal{P}(s)$ に含まれる異なるノードラベルをそれぞれ異なる 1 文字に変換して表せば，$\mathcal{P}(y)$ と $\mathcal{P}(s)$ の比較を文字列の比較に転化させることができる．ノードラベルの文字への変換方法としては，図 4(A)line 1 と line 2 に示すように，2 つのパスに含まれる同じノードラベルには同じ文字を割り当てればよい．例えば，$\mathcal{P}(y)$ の任意の 1 つのノードラベルに対して，$\mathcal{P}(s)$ のノードラベルとの間の類似度を計算し，その類似度が閾値 α を超えるノードラベルが見つかったならば，それらのノードラベルには同じ文字を割り当てる．ノードラベル間の類似度の計算は **4.2(2)** で説明した言語名の類似度の計算法を用いる．図 4(A) の例では，ABCD と ABED という 2 つの文字列が得られる．この変換処理を $\mathcal{F}(\mathcal{P}(y), \mathcal{P}(s))$→(ABCD, ABED) で表す．次に，変換後得られた 2 つの文字列の類似度を計算するが，ここでは **4.2(1)** で説明した語順類似度の計算法とは異なる．

図 4(B) の line 1 と line 2 は，$\mathcal{P}(y)$ と $\mathcal{P}(s)$ がノードラベルが一致するノード {Algic}, {Algonquian}, {Arapaho} に合わせて，揃えられていることを示している．つまり，レベル 1，レベル 2，レベル 4 のノードラベルは一致で，レベル 3 の {Algonquian, Proper} と {Plains} はノードラベルの不一致，言い換えれば $\mathcal{P}(y)$ の {Algonquian, Proper} から $\mathcal{P}(s)$ の {Plains} への置換があった，とみている．それに対し，図 4(B) の line 3 と line 4 は，レベル 3 は {Algonquian, Proper} と {Plains} の不一致（置換）ではなく，$\mathcal{P}(y)$ は，{Algonquian, Proper} の削除および {Plains} の挿入という 2 つの編集操作で $\mathcal{P}(s)$ になった，とみている．我々は，系統分類の比較は図 4(B) の line 1 と line 2 に示している方が妥当だと考える．

一方，2 本のパスを 2 つの文字列に変換した後は，その 2 つの文字列の編集距離を求めるが，その求め方としては，両パス中において 1 つでも同じノードラベルがあれば，その一致を見逃してはいけないことから，**4.1(2)** で説明したように，LCS を求めるべきである．LCS を求めるためには，置換は考慮しないため，図 4(B) の line 3 と line 4 に示すようなアラインメントになる．ここで矛盾が生じるが，その解決法を次において述べる．

(2) 系統分類の類似度

言語 y と s のパス $\mathcal{P}(y)$ と $\mathcal{P}(s)$ に対し，$\mathcal{F}(\mathcal{P}(y), \mathcal{P}(s))$→$(v, w)$ の変換処理を行い，それぞれ文字列 v と w に変換する．v と w によって y と s の系統分類の類似度を求める．上記の例 v=ABCD, w=ABED を用いて，系統分類の類似度の求め方について述べる．

(i) v=ABCD, w=ABED に対し，挿入と削除のみを許すように（コストはいずれも 1 とする），動的計画法にしたがって，v から w に変形する．アラインメント (v'=ABC−D, w'=AB−ED) が得られる．前述のように，アラインメントは複数通り可能である．ここでは v から w への変形過程において，文字の不一致が現れたら，v の文字 (C) の削除と w の文字 (E) の挿入という順に操作するとする．

(ii) v' と w' の 2 行の文字列を 1 行の文字列に変換する．アラインメント (ABC−D, AB−ED) の同じ列の 2 つの文字につき，文字が一致している場合は $*$, v' の文字がギャップ（−）である場合は −, w' の文字がギャップ（−）である場合は +，の記号にそれぞれ置き換える．この例では，$**+-*$ になる．

(iii) (ii) で得られた文字列 $**+-*$ では置換は考慮されていない．$**\underline{+-}*$ に対し，置換を許すように，下線部分の $\underline{+-}$ を X に変換し（X はアラインメントの 2 つの文字の不一致を意味する），新たな文字列 $**X*$ を得る．この再構成後の文字列 $**X*$ を**言語系統分類の類似特性記号列** (Similarity feature string of language classification) とよび，

$SFSLC(v,w)$ で表す．また，$SFSLC(v,w)$ の長さを $l_{SFSLC}(v,w)$ で表す．

$SFSLC$ は．$*,+,-,X$ という 4 つの記号を使って，任意の $y{\in}leaf(T_Y)$ に対し，T_S での系統分類を基準にした T_Y での系統分類の変化を表現するための記号列だとみることができる．また，$SFSLC$ という 1 つの文字列で表すことによって，T_Y と T_S の 2 つの言語系統木での系統分類の相違を視覚的に捉えることもできる．

言語 $y{\in}V_{leaf}(T_Y)$ と $s{\in}V_{leaf}(T_S)$ の系統分類の類似度を次のように定義する．

[定義 6]　次の式を満たす $sd\_lc(y,s)$ は言語 $y{\in}V_{leaf}(T_Y)$ と $s{\in}V_{leaf}(T_S)$ の系統分類の類似度である．

$$sd\_lc(y,s)=\frac{l_{LCS}(v,w)}{l_{SFSLC}(v,w)} \tag{4}$$

ただし，v と w は $\mathcal{F}(\mathcal{P}(y),\mathcal{P}(s))\rightarrow(v,w)$ によって，言語 y と言語 s のそれぞれの系統分類を表すパス $\mathcal{P}(y)$ と $\mathcal{P}(s)$ から変換された文字列である．　□

4.4　ゆれのある同一言語の検出

$y{\in}V_{leaf}(T_Y)$ と $s{\in}V_{leaf}(T_S)$ の 2 つの言語について，言語名が一致していなくても，一定の条件を満たす類似関係をもっていて，かつその上系統分類も一定の条件を満たす類似関係をもっているならば，y と s は同一言語と判定する．以下では，完全一致言語の検出処理では検出できなかった言語に対する検出法について述べる．

任意の $y{\in}V_{leaf}(T_Y)$ に対して，T_S での同一言語の検出処理は次のように行う．

(1) まず，任意の $y{\in}V_{leaf}(T_Y)$ に対し，T_S のすべての言語から，言語名の類似度が最大となる言語を検索する．ここで得られる言語名の類似度の最大値を $sd\_ln_{max}(\mathcal{L}_y)$ で表す．なお，y と $s{\in}V_{leaf}(T_S)$ の言語名の類似度の比較処理は，y の第一言語名 $\mathcal{L}_y$ と s の第一言語名 $\mathcal{L}_s$ に対し，また y の第一言語名 $\mathcal{L}_y$ と s の（複数の）別名 $A_s{=}\{\mathcal{A}_1^s,\mathcal{A}_2^s,\cdots\}$ の中の各々の別名に対しても行う．$sd\_ln_{max}(\mathcal{L}_y){\leq}\alpha$ ならば，y と同一の言語は T_S には存在していないことになる．そうでないならば，次は系統分類の類似度によって判定を行う．

(2) y に対し，$sd\_ln_{max}(\mathcal{L}_y)$ をもつ T_S の言語は複数になる可能性がある．次に，これらの複数の言語 $s_1,s_2,\cdots$ の各々に対し，y との系統分類の類似度 $sd\_lc(y,s_1)$, $sd\_lc(y,s_2)$, $\cdots$ を計算し，その中から系統分類の類似度が最大となる言語を検索する．ここで得られる系統分類の類似度の最大値を $sd\_lc_{max}(y)$ で表す．なお，系統分類の類似度の算出結果は閾値 α の値に関連していることに注意されたい（**4.3(1)** 参照）．

(3) 最後の判定として，(i) $sd\_lc_{max}(y){>}\beta$, (ii) $sd\_lc_{max}(y){>}\beta$ を満たす言語 $s{\in}V_{leaf}(T_S)$ が唯一であること，という 2 つの条件を満たすならば，(y,s) は同一言語と判定する．

なお，α，β の値の決定については，5 節で議論する．

5.　手法の実装および処理結果

本節では，**3.** ～ **4.** で述べた手法の実装を行い，Yamamoto-Data と SilGIS-Data に対する言語の同一性判定処理およびその結果について述べる．

5.1　処理全体の流れ

Yamamoto-Data と SilGIS-Data のそれぞれに含まれる同一言語の検出処理は，T_Y と T_S の 2 つの言語系統木を処理データとし，次の処理 I と処理 II の 2 つに分けて行う．

[処理 I]（完全一致言語の検出）

Step1 効率よく探索を行うために，T_Y と T_S に対し根から左優先の深さ優先探索を行い，$\mathcal{P}(y){=}\mathcal{P}(s)$ を満たすパス対 $(\mathcal{P}(y),\mathcal{P}(s))$ を見つける．

Step2 1 つのパス対 $(\mathcal{P}(y),\mathcal{P}(s))$ に対し，複数の言語対 $\{(y,s)\}$ が存在しうる（$\mathcal{P}(y)$ と $\mathcal{P}(s)$ に複数の言語がぶら下がっている）．これらの (y,s) に対し，$\mathcal{L}_y{=}\mathcal{L}_s$ または $\mathcal{L}_y{\in}\mathcal{A}_s$ を満たした言語 y と s を完全一致言語として出力する．

Step3 以降の処理が効率よく行われるように，完全一致言語 y と s のノードまでの部分パス（子が 1 つだけのノードからなる y と s までの $\mathcal{P}(y)$ と $\mathcal{P}(s)$ の部分パス）を削除し，T_Y と T_S を更新する．

[処理 II]（ゆれのある同一言語の検出）

更新された T_Y と T_S において，すべての $y{\in}V_{leaf}(T_Y)$ に対し，次の処理を行う．

Step1 任意の $y{\in}V_{leaf}(T_Y)$ に対し，**4.4** で述べたゆれのある同一言語の検出処理を実行し，得られた SLP を同一言語ペアとして出力する．

Step2 同一言語ペア (y,s) について，処理 I Step3 と同様に，T_Y と T_S を更新する．

5.2　処理結果および考察

処理結果を表 2 に示す．処理 I と処理 II を合わせると，Yamamoto-Data の総言語数 2,870 中の 2,522 言語（約 88% ）につき，SilGIS-Data の言語との対応づけが判明した．また，検出された 2,522 言語のうち，処理 I で得られた完全一致言語が 1,034（約 36% ），処理 II で得られたゆれのある言語が 1,488（約 52% ）であった．

木構造を取り入れたことにより，**2.2** で述べた (d) のケースの問題は解消され，表 1(B) の Bai 言語は表 1(A) の No=212 の BAI に対応していることが判明した．

処理 II で用いた閾値は α=0.75 と β=0 とした．その理由は次の通りである．同一言語の検出が漏れることがあっても，異なる言語を同一言語と判定することは避けたいという方針のも

表 2 処理結果
Table 2 処理結果

処理	判定できた言語数	比率（2,870に対する比率）
処理 I	1,034	36%
処理 II	1,488	52%
合計	2,522	88%

とで，一部のデータを対象に，α=0.65, 0.70, 0.75, 0.80, 0.85, β=0.10, 0.15, 0.20, 0.25, 0.30 のときのシミュレーションをした結果，α=0.75 と β=0 と設定したときが最もよい結果が得られたためである．

検出された同一言語ペアの正当性について確認作業をした結果，異なる言語を同一言語と判定した例はなかった．それに対し，算出された言語名の類似度 $\leq\alpha$(=0.75) となったため，同一言語として判定されなかった言語が 10 言語以上あった．いずれも系統分類が完全一致の言語である．判定漏れ（本論文で提案した手法を用いて本来検出されるべき同一言語ペアが，実際結果として検出されなかったことを指す）には，次の主なケースが挙げられる．

(i) URARTIAN と Urarina のような言語名が 1 つの語の場合である．類似度が 0.67 で，検出されなかった．

(ii) ‘‘CHONTAL OF OAXACA, HIGHLAND’’ と ‘‘Chontal, Highland Oaxaca’’ のような言語名に助詞が入っている場合である．類似度が 0.75 である．ほかには，類似度が 0.67 の ‘‘CHONTAL OF TABASCO’’ と ‘‘Chontal, Tabasco’’ のような場合も判定漏れになった．下線部分の OF のような助詞が入っている言語名は多数で，検出された言語も少なくなかった．例えば，‘‘MAZATECO, SAN JUAN CHIQUIHUITLA’’ と ‘‘Mazateco de San Juan Chiquihuitlan’’ についても，下線部分の de は助詞だと思われるが，こちらの場合は言語名を構成する語の数が多いため，類似度が 0.78 になり，検出漏れとはならなかった．この問題に対し，助詞のリストを作成し，あらかじめ言語名から助詞を削除する方法も考えられるが，そもそも両言語データとも多種の言語の文献を参考にし作製されたことがあるため，助詞をリストアップすること自体，筆者らにとって困難性が高いことであると予想し，あえて例外処理を行わないことにした．

β については，0.10 以上の値に設定すると判定漏れがあったため，β=0 にしたことは効果があったといえる．

また，Yamamoto-Data の約 88% の言語が SilGIS-Data の言語との同一性が判明できたことは，それらの言語に関連する空間データの利用が可能になることにもつながる．すなわち，GIS を用いた語順研究を展開するために重要な役割を果たすことができる．

6. おわりに

本研究では，我々は言語名と系統分類の類似度を導入し，木構造をなす言語系統木に加え，類似度を考慮した言語の同一性判定の手法を提案した．その結果，合わせて 88% の言語の同一性が判明できた．また，GIS を用いた語順研究の展開を可能にならしめることは重要な意義をもつ．このことからも，本研究は有用であることを示す．

今後は，(1) OF などの助詞に対し，オントロジーアラインメントなどで広く用いられている外部知識源に基づく語の類似度の評価手法の取り入れなどの手法について検討し，(2) 今回の処理で同一性が判明できなかった原因をさらに調査し，ゆれのある言語の検出率の向上を図るなど，本手法をさらに発展させていきたい．

参考文献

1) 池田潤，GIS と言語研究, 一般言語学論叢, 第 9 号, pp.1-10 (2006).
2) 呉靭, 山本秀樹，乾秀行, 杉井学, 松野浩嗣：語順地図作成に必要なデータ及び語順地図に現れる語順分布，一般言語学論叢, 第 10 号, pp.31-49 (2007).
3) 呉靭，乾秀行, 杉井学, 松野浩嗣：言語研究のための GIS データの生成について-Ethnologue GIS データを言語特徴の地図化に用いる一手法，人文科学とコンピュータシンポジュウム論文集, pp.253-258 (2007).
4) Gordon, R.G. (ed.): *Ethnologue: Languages of the World, 15th ed.*, Dallas, SIL International, Texas (2005).
5) http://www.ethnologue.com/web.asp
6) 山本秀樹：世界諸言語の地理的・系統的語順分布とその変遷，渓水社，広島 (2003).
7) Neil C. Jones, Pavel A. Pevzner 著，渋谷哲朗ほか訳，バイオインフォマティクスのためのアルゴリズム入門，共立出版，東京 (2007).
8) Gonzalo Navarro: A guided tour to approximate string matching, ACM Computing Surveys (CSUR), vol.33，no.1, pp.31-88 (2001).
9) 市瀬龍太郎：情報の意味的な統合とオントロジー写像，人工知能学会誌. Vol.22，No.6, pp.818-825 (2007).
10) 高橋良平，小山聡，田中克己：恣意的に名前付けされたオブジェクトの識別手法，日本データベース学会論文誌，Vol. 8, No. 1, pp.5-10 (2009).
11) 呉靭, 乾秀行, 杉井学, 松野浩嗣：*Ethnologue15th* 言語属性データと言語系統データの生成および言語同定における利用，コンピュータ&エデュケーション, vol.25, pp.70-73 (2008).
12) 呉靭，冨永理恵，乾秀行, 杉井学, 松野浩嗣：オープンソース可視化ツールを用いた言語系統樹の図式表現，人文科学とコンピュータシンポジュウム論文集, pp.333-340 (2008).