博　士　論　文

言語系統木と文字列類似度に基づく言語同一性判定に関する研究

（A Study on the Identification of the World's Languages Based on Languages Tree and String Similarities）

2013年8月

呉 靭（Ren WU）

山口大学大学院理工学研究科

学位論文内容要旨

学位論文題目：言語系統木と文字列類似度に基づく言語同一性判定に関する研究
A Study on the Identification of the World's Languages
Based on Languages Tree and String Similarities

指導教員　　：松野　浩嗣　教授

申請者名　　：呉　靭

山口大学大学院理工学研究科
自然科学基盤系専攻（数理複雑系科学領域）

世界には数千種類の言語がある．世界の諸言語はそれぞれ違い，多様性に富んでいる．言語学者は，世界の諸言語にはどのような多様性が見られ，またその多様性の中にどのような普遍性が潜んでいるのかについて古くから探求してきており，言語類型論という学問体系ができている．近年はIT（情報技術）の著しい発展により，文理融合の新たな手法によってさらなる発見をもたらすことが期待されている．

ITを応用した言語類型論的研究では，特に言語特徴に関する情報が含まれている言語データが欠かせない．また，効果的に研究を展開するためには，しばしば他の言語学者が収集した言語資料をデータ化し，研究に使うことがある．その際，複数の異なる言語学者による言語データを一つにし，新たな言語データを生成してから使うこともよく行われる．

言語学者による言語データ（ここでは2つの表形式のデータを想定する）では，普通，言語名によって言語を識別している．しかし，1つの言語には，複数の名前が付いている場合がよくある．また，言語の別名の存在や表記ゆれなどが含まれているため，言語の名前だけでは言語を識別できないケースが多い．つまり，世界諸言語に関するデータでは言語の一意識別子が含まれていないことがあり，このことが，言語データをマッチングする際に問題となる．

言語の一意識別子として，国際標準化機構による言語コード（ISO639 シリーズ）がある．この言語コードの標準化は 1980 年代から始まったが，以来頻繁にコード体系の変更が発生しており，コード体系設計上一般的に要求される安定性や恒久性が具備されているとはいえない．そのためか，言語コードは未だに，言語学者の間でコンセンサスが得られ，確立されたコード体系が共有され，標準として使われる段階に至っているとはいえない．

言語コードが付与されていない価値ある言語資料は数多く存在する．言語同一性の問題が障害となり，言語研究に活かせないのならば，それは大変残念なことである．人類最大の文化遺産ともいえる言語に関する資料を研究に活かせるようにすることは，重大な意義を持つ．一方，世界諸言語に関する言語データは言語数が千単位にのぼるため，手作業によって言語を特定するのは，莫大な作業量を要するうえ，専門知識も必要とするため，大変困難なことである．そこで，本研究では異なるデータ中の言語同一性をコンピュータ自動処理によって判定することに取り組む．特に，2 つの異なる学者による表形式の言語データの一方に言語コードが付けられていない場合における言語同一性の問題に焦点を当て，解決を図る．

本研究が目指す問題解決は，著者の知る限りにおいて，今まで研究が行われていない．本研究では，アプローチとして言語系統木という概念を導入する（言語系統木では，言語は最下位のレベルに位置するリーフノードとなる）．これは，言語名だけでは言語の同一性を判定するための情報が足りないため，別の角度からの情報として，言語系統分類を取り入れるためである．言語系統分類に関するモデルはいくつか提唱されてきたが，そのなかの 1 つとして，系統樹モデルがある．系統樹モデルは同じ語族に属する言語は，はるか過去に話されていた 1 つの言語から分かれて発展してきたと主張し，言語の分化の過程を一本の樹となる系統樹にたとえている．1 つの系統樹は 1 つの語族に含まれる言語から構成され，言語と言語の間の親族関係を表している．本研究では，系統樹モデルに基づき，世界諸言語のデータ構造を系統樹の森となる言語系統木として定義する．言語系統木の導入により，2 つの表形式の言語データに含まれる言語同一性判定の問題は 2 つの木構造のリーフノードの間のマッチング問題として転化する．また，言語系統分類も言語名と同様に，曖昧な性質をもつため，本研究では言語名の類似度と言語系統分類の類似度という概念を提案し，言語類似性の定量化を試みる．

木構造上でのデータマッチングに関する研究は広く行われている．研究対象の概念を明示的に表現し，それらの関係を体系的に記述したオントロジーを構築し，異

なるオントロジー間の対応関係を見つけ出すオントロジー・マッピングや，木編集距離などを利用した木構造パターン・マッチングなど数多くの手法が提案されている．しかし，言語学における言語系統分類の学問分野自身がまだ確立された体系を樹立するまでに至っていないため，オントロジー構築が困難である．さらに，本研究では２つの言語系統木に含まれる言語（リーフノード）のマッチングのみに限定しており，木構造全体のマッチングまでは考慮する必要がないことから，それらの手法は本研究に必ずしも適しているとはいえない．また，木構造上でのデータマッチングに関するテーマではないが，近年ソーシャルネットワークに関連して，人の名前を特定する人名マッチングの研究が盛んである．人名は，本研究が扱う言語名に類似している．オントロジー・マッピング，木構造パターン・マッチングおよび人名マッチングなどに共通して用いられている基本手法がある．それは，文字列類似度に基づく手法である．

文字列類似度にも多くの手法がある．本研究では編集距離を基本とし，文字列類似度計算構造化手法 Monge-Elkan 法を言語名の類似度計算に取り入れる．また，言語系統分類の類似度についても定量化を行い，同一言語ペアの検出法についても提案を行う．さらに，実験を行い，その手法の有効性を示す．本論文は以下のように構成される．

第１章では，言語学的な背景について述べたうえで，本論文の目的および構成を示す．

第２章では，言語データの例を示し，言語同一性判定の問題点を分析する．

第３章では，準備として，系統樹モデルおよびオントロジー・マッピングなどの関連研究について述べる．また，文字列類似度の指標である編集距離と最長共通部分列および文字列類似度計算構造化手法 Monge-Elkan 法などについて述べる．

第４章では，言語系統木について定義したのち，XML を用いた言語系統木データとその構築について述べる．

第５章と第６章では，言語同一性判定の手法として２つの手法（手法 I と手法 II）を提案する．第５章では，手法 I として，まず木構造に基づき，言語名と言語系統分類についてゆれのない完全一致言語の検出法について述べる．次に，言語名や言語系統分類についてゆれのある言語にも対応した木構造と文字列類似度に基づく手法を提案する．実験の方法と結果を提示し，考察を与える．

第６章では，まず手法 I の問題点を指摘する．それは，２つの言語系統木のそれぞれに含まれる２つの言語の言語名，または言語系統分類のどちらか一方が完全に一

致しているにもかかわらず，そのことがその２つの言語の同一性判定にまったく考慮されていない，ということである．その問題点をカバーできる言語名と言語系統分類の総合的尺度に基づく手法を提案する．さらに実験の方法と結果を提示し，考察を与える．

第７章では，言語同一性判定問題に関する今後の課題について述べたのち，本論文をまとめる．

Abstract

Title of Thesis: A Study on the Identification of the World's Languages
Based on Languages Tree and String Similarities

Name of degree candidate: Ren WU

Degree and Year: Ph.D. in Science, 2013

Thesis directed by: Dr. Hiroshi Matsuno
Professor
Graduate School of Science and Engineering
Yamaguchi University, Japan

Thousands of language are spoken in the world. There are many differences in these languages, while they share common features such as vocabularies and word orders. Linguistic typology study, which systematically deals with the language universalities, and the implication and correlation among the linguistic features, has been established. Recent advances of information technology (IT) are expected to bring new discoveries in the linguistic typology along with the progress of an integrated study of literature and science.

Language data are essential in the study of linguistic typology applying IT, especially such data that linguistic features are included in. To conduct these studies, language data based on these materials collected by the linguists are often used. In such a case, language data provided from different researchers are integrated into one in order to perform the study more effectively.

Language researchers usually identify languages by these names. However, it is commonly found that two or more names are assigned to one language. Moreover, in many cases, alternate names and/or different writings make it difficult to discriminate languages only by the language name. Namely, no inclusion of the unique identifier of languages may cause a problem in the matching operation among different language data. The standardization of the language code ISO639 series began in the 1980s. However, since changes of the code system have occurred frequently, the

stability and permanency, which are generally demanded on a code system design possess, are not realized yet. Hence, we cannot say that the language code ISO639 has built consensus among linguists, being shared as a standard code. Not a few valuable language data do not have these language codes. It is regrettable if such valuable language data are unavailable for the linguistic research due to such a code problem in identifying languages. Language is one of the most important human cultural heritages. It is very much meaningful to make all language data available for linguistic studies.

On the other hand, it is quite hard to carry out that matching operations in several thousands of language data manually. In order to resolve this problem, by automated matching of language names, we developed a new method to integrate two different sets of language data, the one has the language code, the other not.

As far as we know, such a work has not be carried out. To do this, we take an approach to introduce a concept of languages tree in which languages are assigned at the lowest level as leaf nodes. Introduction of the languages tree allows us to enhance the information for identifying languages with the incorporation of language classification.

Several models for language classifications have been proposed, including ‘‘family tree’’. The family tree models the process of development of languages as a structure of tree, where a node and an edge of the family tree represent a language and a development between two languages, respectively. The family tree reflects the concept that languages in the same family tree are developed from the same source language, which was spoken long time ago, represented by the root of the family tree. Hence, one family tree corresponds to one language family.

In this study, the structure of the world's languages is defined as ‘‘Languages Tree’’ which is a forest of family trees. By the introduction of Languages Tree, the problem of language matching between two data can be dealt with a matching problem between leaf nodes of the Languages Tree. An ambiguity is a problem which resides in language classifications as well as language names. To cope with this problem, the concepts of the language classification similarity and the language name similarity, are proposed.

Data matchings based on a tree structure which employ the same technique of

string similarity have been widely studied. Ontology mapping is technology that find out the corresponding relation between different ontology. In this technology, the ontology need to be built systematically to describe the relations among the concepts of the aimed research. However, this technique is difficult to apply to this study because systematic classification of languages has not been established yet. In addition, this technique is not suitable for our study because, in this study, matching processes are restricted to language (leaf-node) matching, not for entire language family (a tree).

On the other hand, the studies on personal name matching are quite active in association with the social network, although it is not in the category of tree data matching. Personal names are similar to language names dealt with in this study. The technique commonly used in ontology mapping, the tree pattern matching and the personal name-matching is string similarity.

There are many methods in string similarity. In this study, we use Monge-Elkan method which is a general text string comparison method base on an internal character-based similarity measure. At the same time, we define a measure for language classification to propose a method identifying the same language. The experimental results are shown to prove the efficiency of our method.

This paper is organized as follows.

In Chapter 1, after describing linguistic background, the purpose and organization of this paper are presented.

In Chapter 2, two sets of language data are shown as samples and the problem in identifying language is discussed.

In Chapter 3, after illustrating the concept of the family tree model, related researches including ontology mapping, personal name-matching, etc. are described.

In Chapter 4, after defining Languages Tree, languages tree data using XML and its construction are explained.

In Chapters 5 and 6, two methods named Method I and Method II are proposed as novel methods for the identification of the world's languages. At first, the method of detecting the full match without ambiguities in language names and language classifications is proposed based on the tree structure. Then, Method I is proposed based on tree structure and string similarity, which can absorb the ambiguities con-

tained within language names and language classifications. After this, experimental results are shown together with methodologies.

The problem of the Method I are, in the case that either language names or language classifications of two Languages are the same, this information is not taken into consideration in the identification process at all. Chapter 6 resolves this problem by a technique named Method II, in which the general similarities, a total measure of language name and language classification, is defined. After this, experimental results with its methodologies are presented.

In Chapter 7, the future works on the identification of the world's languages, and the concluding remarks are given.

目 次

図 目 次

表 目 次

記 号

$x \in X$: x は集合 X の要素である．

$x \notin X$: x は集合 X の要素ではない．

$X \subset Y$: 集合 X は集合 Y の部分集合である．

$X \cup Y$: 集合 X と Y の和集合である．

$X \cap Y$: 集合 X と Y の積集合である．

$X - Y$: 集合 X と Y の差集合である．

$X \leftarrow Y$: 集合 Y の要素を集合 X の要素とする．

$\sum x_i$: 全ての要素 x_i の和である．

ϕ : 空集合である．

$|X|$: 集合 X の要素数である．

$\max X$: 集合 X の最大値である．

$|L|$: 文字列 L の長さである．特定の場合，文字列 L におけるカンマ (,) と空白で区切られた部分文字列の数を表す．

第1章 はじめに

1.1 本研究の背景

言語の意味は多様であるが，本論文で扱うのは，我々人間が話すことばの言語（natural language）である．

人間は，この世に生まれてからある特定の言語（個別言語 : particular languages）に触れるが，特別な教育・訓練を経ることがなくても，それを母語として習得する．このことから，人間の言語には何らかの共通の基盤があると考えられる．世界諸言語の数は千単位を数え，多様性に富んでいるが，共通の基盤に基づいて育まれてきたゆえ，何らかの共通の特徴があるのではないかと考えられる [中島 05].

言語学者は世界の諸言語にはどのような多様性が見られ，さらにその多様性の中にどのような普遍性が潜んでいるのかについて古くから探求してきており，言語類型論（language typology）という学問体系ができている．言語類型論は音韻論，形態論，統語論，意味論，社会言語学など言語のさまざまな側面を扱い，個々の個別言語の言語特徴を調べ比較することにより，言語普遍性（language universal，世界の諸言語に共通の特徴）[1]を追求する学問である [中島 05, 風間 04].

従来の言語類型論研究では，真の言語普遍性（絶対的言語普遍性）を見つけるために，地理的および系統的な要因は捨象され，主に言語内部の要因に着目して，言語普遍性や言語特徴の間に成り立つ含意ないし相関関係等を見出すことを目標として探求されてきた．近年，地理的分布や歴史的ないし系統的分布を考慮した研究も次第に行われるようになり，それらの要因が言語現象に大きく関係していることが明らかになりつつある [山本 03, 松本 06, Greenberg 63].

言語類型論研究分野では，世界全域にわたり，種々の言語特徴について，その地理

[1]言語普遍性は，絶対的普遍性と統計的普遍性や含意普遍性と非含意普遍性などさまざまに分類できるが，ここでは，広義の言語普遍性を意味するものとする．

的分布の角度から考察する研究手法が注目されている．自然科学の研究分野では地理的な研究に関連するツールとして GIS (地理情報システム) のニーズが高まっている．GIS は多角的な時空間検索と分析の機能を持っており，言語類型論や方言研究などの分野でも GIS の導入が始まっている [池田 06, 山本 06, 乾 06, 杉井 06, 小西 07, 大西 10]．GIS の導入により，文理融合の新たな手法によって新発見をもたらすことが期待されている．

GIS を用いた言語類型論研究を行う際に，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提供する空間データが必要であるが，世界諸言語に関する GIS 空間データは，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．我々が調査した結果，2 つの表形式の言語属性データを処理することにより，間接的に GIS 空間データを生成することが可能であることが分かった [呉 07]．しかし，その前提として，以下に述べるような 2 つの表形式の言語データ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．

言語学者による言語データでは普通，言語名によって言語を識別している．しかし，1 つの言語に対して近隣民族が様々な呼称を用いるため，違った呼び名が複数存在している場合がよくある．これは言語の別名と呼ばれる．また，世界諸言語に関するデータでは，データを編制するうえで用語に関し統一した基準となるものが存在しているわけではなく，各々の言語学者が自らの立場により言語名などを用いていることが現状であるため，表記ゆれなどが含まれていることも少なくない．表記ゆれは例として，“Arabic-based Creole” と “Creole” / “Arabic based”, “Unclassified” と “Language Isolate” などがある．そのため，世界諸言語に関するデータでは言語の名前だけでは言語を識別できないケースが多い．言語を一意に識別できる識別子がないことが，異なる言語学者による言語データを一つにし，新たな言語データを生成するうえで障害となりうる．

言語の一意識別子として，国際標準化機構による言語コード（ISO639 シリーズの言語コードを指す [URLe]）がある．言語コード体系の標準化は 1980 年代から始まり，以来頻繁にコード体系の変更が発生しており，安定性に欠けているといえる．そのためか，近年は情報科学を用いた言語研究を意識して編制または発表された言語資料には，言語コードが付与されるようになってきたものの，比較的に最近発表

されたものでも言語コードが必ず付与されているわけではなく[2]，言語に独自のコードを付ける言語学者もいる[3]

つまり，言語コードは未だに言語学者の間でコンセンサスが得られ，確立された体系が共有され，標準として使われる段階に至っているとはいえないのが現状である．

このように，世界諸言語に関するデータでは言語の一意識別子がないことがしばしばあり，我々が直面した 2 つの表形式の言語データのそれぞれに含まれる言語の同一性判定の問題は決して偶然なことではなく，一例に過ぎないのである．

また，従来の言語類型論研究では，音韻論，形態論，統語論，意味論，社会言語学など各専門分野の言語学者が自らの専門分野において言語特徴に関する属性情報を収集し，分析する研究スタイルが主であった．情報科学（information science）と情報技術（information technology）が著しく発達している今日では，コンピュータの機能を活かし，多方面にわたる言語特徴を横断的に調査し分析することが可能であり，これは過去には成し得なかった研究ができるようになることにもつながる．このような研究を効果的に展開するためには，他の言語学者が収集した言語資料をデータ化し，研究に使う必要がある．その際，複数の異なる言語学者による言語データを一つにし，新たな言語データを生成してから使うことが有効である．しかし，そこでも同様に言語同一性の問題が存在する．これは GIS に限ったことではなく，IT ツールを用いて言語類型論研究を展開するうえで一般的にいえることである [Bickel 07, Bickel 08b, Bickel 08a, Croft 08b, Croft 08a, Croft 09, Cysouw 07, Nichols 08, DeGraff 01, Johnson 11, Donohue 11, Janssen 06].

言語コードが付与されていない，価値ある言語資料は数多く存在する．言語同一性の問題が障害となり，言語研究に活かせないのならば，それは大変残念なことである．人類最大の文化遺産ともいえる言語に関する資料を研究に活かせるようにすることは大きな課題で，本研究はまさにその課題を解決しようとするものである．

[2]その例として，Routledge 社の *Atlas of the world ' s languages* を挙げる．この文献には世界諸言語の地理的分布図が掲載されており，類型論研究分野では価値の高い資料といわれている．初版は 1993 年に出た．2007 年に第 2 版に改訂されたが，第 2 版にも言語コードは付与されていない [Asher 07].

[3] *WALS (The World Atlas of Language Structures)* もその一例である．*WALS* は世界諸言語に関し，音韻，語彙，形態，統語などの言語特徴の地理的分布を地図化したもので，発売当初は書籍と CD-ROM の媒体で提供されていたが，現在は Web サイトで情報提供している [Haspelmath 05, Horie 06].

1.2 本研究の目的

前節で述べたように，情報科学と情報技術を活用して言語類型論研究を展開するためには，複数の言語学者によって編制された言語データに含まれる言語の同一性問題を解決することが必要である．言語コードという言語の一意識別子が含まれている言語データもあるが，言語の同一性が問題となるのは，言語を識別するのにこの言語コードが付けられていないデータを扱う場合である．

一方，世界諸言語に関する言語データは言語数が千単位にのぼるため，手作業によって言語を特定するのは，莫大な作業量を要するうえ，専門知識も必要とし，大変困難である．

複数の言語データのいずれにも言語コードが含まれないこともありうるが，本研究では問題の複雑化を避けるため，次のように問題設定を行う．2 つの表形式の言語データに照準を合わせ，この 2 つの表形式の言語データでは，一方には言語コードが付けられていないが，他方には言語コードが付けられているとする．本研究は，言語コードが付けられていない言語データに含まれる言語について，自動処理によって，その他方の言語データからその同一言語を見つけ出す手法を提案し，なるべく多くの同一言語ペアを検出することを目的とする．

1.3 本論文の構成

本論文は次のように構成される．

第 2 章では，言語同一性問題について述べる．2 つの表形式の言語データを例にとり，言語の名前だけでは言語を識別できないことについて説明したうえで，問題点を分析し次のようにケース分けする．(i) 1 つの言語データに同じ言語名の言語が複数含まれる，すなわち言語名の重複出現，(ii) 別名の存在，(iii) 言語名のゆれ．このうち，(i) の言語名の重複出現の問題に対しては，言語名に加えて，言語系統分類を導入することにより，(ii) 別名の存在の問題に対しては，別名の属性により，(iii)

言語のゆれ（言語名または言語系統分類のゆれ）の問題に対しては，言語の類似性を定量評価するための指標を導入することにより，解決を図る．

第 3 章では，準備として，まず言語学分野における言語系統分類に関する系統樹モデルについて述べる．また，オントロジー・マッピングや人名マッチングなど，本研究と関連のある手法について述べる．その後，文字列類似度の基本手法である編集距離と最長共通部分列（LCS）および文字列類似度計算構造化手法 Monge-Elkan 法について述べる．

第 4 章では，まず言語系統木という木構造について定義し，2 つの表形式言語データに対応する 2 つの言語系統木について述べる．次に，2 つの言語系統木を構成する言語系統情報の取得および XML を用いた言語系統木データの構築について述べる．

第 5 章と第 6 章では，言語同一性判定の手法として 2 つの手法（手法 I と手法 II）を提案する．第 5 章では，手法 I として，まず木構造に基づき，言語名と言語系統分類についてゆれのない完全一致言語の検出法について述べる．次に，Monge-Elkan 法に基づく言語名の類似度および言語系統分類の類似度について定義を行い，言語名や言語系統分類についてゆれのある言語にも対応した木構造と文字列類似度に基づく同一言語ペアの検出手法を提案する．さらに，実験の方法および結果と考察について述べる．

第 6 章では，同一言語ペアの検出率をさらに向上させる目的で，手法 II を提案する．手法 I では，2 つの言語系統木のそれぞれに含まれる 2 つの言語の言語名，または言語系統分類のどちらか一方が完全に一致しているにもかかわらず，このことがその 2 つの言語の同一性判定にまったく考慮されていないという問題点がある．この問題点を指摘し，言語の類似性評価においてその問題点をカバーできる言語名と言語系統分類の総合的尺度に基づく手法を提案する．さらに，実験の方法および結果と考察について述べる．

第 7 章では，言語同一性判定問題に関する今後の課題について述べたのち，本論文をまとめる．

第2章 言語同一性問題

2.1 2つの表形式の言語データの例

第1章 で述べたように，本研究では異なる学者によって編制された2つの言語データの一方に言語コードが付けられていない場合に焦点を絞って，その2 つの言語データに含まれる言語の同一性判定をコンピュータ自動処理によって行うことを狙いとする．本論文では，表2.1 の (A) と (B) に示す言語データをうえで述べた2つの表形式の言語データの例として，問題分析を展開していく．

表2.1 (A) は文献 [山本 03] 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），本研究では方言についての区別を捨象し，一部の言語を削除し2,869 言語を対象とすることにした．一方，表2.1 (B) は *Ethnologue* 第15版 Web サイト [URLa, Gordon 05] 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を指し，言語数は7,299 である．表2.1 の (A) と (B) に示す表をそれぞれ Yamamoto-Data と SilGIS-Data と呼ぶことにする．

表2.1 の (A) と (B) のいずれも，各行のレコードは 1 言語を表す．表2.1 (A) にある 3 つのフィールドの「*No.*」，「第一言語名」，「属性」は表2.1 (B) にもある．*No.* は各々のデータのレコード番号である．**第一言語名** (primary langague name) [Gordon 05, URLa] は言語の名前の 1 つである．**属性**は複数フィールドを含む場合があり，節語順などの言語特徴 [亀井 96, 山本 03] や話者人口や言語使用状況等の言語に関する属性情報を示すもので，表2.1 の (A) と (B) ではその内容が異なることがある．なお，表2.1 において，またこれ以降においても，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** (language code) [URLe, Gordon 05] と**別名リスト** (alternate

表 2.1: 2 つの表形式の言語データの例（Yamamoto-Data と SilGIS-Data）

(A) Yamamoto-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	・・・・・・

(B) SilGIS-Data

No.	第一言語名	言語コード	別名リスト	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	・・・・・・

names) [Gordon 05, URLa] は表 2.1 (B) のみに含まれている．ここでの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 [Gordon 05, URLa] を指し，アルファベット 3 文字から構成され（たとえば，日本語は jpn），言語の一意識別子となる．一方，別名リストは複数の**別名** (alternate name) [Gordon 05, URLa] をカンマ (,) でつなぎ，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（たとえば,「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 [Gordon 05, URLa] 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．1つの言語に付けら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.

世界諸言語の言語データでは，第一言語名と別名の指定は学者独自に行われている．例として，表2.1 (B) *No*=5763の言語はNaroが第一言語名で，Nharo, Nharon, Nhauru, ··· などの複数の別名がカンマ (,) でつないで列挙されている．これに対し，(A) *No*=1855の言語についてはNHARONが第一言語名となっており，別名に関する情報は書かれていない．この2つの言語は第一言語名が異なっているが，実際は同一言語である．

このように，異なる学者によって編制された言語データでは，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．次節では，Yamamoto-DataとSilGIS-Dataを例に取り，2つの言語データに含まれる言語に対し，言語名による対応づけを行う際の問題点について述べる．

2.2 言語名による言語識別の問題点

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（本論文ではこのような文字列を**語**と呼ぶことにする），(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**と呼ぶ）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．

別名リストは複数の別名をカンマでつないだ文字列になっている．表2.1 (B) *No* = 6262の別名リスト“Hnatho, Otomi del Estado de Mexico, ···, State of Mexico Otomi”は，それぞれ下線部分の別名をカンマでつないでいる．従って，カンマを検出すれば，複数の別名に分割することが可能である．

以下では，表 2.1 のサンプルデータを例に，2 つの言語データに含まれる言語の対応付けおよびそこに存在する言語名による言語の識別の問題点について，ケース分けして説明していく．

(a) 第一言語名による判定

(A) No=1015 と (B) No=3295 の言語は，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 の言語は，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，(A) NHARON と (B) 別名リスト “Nharo, Nharon, Nhauru, $\cdots$” の下線部分の語との一致が認められるため，同一言語と判定できそうである．

(c) 第一言語名と別名による判定その 2

(A) No=1959 と (B) No=6262 の言語は上記 (a) と (b) のいずれの方法によっても同一性判定ができないが，(A) の 第一言語名 “OTOMI, STATE OF MEXICO” と (B) の 別名リスト “Hnatho, Otomi del Estado de Mexico, $\cdots$, State of Mexico Otomi” を比較すると，下線部分の語のリストが何らかの方法で一致が認められそうなため，こちらも同一言語と判定できそうである．

上記 (a) $\sim$ (c) のケースは少なからず同一言語の判定ができそうである．つまり，第一言語名または別名は言語の同一性を判定するうえ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**となってしまうケースもある．この例を次の (d) と (e) に示す．

(d) 言語名の重複出現

(A) No=212 と No=213 の言語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 の言語が，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になってしまう．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 の言語は，第一言語名がそれぞれ‘‘CHINANTECO, LALANA’’ と ‘‘Chinantec, Lalana’’ である．下線部分の CO と c は表記上の違いなどによる変化と直感的にわかるもので，本来は同じ言語名なのではないかとの推測がつくが，しかし，上記 (b) または (c) の方法，すなわち第一言語名が一致するかどうか，または第一言語名と別名リストのなかのどれかの別名とが一致するかどうかによる方法では判定できない．

表記ゆれとは，同音・同意味の語句について異なる文字表記が付されることである．日本語では,「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 [URLf].

一般的に，世界諸言語データでも言語名に表記ゆれが含まれることが少なくない．世界諸言語に関する言語資料は文献調査による成果であることがしばしばで，言語名の多くはデータ編制者にとっては外国語で書かれていることが珍しくない．また，使用する言語の指定や用語基準などが存在するわけではない．そのため，表記ゆれが含まれることもよくある．表記ゆれとして，たとえば‘‘Proper’’ と ‘‘Plains’’, ‘‘of’’ と ‘‘de’’, ‘‘Arabic-based Creole’’ と ‘‘Creole’’/‘‘Arabic based’’, ‘‘Unclassified’’ と ‘‘Language Isolate’’ などを例として挙げることができる．

うえで述べた (a) ∼ (e) のなかで，(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．(a) ∼ (c) または (e) のケースについても，2 つの言語が同じ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，その同一性判定がより正確なものになる．

第3章 準備

本章では，まず 3.1 において，言語系統分類および系統樹モデルについて述べる．次に，3.2 においてオントロジー・マッピングや人名マッチングなどの関連研究について簡単に述べたのち，3.3 において文字列類似度計算の指標である編集距離と最長共通部分列および文字列類似度計算構造化手法 Monge-Elkan 法について述べる．

3.1 言語系統分類と系統樹モデル

世界には数千種の言語があるといわれている[1]．これらの言語の分類には，語形の形態論的な特徴をもとにした類型論的な方法と言語の系統による分類の方法がある．人間に系統があると同じように，言語にも系統があり，家族に似た語族があるはず，つまり親となる言語とその子供となる言語があるはず，という発想に基づき，歴史的・通時的な観点から世界諸言語を分類するのが後者である[2]．

[1]言語は常に変化している．使われなくなった言語は消滅することも，新しい言語が生まれてくることもある．ゆえに，現用言語 (living language) の数を正確に知ることは不可能である．たとえば，*Ethnologue* 第 15 版 [Gordon 05, URLa] では現在話されていることが知られている 6,912 言語に関する情報が掲載されている．*Ethnologue* は国際 SIL(International Summer Institute of Linguistics) という言語研究団体が公開している出版物および Web サイトのことを指し，言語の開発と記録を促進するための言語学者の間における言語情報の共有を目的として創刊されたもので，言語に関する目録としては世界屈指の規模を有している。*Ethnologue* は 4,5 年ごとに改訂版を出しているようで，現在の最新版は第 17 版 [URLc] となる．第 16 版 [Lewis 09, URLb] までは出版物があったが，第 17 版は出版物は発売されていなく，Web サイトだけのようである．また，*Ethnologue* 第 16 版と *Ethnologue* 第 17 版に掲載されている現用言語の数はそれぞれ 6,909 と 7,105 言語となっている．このように，*Ethnologue* に掲載されている現用言語の数は版の改訂につれて多くなったり，少なくなったりと，変化してきている．

[2]十八世紀末，イギリスのインド学者で法律家であった Sir William Jones (1746-94) がある講演の中で述べた言葉に始まるといわれている．彼は，ずば抜けた語学力の持ち主で，直感によって「言語はみな 1 つの源から分かれ出たものに違いない」と考えた．彼のその一言がきっかけとなって，系

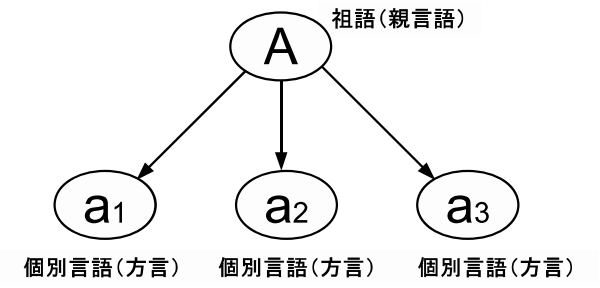


図 3.1: 言語の変化

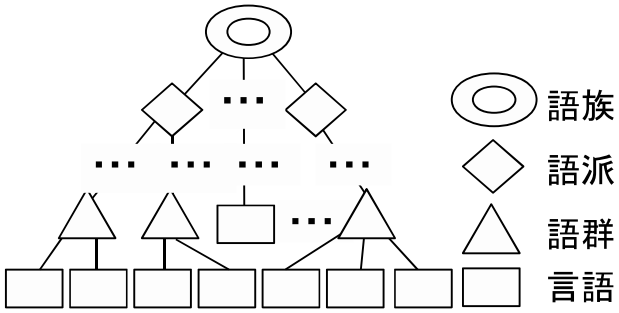


図 3.2: 言語系統樹

語族を設定する条件とともに，同じ系統に属する言語の間の親族関係の解明をしようと，言語の歴史的変化の様子を説くモデルがいくつか提唱されてきた．そのなかの 1 つとして，系統樹説 (family-tree theory) がある．系統樹説は同じ語族に属する言語は，はるか過去に話されていた 1 つの言語から分かれて発展してきたと主張する．そして，言語の分化の過程は一本の樹にたとえられ，それは系統樹 (family tree) と呼ばれる [R.M.W. ディクソン 01, 風間 93, 中島 05].

系統樹説の提唱者はドイツの言語学者 August Schleicher である．系統樹説は，図 3.1 に示すように，A という祖語 (parent language，親言語ともいう) から a_1，a_2，a_3，$\cdots$ という子供の言語（個別言語）へと分かれる言語変化の図式を主張している．

統分類の角度から言語と言語の関係を探る学問が勃興したのである [R.M.W. ディクソン 01].

言語の変化・変遷を表す系統樹説は，インド・ヨーロッパ（印欧）語族を念頭に考え出されたもので，サンスクリット語を中心に印欧語の同系性を確認する形で研究が進められ構想されたものである．系統樹説は，ヨーロッパの言語をサンスクリット語と関連付け，言語同士の関係を解釈する古典的な学説であり，印欧語族に適合したモデルといわれている．印欧語族の系統樹の頂点にある印欧祖語は，印欧語族に属する諸言語の起源として存在するはずの単一の言語であり，印欧語はその 1 本の木の幹となる印欧祖語から枝分かれした言語と考えられている．しかし，これは理論的な要請によるもので，実際の言語のように歴史性をもつものではなく，現存の資料に基づいて再建 (reconstruction) されたものである [亀井 96, R.M.W. ディクソン 01].

系統樹説は言語の分裂や言語関係を示すのに理想的な枝分かれの図式を提供しているが，しかし実際は世界中のどの地域の，どのタイプの言語にも当てはまるわけではない．それ以後，波動説 (wave-theory) や 言語圏説 (linguistic area theory)，断続平衡説 (punctuated equilibrium theory) などの学説 [亀井 96, R.M.W. ディクソン 01, 池上 80] [3]も提唱されてきた．しかし，なかでも系統樹説がやはり最もメジャーな学説のようで，現在言語学者によって提供されている言語系統分類に関するデータは系統樹モデルで構築されているものが多い．たとえば，*Ethnologue* で提供している言語系統分類データの構造も系統樹モデルに従っている [URLa]．また，情報科学分野における言語系統分類の言語モデルに関する研究の多くも，系統樹モデルを視野に行われている [北 97, Gray 03, Dunn 05, Nichols 08, Nicholls 08].

現在用いられている言語系統分類に関するデータの構造は図 3.2 に示すようになっている．同系の言語は 1 つの語族を構成する．語族は系統樹の最大の分類で，語派と語群は同じ語族のなかでの中分類と小分類であり，最下位にあるのが言語である．

[3]系統樹説は同系の諸言語は一般に分岐的 (divergent) な方法によって発達すると考える．つまり，今日の英語とドイツ語の相違は，1000 年前よりずっと大きいことを意味する．しかし，言語の収束的 (convergent) な発達や，混成言語 (mischsprachen) が現に存在する，という事実から，系統樹モデルは多くの状況で適切かつ有効であるものの，どれにも当てはまるものではなく，言語同士の関係のすべてのタイプを説明できるものではないとも指摘されている．

3.2 関連研究について

本研究が目指す問題解決は，著者の知る限りにおいて，今まで研究が行われていない．本研究では，アプローチとして言語系統木という概念を導入する（言語系統木では，言語は最下位のレベルに位置するリーフノードとなる）．これは，言語名だけでは言語の同一性を判定するための情報が足りないため，別の角度からの情報として，言語系統分類を取り入れるためである．言語系統分類に関するモデルはいくつか提唱されてきたが，そのなかの１つとして，系統樹モデルがある．系統樹モデルでは１つの系統樹は１つの語族に含まれる言語から構成され，系統樹は言語と言語の間の親族関係を表している．本研究では，系統樹モデルに基づき，世界諸言語のデータ構造を系統樹の森となる言語系統木として定義する．言語系統木の導入により，２つの表形式の言語データに含まれる言語同一性判定の問題は２つの木構造のリーフノードの間のマッチング問題となる．また，言語系統分類も言語名と同様に，曖昧な性質を持つため，言語名の類似度と言語系統分類の類似度という概念を導入し，言語類似性の定量化を試みる．

木構造におけるデータマッチングに関する研究は広く行われており，異なるオントロジー間の対応関係を見つけ出すオントロジー・マッピング (ontology mapping) の技術開発が盛んに行われてきている [市瀬 02, 市瀬 04, 市瀬 07, 市瀬 08, Ichise 08, Euzenat 04b, Euzenat 04a, Euzenat 07, Stoilos 05, 田村 07, 星合 05, 伊藤 00].

オントロジー (ontology) は，「存在論」を指す哲学用語としても使われているが，情報科学分野では，オントロジーは「概念（化）の明示的仕様」と定義されており，簡単に表現すると「言葉の階層構造とネットワーク」を意味する．オントロジーはある知識ベースが前提としている対象世界の概念化を明示化した物であり，オントロジーと知識ベースとは同じではない．また，オントロジーといえるためには，(a) その分野の人々が合意した知識について表現したものかどうか，(b) 人々はそれを正確に定義された用語として参照しているか，(c) 安定しているか，などの条件がある [溝口 99a, 溝口 99b, 溝口 99c].

言語学における言語系統分類の学問分野自身がまだ確立された体系を樹立するまでに至っておらず，曖昧なところが多い．これらの条件に照らすと，言語系統分類

の木構造に関しては，現状では，まだオントロジー構築が可能な段階には至っていないといえる．言語系統分類のなす木構造に基づいてオントロジー構築が困難である以上，オントロジー・マッピングの手法は直接には使えない．

一方，木構造におけるデータマッチングに関する研究として，ほかに木編集距離などを使った木構造パターン・マッチングがあり，数多くの手法が提案されている [Akutsu 06, Akutsu 10, Bille 05, 浅井 04, 久保山 04, 久保山 05, 久保山 06, 齋藤 06, Wang 01, Yan 02, 深川 04, Zaki 02, Zhang 93, Jiang 95, Valiente 01]．しかし，本研究では 2 つの言語系統木に含まれる言語（リーフノード）のマッチングのみに限定しており，木構造全体のマッチングまでは考慮する必要がないことから，それらの手法も本研究に必ずしも適しているとはいえない．

また，木構造におけるデータマッチングに関するテーマではないが，近年ソーシャルネットワークに関連して，人の名前を特定する人名マッチング（personal name matching）の研究が盛んである [Christen 06, Galvez 07, Piskorski 08, Cohen 03]．人名は，本研究が扱う言語名に類似しており，言語名を比較する手法として参考になることが期待できそうである．

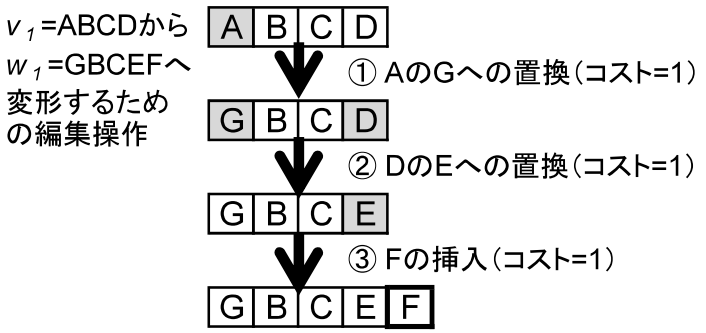
上で述べたオントロジー・マッピングや木構造パターン・マッチングおよび人名マッチングなどに共通して，文字列類似度に関する手法が用いられている．文字列類似度に関する手法は，ほかにバイオインフォマティクスやパターン認識など，多くの分野において応用されている [Navarro 01, Sellers 80, Tiedemann 99, 江里口 96, 高橋 02, 箱田 06, 川上 06, Neil C. Jones 07, 斎藤 05, 高橋 09, 小嶋 93, Islam 08, Islam 09]．文字列パターンに基づく類似度計算の指標としては，文字列の間の距離を測るための編集距離（edit distance）および文字列同士が類似している度合いを測るための最長共通部分列（Longest Common Subsequence, LCS）, N-gram，Jaro-Winkler などがある [Galvez 07]．また，このほかに Monge-Elkan 法という文字列類似度計算構造化手法がある．

本研究では，編集距離を基本とし，文字列類似度計算構造化手法 Monge-Elkan 法を言語名の類似度の計算に取り入れる．このあとの 3.3 において，編集距離, 最長共通部分列，Monge-Elkan 法について説明する．

3.3 文字列類似度評価の指標および基本的手法

3.3.1 編集距離

2 つの文字列がどの程度異なっているかの距離 (非類似性) を測る指標として，1966 年にロシア学者の Vladimir Levenshtein によって考案された**編集距離 (edit distance)** [Levenshtein 66] という概念がある．編集距離は Levenshtein 距離 (Levenshtein distance) とも呼ばれ，2 つの文字列の間の距離を 1 つの記号の挿入，1 つの記号の削除，および 1 つの記号から別の記号への置換という編集操作のもとで，片方の文字列からもう片方の文字列に変形するために必要最少の編集操作の回数として提案されている．編集距離はバイオインフォマティックスを始めとする多くの分野で広く使われている [Neil C. Jones 07, 斎藤 05, 中村 95].



置換，挿入，削除のコストをいずれも
同じ値の 1 とした場合，編集距離 $ed(v_1, w_1) = 3$

図 3.3: 編集距離

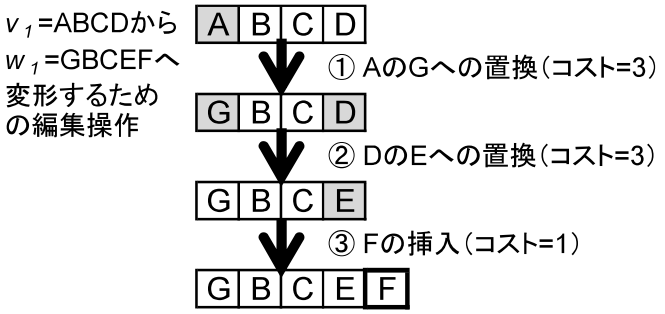
2つの文字vとwの編集距離(edit distance)を$ed(v,w)$と表記し，文字列v_1=ABCD(長さ$|v_1|$=4)と 文字列w_1=GBCEF(長さ$|w_1|$=5) という2つの文字列を例にとり，編集距離について説明する．図3.3に示すように，文字列v_1から文字列w_1に変形するための編集操作のコスト合計の最小値は3となるため，編集距離$ed(v_1,w_1)=3$となる．図3.3に示す編集距離の計算では，1文字の挿入，1文字の削除，または1文字から別の1文字への置換という3つの編集操作のコストをいずれも1としているが，たとえば，1文字の挿入と1文字の削除のコストはそれぞれ1とし，1文字から別の1文字への置換のコストは3とするならば，図3.3と同じ編集操作をした場合のコストは7になり，コストの合計=7となる．これを図3.4に示す．図3.4に示す編集操作の場合のコストは大きく，むしろ図3.5に示している編集操作のほうがコストが小さくなる．この場合，コストの合計=5となる．

編集距離(edit distance)の用語は，特に図3.3に示すような挿入，削除，置換の3つの編集操作のコストのいずれも1とするLevenshtein距離を指し，simple edit distanceとも呼ばれる．これに対し，図3.4や図3.5に示すような挿入，削除，置換の3つの編集操作のなかで，どれかの編集操作のコストが1ではない場合や特定の文字にコストを加重するような場合はgeneral edit distanceと呼ばれる[Navarro 01]．一般的に，編集距離(edit distance)というときは前者を指すことが多い．本論文においてもこの用語を同様に用いる．

編集距離の計算は動的計画法(dynamic programming)に基づいている．編集距離は，編集操作のコストを定めた下で，編集グラフとよばれるグリッドを通る最短距離となる[Neil C. Jones 07]．文字列vとwの編集距離$ed(v,w)$は式(3.1)の漸化式を満たす$f_{min_{|v|,|w|}}$の値となる．

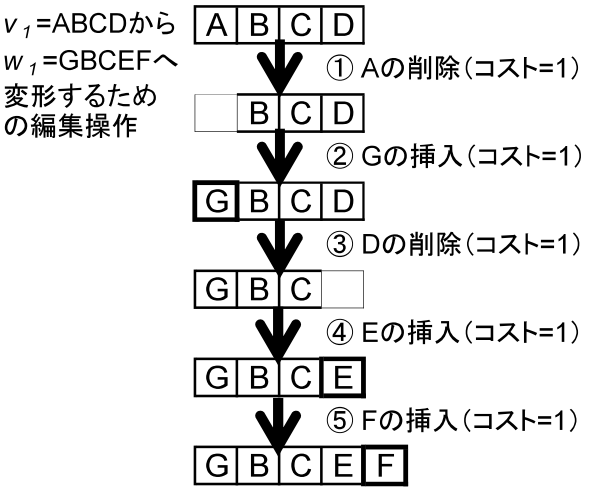
$$f_{min_{i,j}} = \min \begin{cases} f_{min_{i-1,j}} + 1 \\ f_{min_{i,j-1}} + 1 \\ f_{min_{i-1,j-1}} + d_{min}(v_i, w_j) \end{cases} \tag{3.1}$$

式(3.1)では，(i) $|v|$と$|w|$をそれぞれ文字列vとwの長さとし，$1 \leq i \leq |v|$, $1 \leq j \leq |w|$; (ii) $f_{min_{0,0}}=0$, $f_{min_{i,0}}=i$, $f_{min_{0,j}}=j$; (iii) v_iとw_jをそれぞれ文字列vとwのi番目とj番目の文字とし$v_i=w_j$ならば，$d_{min}(v_i,w_j)=0$; $v_i \neq w_j$ならば，$d_{min}(v_i,w_j)=1$となる．



置換，挿入，削除のコストをそれぞれ 3, 1, 1 とした場合，
編集操作のコストの合計=7

図 3.4: 編集操作のコストの変更による文字列間距離の変化の例



置換，挿入，削除のコストをそれぞれ 3, 1, 1 とした場合，
編集操作のコストの合計=5

図 3.5: 編集操作の変更による文字列間距離の変化の例

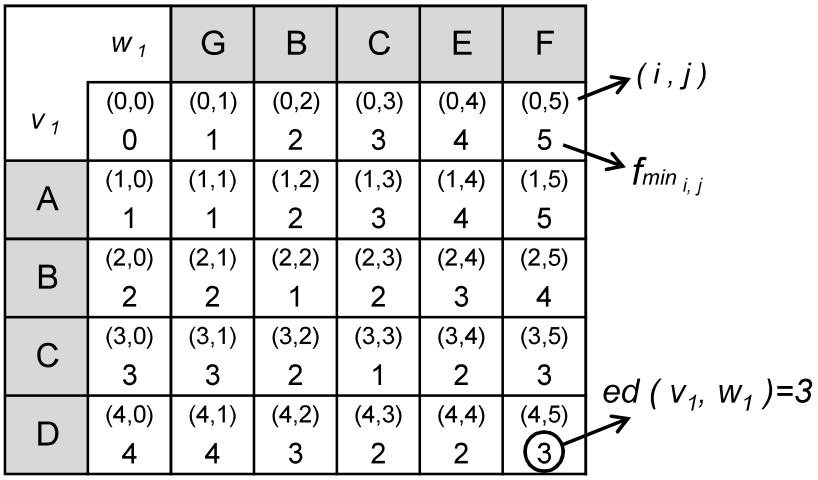
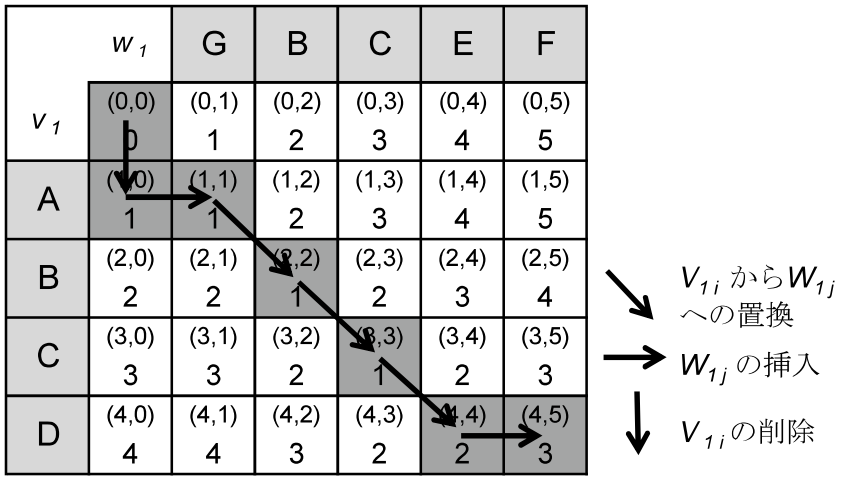


図 3.6: 動的計画法による編集距離 $ed(v_1, w_1) = 3$ の計算

先ほどの v_1=ABCD, w_1=GBCEF の例を用いて，動的計画法による編集距離の計算アルゴリズムを説明していく．ここで, $v_1 = v_{1_1}v_{1_2}\cdots v_{1_i}\cdots$，$w_1 = w_{1_1}w_{1_2}\cdots w_{1_j}\cdots$ とするならば，v_{1_1}=A, v_{1_2}=B, v_{1_3}=C, v_{1_4}=D および w_{1_1}=G, w_{1_2}=B, w_{1_3}=C, w_{1_4}=E, w_{1_5}=F となる．図 3.6 に示しているように，図中の各マス目の上方に書いてある $(0, 5)$ のような丸括弧つきの数字 (i, j) はそれぞれ文字列 v_1 と文字列 w_1 に対し先頭から数えた文字の順番であり，各マス目の下方に書いてある数字は式 (3.1) によって計算される $f_{min_{i,j}}$ の値である．この例では $ed(v_1, w_1) = f_{min_{|v_1|,|w_1|}} = f_{min_{4,5}} = 3$ となることが 図 3.6 から読み取れる．

一方，図 3.7 (A) に示すような編集グラフのグリッド上の走査パスは，文字列 v_1 から w_1 に変形するための操作に対応している．この場合の編集操作は，図 3.7 (B) に示しているように，① $(0, 0)$ から $(1, 0)$ への記号 ↓ は v_{1_1}=A の文字の削除，② $(1, 0)$ から $(1, 1)$ への記号 → は G の文字の挿入，また $(1, 1)$ から $(2, 2)$ へ，$(2, 2)$ から $(3, 3)$ への記号 ↘ は v_{1_2}=w_{1_2} および v_{1_3}=w_{1_3} となるため，無操作となる．これに対して，③ $(3, 3)$ から $(4, 4)$ への記号 ↘ は v_{1_4}≠w_{1_4} となるため，v_{1_4}=D の w_{1_4}=E



(A) 編集グラフ上の走査パス

(A) のように走査したパスに対応する
v_1 から w_1 への編集操作：
① A の削除
② G の挿入
③ D からEへの置換
④ Fの挿入

(B) 編集操作

(A) のように走査したパスに対応する
v_1 と w_1 のアラインメント

v_1	A	−	B	C	D	−
w_1	−	G	B	C	E	F

(C) アラインメント

図 3.7: 編集グラフ上の走査パスとアラインメント

への置換となる．また，$(4,4)$ から $(4,5)$ へは ② と同様に F の文字の挿入となる．

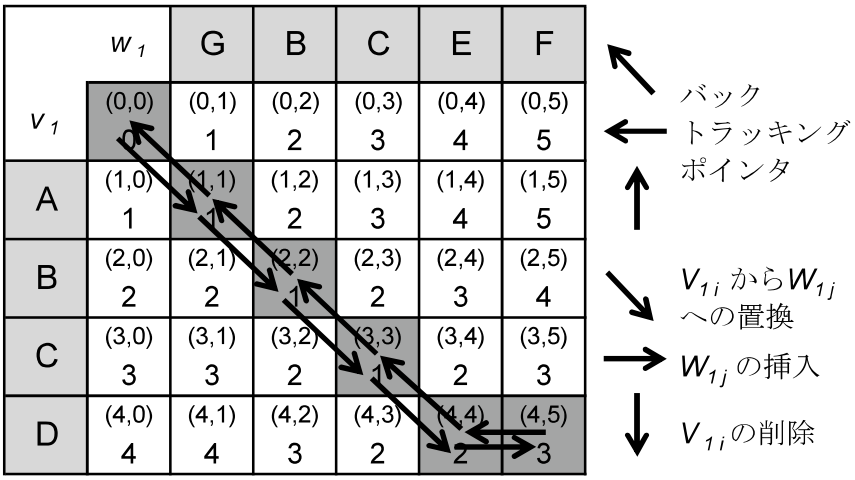
図 3.7 (A) に示す $(0,0)$ から $(4,5)$ へのパスに対応して，図 3.7 (C) に示すような 2 行の文字列 $\begin{pmatrix} \mathrm{A} & - & \mathrm{B} & \mathrm{C} & \mathrm{D} & - \\ - & \mathrm{G} & \mathrm{B} & \mathrm{C} & \mathrm{E} & \mathrm{F} \end{pmatrix}$ が得られ，これは v_1 と w_1 の**アラインメント (alignment)** と呼ばれる．アラインメントは編集グラフ上の走査パスに対応しており，走査パスが複数通りあるため，アラインメントもそれに対応して複数通り可能である．どんなアラインメントも編集グラフ上のパスに対応し，逆に編集グラフ上のどんなパスも，次のように，パス上の各辺がアラインメントの 1 つの列に対応するようなアラインメントに対応する．頂点 (i,j) を終点にもつパス上の記号 $\searrow$,

$\rightarrow$, $\downarrow$ はそれぞれ $\begin{pmatrix} v_{1_i} \\ w_{1_j} \end{pmatrix}$, $\begin{pmatrix} - \\ w_{1_j} \end{pmatrix}$, $\begin{pmatrix} v_{1_i} \\ - \end{pmatrix}$ に対応する [Neil C. Jones 07]．つまり，図 3.7 (A) に示すようなパスで走査したときのアラインメントは次のように書くことができる．

$$\begin{array}{c|ccccccc} v_1 & & \mathrm{A} & - & \mathrm{B} & \mathrm{C} & \mathrm{D} & - \\ \begin{pmatrix} i \\ j \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \begin{pmatrix} 2 \\ 2 \end{pmatrix} & \begin{pmatrix} 3 \\ 3 \end{pmatrix} & \begin{pmatrix} 4 \\ 4 \end{pmatrix} & \begin{pmatrix} 4 \\ 5 \end{pmatrix} \\ w_1 & & - & \mathrm{G} & \mathrm{B} & \mathrm{C} & \mathrm{E} & \mathrm{F} \end{array}$$

また，v_1=ABCD, w_1=GBCEF に対して図 3.7 (B) に示す編集操作は，図 3.3 に示す v_1 と w_1 の編集距離 $ed(v_1, w_1)$=3 を求めるときの編集操作とは異なっており，またアラインメントも異なっている．では，図 3.3 に示すような最少の編集操作をするための編集グラフの走査パスとそのときのアラインメント（このアラインメントを最適アラインメントと呼ぶことにする）はどのように求めることができるのか．以下では，このことについて説明していく．

最適アラインメントは図 3.8 (A) に示すバックトラッキングポインタを用いることで求めることが可能になる．図 3.8 (A) では，前に述べた記号 $\searrow$, $\rightarrow$, $\downarrow$ と方向が反対となる $\nwarrow$, $\leftarrow$, $\uparrow$ の記号がバックトラッキングを意味する．編集グラフの走査グリッド上の終点 $(|v|, |w|)$ から出発し，始点 $(0, 0)$ に向けて逆に走査していく．この際に，左と左斜め上と上との 3 つのマス目のなかで $f_{min_{i,j}}$ の値が最小となるマス目に向けてバックする．$(4, 5)$ からバックする際に，$(4, 4)$ と $(3, 4)$ と $(3, 5)$ のマス目のなかから $f_{min_{4,4}}$=2 または $f_{min_{3,4}}$=2 が同じ値で，ここでは最小となるため，このどちらかに向けてバックする．図 3.8 (A) では $(4, 5)$ から $(4, 4)$ へのパスを選んでいる．次に，$(4, 4)$ からは $(3, 3)$ に向けてしか走査できない．以降同様にして，$(0, 0)$ に到達する．その後は，バックトレースしてきたパスと反対方向に $(0, 0)$ から $(4, 5)$ へ走査していく．図 3.8 (A) に示すパスに対応する編集操作は図 3.8 (B) に示すようになり，これは図 3.3 に示す v_1 と w_1 の編集距離 $ed(v_1, w_1)$=3 を求めるときの編集操作と同じである．そして，このときのアラインメントは最適アラインメントとなり，図 3.8 (C) に示すように $\begin{pmatrix} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} & - \\ \mathrm{G} & \mathrm{B} & \mathrm{C} & \mathrm{E} & \mathrm{F} \end{pmatrix}$ となる．2 つの文字列 v と w のアライ



(A) バックトラッキングポインタの導入による走査パス

(A) のように走査したパスに対応する
v_1 から w_1 への編集操作：
① A からG への置換
② D からEへの置換
③ Fの挿入

(B) 編集操作

(A) のように走査したパスに対応する
v_1 と w_1 のアラインメント

v_1	A	B	C	D	−
w_1	G	B	C	E	F
	X	∗	∗	X	−

∗: 文字の一致，X: 文字の不一致，−: ギャップ

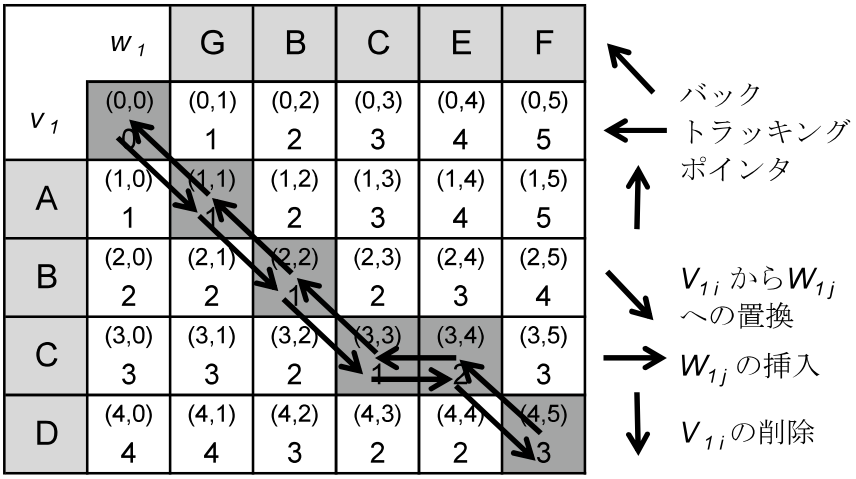
‘X’ の個数 + ‘−’ の個数 $=ed(v_1, w_1)$=3

(C) 最適アラインメント

図 3.8: バックトラッキングと最適アラインメント

ンメントの長さを $l_A(v, w)$ と表記するならば，この v_1 と w_1 の例では，$l_A(v_1, w_1)$=5 となる．

図 3.8 (C) に示す最適アラインメントの 2 行の文字列について，同じ列の 2 つの文字が一致するならば‘∗’，同じ列の 2 つの文字が不一致（置換）であるならば‘X’，また同じ列の 2 つの文字のどちらかに‘−’（ギャップ）が入っている場合は，それは削除または挿入によるものであり，そのときは‘−’とする．‘X’ の個数と‘−’ の個数



(A) 図 3.8 と異なる走査パス

(A) のように走査したパスに対応する
v_1 から w_1 への編集操作：
① A から G への置換
② E の挿入
③ D から F へ置換

(B) 編集操作

(A) のように走査したパスに対応する
v_1 と w_1 のアラインメント

v_1	A	B	C	−	D
w_1	G	B	C	E	F
	X	∗	∗	−	X

∗: 文字の一致，X: 文字の不一致，−: ギャップ

‘X’の個数＋‘$-$’の個数 $=ed(v_1, w_1)$=3

(C) 最適アラインメント

図 3.9: 最適アラインメントのもう一つの例

の合計＝3 となり，これがすなわち編集距離 $ed(v_1, w_1) = 3$ である．

また，図 3.8 (A) のバックトラッキングを図 3.9 (A) のようにすることも可能である．このときの編集操作とアラインメントはそれぞれ図 3.9 (B) と図 3.9 (C) に示すようになり，このときの最適アラインメントは $\begin{pmatrix} \mathrm{A} & \mathrm{B} & \mathrm{C} & - & \mathrm{D} \\ \mathrm{G} & \mathrm{B} & \mathrm{C} & \mathrm{E} & \mathrm{F} \end{pmatrix}$ となる．

2 つの文字列 v と w のアラインメントの長さを $l_A(v, w)$ とし，アラインメントの 2

行の文字列のなかで‘∗’（一致）の個数を $ss(v,w)$ とするならば，$ss(v,w)$ は式 (3.2) のように，2 つの文字列のアラインメントの長さ $l_A(v,w)$ から編集距離 $ed(v,w)$ を引いた値となる．

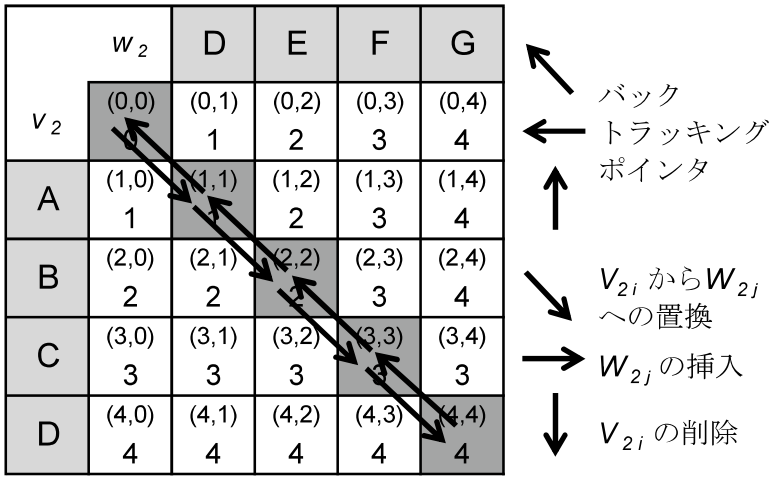
$$ss(v,w) = l_A(v,w) - ed(v,w) \tag{3.2}$$

ここで，$ss(v,w)$ を v と w の類似性スコアと呼ぶことにする．v_1=ABCD, w_1=GBCEF の例では，アラインメントの $\begin{pmatrix} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} & - \\ \mathrm{G} & \mathrm{B} & \mathrm{C} & \mathrm{E} & \mathrm{F} \end{pmatrix}$ と $\begin{pmatrix} \mathrm{A} & \mathrm{B} & \mathrm{C} & - & \mathrm{D} \\ \mathrm{G} & \mathrm{B} & \mathrm{C} & \mathrm{E} & \mathrm{F} \end{pmatrix}$ のどちらも同じく $l_A(v_1,w_1) = 5$ で，またどちらも‘X’（置換）と‘$-$’（削除または挿入）の合計個数が 3 であり，‘∗’（一致）の個数が 2 となるため，この 2 つのアラインメントは等価的であり，いずれも編集距離を求める最適アラインメントとなる．このように，2 つの文字列の最適アラインメントも 1 通りとは限らない．

3.3.2 最長共通部分列 (LCS)

2 つの文字列 v と w の編集距離 $ed(v,w)$ 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．3.3.1 では，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 つの編集操作のコストをいずれも 1 とした場合に，編集距離の計算は，最適アラインメントにおける‘X’（置換）と‘$-$’（削除または挿入）の個数の合計，すなわち‘∗’（一致）以外の個数を数えている．つまり，編集距離は文字列間の距離（非類似性）に着目している．一方，式 (3.2) における類似性スコア $ss(v,w)$ は逆にアラインメントにおける文字の‘∗’（一致）を数えており，これはつまり 2 つの文字列の類似性に着目している．

編集距離と類似性スコアによる文字列類似性評価の違いについて，文字列 v_2=ABC<u>D</u> と w_2=<u>D</u>EFG の例を用いて説明していく．この 2 つの文字列について，下線部分の同じ文字 <u>D</u> が双方の文字列に含まれている．この <u>D</u> による類似性を評価したい．ところが，編集距離を用いる場合の編集グラフとアラインメントはそれぞれ図 3.10 (A) と図 3.10 (B) に示すようになる．図 3.10 から分かるように，編集距離では v_2 と w_2 の



(A) 編集グラフ上のパス

(A) のように編集距離を求めた場合の
v_2 と w_2 のアラインメント

v_2	A	B	C	D
w_2	D	E	F	G
	X	X	X	X

*: 文字の一致，X: 文字の不一致，ー: ギャップ

(B) アラインメント

図 3.10: 編集距離では文字の一致が評価されない例

双方に含まれる同じ文字 D の一致は考慮されることがなく，$ss(v_2, w_2) = 0$ という結果になっている．2 つの文字列の類似性を評価するにあたって，非類似性に着目するならば編集距離は有用な指標となるが，評価する角度によっては編集距離では必ずしも適切とはいえない結果になることがある．

このような場合の類似性を評価したいとき，最も簡単な方法として，最長共通部分列 (Longest Common Subsequence, 以降 LCS と省略して書くことがある) [Navarro 01] を使うことができる．文字列 v_2 =ABC<u>D</u> と w_2 =<u>D</u>EFG の例では，LCS は下線部分

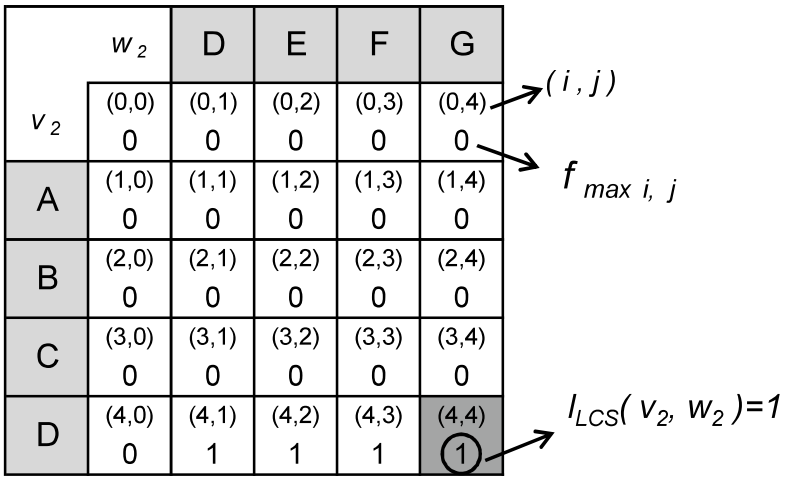


図 3.11: LCS 編集グラフ

の D を指す.

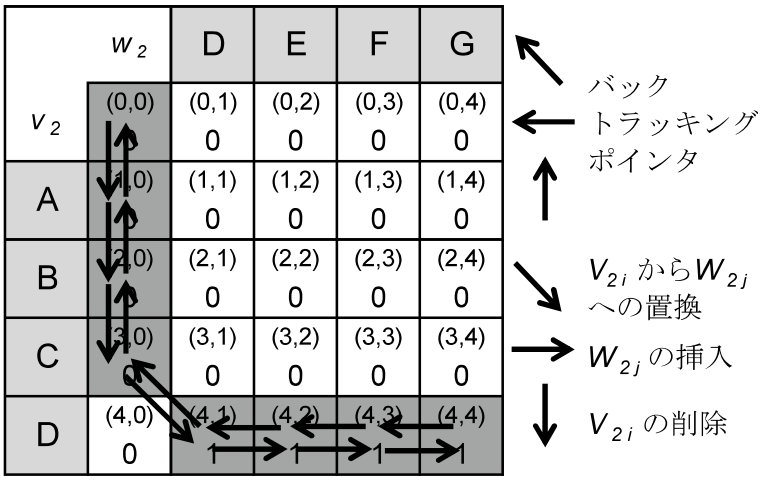
2 つの文字列 v と w の LCS を求める方法は編集距離と類似しており，編集操作のコストを定めた下での LCS 編集グラフとよばれるグリッドを通る最短距離となる．ただし，編集操作については編集距離と異なり，置換を排除し，挿入と削除のみを許し，それらのコストをいずれも 1 としている．また LCS 編集グラフのグリッド上の値は次の式 (3.3) の漸化式を満たすこととしている．

$$f_{max_{i,j}} = \max \begin{cases} f_{max_{i-1,j}} \\ f_{max_{i,j-1}} \\ f_{max_{i-1,j-1}} + 1 \qquad (v_i = w_j) \end{cases} \tag{3.3}$$

式 (3.3) では，(i) $|v|$ と $|w|$ をそれぞれ文字列 v と w の長さとし，$1 \leq i \leq |v|$, $1 \leq j \leq |w|$; (ii) $f_{max_{0,0}}=0$, $f_{max_{i,0}}=0$, $f_{max_{0,j}}=0$; (iii) v_i と w_j をそれぞれ文字列 v と w の i 番目と j 番目の文字とする．

文字列 v_2 と w_2 の例についての LCS を求めるための編集グラフを図 3.11 に示す.

2 つの文字列 v と w の LCS を求める際の最適アラインメントも，図 3.8 (A) に示す方法と同様に，バックトラッキングポインタを導入することで求めることができる．



(A) LCS 編集グラフ上のパス

(A) のようにLCSを求めた場合の
v_2 と w_2 のアラインメント

v_2	A	B	C	D	−	−	−
w_2	−	−	−	D	E	F	G
	−	−	−	∗	−	−	−

∗: 文字の一致，X: 文字の不一致，−: ギャップ

$l_A(v_2, w_2)$=7, $ss(v_2, w_2)$=$l_{LCS}(v_2, w_2)$=1

(B) 最適アラインメント

図 3.12: LCS 編集グラフ上のパスとアラインメント

それと異なるところは，LCS 編集グラフ上の $(|v|, |w|)$ から $(0, 0)$ へバックトレースするときに，左と左斜め上と上のマス目のなかから，$f_{max_{i,j}}$ の値が最大となるマス目を選択し，バックする．文字列 v_2 と w_2 の例を図 3.12 (A) に示す．そして，得られるアラインメントは図 3.12 (B) に示すようになり，$\begin{pmatrix} \mathrm{A} & \mathrm{B} & \mathrm{C} & \underline{\mathrm{D}} & - & - & - \\ - & - & - & \underline{\mathrm{D}} & \mathrm{E} & \mathrm{F} & \mathrm{G} \end{pmatrix}$ とな

る．このアラインメントのなかの下線部分の D が最長共通部分列 (LCS) となる．文字列 v と w の LCS の長さを $l_{LCS}(v,w)$ で表記するならば，この例では $l_{LCS}(v_2,w_2)$=1 となる．

文字列 v_2 と w_2 のアラインメントの長さは $l_A(v_2,w_2)$=7, 類似性スコア $ss(v_2,w_2)$ は LCS の長さであり，$ss(v_2,w_2)$=$l_{LCS}(v_2,w_2)$=1 となる．なお，部分列は連続している必要はないことに注意されたい．

3.3.3 文字列類似度計算構造化手法：Monge-Elkan 法

2 つの文字列ないしテキストの類似度を計算する手法として，Monge-Elkan 法 [Monge 96] があり，近年広く応用されている [Cohen 03, Jimenez 09].

Monge-Elkan 法は 2 つの文字列，たとえば “Dept. of Comput. Sci. and Eng., University of California, San Diego, 9500 Gilman Dr. Dept. 0114, La Jolla, Ca 92093” と “UCSD, Computer Science and Engineering Department, CA 92093-0114” のような文字列に対し，その類似度を計算するために開発された手法である．

2 つの文字列 $A = A_1A_2\cdots A_i\cdots$ と $B = B_1B_2\cdots B_j\cdots$（ここで，$A_1, A_2, \cdots, A_i, \cdots$ と $B_1, B_2, \cdots, B_j, \cdots$ はそれぞれカンマ (,) や空白で区切られた A と B の部分文字列である）について，Monge-Elkan 法では，A と B の類似度は次の式 (3.4) [Monge 96] によって定義されている．

$$match(A,B) = \frac{1}{|A|}\sum_{i=1}^{|A|}\max_{j=1}^{|B|} match(A_i,B_j) \tag{3.4}$$

式 (3.4) では，$match(A_i,B_j)$ は A と B のそれぞれの部分文字列である A_i と B_j の類似度を計算するのに使われている．$|A|$ と $|B|$ はそれぞれ文字列 A と B の部分文字列の数を表している．

Monge-Elkan 法が注目されてきたのは，この $match(A_i,B_j)$ 関数に使う手法よりも，その重畳構造にある．つまり，式 (3.4) では $match(A_i,B_j)$ 関数を切り離して考えることができ，この点に特徴がある．文献 [Monge 96] では，$match(A_i,B_j)$ につ

いては Simth-Waterman 法 [Euzenat 07, 廣安 11] を用いているが，この手法にこだわる必要はなく，ほかの文字列類似度計算の手法を組み入れることができる．次に，2 つの文字列 $A1$ と $B1$ があるとする．

$A1$=“XXXX, YXXX, ZXXX” ($A1_1$ =XXXX, $A1_2$ =YXXX, $A1_3$ =ZXXX)
$B1$=“xxxx, yyyy, zzzz” ($B1_1$ =xxxx, $B1_2$ =yyyy, $B1_3$ =zzzz)
（アルファベットの大文字と小文字は区別しない．以降同様）

式 (3.4) における関数 $match(A_i, B_j)$ について，次の (1) と (2) の 2 通りの計算法を用いる場合の $match(A, B)$ の計算について，この 2 つの文字列を例にとり，説明していく．

(1) 完全一致

関数 $match(A1_i, B1_j)$ について，文字列が一致するか否かによる手法を用いる場合の $match(A1, B1)$ の計算結果について説明する．文字列 $A1$ の部分文字列 $A1_i$ と文字列 $B1$ の部分文字列 $B1_j$ とが完全に一致するならば，$match(A1_i, B1_j) = 1$ とし，そうでないならば，$match(A1_i, B1_j) = 0$ とする．この例では，$|A1| = |B1| = 3$ となり，$match(A1, B1)$ の計算は次のようになる．

① $match(A1_1,\ B1_1) = match(XXXX, xxxx) = 1,$

② $match(A1_1, B1_2) = match(XXXX, yyyy) = 0,$

③ $match(A1_1, B1_3) = match(XXXX, zzzz) = 0,$

よって，(i) $\max\{$ ①, ②, ③ $\} = \max\{\, 1, 0, 0 \,\} = 1$ となる．

④ $match(A1_2, B1_1) = match(YXXX, xxxx) = 0,$

⑤ $match(A1_2, B1_2) = match(YXXX, yyyy) = 0,$

⑥ $match(A1_2, B1_3) = match(YXXX, zzzz) = 0,$

よって，(ii) $\max\{\,④, ⑤, ⑥\,\} = \max\{\,0, 0, 0\,\} = 0$ となる．

⑦ $match(A1_3, B1_1) = match(ZXXX, xxxx) = 0,$

⑧ $match(A1_3, B1_2) = match(ZXXX, yyyy) = 0,$

⑨ $match(A1_3, B1_3) = match(ZXXX, zzzz) = 0,$

よって，(iii) $\max\{\,⑦, ⑧, ⑨\,\} = \max\{\,0, 0, 0\,\} = 0$ となる．

したがって，$match(A1, B1) = \frac{1}{3}((\mathrm{i}) + (\mathrm{ii}) + (\mathrm{iii})) = \frac{1}{3}(1 + 0 + 0) = \frac{1}{3}$ となる．

(2) 編集距離

関数 $match(A1_i, B1_j)$ について， 3.3.1 で説明した編集距離を用いる場合の $match$ $(A1,\ B1)$ の計算結果について説明する．文字列 A の部分文字列 A_i と文字列 B の部分文字列 B_j の類似度 $match(A_i, B_j)$ を式 (3.5) のように定義する．

$$match(A_i, B_j) = \frac{l_A(A_i, B_j) - ed(A_i, B_j)}{l_A(A_i, B_j)} \tag{3.5}$$

式 (3.5) にしたがって，このときの文字列 $A1$ と $B1$ の $match(A1, B1)$ の計算は次のようになる．

① $match(A1_1, B1_1) = match(XXXX, xxxx) = 1,$

② $match(A1_1, B1_2) = match(XXXX, yyyy) = 0,$

③ $match(A1_1, B1_3) = match(XXXX, zzzz) = 0,$

よって，(i) $\max\{\,①, ②, ③\,\} = \max\{\,1, 0, 0\,\} = 1$ となる．

④ $match(A1_2, B1_1) = match(YXXX, xxxx) = \frac{3}{4},$

⑤ $match(A1_2, B1_2) = match(YXXX, yyyy) = \frac{1}{4},$

⑥ $match(A1_2, B1_3) = match(YXXX, zzzz) = 0,$

よって，(ii) $\max\{$ ④, ⑤, ⑥ $\} = \max\{\frac{3}{4}, \frac{1}{4}, 0\} = \frac{3}{4}$ となる．

⑦ $match(A1_3, B1_1) = match(ZXXX, xxxx) = \frac{3}{4},$

⑧ $match(A1_3, B1_2) = match(ZXXX, yyyy) = 0,$

⑨ $match(A1_3, B1_3) = match(ZXXX, zzzz) = \frac{1}{4},$

よって，(iii) $\max\{$ ⑦, ⑧, ⑨ $\} = \max\{\frac{3}{4}, 0, \frac{1}{4}\} = \frac{3}{4}$ となる．

したがって，$match(A1, B1) = \frac{1}{3}((\mathrm{i}) + (\mathrm{ii}) + (\mathrm{iii})) = \frac{1}{3}(1 + \frac{3}{4} + \frac{3}{4}) = \frac{5}{6}$ となる．

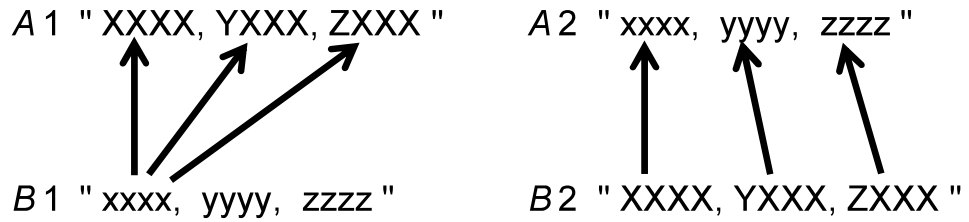
このように，式 (3.4) における $match(A_i, B_j)$ として (1) 完全一致 と (2) 編集距離 を用いる場合の文字列 $A1$ と $B1$ の類似度を計算する関数 $match(A1, B1)$ の結果は異なる値となり，明らかに後者の編集距離の手法のほうがこの 2 つの文字列の類似度計算により適合しているといえる．ここでの注目点は，2 つの文字列 A と B の類似度計算について，A と B の部分文字列同士 A_i と B_j の類似度計算は内部関数 $match(A_i, B_j)$ として分離して定義できる，という構造的な特徴にある．我々は，Monge-Elkan 法のこのような構造を重畳構造と呼んでいる．この重畳構造により，扱う問題の特性に合わせて，$match(A_i, B_j)$ 関数を変化させることができる．

一方，Monge-Elkan 法の欠点としてはその非対称性にある．これは文献 [Monge 96] 自身においても指摘されている．次において，その非対称性について説明する．

さきほど例として挙げた文字列 $A1$ と $B1$ の内容を逆にした文字列 $A2$ と $B2$ を例にとり，説明する．式 (3.4) における $match(A_i, B_j)$ としては，上記 (2) の編集距離を用いる．

$A2$=“xxxx, yyyy, zzzz” ($A2_1$ =xxxx, $A2_2$ =yyyy, $A2_3$ =zzzz)
$B2$=“XXXX, YXXX, ZXXX” ($B2_1$ =XXXX, $B2_2$ =YXXX, $B2_3$ =ZXXX)

このときの $match(A2, B2)$ の計算は次のようになる．



(A) $match(A1, B1) = \frac{5}{6}$ (B) $match(A2, B2) = \frac{1}{2}$

説明：図中の矢印は，類似度が最大となる部分文字列の組合せとその対応関係を表している．また，$match(A1_i, B1_j)$ と $match(A2_i, B2_j)$ については同じく編集距離に基づく類似度計算法を用いている．

図 3.13: Monge-Elkan 法の非対称性

① $match(A2_1, B2_1) = match(xxxx, XXXX) = 1,$

② $match(A2_1, B2_2) = match(xxxx, YXXX) = \frac{3}{4},$

③ $match(A2_1, B2_3) = match(xxxx, ZXXX) = \frac{3}{4},$

よって，(i) $\max\{ ①, ②, ③ \} = \max\{ 1, \frac{3}{4}, \frac{3}{4} \} = 1$ となる．

④ $match(A2_2, B2_1) = match(yyyy, XXXX) = \frac{0}{4} = 0,$

⑤ $match(A2_2, B2_2) = match(yyyy, YXXX) = \frac{1}{4},$

⑥ $match(A2_2, B2_3) = match(yyyy, ZXXX) = \frac{0}{4} = 0,$

よって，(ii) $\max\{ ④, ⑤, ⑥ \} = \max\{ 0, \frac{1}{4}, 0 \} = \frac{1}{4}$ となる．

⑦ $match(A2_3, B2_1) = match(zzzz, XXXX) = \frac{0}{4}$=0,

⑧ $match(A2_3, B2_2) = match(zzzz, YXXX) = \frac{0}{4}$=0,

⑨ $match(A2_3, B2_3) = match(zzzz, ZXXX) = \frac{1}{4},$

よって，(iii) $\max\{ ⑦, ⑧, ⑨ \} = \max\{ 0, 0, \frac{1}{4} \} = \frac{1}{4}$ となる．

したがって，$match(A2, B2) = \frac{1}{3}((\mathrm{i}) + (\mathrm{ii}) + (\mathrm{iii})) = \frac{1}{3}(1 + \frac{1}{4} + \frac{1}{4}) = \frac{1}{2}$ となる．

文字列 $A2$ と $B2$ は文字列 $A1$ と $B1$ の内容を逆にしただけであるが，図 3.13 に示すように，それらの部分文字列間の対応が変化し，最終的に得られる 2 つの文字列の類似度も $match(A1, B1) = \frac{5}{6}$ と $match(A2, B2) = \frac{1}{2}$ という異なる結果となっている．Monge-Elkan 法の非対称性とはこのことを指す.

2 つの文字列の類似度計算において，この非対称性が妥当性を欠くことになることがある．本研究では，Monge-Elkan 法の重畳構造を取り入れつつ，この非対称性について改善を図る．

第 4 章 言語系統木の構造と言語系統木データ

4.1 はじめに

3.1 では言語系統分類についての系統樹モデルについて説明した．図 3.2 に示しているように，系統樹モデルでは同系の言語は 1 つの語族となり，1 つの系統樹を構成する．世界諸言語は多くの語族，多くの系統樹を構成する．このように，世界諸言語は系統樹の森となる．

図 3.2 に示す系統樹はデータ構造のうえでは，根付き木となる．ここで，語族がルートノード (root node) となり，言語がリーフノード (leaf node) となる．根付き木では，ルートノードの語族からリーフノードの言語までのパスは常に一意という性質を持つ [斎藤 98]．このことから，言語名に加え，言語系統分類も考慮すれば，2.2 (d) で述べたような，言語名が重複している場合でも，言語の対応関係が判定可能になる．

そこで，本研究では表 2.1 のような表形式データの情報不足を補うため，言語名に加えて言語系統分類の情報も言語の同一性判定に取り入れることを提案し，2 つの表形式の言語データに含まれる言語の同一性判定の問題を 2 つの木構造である言語系統木のデータに含まれる言語の同一性判定の問題として扱う．

以下， 4.2 では，言語系統木という木構造について定義し，Yamamoto-Data と SilGIS-Data に対応する 2 つの言語系統木 T_Y と T_S について述べる． 4.3 では，2 つの言語系統木 T_Y と T_S を構成する言語系統情報の取得および XML を用いた言語系統木のデータ構築について述べる．

4.2 言語系統木

4.2.1 言語系統木の定義

本研究では，まず図 3.2 に示す世界諸言語の言語系統分類データのモデルとなる言語系統樹について，中分類を表わす語派と小分類を表わす語群をまとめて**言語グループ**とし，データ構造の抽象化を行う [石畑 89, 落水 93]．それは図 4.1 に示すようになる．また，世界諸言語は多くの語族に分類されているが，本研究では図 4.2 に示すように，語族の森を 世界諸言語 (World Languages) というルートの下にまとめ，1 本の**木**として扱うことにする．言語名（語族名や言語グループ名を含む）を木構造のノードのラベルとし，また別名リストと言語コードを最下位にあるリーフノードの言語の属性情報として，それぞれもたせることにする．この構造を言語系統木と呼び，そのイメージを図 4.3 に示す．以下では，まず木について定式化したうえで，図 4.3 に示す言語系統木について定義する．

定義 4.1. 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**と呼ぶ．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**と呼ぶ．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 4.1 (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 4.2 で与えている．

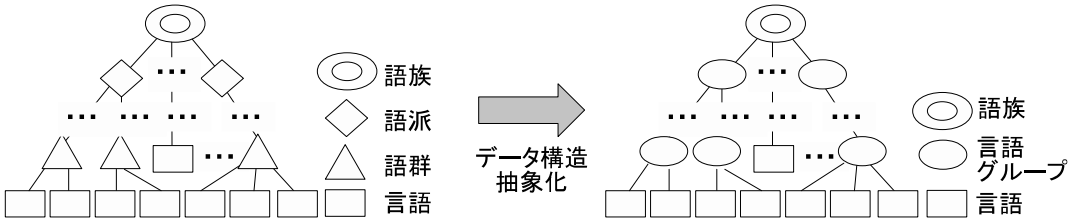


図 4.1: 言語系統樹データ構造の抽象化

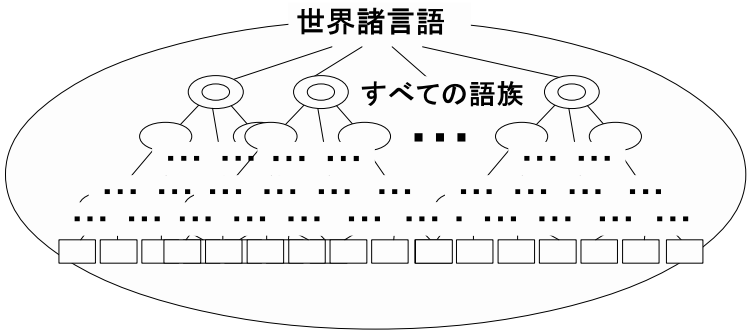


図 4.2: 世界諸言語

定義 4.2. 次の条件を満たす T を**言語系統木**と呼ぶ.

(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，**別名リスト**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 $\mathcal{L}_{r(T)} = \{\text{World, Languages}\}$ 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.

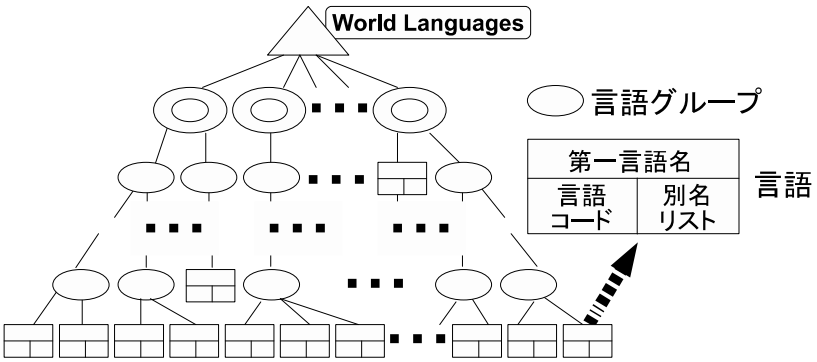


図 4.3: 言語系統木

(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≤w_2^x≤$\cdots$) , $\mathcal{L}_z$ =$\{w_1^z, w_2^z, \cdots\}$ (w_1^z≤w_2^z≤$\cdots$) とする．(i) w_1^x=w_1^z, w_2^x=w_2^z,$\cdots$, w_{i-1}^x=w_{i-1}^z, w_i^x<w_i^z となるような $i(\geq 1)$ が存在する，または (ii) l=$|\{w_1^x, w_2^x, \cdots\}|$<$|\{w_1^z, w_2^z, \cdots\}|$ としたときに，w_1^x=w_1^z, w_2^x=w_2^z,$\cdots$,w_{l-1}^x=w_{l-1}^z,w_l^x=w_l^z が成立するならば，$L(x)$≺$L(z)$ である．$L(x)$≻$L(z)$ が成立しないとき，$L(x)$⪯$L(z)$ と書く．ここで，w_i^x<w_i^z は w_i^x と w_i^z の辞書式順序を表す. □

なお，本研究の言語系統木を構成する言語データでは，$L(x)$=$L(z)$ となるような 2 つの異なる兄弟 x と z は存在しないとしている．

4.2.2 2 つの言語系統木 T_Y と T_S

本研究では 2 つの言語系統木を処理対象とする．この 2 つの言語系統木を T_Y と T_S とし，図 4.4 に示す．y∈$V_{leaf}(T_Y)$，つまり y は言語系統木 T_Y に含まれる 1 つの言語で，s は本来 y と同一の言語と仮定すれば，言語系統木 T_S での s の有無または有り方としては，(i) s∈$V_{leaf}(T_S)$，かつ言語名と言語系統分類がともに y とまったく同じである．(ii) s∈$V_{leaf}(T_S)$ となるが，言語名または言語系統分類のどちらかが y と異なっている．(iii) s∉$V_{leaf}(T_S)$，つまり s は言語系統木 T_S には含まれていない，

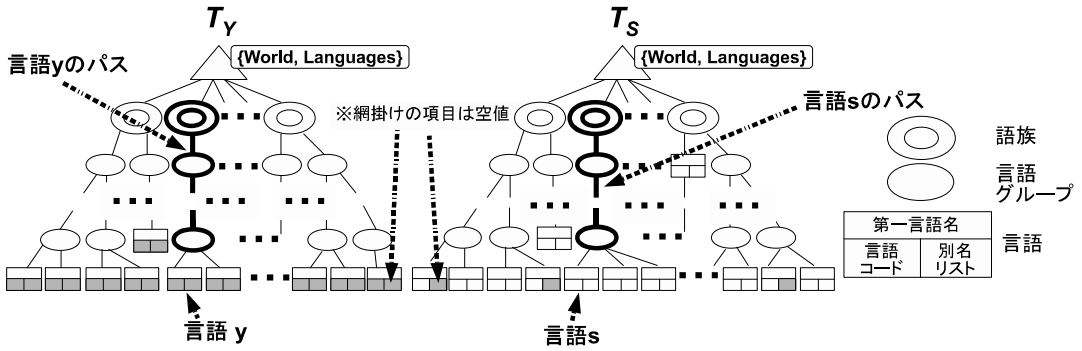


図 4.4: 2 つの言語系統木 T_Y と T_S

という 3 つの可能性がある．本研究の狙いは，このような (y, s) の同一言語ペアをなるべく多く検出することである．

ここでの T_Y と T_S は定義 4.2 の定める言語系統木の条件を満たす木構造で，次のような特徴がある．

定義 4.2 で定式化されているように，言語系統木 T_Y と T_S のどちらでも，任意のリーフノード $x \in (V_{leaf}(T_Y) \cup V_{leaf}(T_S))$ には言語コード C_x と別名リスト A_x との 2 つ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）．つまり，T_Y では，言語コード C_y も別名リスト A_y もデータとして存在していない．(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，別名リストは必ず存在しているわけではなく，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．また，言語数について，T_S が T_Y よりはるかに上回ることが通常である．

本研究では，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 $s \in V_{leaf}(T_S)$ を見つけ出すための手法を提案する．

4.3 言語系統木データ

4.3.1 言語系統木のデータソース

本章では言語名の類似度と言語系統分類の類似度および同一言語ペアの検出手法の有効性などを確認するため，図 4.4 に示している 2 つの言語系統木 T_Y と T_S の構造を持つ 2 つの言語系統データを用いて実験を行う．この 2 つのデータをそれぞれ $T_{Y_{XML}}$ と $T_{S_{XML}}$ と表記する．これらはそれぞれ Yamamoto-Data と SilGIS-Data に関連する言語系統情報から作られるデータであり，以下ではそれらのデータソースについて述べる．

(i) Yamamoto-Data に対応した言語系統データは文献 [山本 03] にある「系統別語順分布表」であり，言語を系統分類の観点から整理した語順データである．$T_{Y_{XML}}$ は，この文献にある紙ベースのデータを言語系統木の定義（定義 4.2）に従って XML 形式に変換したデータのことを指す．$T_{Y_{XML}}$ に含まれる言語数は 2,869 で，117 語族で構成されている（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，そのような言語を除外して，2,869 言語を取り入れることにした）．

(ii) SilGIS-Data のデータソースである *Ethnologue* 第 15 版 Web サイト [URLa] には世界諸言語の系統分類の情報も掲載されている．$T_{S_{XML}}$ は，この Web サイトから言語系統分類上の親子情報を含め，言語名や言語コードや別名などを取得し，言語系統木の定義に従って XML 形式に変換したデータのことを指す．$T_{S_{XML}}$ に含まれる言語数は 7,299 で, 108 語族で構成されている．

なお，*Ethnologue* 第 15 版 Web サイトでは，言語名表記に Unicode でないと表現できない文字が使われているところがある．$T_{Y_{XML}}$ での言語名表記は ASCII コード体系に従っているため，$T_{S_{XML}}$ も ASCII コードに変換した．また，Yamamoto-Data と SilGIS-Data では，言語名（言語グループ名などを含む）にアルファベットと区切り記号以外の '|' や '=' などの記号も使われているが，$T_{Y_{XML}}$ と $T_{S_{XML}}$ ではこれらの記号を削除した．

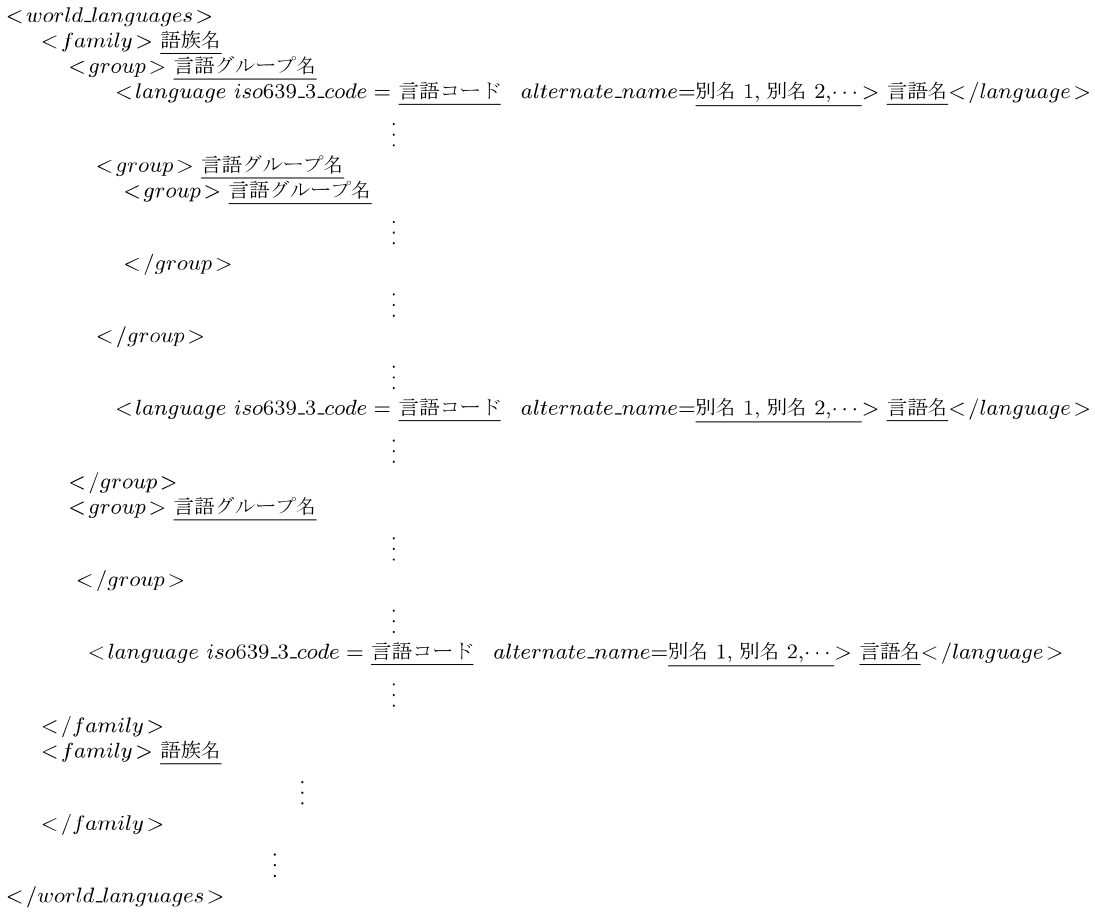


図 4.5: 言語系統木データの XML 構造

4.3.2 XML に基づく言語系統木データの構造

XML (Extensible Markup Language) は文書やデータの意味や構造を記述するためのマークアップ言語の一つで，ユーザが独自のタグを指定でき，様々な場面で利用されている．XML は木構造の表現に優れており，本研究では XML を用いて言語系統木データ $T_{Y_{XML}}$ と $T_{S_{XML}}$ を構築する [猿橋 08, 山田 01, オフィス 01, 立川 04]. XML を用いた言語系統木データ $T_{Y_{XML}}$ と $T_{S_{XML}}$ の構造を図 4.5 に示す.

図 4.5 に，タグ名 $<world\_languages><\/world\_languages>$, $<family><\/family>$, $<language><\/language>$ および $<group><\/group>$ で囲んでいるの

はそれぞれ言語系統木のルートのノード（世界諸言語），ルートの子ノード（語族），リーフノード（言語）とインナーノード（言語グループ）である．図 4.5 から判るように，$<group></group>$ ノードは入れ子構造になっており，言語グループの中に言語グループを包含することができる．また，$<language></language>$ ノードは $<group></group>$ ノードの下位レベルに存在する以外に $<family></family>$ ノードの直下レベルに存在することもある．次節では，XML 形式の言語系統木データ $T_{Y_{XML}}$ と $T_{S_{XML}}$ の生成について述べる．

4.3.3 言語系統木データの生成

$T_{Y_{XML}}$ は，4.3.1 (i) で述べたように Yamamoto-Data に関連する言語系統データから変換して得られた．そのデータは Microsoft Excel 形式の電子データであり，著者自ら作成した自動変換ツールによって，Excel のデータから図 4.5 の構造を持つ XML 形式のデータに変換した．その自動変換ツールについての説明は割愛するが，本節では主に後者の $T_{S_{XML}}$ の生成について述べる．

4.3.1 で述べたように，$T_{S_{XML}}$ は *Ethnologue* 第 15 版 Web サイト [URLa] から情報を取得して，XML 形式に変換して得られたデータである．以下において，*Ethnologue* 第 15 版 Web サイトからの情報取得などについて説明する [韓 06, Suzuki 03, Suzuki 04, 後藤 07, Jabbour 01].

(1) *Ethnologue* 第 15 版 Web サイトの構成およびページリンクの特徴

Ethnologue 第 15 版 Web サイトのトップページは総目次になっており，その Web ページに，言語名別目次ページ (`http://archive.ethnologue.com/15/language_index.asp`) と語族別目次ページ (`http://archive.ethnologue.com/15/family_index.asp`) のリンクがある．*Ethnologue* 第 15 版は書籍版 [Gordon 05] もあるが，語族別目次は Web サイトにのみ提供されている．以降では，*Ethnologue* 第 15 版 Web サイトを Web サイト，Web ページをページと省略して呼ぶことにする．

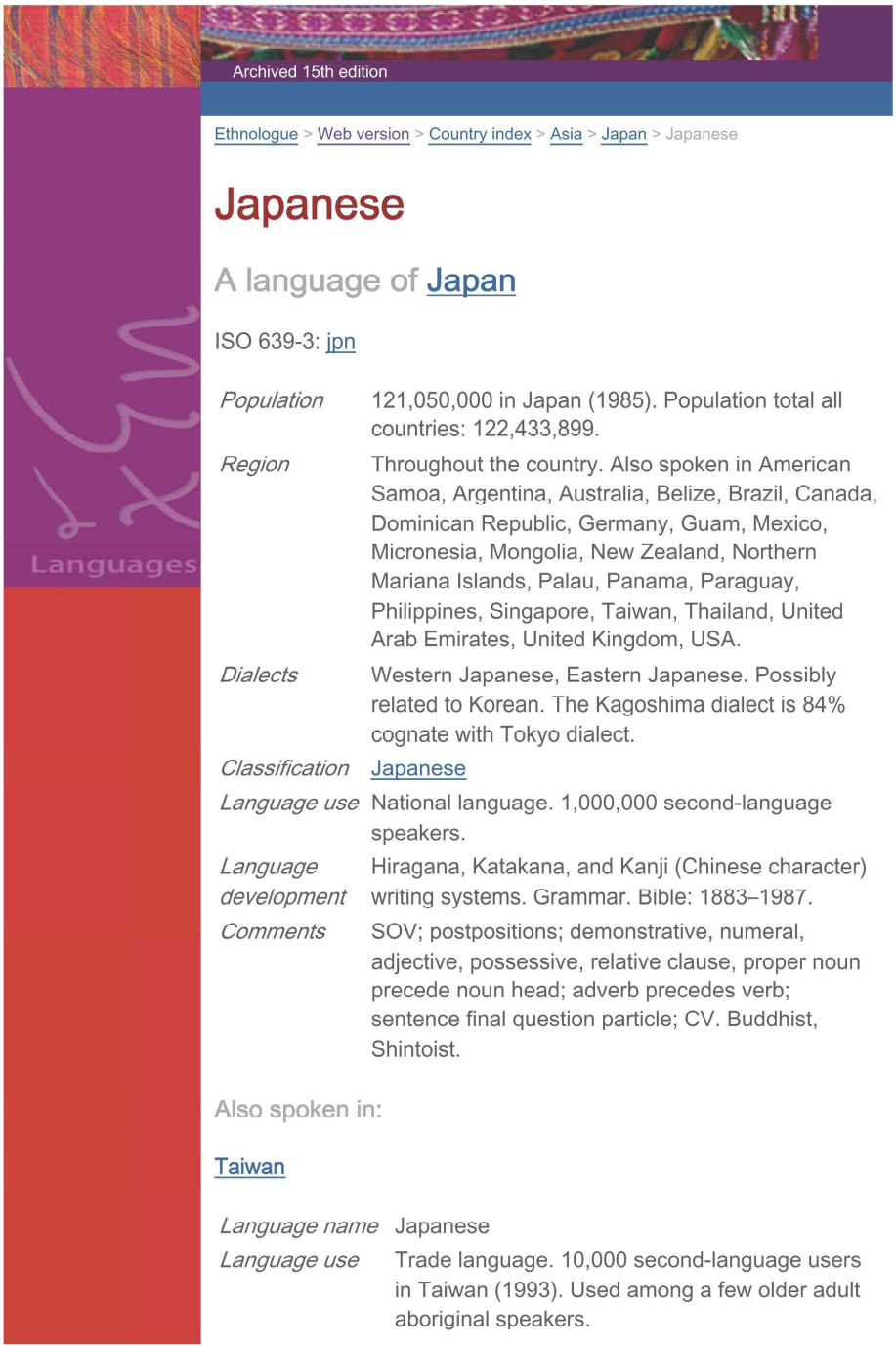


図 4.6: 属性ページ（*Ethnologue* 第 15 版 Web サイト [URLa] より引用）

Web サイトにある言語名別目次ページには A から Z までのアルファベットがあり，そのアルファベットで始まる言語名のページにリンクされている．例として，J をクリックすると Japanese などの J で始まる言語名の一覧となるページ (以降，2 段階目の目次ページと呼ぶ) が表示される．2 段階目の目次ページにある各々の言語名にはその言語に関する属性情報のページ (以降，属性ページと呼ぶ) がリンクされている．1 つの言語は 1 つ以上の国または地域で話されている．その場合，それらの国または地域では，その言語に関する属性情報が違う可能性がある．そのため，各属性ページにはその言語が話されている国別に属性情報が配置されている．その中で，第一国 (primary country) と呼ばれる国または地域があり，それは言語の発祥地または最も多くの話者がいる国または地域である [Gordon 05]．図 4.6 には，例として言語 Japanese の属性ページを示している．‘‘A language of Japan’’ 以下が第一国に関する情報で，‘‘Also spoken in:’’ 以下が第一国以外に関する情報である．

各属性ページに，第一国に関する言語の属性情報は必ず記載されている．また，図 4.6 から分かるように，属性情報の項目として，人口 (population)，地域 (region)，方言 (dialects)，系統分類 (classification) などがある．また，図 4.6 には現れていないが，言語によっては別名リスト (alternate names) という項目もある． 4.2 でも述べたように，別名は必ずしもすべての言語に付いているわけではない．これに対し，系統分類 (classification) の項目は必ず表示される項目であり，この後に述べる語族ページにリンクされている．また，系統分類 (classification) の項目は第一国のみにある項目である．

一方，語族別目次ページには，すべての語族の語族名がアルファベット順に配置されていて，各々の語族名にはその語族の言語系統樹を表したページ (以降，語族ページと呼ぶ) がリンクされている．語族ページにある言語系統樹のノードは，(i) 言語グループ，(ii) 言語，という 2 種類に分けて区別できるように表現されている．

そのノードのテキストの形式および特徴を表 4.1 に示す．それによりノードのタイプが言語グループか，それとも言語かの識別ができる．図 4.3 からも分かるように，言語グループは本来必ず子をもつ．しかし，語族ページは横スクロールしないように画面設計されているため，各々の語族ページにおいて，言語系統樹の木が高い場合には，言語グループの下位レベルの情報はさらにリンクしている下位の語族

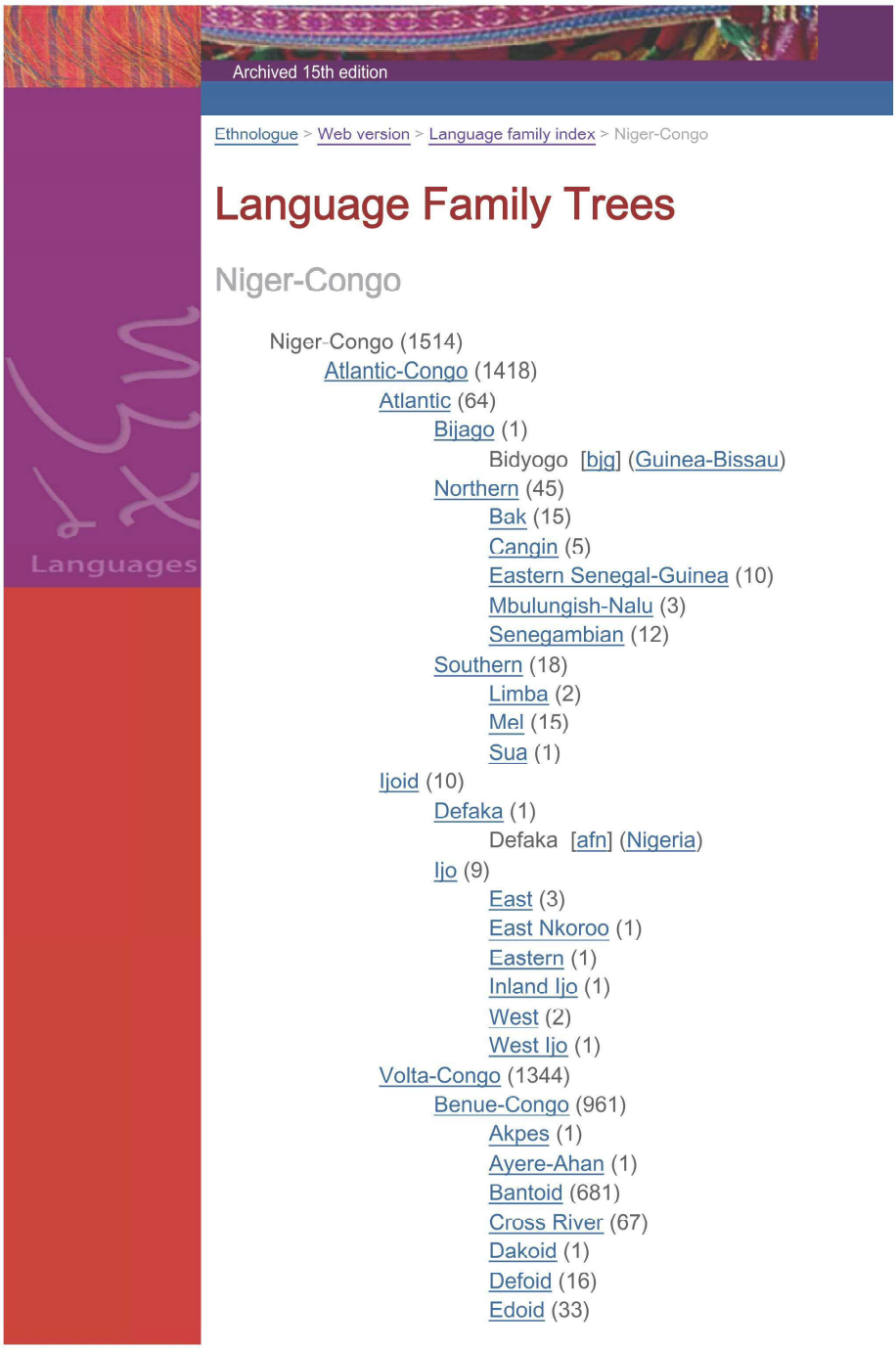


図 4.7: 語族ページ（*Ethnologue* 第 15 版 Web サイト [URLa] より引用）

表 4.1: 言語系統樹のノードのテキストの形式と特徴

ノードタイプ	ノードのテキスト	ノードのテキストの特徴
言語 グループ	言語グループ名（下位レベルに含まれている言語の数）[注1] 例：Afro-Asiatic(375)	a. 最後が必ず右丸括弧「)」 b. 左丸括弧「(」が必ず1つ以上含まれる c. 左丸括弧「(」と右丸括弧「)」の間は必ず数字
言語	言語名 言語コード(第一国の国名)[注1] 例：Awjilah [auj](Libya)	丸括弧 ( ) と鍵括弧 [ ] 自体は不変要素であり，必ずそれぞれ1組以上含まれる

注1：下線付き部分はいずれも可変要素であることを指す．

ページに含まれていることがある．例として，図 4.7 に示している語族 Niger-Congo は言語数が多く，階層が深い．図中の

Bak (15)

Cangin (5)

Eastern Senegal-Guinea (10)

Mbulungish-Nalu (3)

Senegambian (12)

の部分がこのケースに該当する．丸括弧の中の数字は下位の言語数を指している．つまり，これらのノードはこのページにおいてリーフノードの言語に見えるが，実際はインナーノードの言語グループである．ある言語グループが見掛け上リーフノードになっている場合は，必ずリンク先が設定されているため，リンクを辿っていけば，各々の語族の全構成要素を表示することができる．

(2) Web ページの HTML ソース解析

Web サイトの関連ページの URL にいずれも .asp が含まれ，また各々のページが HTML 形式 [アン 02, ノマド 00] であることから，このサイトの作成には Microsoft 社の ASP 技術 [URLd] が使われていることがわかる．ページの情報の表示形式やリンク先の設定などに規則性があると考え，ページのソースを解析したところ，ページの構成および HTML タグ [アン 02, ノマド 00] の使用に関し次の特徴があることが明らかになった．

(i) 言語別目次ページと属性ページの特徴

A) 言語名別目次ページにある A から Z までのアルファベットのリンク先 URL は“···/language_index.asp?letter=_”の形式になっている．ここで，_ は 1 桁のアルファベットを意味する．2 段階目の目次ページも類似の構造となっているため，説明は省略する．

B) 属性ページにある言語名 A，第一国の国名 B，言語コード zzz はそれぞれ“$< H1 >$A$< /H1 >$”，“$< h2 >$A language of$< A\ HREF =$‘···’ $>$B$< /A >< /h2 >$”，“$< p >$ ISO639-3: $< a\ href =$‘···’$>$ zzz$< /a >< /p >$” の文字列を分割することにより取得できる．

C) 第一国における属性情報はタグ $< TABLE >< /TABLE >$ にまとめられている．1 行 (タグ$< TR >\cdots< /TR >$ が囲んでいる部分) が 1 項目の情報を表わしていて，左列 (タグ $< TD > \cdots < /TD >$ が囲んでいる部分) が項目名 (例として，別名リスト: alternate names) で，右列がその内容となっている．

D) 第一国以外の国の有無は“$< H3 >$Also spoken in:$< /H3 >$” 要素の有無により判断できる．有の場合は，その国名 C は$< h4 >< A\ HREF =$‘···’$>$C $< /$A$>< /h4 >$ の文字列を分割することにより取得できる．他の属性情報は第一国と同様に表形式になっている．

(ii) 語族別目次ページと語族ページの特徴

A) 語族別目次ページにある各々の語族名にリンク設定されている各語族ページの URL は“···/show_family.asp?subid=XXXXX}”の形式になっている．ここで XXXXX は 5 桁の数字を意味しており，語族ページ番号となる．

B) 語族ページの言語系統樹のノードのテキストは $<dt><\!/dt>$ を用いて表現されていて，$<dt>\cdots<\!/dt>$ 要素を分割することにより取得できる (ノードのタイプの識別については 表 4.1 を参照されたい).

C) 語族ページの言語系統樹の階層構造は $<dl><\!/dl>$ を用いて表現されている．言語系統樹のノードのレベルは “ノードのレベル = ノードのテキストを表現している $<dt>\cdots<\!/dt>$ 要素を囲んでいる $<dl><\!/dl>$ の繰り返しの数 + 1” によって取得できる．

(3) データ取得処理の流れと結果

言語属性情報と言語系統情報の取得処理の全体の流れをそれぞれ図 4.8 と図 4.9 に示す．流れ図の中の定義済処理については図の下にその処理概要を示している．両処理はともに MS-Excel (97-2003 ブック形式) VBA マクロより実装し，取得情報を Excel ブックに出力した [鍛治 07, 西沢 07, プロジェクト 03, 土屋 06] .

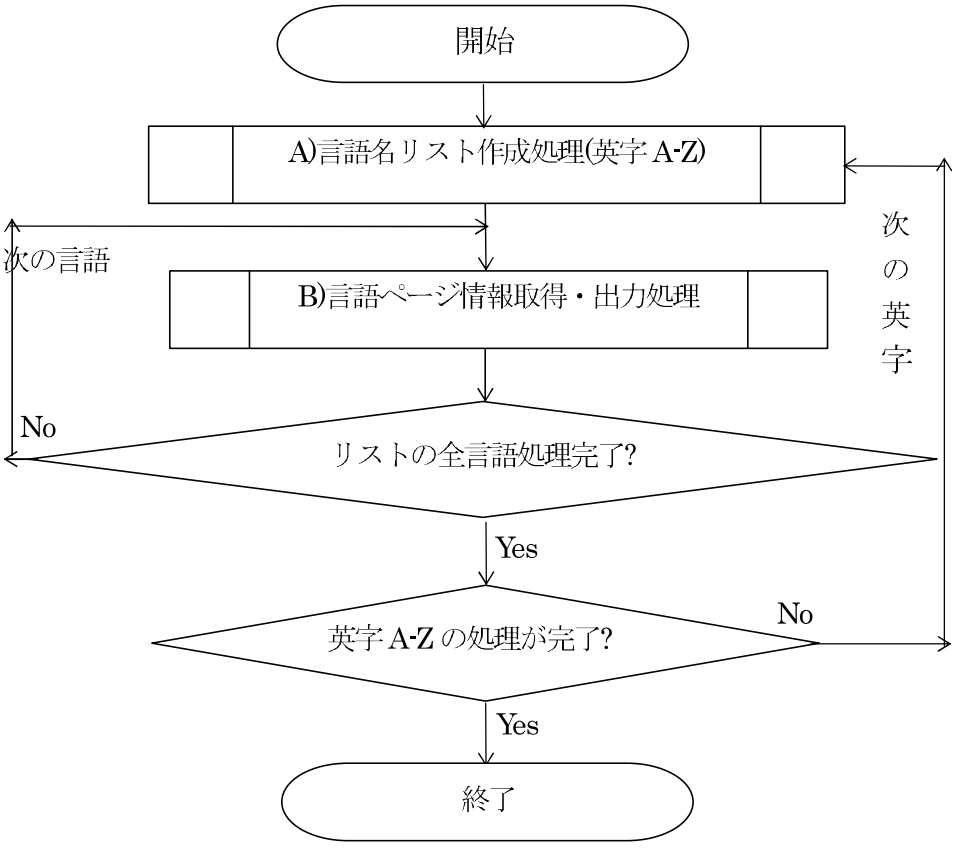
言語属性情報に関しては，(i) すべての言語コードを 1 つのシートに書き込む．(ii) すべての言語コードにつき，リンクしているページにアクセスし，そこに表示されている属性情報を 1 レコードとして書き込んでいる（図 4.10）.

言語系統情報に関しては，1 つのシートに 1 つの語族を書き込むことにし，シート名はその語族で名前を付けている（図 4.11）．子である下位のノードは親ノードより 1 列右に書き込んでいる．

(4) XML 形式データへの変換処理と結果

図 4.11 に示すように，言語系統情報は 1 シートが 1 つの語族になっている．ルート（タグ名 $<world\_languages><\!/world\_languages>$）だけを配置した XML ファイルを予め生成しておき，図 4.11 の 1 シートずつの情報をルートの子として挿入していく．処理はトップダウン的に行う．XML 形式に変換した後のデータを XML Notepad というツール [打越 08] を用いて表示したイメージを図 4.12 に示す．この図からも分かるように，ここでは言語ノード $<language><\!/language>$）に別名がまだ含まれていない．最後の処理として，図 4.12 に示している XML 形式の木構造全体に

含まれているすべての言語ノード（$< language >< /language >$）について，その言語ノードの言語コード（$< language >< /language >$ ノードの $iso639\_3\_code$ という属性）を読み取り，図 4.10 に示しているデータから，言語コードと国が一致するレコードを見つけ出し，そのレコードに含まれている別名の情報（別名リスト）を取得する．次に，この別名に関する情報を，さきほどの言語ノードの別名リストの属性（$< language >< /language >$ ノードの $alternate\_name$ という属性）として挿入する．このように，図 4.12 のすべての言語ノードについて処理し，最後に得られたデータが $T_{S_{XML}}$（図 4.13）となる．



A) 言語名リスト作成処理 (アルファベット A ～ Z)

言語名の先頭 1 文字が A から Z までのアルファベットに対応する言語のリストを作成する。

B) 属性ページ情報取得・出力処理

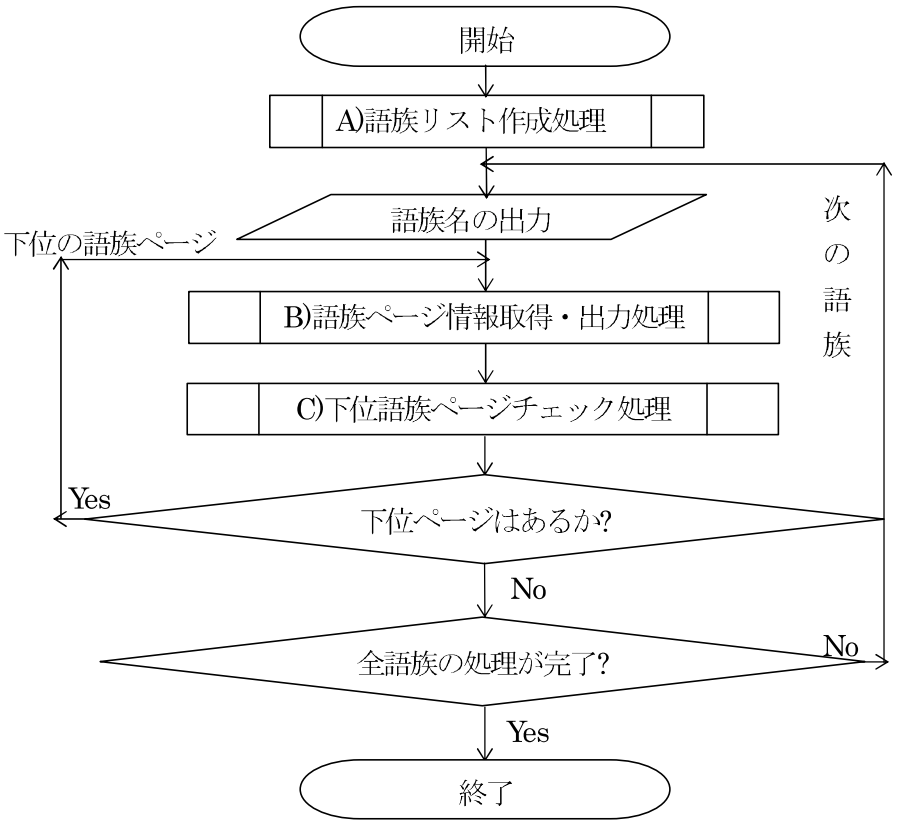
各々の属性ページにおける処理は次の流れで情報を取得した後，出力する。

a) 言語名，第一国の国名，言語コードを取得する。

b) 第一国における情報は 1 つ目の<*TABEL*>･･･</*TABLE*>要素に含まれている。<*TD*>･･･</*TD*>要素の中に含まれている項目名を検索し，あればその内容を取得する。

c) 第一国以外の国があった場合は，まずその国名を取得し，他の情報は第一国と同様に取得する。

図 4.8: 言語属性情報取得処理の流れ



A) 語族リスト作成処理

語族別目次ページを保存し，語族名および対応するリンク先ページ番号のリストを作成する。

B) 語族ページ情報取得・出力処理

各々の語族について，その語族ページにアクセスし，ソースを取得する。すべてのノードのテキストとレベルを出力する。

C) 下位語族ページチェック処理

表 4.1 に従ってノード N_i, N_{i+1} (ただし, i はノードの出現順番であり, $i = 1, 2, \cdots$) が言語グループであるかどうかを判定し，(i) N_i と N_{i+1} がともに言語グループである，(ii) $L_i = L_i + 1$ または $L_i > L_i + 1$ (ただし，L_i はノード N_i のレベル,・・・)，との 2 つの条件をともに満たした場合は下位ページがあると判定する。

図 4.9: 言語系統情報取得処理の流れ

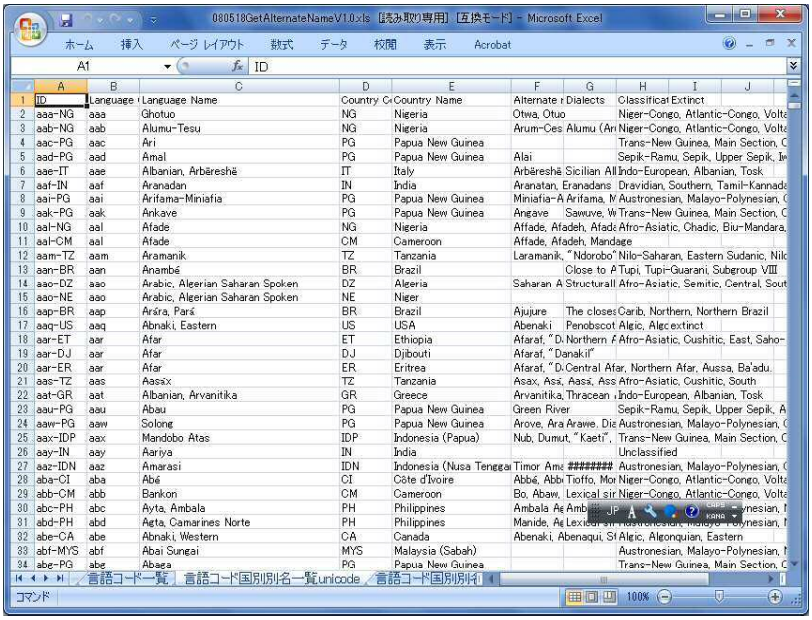


図 4.10: 言語属性情報出力イメージ

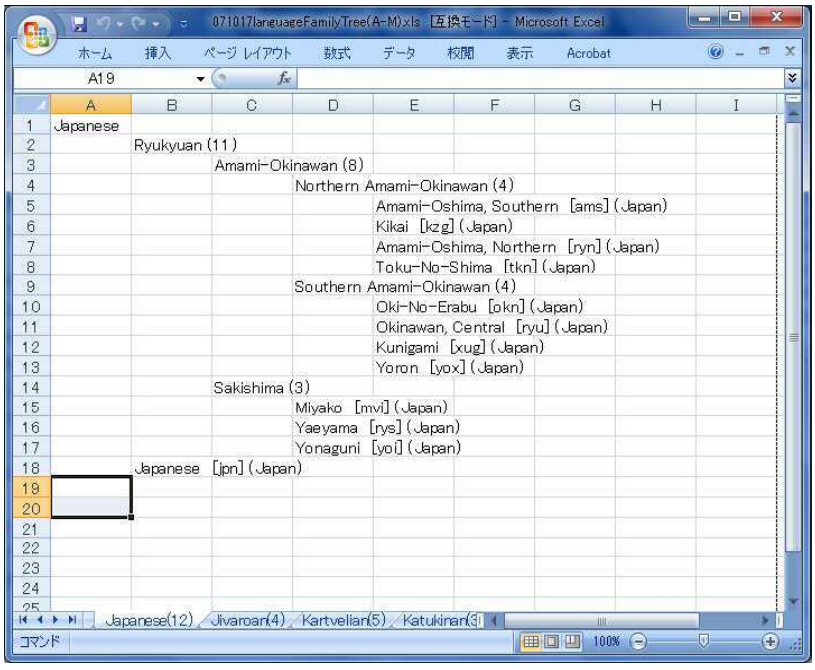


図 4.11: 言語系統情報出力イメージ

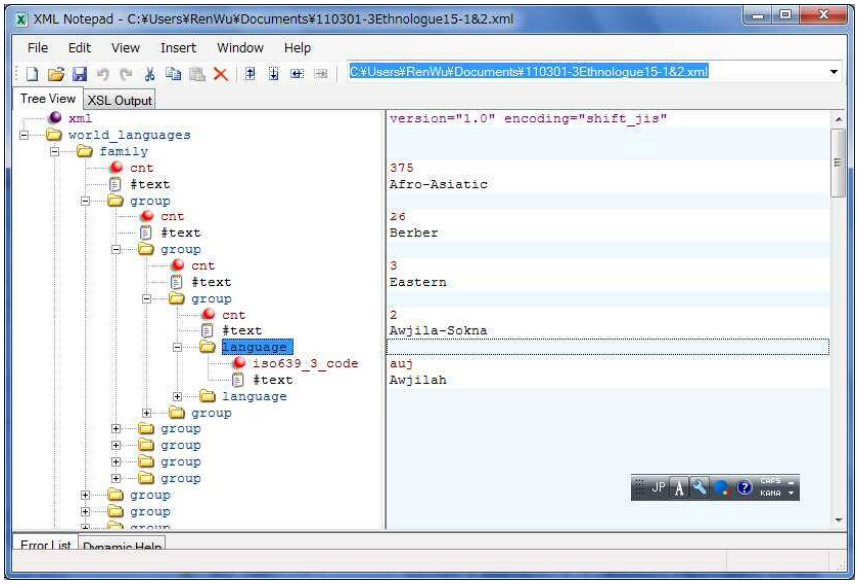


図 4.12: XML 形式に変換した言語系統情報

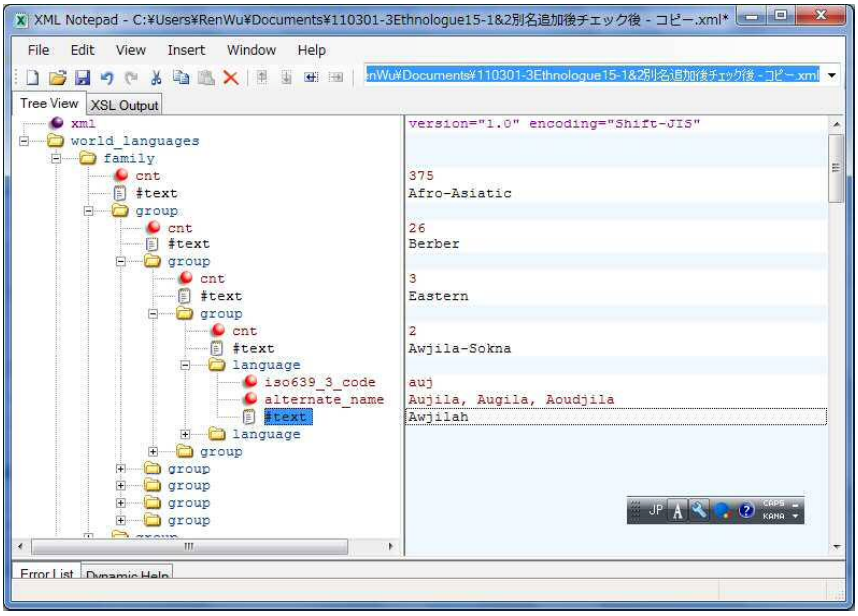


図 4.13: T_{SXML}

第5章 手法I: 木構造と文字列類似度に基づく手法

5.1 はじめに

本章では，4.2 で定義した言語系統木に基づき，まず 2 つの言語系統木 T_Y と T_S に含まれる，ゆれのない同一言語ペアを検出する方法を提案する．

次において，この方法では解決できない 2.2 (e) で述べたような言語名が類似しているケース（言語系統分類は一致または不一致）を取り扱うために，言語名の類似度という概念を導入する．文字列の類似度計算の手法はいろいろ考えられるが，本研究では，Monge-Elkan 法に基づいて，言語名の類似度を定義する．

言語名と同様に，言語の系統分類も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（ここでは，語族名なども含む広義の言語名を指す）表記がともなうため，表記ゆれなど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，言語の系統分類の類似度についても定量化を行う．そして，言語の系統分類は言語名に次ぐ有益な情報として利用する．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．さらに，これらの類似度に基づく言語の同一性判定のルールを定め，2 つの言語系統木に含まれる，ゆれのある同一言語ペアを見つけ出す手法を提案する．

以下，5.2 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．5.3 では言語名の類似度と言語系統分類の類似度について定義を行い，2 つの言語について，それらの言語名の類似度と言語系統分類の類似度の計算法について述べる．5.4 では，2 つの言語系統木からゆれのある同一言語ペアを見つけ出す手法について述べる．その後，5.2 で述べた完全一致言語の検出法を含めた同一言語ペアの検出処理の全体の流れについて述べる．5.5 では，閾値 α と β の値設定について説明したうえで，4.3 で構築したデータを用いた実験結果を提示する．さらに本章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 5.6 で本章をまとめる。

5.2 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，言語系統分類が一致し，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．言語系統木 T の言語はリーフノードにあたり，言語系統分類はパス，言語名はノードラベルなどで表せる．

定義 5.1. T_Y と T_S は 2 つの異なる言語系統木であり，言語 y と言語 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) $\mathcal{L}_{x_i}{=}\mathcal{L}_{y_i}$（$i{=}1,2,\cdots,k$）が成立つとき y と s は**言語系統分類一致**といい，$\mathcal{P}(y){=}\mathcal{P}(s)$ で表わす．

(2) $\mathcal{L}_y{=}\mathcal{L}_s$ または $\mathcal{L}_y{\in}A_s$ が成立つとき，すなわち y と s のノードラベルが一致するか，または y のノードラベルが s の別名リストに含まれるとき，y と s は**言語名一致**という．

(3) y と s が言語系統分類一致で，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**と呼ぶ． □

$y \in V_{leaf}(T_Y)$ に対し，$\mathcal{P}(y)$ と言語系統分類一致の $\mathcal{P}(s)$ 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 $\mathcal{L}_y \in A_s$ を満たす s を見つければよい．T_Y と T_S が順序木（定義 4.1 (3)，定義 4.2 (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言語系統分類が一致ではなく，類似しているにすぎない場合は，検出されない．これ以降この問題を解決するための方法について述べる．

5.3 言語名の類似度と言語系統分類の類似度

5.3.1 Monge-Elkan 法の非対称性の解消について

定義 4.2 (1)(i) で定めているように，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．例として，T_Y と T_S のそれぞれに含まれる 2 つの言語の言語名 $\mathcal{L}_1^{T_Y}=\{\text{CHINANTECO, LALANA}\}$，$\mathcal{L}_1^{T_S}=\{\text{Chinantec, Lalana}\}$ を考える．言語名 $\mathcal{L}_1^{T_Y}$ と $\mathcal{L}_1^{T_S}$ の文字列の特徴として，カンマ (,) や空白などの区切り記号によって分割可能な複数の部分文字列から構成されている．これは 3.3.3 で述べた Monge-Elkan 法が処理対象としている 2 つの文字列の構成と同じである．つまり，言語名の類似度の定義に際し，類似度計算を重畳的に定義する Monge-Elkan 法が応用できそうである．しかし，図 3.13 に示している Monge-Elkan 法の非対称性が言語名の類似度計算には妥当性を欠くと思われる．そこで本研究では，Monge-Elkan 法の類似度計算の重畳構造を取り入れ，その非対称性の問題については改善を行い，そのうえで言語名の類似度を定義する．

言語名の類似度の計算を次の 2 つのステップに分けて行う．(1) $\mathcal{L}_1^{T_Y}$ に含まれる語と $\mathcal{L}_1^{T_S}$ に含まれる語との間の語類似度を計算する．これは Monge-Elkan 法の式 (3.4) の

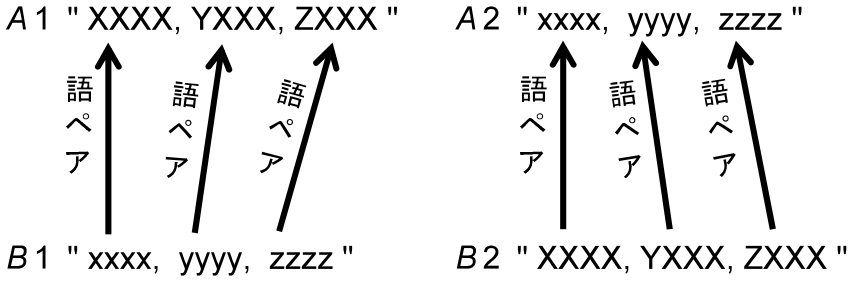


図 5.1: 語ペアの導入による Monge-Elkan 法の非対称性の解消

なかの $match(A_i, B_j)$ に相当する．(2) (1) で計算された語類似度により，言語名の類似度を計算する．

ステップ (1) では，Monge-Elkan 法の $match(A_i, B_j)$ に相当する語類似度の計算手法としては，編集距離を用いる．文字列類似度計算の基本的な手法は数多くあるが，どの手法が本研究の言語名の類似度により適合しているかが未知であるため，もっとも広く使われている編集距離による方法を用いることにする．

また，ステップ (2) の言語名の類似度の計算にあたって，Monge-Elkan 法の欠点である非対称性を克服するため，語ペアという概念を導入する．Monge-Elkan 法の非対称性に対し，語ペア導入の意図する効果について，図 3.13 の例で説明する．

図 3.13 では，2 つの文字列 $A1$ と $B1$ およびそれらの内容を逆にした 2 つの文字列 $A2$ と $B2$ がある．$A1$ の各部分文字列に対し，文字列の類似度が最大となる $B1$ の部分文字列が集中することがある．つまり，$A1$ の 3 つの部分文字列 XXXX, YXXX, ZXXX のいずれに対しても，$B1$ の 3 つの部分文字列のなかの xxxx が類似度が最大となっている．語ペアを導入することで，図 5.1 に示すように，$A1$ の部分文字列 XXXX, YXXX, ZXXX 対し，それぞれ $B1$ の部分文字列 xxxx, yyyy, zzzz が類似度が最大となり，また $A1$ と $B1$ の内容を逆にした $A2$ と $B2$ の場合も，この対応関係は変わらない．ここで，$A1_1$ =XXXX と $B1_1$ =xxxx，$A1_2$ =YXXX と $B1_2$ =yyyy，$\cdots$, のような部分文字列の組合せのことを語ペアと呼ぶ．

本研究では，語ペアを見つける手法を提案し，Monge-Elkan 法の重畳構造に基づく言語名の類似度の計算に適用することにより，Monge-Elkan 法の非対称性を解消することを狙う．

5.3.2 言語名の類似度

(1) 語類似度

3.3.1 で説明した編集距離は，図 3.3 に示しているように，置換，挿入，削除という 3 つの編集操作のコストをいずれも 1 とした下で，片方の文字列からもう片方の文字列に変形するために必要最少の編集操作の回数とされている．

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 5.2. 2 つの語 v と w の**語類似度** $sd\_w(v,w)$ は，以下の式で算出される値である．

$$sd\_w(v,w) = \frac{l_A(v,w) - ed(v,w)}{l_A(v,w)} \tag{5.1}$$

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． □

前に述べた例について計算すると，(i) CHINANTECO と Chinantec については，l_A=9, ed=1，sd_w=8/9 となり．(ii) PEI と Bei について，l_A=3, ed=1，sd_w=2/3 となる．

(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 (5.1) 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**と呼ぶことにする．

2 つの言語名の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 $\mathcal{L}_1$=$\{v_1, v_2, \cdots, v_m\}$ と $\mathcal{L}_2$=$\{w_1, w_2, \cdots, w_n\}$ $(m \geq n)$ の類似度 $sd\_ln(\mathcal{L}_1, \mathcal{L}_2)$ を次のように定義する．

定義 5.3. $\mathcal{L}_1$=$\{v_1, v_2, \cdots, v_m\}$ 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 w_i'=$NULL$ である．言語名 $\mathcal{L}_1$ と $\mathcal{L}_2$ の**言語名の類似度** $sd\_ln(\mathcal{L}_1, \mathcal{L}_2)$ は以下の式で算出される値である．

$$sd\_ln(\mathcal{L}_1, \mathcal{L}_2) = \frac{\sum_{i=1}^{m} sd\_w(v_i, w_i')}{m} \tag{5.2}$$

□

$\mathcal{L}_1^{T_Y}$=$\{$CHINANTECO, LALANA$\}$ と $\mathcal{L}_1^{T_S}$=$\{$Chinantec, Lalana$\}$ の例では，sd_ln $(\mathcal{L}_1^{T_Y}, \mathcal{L}_1^{T_S})$=$\frac{1+0.88}{2}$=0.94 となる．

5.3.3 言語系統分類の類似度

言語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（4.2 を参照されたい）．

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 5.2 に示すように，$\mathcal{P}(y)$={Algic}{Algonquian}{Algonquian, Proper} {Arapaho} となる．言語 s のパスは $\mathcal{P}(s)$={Algic}{Algonquian}{Plains}{Arapaho} である．

$\mathcal{P}(y)$ 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 5.2 (A) 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 5.3.2 (2) で説明した言語名の類似度の計算法を用いる．図 5.2 (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 $\mathcal{F}(\mathcal{P}(y), \mathcal{P}(s)) \rightarrow$(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，これは 5.3.2 (1) で説明した語類似度の計算法とは異なる．

図 5.2 (B) の上半分の line 1, 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．すなわち，レベル 1，レベル 2，レベル 4 のノードラベルは一致しているが，レベル 3 の {Algonquian,

Proper} と {Plains} は不一致である．つまり，これは $\mathcal{P}(y)$ の {Algonquian, Proper} から $\mathcal{P}(s)$ の {Plains} への置換があった，としている．それに対し，図 5.2 (B) の下半分の line 3, 4 では別の見方をしており，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，としている．我々は，言語系統分類の比較は図 5.2 (B) の line 1, 2 に示している方が妥当であると考える．

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，3.3.2 で説明したように最長共通部分列 (LCS) を求める．LCS を求めるためには，置換は考慮しないため，図 5.2 (B) の line 3, 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) 言語系統分類の類似度

言語 y と言語 s のパス $\mathcal{P}(y)$ と $\mathcal{P}(s)$ に対し，$\mathcal{F}(\mathcal{P}(y), \mathcal{P}(s)) \rightarrow (v, w)$ の変換処理を行い，それぞれ 2 つの文字列 v と w に変換する．v と w は言語 y と言語 s の系統分類の比較を行うための文字列となっている．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），動的計画法による LCS 問題を解決する手法にしたがって，v から w に変形すると，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．また，後述する系統分類の類似度の定義からわかるように，このような限定を加えても，系統分類の類似度の値に影響を及ぼすことはない．

(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

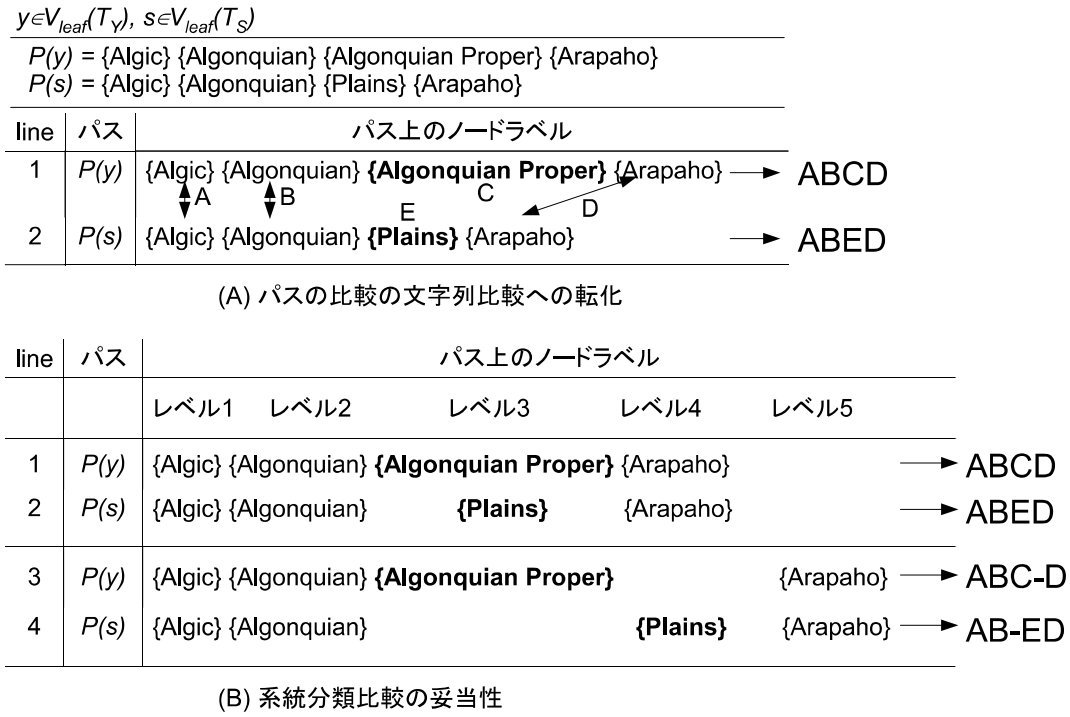


図 5.2: 言語系統分類の比較

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，$**+-*$ となる．

(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のとりうる様相を推定し，視覚的に捉えることもできる．なお，$SFSLC$ の求め方は上記提案した手続きによるほか，例えば，置換コストを 2 として，アラインメントを求めるトレースバック時には，置換を挿入・削除より優先してたどる方法によっても得ることができる．

言語 $y \in V_{leaf}(T_Y)$ と $s \in V_{leaf}(T_S)$ の系統分類の類似度 $sd\_lc(y,s)$ を次に定義する．

定義 5.4. 言語 $y \in V_{leaf}(T_Y)$ と $s \in V_{leaf}(T_S)$ の**言語系統分類の類似度** $sd\_lc(y,s)$ は，以下の式で算出される値である．

$$sd\_lc(y,s) = \frac{l_{LCS}(v,w)}{l_{SFSLC}(v,w)} \tag{5.3}$$

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． □

図 5.2 の例では，v=ABCD, w=ABED, $l_{LCS}(v,w)$=3, $SFSLC(v,w)$=$**X*$, $l_{SFSLC}(v,w)$=4, $sd\_lc(y,s)$=3/4 となる．

5.4 同一言語ペアの検出

5.4.1 ゆれのある同一言語ペアの検出

5.2 で述べた完全一致言語の検出方法では，言語名のゆれと系統分類のゆれには対応できない．つまりこの方法では，$y \in V_{leaf}(T_Y)$ と $s \in V_{leaf}(T_S)$ の 2 つの言語が本来同一言語であっても，言語名または系統分類にゆれがあり，あるいはこれらの両方にゆれがある場合は，(y,s) を同一言語ペアとして検出することはできない．

ここでは，y と s の言語名と系統分類がともに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るならば，y と s を同一言語と判定する方法について

述べる．式 (5.2) にしたがって言語名の類似度 $sd\_ln(\mathcal{L}_y, \mathcal{L}_s)$ （または $sd\_ln(\mathcal{L}_y, \mathcal{A}_i^s)$, $\mathcal{A}_i^s$ は s の（複数の）別名の中の任意の 1 つの別名である）の最大値, また式 (5.3) にしたがって言語系統分類の類似度 $sd\_lc(y, s)$ の最大値を計算し，それらの最大値がそれぞれ閾値 α と閾値 β を超えるならば，(y, s) を同一言語ペアとして検出する．閾値 α と β はあらかじめ値設定をしておく必要がある．閾値 α と β の値の決め方については 5.5.1 で議論する．

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．全ての $y \in V_{leaf}(T_Y)$ に対し，$sd\_ln_{max}(\mathcal{L}_y) \leq \alpha$ 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) (1) で得られた，y との言語名の類似度の最大値 $sd\_ln_{max}(\mathcal{L}_y)$ をもつ T_S の言語は複数得られる可能性がある．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（5.3.3 (1) を参照されたい）．

(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．

以上の処理のアルゴリズムを図 5.3 に示す．

アルゴリズム：FSLV

入力：$y \in V_{leaf}(T_Y)$, T_S, α, β
出力：y の同一言語ペア SLP
手法：
0° $SLP \leftarrow \phi$
1° $S^y \leftarrow \phi$, 次の (i) ∼ (iv) を行う.

(i) すべての $s \in V_{leaf}(T_S)$ に対し，$\mathcal{L}'_s \leftarrow \mathcal{L}_s \cup \mathcal{A}_1^s \cup \mathcal{A}_2^s \cup \cdots$ とする.

(ii) $L' \leftarrow \{\mathcal{L}'_s | s \in V_{leaf}(T_S)\}$ とする.

(iii) $sd\_ln_{max}(\mathcal{L}_y) = \max\{sd\_ln(\mathcal{L}_y, \mathcal{L}) | \mathcal{L} \in L'\}$ を計算する.

(iv) $S_y \leftarrow \{s | sd\_ln(\mathcal{L}_y, \mathcal{L}'_s) = sd\_ln_{max}(\mathcal{L}_y), sd\_ln_{max}(\mathcal{L}_y) > \alpha\}$ とする.

2° 次の (i) ∼ (iii) を行う.

(i) $sd\_lc_{max} = \max\{sd\_lc(y, s) | s \in S_y\}$ を計算する.

(ii) $SLP \leftarrow \{(y, s) | sd\_lc(y, s) = sd\_lc_{max}, sd\_lc_{max} > \beta\}$ とする.

(iii) $|SLP| > 1$ ならば，$SLP \leftarrow \phi$ とする.

3° SLP の要素を出力し，停止する.

図 5.3: アルゴリズム：FSLV

5.4.2 同一言語ペア検出処理全体の流れ

T_Y と T_S の 2 つの言語系統木に含まれる同一言語ペアの検出は次に処理 I と処理 II の 2 つの手順に分けて行う.

[処理 I]（完全一致言語の検出）

すべての $y \in V_{leaf}(T_Y)$ に対し，次の処理を行う.

Step1

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.

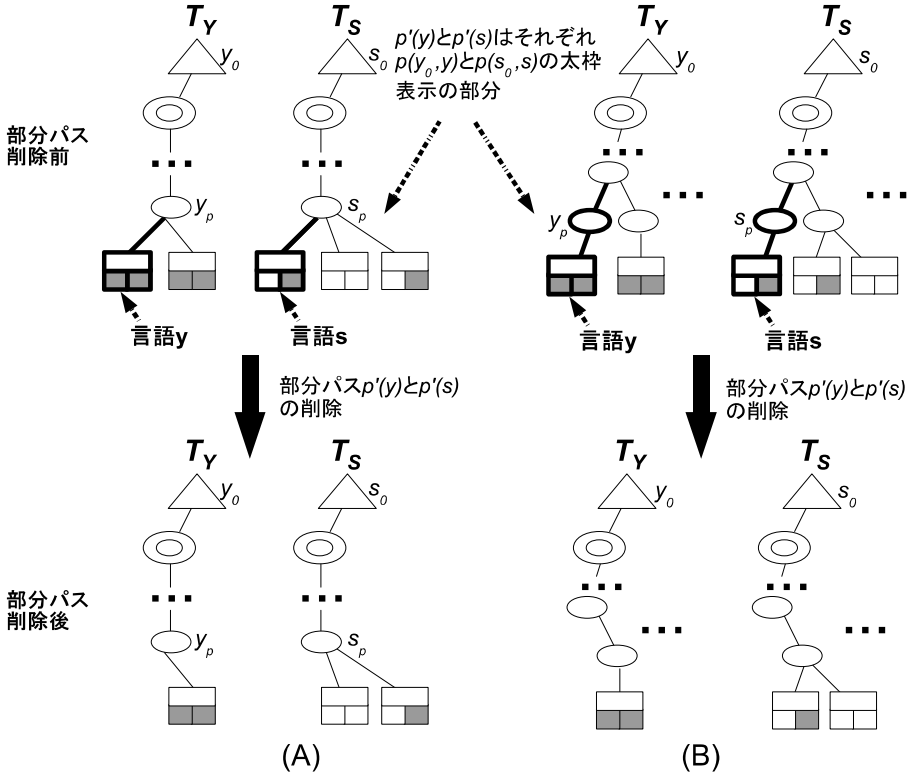


図 5.4: 部分パス削除の例

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 $\mathcal{L}_y{\in}A_s$ 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

完全一致言語として検出された同一言語ペア (y, s) について，パス $p(y_0, y)$ $(y_0{=}r(T_Y))$ とパス $p(s_0, s)(s_0{=}r(T_S))$ のそれぞれの部分パスである $p'(y)$ と $p'(s)$ を削除し，T_Y と T_S を更新する．y の部分パス $p'(y)$ は次の (i) と (ii) を満たす $p(y_0, y)$ の最も長い部分パスである．(i) 部分パス $p'(y)$ は $(y_p, y){\in}E(T_Y)$ を含む．(ii) 部分パス $p'(y)$ 上のノードの子の数は 1 つである（ただし，ノード y の

場合は子の数は 0）．s の部分パス $p'(s)$ についても同様に定められる．図 5.4 に (A) と (B) の 2 つの例を示す．

図 5.4 からもわかるように，このような操作によって削除される部分パス上のノードには 2 つ以上の子をもつノードは含まれていないことから，削除後に他の言語の系統分類の類似度が変化してしまうことはない．

[処理 II]（ゆれのある同一言語ペアの検出）

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，アルゴリズム FSLV（図 5.3）を実行し，得られた SLP を同一言語ペアとして出力する．

Step2

同一言語と判定した $y \in V_{leaf}(T_Y)$ と $s \in V_{leaf}(T_S)$ について，処理 I Step3 と同様にして，T_Y と T_S を更新する．

5.5 実験結果および考察

5.5.1 閾値 α と β の値設定

5.4.1 で述べたゆれのある同一言語ペアを検出するための処理（図 5.3）では，2 つの言語系統木 T_Y と T_S の木構造データ以外に，入力値として言語名の類似度の閾値 α と言語系統分類の類似度の閾値 β の値を設定する必要がある．言語系統木データについては，そのデータ形式および生成などについて，4.3 においてすでに述べたが，ここでは閾値 α と β の値設定について述べる．

閾値 α と β の値設定は，次のような考え方に基づいて行われる．閾値 α と β はともに設定値が小さいほど検出される同一言語ペアの数が増える一方，誤判定（異な

る言語を同一言語と判定すること）の言語ペアの数も増える．それはすなわち，処理結果の信頼性の低下を意味することである．閾値の最適な値は，検出される同一言語ペアの数とその中に含まれる誤判定の言語ペアの数を総合評価したうえで，設定すべきと考えられる．T_Y と T_S のそれぞれに含まれる言語の同一性判定という本研究の目的に照らして考えると，それは T_Y と T_S のそれぞれが対応している元の 2 つの表形式の言語データ Yamamoto-Data と SilGIS-Data のマッチングをとることである．SilGIS-Data を媒介とし，Yamamoto-Data の言語の地理的位置情報を取得するなど，本研究の成果を手段として言語研究に使えるようにするためである．このことから，なるべく多くの同一言語ペアを検出することは目的の一つではあるが，信頼性の低下がもたらす弊害が直接言語研究に及ぶかもしれないため，同時に誤判定の言語ペアの数をなるべく抑えることも重要である．検出される同一言語ペアの数を多くすること，と誤判定の言語ペアの数を少なくすることはトレード・オフの関係にある．ここで，誤判定の言語ペアの数をなるべく少なくすることを第一要件とする．このような価値判断のもとで，良い結果をもたらすと思われる閾値 α と β の値を見つけるための実験を行った．

この閾値 α と β の値を決めるための実験のデータとしては，4.3 で述べた 2 つの XML 形式の木構造データ $T_{Y_{XML}}$ と $T_{S_{XML}}$ を使った．処理は図 5.3 のアルゴリズムにしたがって行い，まず，閾値 β をある一定の値に設定したうえで，閾値 α の値を変化させたときの検出同一言語ペア数と誤判定の言語ペア数を集計し，閾値 α の値を決める．次に，逆に閾値 α を既定値に設定したうえで，閾値 β の値を変化させたときの検出同一言語ペア数と誤判定の言語ペア数を集計し，閾値 β の値を決める．言語名の類似度の閾値 α と言語系統分類の類似度の閾値 β の値設定についての実験の経過および結果をそれぞれ表 5.1 (A) と表 5.1 (B) に示し，次において詳細に説明していく．

(A) 言語名の類似度の閾値 α の値設定

まずは β=0，α=0.65 に設定して実験を行った．言語系統分類の類似度の閾値を β=0 という値から出発したのは，次のような理由によるものである．言語系統分類を比較するときに，T_Y と T_S のそれぞれに含まれる 2 つの言語のパス上において，同

表 5.1: 閾値 α と β の値設定に関する実験

(A) α の値設定に関する実験結果(β =0)

実験	1	2	3	4
閾値 α の設定値	0.65	0.7	0.75	0.74
検出同一言語ペア数	1,541	1,512	**1,492**	1,503
誤判定の言語ペア数	23	12	**0**	3

(B) β の値設定に関する実験結果(α =0.75)

実験	1	2
閾値 β の設定値	0.1	0.15
検出同一言語ペア数	1,490	1,488
誤判定の言語ペア数	0	0

じノードラベルが 1 つしかない場合でも，この 1 つのノードラベルの一致をこの 2 つの言語の言語系統分類の類似性評価にポジティブに作用させるように捉える．つまり，1 つのノードラベルの一致は，1 つだけのノードラベルの一致でしかなく，偶然の一致であろうと推定することはせず，1 つのノードラベルが一致しているから，この 2 つの言語の言語系統分類はどこか類似しているところがあると考える．(5.3.3 を参照されたい) ．パスが長い場合，長いパス上の 1 つのノードラベルが一致していても，2 つの言語の言語系統分類の類似度を計算すると，言語系統分類の類似度の値は 0 に近い値になることがある．そのため，言語系統分類の類似度の値は 0 以上であれば，意味がある値と推定する．したがって，言語系統分類の類似度の閾値は最小値 $\beta = 0$ に設定して実験をスタートすることにした．β=0，α=0.65 のときの結果として，検出同一言語ペア数は 1,541，誤判定の言語ペア数は 23 であった．

次も β=0 に設定したもとで実験を行うが，ここで誤判定の言語ペア数が 23 と出ているため，さきほど述べた実験の方針にしたがい，誤判定の言語ペア数を少なくする必要があるため，閾値 α を 0.65 より大きい値に設定したうえで実験を行なってみた．α=0.7, 0.75 のときの検出同一言語ペア数と誤判定の言語ペア数は，それぞれ 1,512 と 12 および 1,492 と 0，という結果となった．ここで，α=0.75 のときの誤判定の言語ペア数が 0 となっている．さらに，β=0，α=0.74 に設定して実験したところ，検出同一言語ペア数は 1,503 に増えた一方，誤判定の言語数も 0 から 3 に増えたため，言語名の類似度の閾値は α=0.75 をもっとも良い値として採用することにした．

表 5.2: 手法 I による同一言語ペアの検出結果

処理	検出同一言語ペア数	比率 （2,869に対する比率）
処理 I	1,034	36%
処理 II	1,492	52%
合計	2,526	88%

α=0.75, β=0

(B) 言語系統分類の類似度の閾値 β の値設定

言語名の類似度の閾値 α の値設定では，β=0 と設定したうえで実験を行った．結果として，β=0，α=0.75 のときに，誤判定の言語ペア数が 0 という値が出ている．ここでは，言語系統分類の類似度の閾値を β=0 に設定することの効果を確認するため，α=0.75 と設定したうえで，β の値を変化させ，β=0.1, 0.15 のときの実験を行った．その結果を表 5.1 (B) に示す．表 5.1 (B) では，β=0.1, 0.15 に設定したときは，β=0 に設定したときと同じように誤判定は出ていないが，検出同一言語ペアの数が減り，β=0 の場合より悪い結果となっている．このことから，言語系統分類の類似度の閾値を β=0 に設定することはやはり効果があるといえる．

5.5.2 同一言語ペアの検出結果

言語名の類似度の閾値は α=0.75，系統分類の類似度の閾値は β=0 と設定したうえで，2 つの言語系統木 T_Y と T_S としては 4.3 で述べた 2 つの XML 形式の木構造データ $T_{Y_{XML}}$ と $T_{S_{XML}}$ を用いて， 5.4.2 で述べた手順にしたがって，実験を行った結果を表 5.2 に示す．表 5.2 では，処理 I と処理 II は，それぞれ完全一致言語とゆれのある同一言語ペアを検出するための処理である．

処理 I と処理 II を合わせると，Yamamoto-Data の総言語数 2,870 中の 2,526 言語（約 88% ）について，SilGIS-Data の言語との対応づけが判明できた．また，検出された 2,526 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 1,492（約 52% ）であった．さらに，完全一致言語の 1,034 言語のうち，Yamamoto-Data の第一言語名と SilGIS-Data の別名の一致による結果が 156 言語であった．ゆれのある言語の 1,492 言語のうち，言語名一致・系統分類類似，言語名類似・系統分類一致，言語名類似・系統分類類似，の言語がそれぞれ 1,367, 81, 44 個の結果となった．なお，処理 II の結果は，言語名の類似度の閾値を α=0.75，系統分類の類似度の閾値を β=0 と設定したときの値である．

5.5.3 考察

本研究では，2 つの表形式の言語データのそれぞれに含まれる言語の同一性を判定する際の情報不足を補うため，言語系統分類に関する情報を取り入れることにした．これにより，扱うデータは表形式データから木構造データに変わった．木構造データを取り入れたことにより，2.2 (d) のような言語名重複の問題は解決できるようになり，表 2.1 (B) の言語 ‘‘Bai’’ は表 2.1 (A) の No=212 の言語 ‘‘BAI’’ と No=213 の言語 ‘‘BAI’’ とのうち，No=212 の言語 ‘‘BAI’’ に対応していることが判明した．また，言語名と言語系統分類がともに類似する例として，T_Y の第一言語名が ‘‘YI, GUICHOU’’，系統分類が ‘‘Sino-Tibetan’’/ ‘‘Tibeto-Burman’’/‘‘Burmese-Lolo’’/‘‘Lolo’’/‘‘Northern’’ の言語と T_S の第一言語名が ‘‘Yi, Guizhou’’，系統分類が ‘‘Sino-Tibetan’’/‘‘Tibeto-Burman’’/‘‘Lolo-Burmese’’/‘‘Loloish’’/‘‘Northern’’/‘‘Yi’’ の言語が，言語名の類似度が 0.93，系統分類の類似度が 0.67，$SFSLC$ が $***X*-$ という結果が得られ，同一言語として検出された．

また，言語名の類似度の閾値を α=0.75，系統分類の類似度の閾値を β=0 と設定したときの結果について，判定漏れ（本来検出されるべき同一言語ペアが検出されなかったことを指す）の原因について調査したところ，次のようなケースがあった．

‘‘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，このときの類似度が 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，検出されたケースも少なくなかった．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，判定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成されており，助詞をリストアップすること自体が困難であると予想できるため，あえて例外処理を行わないことにした．

また，言語名の類似度は閾値 α=0.75 を超えたが，言語系統分類の類似度が 0 となったため，検出されなかった言語は 45 もあった．この中では，全く言語系統分類が異なる言語が多かったが，次のようなケースもあった．T_Y と T_S での系統分類はそれぞれ‘‘French-based Creole'' と ‘‘Creole''/‘‘French based'' である．2 本のパスを 2 つの文字列に変換する際のノードラベルの類似度を閾値 α=0.75 にしているため，T_Y のパス上の唯一のノードラベル {French, based, Creole} が T_S のパス上の 2 つのノードラベルの {Creole} と {French , based} のどちらとも同じノードとならず，言語系統分類の類似度は 0 になり，検出されなかった．これは，2 本のパスを 2 つの文字列に変換する際のノードラベルの類似度の閾値の設定を β と関連して考慮する必要性の検討について，示唆を与えてくれたことになる．

5.6 まとめ

本研究では，言語系統分類に関しては系統樹モデルを取り入れた．系統樹モデルでは，語族は木構造をなす．本研究では，語族のなす木構造を抽象化し，さらに世界諸言語を 1 本の木にまとめた言語系統木について定式化を行った．言語系統木の導入により，言語の同一性判定において，言語名に加えて，言語系統分類も指標として取り入れることができた．木構造に基づき，異なる言語データにおいて言語名と言語系統分類に変化がない言語（完全一致言語と呼ぶ）を検出するための手法を提案した．

次に，異なる言語データにおいて言語名あるいは言語系統分類，またはその両方

に変化がある言語（ゆれのある言語と呼ぶ）を検出するため，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，それらの類似度の評価法について定式化を行い，本来は同一言語であるが，異なる言語データに含まれ，ゆれのある同一言語ペアを検出するための手法を提案した．

さらに，言語系統木のデータ収集および構築を行い，本章で提案した手法の有効性についての実験を行った．その結果，合わせて 88% の言語の同一性が判定できた．そのうち，52% は言語名の類似度と言語系統分類の類似度の適用による結果であった．このことから，本章で提案した言語名の類似度と言語系統分類の類似度とゆれのある言語の検出手法は効果的である，といえる．

5.5.2 で述べた実験結果からわかるように，本章の提案した手法では同一性判定ができない言語がなお残っている．本章の手法で同一性が判明できない言語について調査し，さらなる考察を通してその特徴を分析し，ゆれのある同一言語ペアの検出率の向上を図りたい．

第6章 手法II: 言語名と言語系統分類の総合的尺度に基づく手法

6.1 はじめに

第5章では，言語名と言語系統分類の曖昧な性質に対して，文字列類似度に基づく言語名の類似度と言語系統分類の類似度を導入し，2つの言語系統木に含まれる同一言語を検出するための手法を提案した．この手法は成果を上げ，T_Y の約88%の言語について，T_S における同一言語を検出し，言語コードを付与することに成功した．T_Y の言語が必ず T_S にも含まれているわけではないが，残りの12%に同一言語ペアがなお含まれていることが予測できる．この木構造と文字列類似度に基づく言語の同一性判定手法は，2つの言語系統木のそれぞれに含まれる2つの言語が言語名の類似度と言語系統分類の類似度の両方ともそれぞれ閾値に関する条件をクリアしなければならない，ということを同一性判定の要件としている．しかし，なかには，両方の条件をともには満たしていないが，言語名または言語系統分類のどちらか一方が完全に一致するなど高い類似性を有しているケースもあり，このような場合にも対応する必要がある．

本章では同一言語の検出率を向上させるため，第5章で提案した木構造と文字列類似度に基づく言語の同一性判定手法を発展させる．第一に言語系統分類の類似性評価の改善を試み，兄弟情報を考慮した新しい言語系統分類の類似度を定義する．第二に言語総合類似度（language general similarity）という概念を新たに導入し，言語名の類似度と言語総合類似度に基づく言語の同一性判定の手法を提案する．

以下 6.2 では，言語名または言語系統分類の一方は完全に一致していて，その他方の類似度が閾値に関する条件をクリアできないため，結果として同一言語として

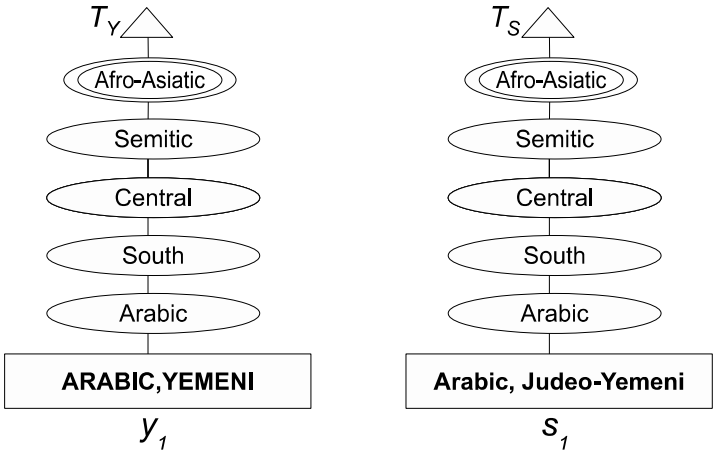
判定されないような例を挙げ，問題の原因を分析する． 6.3 では語族，親，兄弟の情報を考慮した新言語系統分類の類似度，さらに言語総合類似度について定義する．6.4 では， 6.3 で定義した類似度を用いて，T_Y の言語に対し，T_S からその同一言語を検出するための方法について述べ，アルゴリズムを与える． 6.5 では新言語系統分類の類似度と言語総合類似度の計算に必要なパラメータと閾値の値設定について述べる． 6.6 では実験結果を提示し，手法の妥当性および有用性などを考察する．最後に， 6.7 で本章をまとめる．

6.2 手法 I の問題点

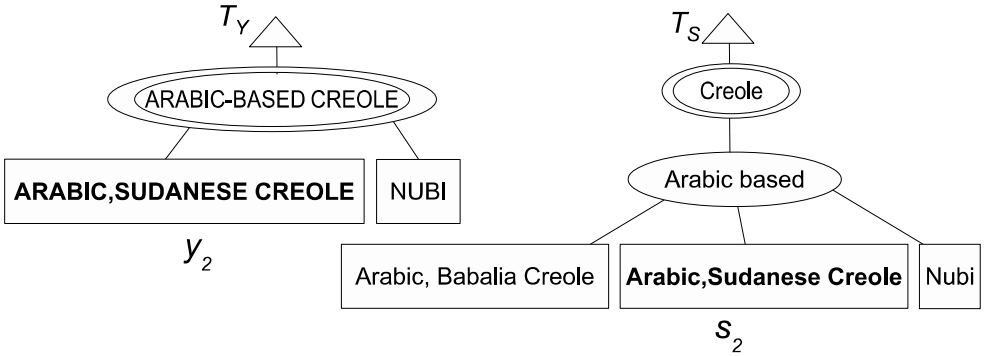
6.2.1 手法 I で言語の同一性が判定できない 3 つの例

図 6.1 に，手法 I では言語の同一性が判定できない 3 つの例を示している．この図の (1), (2), (3) に示している (y_1, s_1), (y_2, s_2), (y_3, s_3) はそれぞれ 3 つの同一言語ペアである．前に述べたように，同一言語ペアとは T_Y と T_S の両方に含まれ，異なる言語名または言語系統分類を示すが，本来は同じ言語の組合せのことをいう． 第 5 章 で提案した手法 I に基づくこの 3 つの同一言語ペアの言語名の類似度と言語系統分類の類似度はそれぞれ次のように計算される．

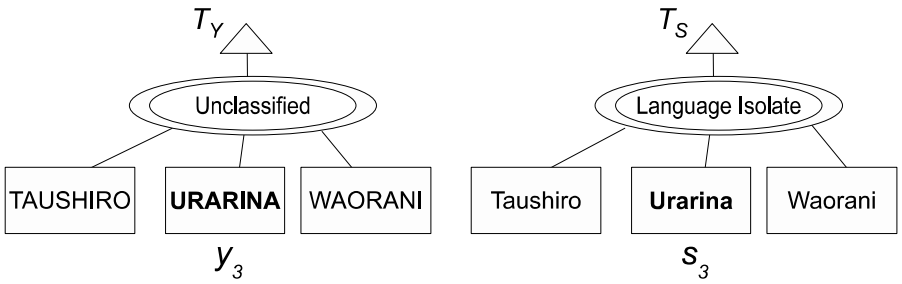
図 6.1 (1) に示す T_Y の言語 y_1 と T_S の言語 s_1 の第一言語名はそれぞれ “ARABIC, YEMENI” と “Arabic, Judeo-Yemeni” である．ここで，前述と同じように，アルファベット表記は特に大文字と小文字を区別しない．言語 $y \in V_{leaf}(T_Y)$ の言語名（第一言語名または別名を指す）を $\mathcal{L}_y^{T_Y}$ と表記するならば，y_1 と s_1 の言語名はそれぞれ，$\mathcal{L}_{y_1}^{T_Y}$={ARABIC, YEMENI}，$\mathcal{L}_{s_1}^{T_S}$={Arabic, Judeo,Yemeni} となる．そして，言語 y_1 と言語 s_1 の言語名の類似度は，$sd\_ln(\mathcal{L}_{y_1}^{T_Y}, \mathcal{L}_{s_1}^{T_S}) = \frac{1}{3}(sd\_w(\text{Arabic, ARABIC}) + sd\_w(\text{Judeo, NULL}) + sd\_w(\text{Yemeni, YEMENI})) = \frac{1}{3}(1+0+1) = 0.67$ となる（ 5.3.2 を参照されたい）．



(1) 言語系統分類は同じであるが，言語名の類似度が閾値αに達しないケースの例



(2) 言語名は同じであるが，言語系統分類の類似度が閾値βに達しないケースの例1



(3) 言語名は同じであるが，言語系統分類の類似度が閾値βに達しないケースの例2

図 6.1: 手法 I では言語の同一性が判定できない 3 つの例

図 6.1 (2) の言語 y_2 と言語 s_2 は，言語名がそれぞれ “ARABIC, SUDANESE CREOLE” と “Arabic, Sudanese Creole” で，言語名が同じであるため，言語名の類似度は $sd\_ln(\mathcal{L}_{y_2}^{T_Y}, \mathcal{L}_{s_2}^{T_S}) = 1$ となる．図 6.1 (3) の言語 y_3 と言語 s_3 は言語名がそれぞれ “URARINA” と “Urarina” で，言語 y_2 と言語 s_2 と同様に言語名が同じであるため，言語 y_3 と言語 s_3 の言語名の類似度 $sd\_ln(\mathcal{L}_{y_3}^{T_Y}, \mathcal{L}_{s_3}^{T_S}) = 1$ となる．

一方，言語系統分類は言語系統木におけるルートの子のノードから言語の親のノードまでのパスの類似度に基づいて計算される．パスはノードラベルのリストであり，ノードラベルは語の集合として定義されているため，パスは語の集合のリストで表せる（ 4.2 を参照されたい）．まず，図 6.1 (2) の言語 y_2 と言語 s_2 の言語系統分類の類似度について計算するが，それらのパスはそれぞれ “ARABIC-BASED CREOLE” と “Creole”/“Arabic based” であり，言語 $y \in V_{leaf}(T_Y)$ のパスを $\mathcal{P}_Y(y)$，言語 $s \in V_{leaf}(T_S)$ のパスを $\mathcal{P}_S(s)$ と表記するならば，y_2 と s_2 のパスはそれぞれ $\mathcal{P}_Y(y_2)$ = {ARABIC, BASED, CREOLE} と $\mathcal{P}_S(s_2)$ = {Creole}{Arabic, based} となる．

この 2 つのパスの類似度を計算するために，まずこの 2 つのパスを 2 つの文字列 v と w に変換処理を施す．変換する際，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，異なるノードラベルには異なる文字を割り当てる．ここでノードラベルが同じかどうかの判定を含めた 2 つのパスの 2 つの文字列への変換方法は次のようになる．

$\mathcal{P}_Y(y_2)$ と $\mathcal{P}_S(s_2)$ について，$\mathcal{P}_Y(y_2)$ にはノードラベルが {ARABIC,BASED, CREOLE} となる 1 つのノードしか含まれていない．このノードラベルに，たとえばアルファベットの 1 文字目の大文字 A を割り当てる．一方，$\mathcal{P}_S(s_2)$ には {Creole} と {Arabic, based} の 2 つのノードラベルがあり，まず前者の {Creole} について，式 (5.2) に従って {Creole} と {ARABIC, BASED, CREOLE} の言語名の類似度を計算し，sd_ln({Creole},{ARABIC, BASED, CREOLE})=0.33 が得られる．ノードラベル間の言語名の類似度が閾値 α を下回らないならば（ここで α=0.75 であり， 5.5.1 に示す実験を通して決定された値），この 2 つのノードラベルを同じとし，{Creole} にも同じく文字 A を割り当てるが，ここでは $0.33 < \alpha (= 0.75)$ となるため，同じノードラベルとは認められず，{Creole} には A 以外の文字，たとえば B を割り当てる．後者の {Arabic, based} については，sd_ln({Arabic, based}, {ARABIC, BASED,

CREOLE})=0.67 < α(=0.75) であり，かつ sd_ln({Arabic, based},{Creole})=0<α (=0.75) である．つまり {Arabic, based} は {ARABIC, BASED, CREOLE} と {Creole} のいずれとも同じノードラベルとは認められない．そのため，{Arabic, based} には A または B 以外の文字，たとえば C を割り当てる．このようにして，$\mathcal{P}_Y(y_2)$ と $\mathcal{P}_S(s_2)$ をそれぞれ 2 つの文字列，たとえば A と BC，に変換できる．この変換処理を $\mathcal{F}(\mathcal{P}_Y(y)$,$\mathcal{P}_S(s)) \rightarrow (v_{y,s}, w_{y,s})$ と表記するならば，$\mathcal{P}_Y(y_2)$ と $\mathcal{P}_S(s_2)$ に対し，変換後得られる 2 つの文字列は v_{y_2,s_2}=A, w_{y_2,s_2}=BC となる．

パスを 2 つの文字列 $v_{y,s}$ と $w_{y,s}$ に変換した後，さらにこの 2 つの文字列を言語系統分類の類似特性記号列 ($SFSLC$) と呼ばれる 1 つの文字列に変換する．この文字列を $SFSLC(v_{y,s}, w_{y,s})$，$SFSLC(v_{y,s}, w_{y,s})$ の長さを $l_{SFSLC}(v_{y,s}, w_{y,s})$ と表記するならば，$SFSLC(v_{y_2,s_2}, w_{y_2,s_2})$=$X-$, $l_{SFSLC}(v_{y_2,s_2}, w_{y_2,s_2})$=2 となる．式 (5.3) に従って，言語 y_2 と言語 s_2 の言語系統分類の類似度を計算すると，$sd\_lc(y_2, s_2)$=$\frac{0}{2}$=0 となる．

また，図 6.1 (3) の言語 y_3 と言語 s_3 のパスはそれぞれ “Unclassified” と “Language Isolate” で，$\mathcal{P}_Y(y_3)$={Unclassified}, $\mathcal{P}_S(s_3)$={Language, Isolate} となる．$\mathcal{F}(\mathcal{P}_Y(y_3)$, $\mathcal{P}_S(s_3)) \rightarrow (v_{y_3,s_3}, w_{y_3,s_3})$ で変換し，得られる v_{y_3,s_3} と w_{y_3,s_3} も図 6.1 (2) の y_2 と s_2 と同様，v_{y_3,s_3}=A, w_{y_3,s_3}=BC にすることができるため，$sd\_lc(y_3, s_3)$=$sd\_lc(y_2, s_2)$=0 となる．なお，図 6.1 (1) の言語 y_1 と言語 s_1 の場合は，言語系統分類がまったく同じであるため，$sd\_lc(y_1, s_1)$=1 となる．

手法 I では，T_Y と T_S に含まれる同一言語ペアの検出処理は，2 つのステップに分けて行なっている．まず完全一致言語，すなわち T_Y と T_S において言語名にも言語系統分類にも違いがない言語，について処理する．次に，ゆれのある言語，すなわち y_1 と s_1 や y_2 と s_2 のような，本来同一言語であるが，言語名または言語系統分類，あるいはその両方に違いがある言語，から同一言語ペアを見つける．本節では後者について説明する．

言語 $y \in V_{leaf}(T_Y)$ と言語 $s \in V_{leaf}(T_S)$ が次の 3 つの条件をともに満たした場合，同一言語ペアと判定される．(i) $sd\_ln(\mathcal{L}_y^{T_Y}, \mathcal{L}_s^{T_S}) \geq \alpha$，すなわち言語名の類似度 $sd\_ln(\mathcal{L}_y^{T_Y}, \mathcal{L}_s^{T_S})$ が閾値 α (=0.75) 以上でなければならない; (ii) $sd\_lc(y, s) > \beta$，すなわち言語系統分類の類似度 $sd\_lc(y, s)$ が閾値 β (=0) を超えなければならない．

(iii) $y \in V_{leaf}(T_Y)$ に対し，T_S において言語 s が言語 y との言語系統分類の類似度の値が最も高い．つまり，y との言語名の類似度が同じである T_S 中の言語は，s を含め，複数存在しうるが，s 以外の言語と y との言語系統分類の類似度の値はいずれも s より低くないといけない．なお，閾値 β=0 も同様に， 5.5.1 に示す実験を通して決定された値である．

図 6.1 にある 3 つの同一言語ペア (y_1, s_1), (y_2, s_2), (y_3, s_3) は，いずれも手法 I では同一言語ペアとして検出できない．ケース 1 として，図 6.1 (1) の (y_1, s_1) は，言語系統分類は同じであるが，言語名の類似度 $sd\_ln(\mathcal{L}_{y_1}^{T_Y}, \mathcal{L}_{s_1}^{T_S})$=0.67$\not\geq \alpha(= 0.75)$ で，条件 (i) を満たしていない．また，ケース 2 として，図 6.1 (2) の (y_2, s_2) は逆に，言語名は同じであるが，言語系統分類の類似度 $sd\_lc(y_2, s_2) = 0 \not> \beta(= 0)$ であるため，条件 (ii) を満たしていない．図 6.1 (3) の (y_3, s_3) は図 6.1 (2) の (y_2, s_2) と同じく，ケース 2 に含まれる．

この 2 つのケースともに，言語名または言語系統分類の一方は完全に一致しているが，その他方の類似度が閾値に満たないため，同一言語として判定されない結果となっている．つまり，手法 I では言語名または言語系統分類の一方が完全一致であることが同一言語としての判定にまったく考慮されていない．

6.2.2 手法 I 問題点の分析

手法 I の問題点は，(1) 言語名の類似度と言語系統分類の類似度を分離して同一言語の判定処理に用いていること，(2) 言語系統分類の類似性の評価が不十分であることである．

6.2.1 に挙げている 2 つのケースの中で，ケース 1 として，図 6.1 (1) の (y_1, s_1) は，T_Y と T_S での言語系統分類は完全に一致しているが，言語名の類似度の値が閾値に達しないため，同一言語ペアとして検出されない例である．逆にケース 2 として，図 6.1 (2) の (y_2, s_2) と図 6.1 (3) の (y_3, s_3) は，言語名は完全に一致しているが，言語系統分類の類似度の値が閾値に達しないため，同様に同一言語ペアとして検出されない例である．

この 2 つのケースに共通しているのは，言語名の類似度と言語系統分類の類似度の一方が最大値 1 となっているにもかかわらず，他方が閾値を下回っているため，同一言語ペアとして棄却されていることである．手法 I では，まず言語名の類似度に関する条件の成立可否をチェックし，そのうえでさらに言語系統分類の類似度に関する条件の成立可否をチェックする，という 2 段階処理を行なっており，それぞれの段階で，言語名の類似度と言語系統分類の類似度という別々の尺度を用いている．その結果として，一方の尺度が高値を示す場合においても，他方の尺度が低ければ別言語と判定してしまうことになる．この点が上に挙げた例での言語同一性判定の失敗原因であり，これを解決するためには，言語名の類似度と言語系統分類の類似度の 2 つの尺度を同時に考慮する必要がある．

また，ケース 2 の図 6.1 (2) の (y_2, s_2) と図 6.1 (3) の (y_3, s_3) の例では，言語系統分類の類似度の値が共に 0 という結果が出ている．手法 I では言語系統分類の類似性はもっぱら語族（言語系統木におけるルートの子）から言語の親までのパス，つまり語族名や言語グループ名の異同，を考慮している．図 6.1 (3) では，y_3 と s_3 のパスはそれぞれ “Unclassified” と “Language Isolate” で，パスの類似度が 0 になるため，言語系統分類の類似度も 0 との評価になっている．この “Unclassified” と “Language Isolate” は，ここでは言語名の綴りはまったく異なるが，同様な意味で使われている．一方，図 6.1 (2) の例で，言語 y_2 と s_2 のパスはそれぞれ “ARABIC-BASED CREOLE” と “Creole” / “Arabic based” になっており，こちらもパスの類似度が 0 となっている．この例では，直観的には同じパスだと判断できるので，言語系統分類の類似性を十分に引き出せていないと言わざるをえない．

6.3 新言語系統分類の類似度と言語総合類似度

6.3.1 概要

本節では上に述べたような問題点を踏まえ，本研究において行う提案の概要について述べる．

第一に，図 6.1 のケース 1 とケース 2 のように，言語名の類似度または言語系統分類の類似度の一方が明らかに高く（完全に一致することはその最たる例である），その他方が閾値には達しないものの，閾値との差がさほど大きくないような場合は，同一言語である可能性が高いと考える．新手法では，手法 I の言語名の類似度と言語系統分類の類似度を 2 段階で考慮する判定方法を改善し，両方の類似性を共に考慮する言語総合類似度（言語名の類似度と言語系統分類の類似度の加重平均値）を新たに導入する．言語総合類似度を言語の同一性判定に用いる．

第二に，言語系統分類の類似性の評価の改善について考える．図 6.1 (2) の (y_2, s_2) と図 6.1 (3) の (y_3, s_3) はどちらも言語系統分類の類似性を十分に引き出せていない例である．前者の図 6.1 (2) の (y_2, s_2) のような例は，言語系統分類を表すパス上のノードラベルの文字列パターンから直観的に言語 y_2 と言語 s_2 は系統分類が同じと判断できるのに対し，手法 I では言語系統分類の類似度が 0 という結果になる．この点に関し，言語系統分類の類似度を表すパスの類似度の計算法に改善の余地があるように思われる．しかし，後者の図 6.1 (3) の (y_3, s_3) のような例は，性質が異なる．この例では，言語系統分類を表すパス上のノードラベルが表す言語名（語族名や言語グループ名を含む）の文字列パターンはまったく異なっているが，同様な意味で使われている．辞書を導入するなどの手法を使えば，同じ言語名との判定が可能になるが，文字列類似度に基づく方針をとる限り，その類似性を引き出すには限界があるように思われる．そこで，我々はこの 2 つの例に共通していることに着目し，パスの類似度の計算よりも，言語系統分類の類似性評価を別の角度から見直すことにする．

図 6.1 (2) の y_2 には兄弟 “NUBI”，s_2 には兄弟 “Nubi” がそれぞれ存在し，これらは同じ言語である．一方，図 6.1 (3) の y_3 と s_3 のそれぞれの兄弟にも同一言語ペア

である (“TAUSHIRO”,“Taushiro”) が存在している．しかも，この他にさらにもう 1 つの同一言語ペア (“WAORANI”,“Waorani”) が存在している．つまり，この 2 つの例に共通しているのは，それぞれ共に兄弟言語が存在し，その兄弟言語もまた同一言語ペアである，ということである．我々は，T_Y と T_S のそれぞれに含まれる 2 つの言語について，互いに兄弟言語が存在し，その兄弟言語同士が同じ言語であるならば，この 2 つの言語も同一言語である可能性が高い，と考える．また，手法 I と同様に，語族と親の異同が T_Y と T_S のそれぞれに含まれる 2 つの言語が同じ言語かどうかを判定するうえで重要な情報である，と考える（語族と親の情報はパスに含まれているため，この点は手法 I ではすでに考慮している）．新手法では，系統分類の類似性評価の対象を広め，語族，親，兄弟の情報を考慮した新たな言語系統分類の類似度を導入する．

6.3.2 新言語系統分類の類似度と言語総合類似度

言語系統分類の類似度計算の新手法は語族類似度，親類似度，兄弟類似度に基づくものである．本節では，まず言語名の類似度について再定義を行う．次に語族類似度，親類似度，兄弟類似度について説明したうえで，言語系統分類の類似度の計算法について述べる．さらに言語総合類似度について定義し，それに基づく同一言語ペアの検出方法を示す．

(1) 言語名の類似度の再定義

手法 I では，言語名の類似度の計算式（式 (5.2) ）は，言語名（第一言語名または別名）を引数とする形式で定義されている．一方，言語 $y \in V_{leaf}(T_Y)$ と言語 $s \in V_{leaf}(T_S)$ の言語名の類似度は，y と s のそれぞれの第一言語名同士の類似度および y の第一言語名と s のすべての別名との類似度の中での最も高い値とされているが，これについては定義されていない．なお，s は別名を持たない場合もあることに注意されたい．式 (5.2) に基づいて，次のように言語名の類似度について言語ノードを引数とする形式に変更する．

定義 6.1. 言語 $y \in V_{leaf}(T_Y)$ と $s \in V_{leaf}(T_S)$ の第一言語名をそれぞれ $\mathcal{L}_y^{T_Y}$ と $\mathcal{L}_s^{T_S}$ とする．言語 s の別名の数を m とし，$m > 0$ のときのすべての別名を $\mathcal{A}_1^s, \mathcal{A}_2^s, \cdots, \mathcal{A}_m^s$ とする．y と s の**言語名の類似度** $sd\_ln_{new}(y, s)$ は，以下の方法で算出される値である．

$$sd\_ln_{new}(y,s) = \begin{cases} sd\_ln(\mathcal{L}_y^{T_Y}, \mathcal{L}_s^{T_S}) & (m{=}0) \\ \max\{sd\_ln(\mathcal{L}_y^{T_Y}, \mathcal{L}_s^{T_S}), sd\_ln(\mathcal{L}_y^{T_Y}, \mathcal{A}_1^s), & \\ \qquad sd\_ln(\mathcal{L}_y^{T_Y}, \mathcal{A}_2^s), \cdots, sd\_ln(\mathcal{L}_y^{T_Y}, \mathcal{A}_m^s)\} & (m{>}0) \end{cases} \tag{6.1}$$

□

(2) 兄弟情報を考慮した言語系統分類の類似度

ここでは，図 6.1 (2) の同一言語ペア (y_2, s_2) を例に説明していく．言語 y_2 と言語 s_2 が分類されている語族の名前は，それぞれ “ARABIC-BASED CREOLE” と “Creole” である．y_2 と s_2 の語族類似度を {ARABIC, BASED, CREOLE} と {Creole} という 2 つの語族名の言語名の類似度として定義する．

定義 6.2. 言語 $y \in V_{leaf}(T_Y)$ と言語 $s \in V_{leaf}(T_S)$ の語族名をそれぞれ $\mathcal{FN}_y^{T_Y}$ と $\mathcal{FN}_s^{T_S}$ とする．ただし，$\mathcal{FN}_y^{T_Y}$ と $\mathcal{FN}_s^{T_S}$ はそれぞれ語の集合である．y と s の**語族類似度** $sd\_fn(y, s)$ は，以下の方法で算出される値である．

$$sd\_fn(y,s) = sd\_ln(\mathcal{FN}_y^{T_Y}, \mathcal{FN}_s^{T_S}) \tag{6.2}$$

□

図 6.1 (2) の同一言語ペア (y_2, s_2) の例では，$sd\_fn(y_2, s_2)$=sd_ln({ARABIC, BASED, CREOLE}, {Creole})=0.33 となる．

また，図 6.1 (2) の言語 y_2 の親は “ARABIC-BASED CREOLE” であり，このノードは語族でもある．一方，言語 s_2 の親は “Arabic Based” である．語族類似度と同様に，y_2 と s_2 の親類似度を {ARABIC, BASED, CREOLE} と {Arabic, Based} との 2 つの言語グループ名の類似度として定義する．

定義 6.3. 言語 $y \in V_{leaf}(T_Y)$ と言語 $s \in V_{leaf}(T_S)$ の親の言語グループ名をそれぞれ $\mathcal{PN}_y^{T_Y}$ と $\mathcal{PN}_s^{T_S}$ とする．ただし，$\mathcal{PN}_y^{T_Y}$ と $\mathcal{PN}_s^{T_S}$ はそれぞれ語の集合である．y と s の**親類似度** $sd\_pn(y, s)$ は，以下の方法で算出される値である．

$$sd\_pn(y, s) = sd\_ln(\mathcal{PN}_y^{T_Y}, \mathcal{PN}_s^{T_S}) \tag{6.3}$$

□

図 6.1 (2) の同一言語ペア (y_2, s_2) の例では，$sd\_pn(y_2, s_2)$=sd_ln({ARABIC, BASED, CREOLE}, {Arabic, Based})=0.67 となる．

さらに，兄弟類似度の定義に当たっては，まず T_Y と T_S に含まれる同一言語の兄弟の存在状況を分析する．言語 $y \in V_{leaf}(T_Y)$ と言語 $s \in V_{leaf}(T_S)$ の兄弟について，次の 3 つのケースに分けられる．

Case（i）　y と s が共に一人だけの子であり，兄弟はいない．図 6.2 （i）に示す．

Case（ii）　y は一人だけの子であり，s は複数の兄弟がいる．またその逆も同じである．図 6.2 （ii）に示す．

Case（iii）　y と s が共に複数の兄弟がいる．図 6.2 （iii）に示す．

y と s の兄弟類似度の計算法は，この 3 つのケースに分けて，それぞれ次のように考える．

Case（i）　y と s の言語名の類似度を兄弟類似度とする．

Case（ii）　兄弟類似度を 0（ゼロ）とする．

Case（iii）　この場合の兄弟類似度の計算は，次に示す (a) と (b) の 2 つのステップに分けて行う．(a) y と s のそれぞれの兄弟の中から対応する兄弟言語ペアを確定し，それらの言語名の類似度を計算する，(b) すべての兄弟言語ペアの類似度の総和を算出し，これを y と s の兄弟数の平均値で割った値を y と s の兄弟類似度とする．

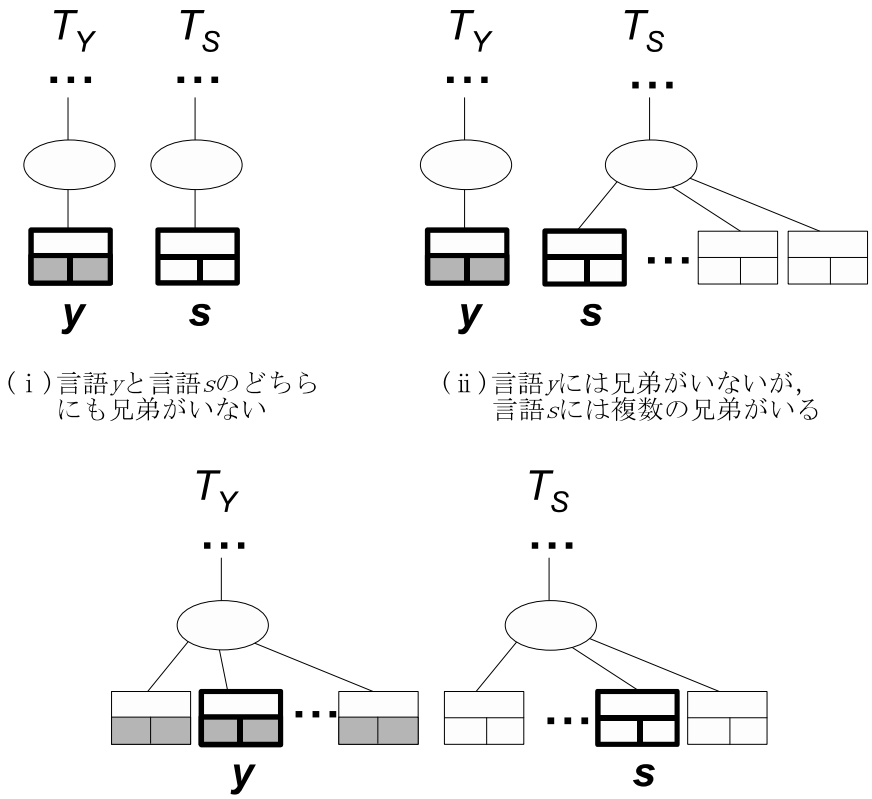


図 6.2: T_Y と T_S に含まれる同一言語の兄弟の存在状況

以下では，まず兄弟言語ペアの確定について説明し，次に兄弟類似度の定義を与える．

図 6.1 (2) の同一言語ペア (y_2, s_2) は言語名が同じで，y_2 には兄弟として “NUBI” が，s_2 には “Arabic, Babalia Creole” と “Nubi” の 2 兄弟がいる．y_2 と s_2 のそれぞれの兄弟の中で，明らかに “NUBI” と “Nubi” は言語名の類似度が高く，対応していると推測できる．このような 2 つの言語を y_2 と s_2 に対する**兄弟言語ペア**と呼ぶことにする．もっとも，ここの “NUBI” と “Nubi” は言語名の類似度が 1 で，同じ言語であるが，兄弟言語ペアは必ずしも同じ言語である必要はなく，y_2 の兄弟言語

“NUBI” に対し，s_2 の兄弟言語の中から類似度が最も高い言語であれば兄弟言語ペアとみなす.

兄弟言語ペアの探し方は，まず言語 y_2 と言語 s_2 の中で兄弟言語をより多く持つ s_2 について，s_2 のすべての兄弟に対して y_2 の兄弟の中から類似度が最も高い言語を探し，それらを兄弟言語ペアとする．この例では，s_2 の 2 つの兄弟言語である “Arabic, Babalia Creole” と “Nubi” のうち，“NUBI” と “Nubi” が兄弟ペア言語となる．一方，y_2 は “NUBI” 以外に兄弟言語はいないため，“Arabic, Babalia Creole” に対応する兄弟言語も存在しない．このような場合は，s_2 の兄弟言語である “Arabic, Babalia Creole” に対応する y_2 の兄弟言語を NULL とする．よって，y_2 と s_2 の兄弟言語ペアは，(“NUBI”,“Nubi’) と (Null,“Arabic, Babalia Creole”) の 2 つの組合せになる．兄弟言語ペアを次のように定義する.

定義 6.4. x_1 と x_2 は，$x_1 \in V_{leaf}(T_Y)$（または $V_{leaf}(T_S)$）ならば，$x_2 \in V_{leaf}(T_S)$（または $V_{leaf}(T_Y)$）であるとし，また x_1 と x_2 の兄弟言語の集合はそれぞれ $BL_{x_1}=\{bl_1^{x_1}, bl_2^{x_1}, \cdots\}$ と $BL_{x_2}=\{bl_1^{x_2}, bl_2^{x_2}, \cdots\}$ $(|BL_{x_1}| \geq |BL_{x_2}| > 0)$ であるとする．次の操作で得られた $BP(x_1, x_2)$ を x_1 と x_2 の**兄弟言語ペア集合**という.

(i) $BP(x_1, x_2) \leftarrow \phi$ とする.

(ii) $sd\_ln_{new}(bl^{x_1}, bl^{x_2}) = \max \{sd\_ln_{new}(bl_i^{x_1}, bl_j^{x_2}) \,|bl_i^{x_1} \in BL_{x_1}, bl_j^{x_2} \in BL_{x_2}\}$ を満たす兄弟言語ペア (bl^{x_1}, bl^{x_2}) を見つける.

(iii) $BP(x_1, x_2) \leftarrow BP(x_1, x_2) \cup \{(bl^{x_1}, bl^{x_2})\}$, $BL_{x_1} \leftarrow BL_{x_1} - \{bl^{x_1}\}$, $BL_{x_2} \leftarrow BL_{x_2} - \{bl^{x_2}\}$ とする.

(iv) $|BL_{x_2}|>0$ ならば，(ii) へ.

(v) $BP(x_1, x_2) \leftarrow BP(x_1, x_2) \cup \{(bl_i^{x_1}, \text{Null}) \mid bl_i^{x_1} \in BL_{x_1}\}$ とし，停止. □

言語 $y \in V_{leaf}(T_Y)$ と言語 $s \in V_{leaf}(T_S)$ の兄弟類似度を，図 6.2 の 3 つのケースに分けて，次のように定義する.

定義 6.5. x_1 と x_2 は，$x_1 \in V_{leaf}(T_Y)$（または $V_{leaf}(T_S)$）ならば，$x_2 \in V_{leaf}(T_S)$（または $V_{leaf}(T_Y)$）であるとする．また，x_1 と x_2 の兄弟言語の集合をそれぞれ BL_{x_1} と BL_{x_2} とし，$(|BL_{x_1}|=m,\ |BL_{x_2}|=n,\ m \geq n \geq 0)$，$BP(x_1, x_2)$ を $n > 0$ のときの x_1 と x_2 の兄弟言語ペア集合とする．x_1 と x_2 の**兄弟類似度** $sd\_bn(x_1, x_2)$ は，以下の方法で算出される値である．

$$sd\_bn(x_1, x_2) = \begin{cases} sd\_ln_{new}(x_1, x_2) & (m{=}n{=}0) \\ 0 & (m{>}0, n{=}0) \\ \frac{\sum_{(\mu,\nu)\in BP(x_1,x_2)} sd\_ln_{new}(\mu,\nu)}{\frac{m+n}{2}} & (m \geq n > 0) \end{cases} \tag{6.4}$$

□

図 6.1 (2) の同一言語ペア (y_2, s_2) の例では，言語 y_2 と言語 s_2 の 2 つの兄弟言語ペア (“NUBI”,“Nubi’) と (Null,“Arabic, Babalia Creole”) の類似度はそれぞれ 1 と 0 である．y_2 と s_2 の兄弟言語の数は全部で 3 であり，平均 1.5 である．よって，$sd\_bn(y_2, s_2) = \frac{1+0}{1.5} = 0.67$ となる．

言語 $y \in V_{leaf}(T_Y)$ と $s \in V_{leaf}(T_S)$ の言語系統分類の類似度を y と s の語族類似度，親類似度，兄弟類似度の加重平均とし，次のように定義する．

定義 6.6. 言語 $y \in V_{leaf}(T_Y)$ と言語 $s \in V_{leaf}(T_S)$ の**言語系統分類の類似度** $sd\_lc_{new}(y, s)$ は以下の方法で算出される値である．

$$sd\_lc_{new}(y, s) = e{*}sd\_fn(y, s){+}f{*}sd\_pn(y, s){+}g{*}sd\_bn(y, s) \tag{6.5}$$

ただし，係数 e, f, g は $1 \geq e \geq 0$, $1 \geq f \geq 0$, $1 \geq g \geq 0$, $e + f + g = 1$ を満たす． □

この係数 e, f, g の値設定については， 6.5 で述べる．

(3) 言語総合類似度

言語 $y \in V_{leaf}(T_Y)$ と言語 $s \in V_{leaf}(T_S)$ の言語総合類似度を y と s の言語名の類似度 $sd\_ln_{new}(y, s)$ と言語系統分類の類似度 $sd\_lc_{new}(y, s)$ の加重平均とし，次のように定義する．

定義 6.7. 言語 $y \in V_{leaf}(T_Y)$ と言語 $s \in V_{leaf}(T_S)$ の**言語総合類似度** $sd\_gen(y, s)$ は以下の方法で算出される値である．

$$sd\_gen(y, s) = a * sd\_ln_{new}(y, s) + b * sd\_lc_{new}(y, s) \tag{6.6}$$

ただし，係数 a, b は $1 \geq a \geq 0$, $1 \geq b \geq 0$, $a + b = 1$ を満たす． □

この係数 a, b の値設定については， 6.5 で述べる．

6.4 同一言語ペアの検出

6.4.1 概要

図 6.3 の例を用いて，同一言語ペア検出処理の概要について説明していく．

T_Y には $y_1, y_2, \cdots, y_m$ の m 言語，T_S には $s_1, s_2, \cdots, s_n$ の n 言語が含まれていると仮定し，図 6.3 (a) に T_Y と T_S の言語をマトリックスで示す．また，ある i, j に関し，言語 y_i と言語 s_j を同一言語ペアとする．(y_i, s_j) の下方にある 2 つの数字は，前者の 0.98 が言語名の類似度 $sd\_ln_{new}(y_i, s_j)$，後者の 0.78 が言語総合類似度 $sd\_gen(y_i, s_j)$ を表している．

図 6.3 （b）に示しているのは，同一言語ペア (y_i, s_j) に関し，(y_i, s_1), (y_i, s_2), $\cdots$, (y_i, s_n) と (y_1, s_j), (y_2, s_j), $\cdots$, (y_m, s_j) のうち，言語名の類似度と言語総合類似度が相対的に高い値を有する言語の組合せである．$sd\_gen(y_i, s_j) = 0.78$ に対し，$sd\_gen(y_i, s_{j'}) = 0.79$ となっており，つまり $s_{j'}$ という言語が y_i の真の同一言語 s_j よりも，y_i との言語総合類似度が高い（以降に述べる言語名の類似度と言語総合類似度の大小逆転現象とはこのことを指す）．実際の T_Y と T_S のデータではこのようなケースも存在しているため，言語総合類似度のみに基づいて同一言語かどうかを判定するのは適切とはいえない．

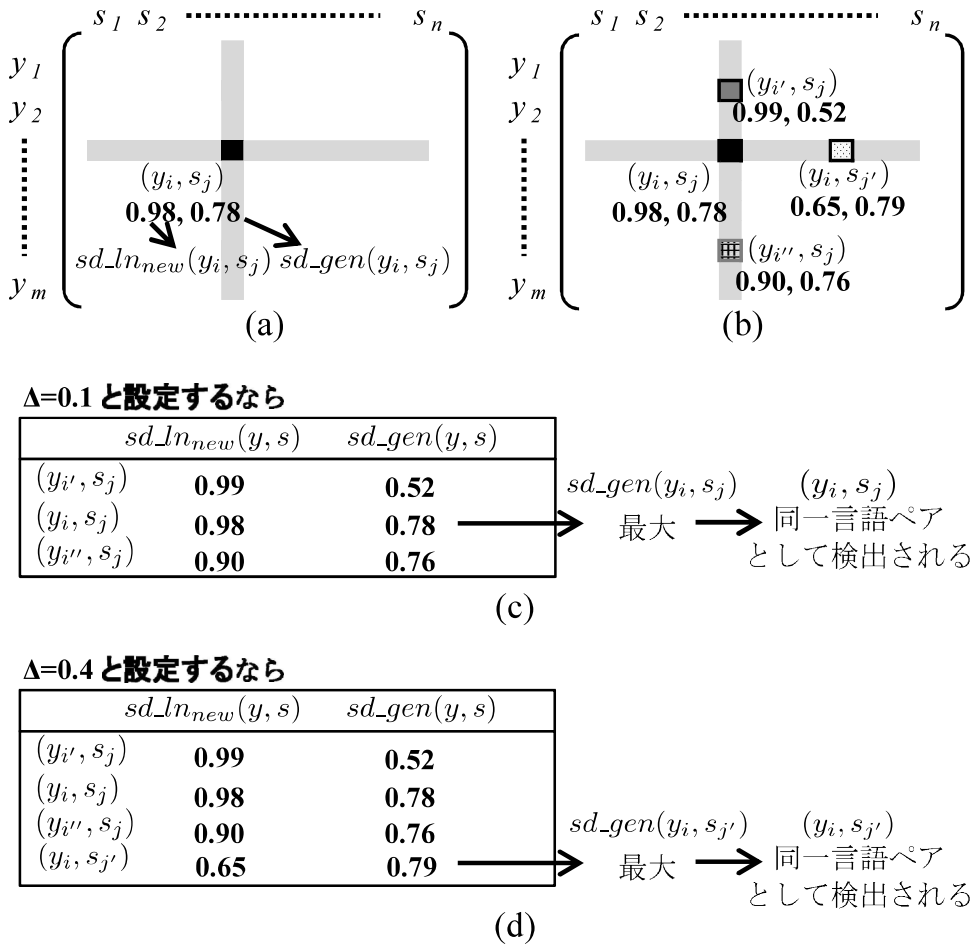


図 6.3: 言語名の類似度と言語総合類似度の大小逆転現象

一方，従来研究から言語名の類似度が言語同定において重要であることが知られている．しかし，言語名の類似度のみに基づいて同一言語かどうかを判定することはできない．この点について，手法 I でもすでに判明している．さらに，図 6.3 (b) に示すように $sd\_ln_{new}(y_i, s_j)$=0.98 に対し，$sd\_ln_{new}(y_{i'}, s_j)$=0.99 となっており，つまり $y_{i'}$ という言語が s_j の真の同一言語 y_i よりも，s_j との言語名の類似度が高いようなケースも存在している．

ここで，T_Y と T_S の言語から，どのように同一言語ペア (y_i, s_j) を探し出すかが問

題となる．言語名の類似度と言語総合類似度の 2 つの指標の優先順位が同一言語ペアの検出結果に影響をおよぼすと考えられる．本研究では手法 I のような 2 つの指標を分離する問題点を克服するため，同一言語ペアかどうかの最終判定は言語総合類似度に従うとする一方，この 2 つの指標を共に使うことを提案する．そのため，γ と Δ という 2 つのパラメータを導入する．言語名の類似度に $\gamma \geq sd\_ln_{new}(y,s) > \gamma - \Delta$ という幅を持たせて，まず言語名の類似度の値がこの範囲に入る言語の組合せを抽出し，次の言語総合類似度による判定処理の対象とする．なお，γ は最大値 1 からスタートし，少しずつ下げていく．Δ の大きさは，$0.3 > \Delta > 0$ と想定しており，実際は実験を通して決定するものとする（6.5 を参照されたい）．

γ と Δ を用いた処理の概要は次のようになる．図 6.3 (c) に示すように，γ=1, Δ=0.1 とするならば，言語名の類似度の高い順に $sd\_ln_{new}(y_{i'}, s_j)$=0.99, $sd\_ln_{new}(y_i, s_j)$=0.98, $sd\_ln_{new}(y_{i''}, s_j)$=0.90, $sd\_ln_{new}(y_i, s_{j'})$=0.65 となるため，$(y_{i'}, s_j)$, (y_i, s_j), $(y_{i''}, s_j)$ の 3 つの組合せが選ばれる．さらに，それぞれの組合せの言語総合類似度が $sd\_gen(y_{i'}, s_j) = 0.52$, $sd\_gen(y_i, s_j)$=0.78, $sd\_gen(y_{i''}, s_j)$=0.76 となるため，そのうち言語総合類似度が最大となる (y_i, s_j) が同一言語ペアとして判定される．

このように γ と Δ を導入することで，6.2.2 で述べたような手法 I の問題点を回避しながら言語名の類似度と言語総合類似度を共に同一言語の判定に用いることによって，そのどちらか一方のみに基づく場合の誤判定のリスクを軽減することができる．もっとも，Δ の値設定がかなり重要で，Δ 値の大きさによっては間違った判定を導くこともある．図 6.3 (d) に示しているのがその一例である．Δ=0.4 と設定するならば，本来の同一言語ペア (y_i, s_j) ではなく，$(y_i, s_{j'})$ が検出される結果となる．

6.4.2　同一言語ペアの検出処理の流れ

(1)　T_Y の任意の言語と T_S の任意の言語とのすべての組合せ，つまり (y_1, s_1), (y_1, s_2), $\cdots$, (y_1, s_n), (y_2, s_1), (y_2, s_2), $\cdots$, (y_i, s_j), $\cdots$, (y_m, s_n) （$y_i \in V_{leaf}(T_Y)$, $s_j \in V_{leaf}(T_S)$, m と n はそれぞれ T_Y と T_S に含まれる言語の数）について，式 (6.1) に従って，言語名の類似度 $sd\_ln_{new}(y_i, s_j)$ を計算し，言語名の類似度行列を生成する．図 6.4 (a) に示しているダミーデータは，T_Y と T_S の言語数

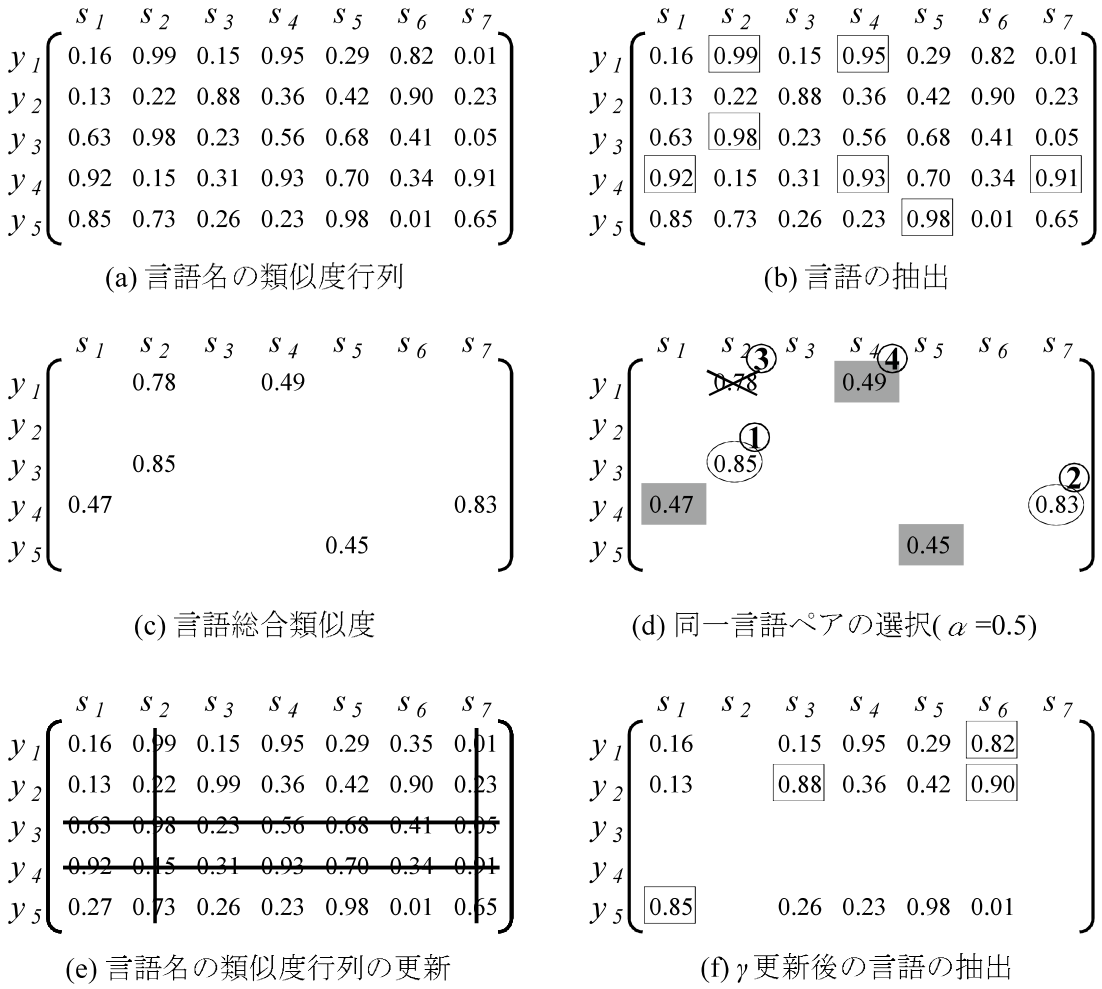


図 6.4: 同一言語ペア検出処理の流れ

をそれぞれ 5 と 7 としており，すなわち m=5, n=7 である．

(2) まず，言語名の類似度の値がある範囲内に入る言語の組合せを抽出する．T_Y と T_S に含まれる言語のすべての組合せの中から，$\gamma \geq sd\_ln_{new}(y_i, s_j) > \gamma - \Delta$ の条件を満たす組合せをすべて抽出する．ここでの γ は言語名の類似度行列 $sd\_ln_{new}(y_i, s_j)$ の要素の最大値であり，γ=1 からスタートする．また，Δ の決め方については 6.5 で議論するが，ここで仮に Δ=0.1 とする．つまり，

$1 \geq sd\_ln_{new}(y_i, s_j) > 0.9$ から処理を開始する．図 6.4 (b) に示すように，この条件を満たす組合せは全部で 6 つあり，四角で囲んでいる (y_1, s_2), (y_1, s_4), (y_3, s_2), (y_4, s_1), (y_4, s_7), (y_5, s_5) である．

(3) 上記（2）で $\gamma \geq sd\_ln_{new}(y_i, s_j) > \gamma - \Delta$ の条件を満たす組合せにつき，それぞれ式 (6.2)，式 (6.3)，式 (6.4) に従って，語族類似度 $sd\_fn\ (y_i, s_j)$，親類似度 $sd\_pn(y_i, s_j)$，兄弟類似度 $sd\_bn(y_i, s_j)$ を計算する．さらに，式 (6.5) に従って，言語系統分類の類似度 $sd\_lc_{new}(y_i, s_j)$ を計算し，式 (6.6) に従って，言語総合類似度 $sd\_gen(y_i, s_j)$ を算出する．そのダミーの値を図 6.4 (c) に示す．

(4) 図 6.4 (c) の 6 つの言語の組合せの中で言語総合類似度が最大となる組合せ $sd\_gen(y_3, s_2)$=0.85 を選び出す．図 6.4 (d) に (y_3, s_2) に対応する言語総合類似度値の右上に丸で囲んでいる数字①は言語総合類似度値の高い順位を表している．もしここでこの言語総合類似度の値が閾値 ρ(=0.5) より低いならば，この範囲内に抽出された言語の組合せでは同一言語ペアは存在しないものとし，上記（2）に戻る．この例では，① $sd\_gen(y_3, s_2)$=0.85 で，$sd\_gen(y_i, s_j) \geq$ 閾値 ρ (=0.5) という条件を満たしているため，(y_3, s_2) を同一言語ペアとして判定する．図 6.4 (d) に示すように，言語総合類似度の値を丸で囲んでいるのは同一言語ペアと判定したことを意味する．同様にして，② $sd\_gen(y_4, s_7)$=0.83 を同一言語ペアとして判定する．しかし，③ $sd\_gen(y_1, s_2)$=0.78 については同一言語ペアとして判定しない．なぜならば，我々は T_Y の言語 y の同一言語は，T_S では一言語のみ存在する（または存在していない可能性もある）と仮定し，y_i または s_j を 2 回以上重ねて選ぶことはしないとする．従って，すでに (y_3, s_2) を同一言語と決定しているため，(y_1, s_2) を同一言語としてみなすことはできない．ここで，③ $sd\_gen(y_1, s_2)$=0.78 の値を一旦削除し，後の処理に委ねる．次に，言語総合類似度値が順位 4 となる④ $sd\_gen(y_1, s_4)$=0.49 について処理する．$sd\_gen(y_1, s_4)$=0.49< 閾値 ρ (=0.5) となるため，(y_1, s_4) を同一

言語と判定しない．ここで言語総合類似度 $sd\_gen(y_i, s_j)$ が閾値 ρ より低くなったため，残りの組合せについての処理は中止し，上記（2）に戻る．図 6.4 (d) において，グレーで塗つぶした箇所が同一言語ペアと判定されなかった組合せを指す．

(5) 決定された同一言語ペアに関し，言語名の類似度行列からその値を消去し，言語名の類似度行列を更新する．図 6.4 (e) に示しているように，決定された同一言語ペア (y_3, s_2) について $sd\_ln_{new}(y_3, s_1)$, $sd\_ln_{new}(y_3, s_2)$, $\cdots$, $sd\_ln_{new}(y_3, s_7)$ と $sd\_ln_{new}(y_1, s_2)$, $sd\_ln_{new}(y_2, s_2)$, $\cdots$, $sd\_ln_{new}(y_5, s_2)$ を消去する．同一言語ペア (y_4, s_7) についても同様に処理する．

(6) 上記（2）の処理では，$\gamma \geq sd\_ln_{new}(y_i, s_j) > \gamma - \Delta$ $(1 \geq sd\_ln_{new}(y_i, s_j) > 0.9)$ の条件を満たす言語の組合せを抽出した．この範囲から Δ の幅を下げ，つまり $\gamma \leftarrow \gamma - \Delta$ $(=1 - 0.1 = 0.9)$ となるように γ を更新する．次に，図 6.4 (f) に示すように，さらに言語名の類似度行列から $\gamma \geq sd\_ln_{new}(y_i, s_j) > \gamma - \Delta$ $(0.9 \geq sd\_ln_{new}(y_i, s_j) > 0.8)$ の条件を満たす言語の組合せを抽出する．このように，言語名の類似度行列に値がなくなる，または $\gamma \leq 0$ となるまで上記 (2) ～ (5) の処理を繰り返す．

以上の処理のアルゴリズムを図 6.5 に示す．

6.5 パラメータの値設定

6.5.1 テストデータと評価方法

本研究では，言語系統分類の類似度を語族類似度，親類似度，兄弟類似度の重み付き加重平均として定義しており，重み係数がそれぞれ e,f,g としている．e,f,g の重み設定に関しては，次のように考える．T_Y と T_S のそれぞれに含まれる 2 つの言語が

アルゴリズム：NEW_FSLV

入力：$V_{leaf}(T_Y)$, $V_{leaf}(T_S)$, e, f, g, a, b, Δ, ρ

出力：すべての $y{\in}V_{leaf}(T_Y)$ の同一言語ペア SLP

手法：

1° $SLP{\leftarrow}\phi$, $P{\leftarrow}\phi$, $\gamma{\leftarrow}1$ とする．すべての $y{\in}V_{leaf}(T_Y)$, $s{\in}V_{leaf}(T_S)$ について，$\psi_{y,s}{=}sd\_ln_{new}(y,s)$ を計算し，言語名の類似度行列 $\Psi{=}(\psi_{y,s})$ を生成する．

2° $\gamma{\leq}0$ ならば，3° へ，そうでなければ，$P{\leftarrow}P{\cup}\{(y,s) \mid \gamma{\geq}\psi_{y,s}{>}\gamma{-}\Delta$, $\psi_{y,s}$ はΨの要素 $\}$ とし，次の (i)∼(v) を行う．

(i) パラメータ e,f,g,a,b に値を設定し，すべての $(y,s){\in}P$ に対して言語総合類似度 $\omega_{y,s}{=}sd\_gen(y,s)$ を計算する．

(ii) $\omega_{max}=\max\{\omega_{y,s} \mid (y,s){\in}P\}$ とする．$\omega_{max}<\rho$ ならば，$\gamma{\leftarrow}\gamma{-}\Delta$ とし，2° へ．

(iii) $\omega_{y',s'}=\omega_{max}$ と $(y',s'){\in}P$ を満たす (y',s') を 1 つ選び，$SLP{\leftarrow}SLP{\cup}(y',s')$ とする．

(iv) $P{\leftarrow}P{-}\{(y',s')\}$ とし，Ψ からそれぞれ y' の行と s' の列を削除する．

(v) $P{\neq}\phi$ ならば，(ii) へ．そうでなければ，$\gamma{\leftarrow}\gamma{-}\Delta$ とし，2° へ．

3° SLP を出力し，停止する．

図 6.5: アルゴリズム：NEW_FSLV

持つ兄弟言語が同じ言語である場合，この 2 つの言語もまた同じ言語である可能性が高いと考える．また，手法 I では言語系統分類の類似度は語族（言語系統木におけるルートの子のノード）から言語の親のノードまでのパスの類似度で表されており，そのパスには語族と親のノードの類似性に関する情報がすでに考慮されている．これに対し，兄弟類似度は本研究で新たに導入されたものである．語族類似度と親類似度に比べ兄弟類似度に付ける重みを大きくするほど，兄弟情報が考慮されやすくなる．それは手法 I では判定できない同一言語ペアの検出に資する，と考えられる．特にこの効果を調べるため，本研究では兄弟類似度を重視し，$e{=}0.25, f{=}0.25, g{=}0.5$ と兄弟類似度のウェイトを高く設定した．

一方，パラメータ a, b, Δ および閾値 ρ は，次のように実験を通して決定する．

表 6.1: 同一言語検出結果の正誤評価

			真の結果 同一言語が存在しているか	
			存在 している	存在 していない
検出結果 同一言語が出力されたか		出力 あり	① TP (True Positive)	② FP (False Positive)
		出力 なし	③ FN (False Negative)	④ TN (True Negative)

T_Y の総言語数 2,869 の中から無作為に 200 言語を抽出し，テストデータとした．この T_Y の 200 言語に関し，対応する T_S での同一言語を予め調査しておき，これを真の結果とした．もっとも，T_Y に含まれている言語が T_S にも必ず含まれているわけではないため，この T_Y の 200 言語に関しても，対応する T_S での同一言語が必ず存在するわけではない．

表 6.1 には同一言語の検出結果の正誤評価を示している．この表の TP, FP, FN, TN [高村 10] は次のように集計した．

パラメータ Δ, a, b および閾値 ρ を変化させ，図 6.5 のアルゴリズムに従って T_Y と T_S に含まれている同一言語の検出処理を行った．テストデータの 200 言語に関し，各 Δ, a, b および閾値 ρ の設定値の下で出力された T_S での同一言語の結果と真の結果とを比較し，①T_Y の言語に対し，T_S での同一言語が出力され，かつ真の結果と一致する言語の数を TP，②T_Y の言語に対し，T_S での同一言語が出力され，かつ真の結果と異なる言語の数を FP，③T_Y の言語に対し，言語総合類似度の値が閾値 ρ より低く，T_S での同一言語の出力はされないが，真の結果では T_Y の言語に対

する同一言語が存在している言語の数を FN，④ T_Y の言語に対し，言語総合類似度の値が閾値 ρ より低く，T_S での同一言語の出力はされないが，真の結果でも T_Y の言語に対する同一言語が存在しない言語の数を TN としてそれぞれ集計した．ここで，$TP + FP + TN + FN = 200$ である．

再現率，適合率，F 値の計算式 [高村 10] を以下に示す．

$$\text{再現率} = \frac{TP}{TP + FN} \tag{6.7}$$

$$\text{適合率} = \frac{TP}{TP + FP} \tag{6.8}$$

$$\text{F 値} = \frac{2 \cdot \text{再現率} \cdot \text{適合率}}{\text{再現率} + \text{適合率}} \tag{6.9}$$

式 (6.7)，式 (6.8)，式 (6.9) によって算出した各 Δ, a, b および閾値 ρ の設定値の下の再現率，適合率，F 値を比較検討し，F 値が最も高く，かつ適合率が高いときの設定値に決定した．

6.5.2 パラメータ値設定実験の経過

まず，Δ=0.01 に設定し，(a, b) の値をそれぞれ $(1.0, 0.0)$, $(0.9, 0.1)$, $(0.8, 0.2)$, $(0.7, 0.3)$, $(0.6, 0.4)$, $(0.5, 0.5)$, $(0.4, 0.6)$, $(0.3, 0.7)$, $(0.2, 0.8)$, $(0.1, 0.9)$, $(0.0, 1.0)$ と 0.1 刻みで重みをずらして，図 6.5 のアルゴリズムに従って計算を行い，次に Δ を 0.01 ずつ増やし，つまり Δ=0.02, 0.03, $\cdots$, 0.25 とそれぞれ設定し，同様に a と b の値を変化させながら，計算を行った．

言語総合類似度の閾値 ρ は，次のような理由により，まず ρ=0.5 と設定した．閾値を低くすればより多くの言語ペアが出力され，真の同一言語ペアがより多く見つかる一方（再現率の上昇），本来同一言語ではない言語も出力されることが増える（適合率の低下）ことが予想される．a, b, Δ の値を決定するためには，まず閾値 ρ

表 6.2: パラメータ a, b, Δ の値設定に関する実験

	a		1.0		0.9		0.8		0.7		0.6		**0.5**		0.4		0.3		0.2		0.1		0.0	
	b		0.0		0.1		0.2		0.3		0.4		**0.5**		0.6		0.7		0.8		0.9		1.0	
Δ	0.01	TP	167	FP 29	176	FP 19	176	FP 18	176	FP 16	176	FP 13	176	FP 9	174	FP 6	160	FP 6	143	FP 6	130	FP 6	115	FP 6
		FN	3	TN 1	3	TN 2	4	TN 2	4	TN 4	7	TN 4	11	TN 4	15	TN 5	29	TN 5	46	TN 5	59	TN 5	74	TN 5
		再現率	0.9824		0.9832		0.9778		0.9778		0.9617		0.9412		0.9206		0.8466		0.7566		0.6878		0.6085	
		適合率	0.8520		0.9026		0.9072		0.9167		0.9312		0.9514		0.9667		0.9639		0.9597		0.9559		0.9504	
		F値	0.9126		0.9412		0.9412		0.9462		0.9462		0.9462		0.9431		0.9014		0.8462		0.8000		0.7419	
	0.05	TP	167	FP 29	176	FP 19	176	FP 18	176	FP 16	176	FP 13	176	FP 9	174	FP 6	160	FP 6	143	FP 6	130	FP 7	115	FP 6
		FN	3	TN 1	3	TN 2	4	TN 2	4	TN 4	7	TN 4	11	TN 4	15	TN 5	29	TN 5	46	TN 5	58	TN 5	74	TN 5
		再現率	0.9824		0.9832		0.9778		0.9778		0.9617		0.9412		0.9206		0.8466		0.7566		0.6915		0.6085	
		適合率	0.8520		0.9026		0.9072		0.9167		0.9312		0.9514		0.9667		0.9639		0.9597		0.9489		0.9504	
		F値	0.9126		0.9412		0.9412		0.9462		0.9462		0.9462		0.9431		0.9014		0.8462		0.8000		0.7419	
	0.10	TP	167	FP 29	176	FP 19	177	FP 18	177	FP 16	177	FP 14	177	FP 11	175	FP 7	161	FP 7	144	FP 7	131	FP 8	116	FP 7
		FN	3	TN 1	3	TN 2	3	TN 2	3	TN 4	5	TN 4	8	TN 4	13	TN 5	27	TN 5	44	TN 5	56	TN 5	72	TN 5
		再現率	0.9824		0.9832		0.9833		0.9833		0.9725		0.9568		0.9309		0.8564		0.7660		0.7005		0.6170	
		適合率	0.8520		0.9026		0.9077		0.9171		0.9267		0.9415		0.9615		0.9583		0.9536		0.9424		0.9431	
		F値	0.9126		0.9412		0.9440		0.9491		0.9491		0.9491		0.9459		0.9045		0.8496		0.8037		0.7460	
	0.13	TP	167	FP 29	176	FP 19	177	FP 18	177	FP 16	177	FP 13	177	FP 11	175	FP 7	161	FP 7	144	FP 7	131	FP 8	116	FP 7
		FN	3	TN 1	3	TN 2	3	TN 2	3	TN 4	6	TN 4	8	TN 4	13	TN 5	27	TN 5	44	TN 5	56	TN 5	72	TN 5
		再現率	0.9824		0.9832		0.9833		0.9833		0.9655		0.9568		0.9309		0.8564		0.7660		0.7005		0.6170	
		適合率	0.8520		0.9026		0.9077		0.9171		0.9316		0.9415		0.9615		0.9583		0.9536		0.9424		0.9431	
		F値	0.9126		0.9412		0.9440		0.9491		0.9491		0.9491		0.9459		0.9045		0.8496		0.8037		0.7460	
	0.15	TP	167	FP 29	176	FP 19	176	FP 18	175	FP 17	175	FP 15	175	FP 11	173	FP 7	160	FP 7	143	FP 7	130	FP 7	115	FP 7
		FN	3	TN 1	3	TN 2	4	TN 2	4	TN 4	6	TN 4	10	TN 4	15	TN 5	28	TN 5	45	TN 5	58	TN 5	73	TN 5
		再現率	0.9824		0.9832		0.9778		0.9777		0.9669		0.9459		0.9202		0.8511		0.7606		0.6915		0.6117	
		適合率	0.8520		0.9026		0.9072		0.9115		0.9211		0.9409		0.9611		0.9581		0.9533		0.9489		0.9426	
		F値	0.9126		0.9412		0.9412		0.9434		0.9434		0.9434		0.9402		0.9014		0.8462		0.8000		0.7419	
	0.20	TP	167	FP 29	176	FP 19	177	FP 18	176	FP 17	176	FP 15	176	FP 14	174	FP 9	161	FP 8	144	FP 8	131	FP 8	115	FP 9
		FN	3	TN 1	3	TN 2	3	TN 2	3	TN 4	6	TN 3	6	TN 4	12	TN 5	26	TN 5	43	TN 5	56	TN 5	71	TN 5
		再現率	0.9824		0.9832		0.9833		0.9832		0.9670		0.9670		0.9355		0.8610		0.7701		0.7005		0.6183	
		適合率	0.8520		0.9026		0.9077		0.9119		0.9215		0.9263		0.9508		0.9527		0.9474		0.9424		0.9274	
		F値	0.9126		0.9412		0.9440		0.9462		0.9437		0.9462		0.9431		0.9045		0.8496		0.8037		0.7419	
	0.25	TP	167	FP 29	176	FP 19	176	FP 18	175	FP 17	175	FP 14	174	FP 13	172	FP 8	159	FP 9	143	FP 9	130	FP 9	115	FP 8
		FN	3	TN 1	3	TN 2	4	TN 2	4	TN 4	7	TN 4	9	TN 4	15	TN 5	27	TN 5	43	TN 5	57	TN 5	72	TN 5
		再現率	0.9824		0.9832		0.9778		0.9777		0.9615		0.9508		0.9198		0.8548		0.7688		0.6952		0.6150	
		適合率	0.8520		0.9026		0.9072		0.9115		0.9259		0.9305		0.9556		0.9464		0.9408		0.9353		0.9350	
		F値	0.9126		0.9412		0.9412		0.9434		0.9434		0.9405		0.9373		0.8983		0.8462		0.7975		0.7419	

パラメータ e=0.25, f=0.25, g=0.5, 閾値 ρ=0.5

表 6.3: 閾値 ρ の値設定に関する実験

閾値 ρ			0.50	0.51	0.52	0.53	**0.54**	0.55	0.60	0.65	0.70	0.75	0.80
Δ	**0.10**	TP FP	177 11	175 9	175 9	175 9	174 6	174 6	171 6	158 4	137 2	114 0	90 0
		FN TN	8 4	12 4	12 4	12 4	14 6	14 6	17 6	31 7	54 7	79 7	103 7
		再現率	0.9568	0.9358	0.9358	0.9358	0.9255	0.9255	0.9096	0.8360	0.7173	0.5907	0.4663
		適合率	0.9415	0.9511	0.9511	0.9511	0.9667	0.9667	0.9661	0.9753	0.9856	1.0000	1.0000
		F値	0.9491	0.9434	0.9434	0.9434	0.9457	0.9457	0.9370	0.9003	0.8303	0.7427	0.6360
	0.13	TP FP	177 11	175 9	175 9	175 9	174 6	174 6	171 6	158 4	137 2	114 0	79 0
		FN TN	8 4	12 4	12 4	12 4	14 6	14 6	17 6	31 7	54 7	79 7	103 7
		再現率	0.9568	0.9358	0.9358	0.9358	0.9255	0.9255	0.9096	0.8360	0.7173	0.5907	0.4341
		適合率	0.9415	0.9511	0.9511	0.9511	0.9667	0.9667	0.9661	0.9753	0.9856	1.0000	1.0000
		F値	0.9491	0.9434	0.9434	0.9434	0.9457	0.9457	0.9370	0.9003	0.8303	0.7427	0.6054

パラメータe=0.25, f=0.25, g=0.5, a=0.5, b=0.5

を仮設定しておく必要があるが，なるべく再現率と適合率が極端にアンバランスになることのないような値を選定するのが良い．本研究では，T_Y の言語に対し，検出された T_S での同一言語との言語総合類似度の最大値が 0.5 未満の場合は，それらが真の同一言語である可能性が小さいと考え，仮に ρ=0.5 とした．閾値 ρ の妥当な値に関しては，a, b, Δ の値が定まった後，さらに検討していく．

このようにテストデータを用いて繰り返し計算を行い，集計した結果の一部を表 6.2 に示す．Δ および a, b による影響の傾向を示すため，Δ=0.01,0.05,0.10,0.15,0.20, 0.25，また後に採用される Δ=0.13 と設定したときの TP, FP, FN, TN および再現率，適合率，F 値を載せている．

表 6.2 では，各 Δ の設定値に対する F 値が最も高く，同じ F 値では適合率が最も高いところをダークグレーで塗りつぶしてある．Δ=0.20 までの範囲では，$(a, b) = (0.5, 0.5)$ のときに，良い結果が出る傾向にあることが分かる．Δ の設定値に関しては，Δ=0.10 と Δ=0.13 のところにピークがあり，その後下がる傾向にある．Δ=0.10 と Δ=0.13 のときに同じ結果が出ているため，Δ=0.10, a=0.5, b=0.5 と Δ=0.13, a=0.5, b=0.5 を候補とする．

Δ=0.10, a=0.5, b=0.5 と Δ=0.13, a=0.5, b=0.5 の設定値の下で，閾値 ρ=0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80 と 0.05 刻みで増やし計算を行った．F 値の結果を見

てみたところ，ρ=0.55 のところに F 値のピークがあったため，さらに ρ=0.51, 0.52, 0.53, 0.54 に設定し，計算を行った．その結果を表 6.3 に示す．

表 6.3 に示すように，ρ は 0.54 と 0.55，Δ は 0.10 と 0.13 のそれぞれに設定したとき，F 値と適合率が同等な結果が出ている．同じ F 値と適合率では，閾値が低いほうが良いため，ρ は 0.54 に決定することにした．また，Δ=0.10 と Δ=0.13 では差が出ていないため，とりあえず両方を採用し，この後に行う T_Y と T_S に含まれる同一言語ペアの検出処理でパラメータの値をこの両方に合わせて設定し変え，得られる結果の良い方を選択することにする（表 5.2 を参照されたい）．つまり Δ=0.10, a=0.5, b=0.5, ρ=0.54 または Δ=0.13, a=0.5, b=0.5, ρ=0.54 である．

6.5.3 パラメータ値設定合理性の検討

本研究では，言語総合類似度をそれぞれ重み係数 a と b とする言語名の類似度と言語系統分類の類似度の重み付き加重平均値として定義している．(a, b) の値を変化させ，テストデータを用いて実験した結果を表 6.2 に示している．表 6.2 の結果から分かるように，再現率と適合率および F 値に傾向が現れている．Δ が同じ値の下で，言語系統分類の類似度の重み係数 b のウェートを大きくするにつれ，再現率は下がり，その一方で適合率は上がる．a=1.0, b=0.0 のときに再現率が最も高く，適合率が最も低い．逆に，a=0.0, b=1.0 のときに再現率が最も低く，適合率が最も高くなっている．これは本研究で提案している言語系統分類の類似度が言語の同一性判定に有用であることを説明している．a=1.0, b=0.0 という値設定は，すなわち言語総合類度において実質上言語系統分類を考慮しないことである．言語名の類似度のみによって同一言語かどうかを決定することになり，言語名が似ていれば，本来同一言語ではない 2 つの言語も同一言語ペアとして出力されることになるため，再現率が高い一方，適合率が低い．

また，本研究ではまず言語名の類似度の値が Δ という幅に入る言語の組合せを抽出し，次に言語総合類似度によって同一言語かどうかの判定をしている（6.4 を参照されたい）．表 6.2 から，Δ が 0.10 と 0.13 のときに F 値が最も高い値 0.9491 を出しており，次は Δ が 0.01, 0.05, 0.20 のときで F 値が同じ 0.9462 の値になっており，さ

らに次は Δ が 0.15, 0.25 のときで F 値が 0.9434 となっている．この変化は軽微ではあるが，この Δ というパラメータを取り入れることの有用性が現れていると言える．Δ の効果が限定されているのは，6.4.1 で言及しているような言語名の類似度と言語総合類似度の逆転現象が多く存在しているわけではないためである．しかし，そのような逆転現象が存在している以上，Δ の導入は必要である．そして，このテストデータを用いたパラメータの値設定実験を通して，その有効性も現れており，また a, b と同様に Δ の値設定が言語の同一性判定に大きな影響をおよぼすことが見えてきた．

6.6 実験結果および考察

6.6.1 同一言語ペアの検出結果

パラメータ e, g, f は e=0.25, f=0.25, g=0.5 に設定し，パラメータ a, b, Δ および閾値 ρ をそれぞれ（i）Δ=0.10, a=0.5, b=0.5, ρ=0.54，または（ii）Δ=0.13, a=0.5, b=0.5, ρ=0.54 に設定し，T_Y と T_S に対し処理を行ったところ，次の結果が得られた．T_Y の総言語数 2,869 のうち，前者の（i）のときは 2,637 言語，後者の（ii）のときは 2,648 言語について，T_S での同一言語が見つかった．つまり，パラメータを Δ=0.13, a=0.5, b=0.5, ρ=0.54 と設定したときに，最も良い結果が得られた．その結果を表 6.4 に示す．

T_Y (総言語数 2,869) と T_S の言語に対し，図 6.5 のアルゴリズムに従って処理し，同一言語ペアとして出力された数は 2,687 であった．この 2,687 の言語ペアが真の同一言語ペアかどうかをチェックしたところ，真の同一言語ペアの数（TP）が 2,648 で，真の同一言語ペアではない数（FP）が 39 であった．検出率 ($= \frac{TP}{\text{総言語数}}$) は約 92.3%($= \frac{2{,}648}{2{,}869}$) で，適合率 ($= \frac{TP}{TP+FP}$) は約 98.5% ($= \frac{2{,}648}{2{,}648+39}$) となる．検出率は手法 I の約 88% より 4%以上増えたことになる．

表 6.4: 手法 II による同一言語ペアの検出結果

総言語数	検出同一言語ペア数	TP	FP	検出率 (TP/総言語数)	適合率 (TP/TP+FP)
2,869	2,687	2,648	39	**92.3%**	98.5%

パラメータ*e=0.25, f=0.25, g=0.5,* Δ =0.13, a=0.5, b=0.5, ρ =0.54

なお，表 6.4 に再現率は掲載していない．それは，T_Y と T_S に含まれる同一言語ペアの総数は未知で，本研究を通してつかみたいことであり，現段階では算出できないためである．

6.6.2 考察

本研究の提案した手法により，従来手法では同一性判定ができなかった図 6.1 (1) の (y_1, s_1)，図 6.1 (2) の (y_2, s_2)，図 6.1 (3) の (y_3, s_3) のような言語が同一言語として判定され，検出できるようになった．

図 6.1 には 第 5 章 で提案した手法 I の問題点として，2 つのケースに分けて指摘している．ケース 1 としての図 6.1 (1) の (y_1, s_1) は，T_Y と T_S での第一言語名がそれぞれ “ARABIC, YEMENI” と “Arabic, Judeo-Yemeni” で，言語名の類似度が 0.67 である．言語系統分類は完全に一致しており，語族類似度と親類似度はそれぞれ 1 で，兄弟類似度は 0.39 になる．言語系統分類の類似度として 0.7 が算出され，言語総合類似度が 0.61 になり，閾値 $\rho\,(=0.54)$ を上回るため，同一言語として判定された．

ケース 1 の例では，ほかにも多数検出されている．たとえば，T_Y と T_S での言語系統分類が同じで第一言語名がそれぞれ “GREEK, CLASSICAL” と “Greek, Ancient” となっている 2 つの言語も同一性が確定された．

次に，ケース 2 の場合について述べる．図 6.1 (2) の (y_2, s_2) は第一言語名が同じであり，本来同一言語ペアである．しかし，言語系統分類がそれぞれ “ARABIC-BASED CREOLE” と “Creole”/“Arabic based” で，手法 I では言語系統分類の類似度が 0 に

なるため，同一性を肯定することができない．本研究では，言語系統分類の類似度の定義を見なおすことによって，語族類似度が 0.3, 親類似度が 0.7，兄弟類似度が 0.67 の加重平均値として算出される新しい言語系統分類の類似度は 0.58 になり，言語総合類似度が 0.79 になるため，閾値 ρ を上回り，同一性が肯定されるようになった．

図 6.1 (3) の (y_3, s_3) も同様である．また，ケース 2 に含まれる例として，T_Y と T_S における第一言語名が同じで，それぞれ “LINGAO” と “Lingao”（言語コードが onb）である 2 つの言語が，言語系統分類がそれぞれ “Daic”/“Kadai” と “Tai-Kadai”/“Kam-Tai”/“Be-Tai”/“Be” で，手法 I では言語系統分類の類似度が 0 になるため，同一言語として判定されなかった．“LINGAO” と “Lingao” の語族類似度は 0.3，親類似度は 0，といずれも低い値である．一方いずれも兄弟はいない．本研究では，共に兄弟がいない場合は言語名の類似度を兄弟類似度としているため，兄弟類似度は 1，言語系統分類の類似度は 0.56 が算出され，言語総合類似度は 0.78 になるため，閾値 ρ を上回り，同一言語として判定されるようになった．

本研究の提案手法により，本来同一言語ではない言語が間違って同一言語として出力されることもあった．表 6.4 にある FP の値 39，つまり 39 の組合せの言語がそれにあたる．その中の一例として，T_Y と T_S での第一言語名がそれぞれ “SARDINIAN” と “Armenian”（言語コードが hye）という 2 つの言語を挙げる．この 2 つの言語の言語名の類似度は 0.67 で，言語系統分類はそれぞれ “Indo-European”/“Italic”/“Latino-Faliscan”/“Romance”/“Southern”/“Sardinian” と “Indo-European”/“Armenian” である．語族が同じ “Indo-European” で，また親の名前がそれぞれ “Sardinian” と “Armenian” で，さらに偶然に共に兄弟のいない一人だけの子である．語族類似度が 1，親類似度と兄弟類似度がそれぞれ 0.67 で，言語系統分類の類似度として 0.75 が算出されたため，言語総合類似度が 0.71 になり，誤って同一言語として判定された．そこで，はたして T_Y の “SARDINIAN’ の同一言語が T_S において本当に存在しているかどうかを調査したところ，次のようなことがわかった．T_S では，“Indo-European”/“Italic”/“Romance”/“Southern”/“Sardinian” の分類の下で，互いに兄弟となる 4 つの言語が存在しており，それらの第一言語名はそれぞれ “Sardinian, Campidanese”, “Sardinian, Gallurese”, “Sardinian, Logudorese”, “Sardinian, Sassarese” である．言語 “SARDINIAN” との言語名の類似度がいずれも 0.5 で，“SAR-

DINIAN” と “Armenian” の言語名の類似度の 0.67 より低く，なお兄弟類似度も低い値が算出されたため，前述のような間違ったペアリングがされてしまった．

本研究では，言語系統分類の類似度を重み係数がそれぞれ e=0.25, f=0.25, g=0.5 となる語族類似度，親類似度，兄弟類似度の重み付き加重平均として定義しており，この中で特に兄弟類似度に大きくウェイトをおいている．表 6.4 に示している T_Y と T_S のすべての言語に対する処理結果から，語族類似度，親類似度に比べ，兄弟類似度に大きくウェイトをおくことが，手法 I では同一性判定ができなかった同一言語ペアも検出されたことから，言語の同一性を評価する要素として兄弟情報を考慮し，かつ偏重することは，方向性が正しく，効果的であるといえる．一方，“SARDINIAN” と “Armenian” のような誤判定の例が出ているように，偶然に共に兄弟がいないが故に，兄弟類似度が過大に評価され，誤判定を導いてしまうようなこともある．このような状況発生の蓋然性を低下させる工夫が必要であることを気づかせてくれた．また，本研究では，語族類似度，親類似度，兄弟類似度の重みがそれぞれ e=0.25, f=0.25, g=0.5 の値設定の下で実験を進めてきたが，この設定値が最適であるかどうかは，さらなる実験を通して検討することが必要であり，より良い結果を導く設定値が見つかる可能性が現段階では否められない．

一方，T_Y の第一言語名が “SARDINIAN” の言語に対し，T_S では，第一言語名がそれぞれ “Sardinian, Campidanese”, “Sardinian, Gallurese”, “Sardinian, Logudorese”, “Sardinian, Sassarese” の 4 つの言語になっている．つまり，T_S では “SARDINIAN” の 4 つの異なる変種（方言）を言語として扱っている [Gordon 05, 亀井 96]．言語の定義は元々曖昧なところがあるといわれており，この例からも，異なる学者による言語データにおいて，言語とするか，方言とするかについて，扱いが分かれていることが確認できた．

本研究ではまず言語名の類似度の値が Δ という幅に入る言語の組合せを抽出し，次に言語総合類似度によって同一言語かどうかの判定をしている．言語名の類似度と言語総合類似度の両指標および同一言語ペアの検出処理において Δ というパラメータを取り入れることの必要性に関しては，6.5.3 においてすでに考察している．これに対し，先に言語総合類似度の値が Δ という幅に入る言語の組合せを抽出し，次に言語名の類似度によって同一言語かどうかを判定する処理手順も考えられる．この点に関しても，さらなる実験を通して検討してみたい．

6.7 まとめ

本章では，言語系統分類の類似度に新たに兄弟情報を考慮し，さらに言語名の類似度と言語系統分類の類似度の加重平均となる言語総合類似度を導入した．言語の同一性判定においては，パラメータの導入により，言語名の類似度と言語総合類似度の 2 つの指標を分離して用いるという従来研究の欠点を克服した新しい手法を考案した．その結果，T_Y の約 92%の言語に関し，T_S での同一言語が見つかった．これは，手法 I より検出率が 4%以上向上している．適合率は約 98%であり，言語の同一性判定の精度もかなり良い結果である．このことから，兄弟情報を考慮した言語系統分類の類似度は，言語の系統分類の比較において，より良くその特徴を捉えており，有用である．また，言語名の類似度と言語総合類似度による同一言語ペアの検出方法も効果的である．

また，本章で提案した言語総合類似度は，パラメータの値設定に大きく左右されるところがあり，値設定が重要なポイントとなる．このことについては，この後の 7.1 において述べる．

第7章 おわりに

本章では，まず今後の課題として次の3つの点について述べる．(1) 手法IIで定義した言語系統分類の類似度と言語総合類似度の計算ではパラメータ e, f, g などが使われているが，7.1 ではこれらのパラメータを調整することの必要性について述べる．(2) 手法IIの実験結果の考察により，T_Y の言語が T_S では言語の親である言語グループになっているケースがあることが判明した．7.2 ではこの対応について述べる．(3) 7.3 では，手法Iで提案した言語名の類似度の計算における語類似度計算手法の検討の必要性について述べる．最後に，7.4 において本論文をまとめる．

7.1 言語系統分類の類似度計算のパラメータ調整

言語同一性判定に関し，本論文で提案した手法IIでは，言語系統分類の類似度は式(6.5)のように，重み係数がそれぞれ e, f, g となる語族類似度，親類似度，兄弟類似度の重み付き加重平均として，また言語総合類似度は式(6.6)のように，重み係数がそれぞれ a と b となる言語名の類似度と言語系統分類の類似度の重み付き加重平均として定義している．このほかに，同一言語ペアの検出処理において，言語名の類似度の値が Δ という幅に入る言語の組合せを抽出し，次に言語総合類似度によって同一言語かどうかの判定手法を取っており，ここでも Δ というパラメータを用いている（6.4 を参照されたい）．

表6.4に示している実験結果は，e=0.25, f=0.25, g=0.5 と設定した前提において，テストデータを用いて，もっとも良い結果をもたらすパラメータ Δ, a, b および閾値 ρ の値を決定し，それらのパラメータの値をもって得られた結果である．もしパラ

メータ e=0.25, f=0.25, g=0.5 という前提となる設定値を変更すれば，ほかのパラメータ Δ, a, b および閾値 ρ の値も変化し，引いては表 6.4 に示している実験結果も変わる可能性がある．これについては 6.6.2 においても述べた．

つまり，本論文で提案した言語総合類似度は，パラメータの値設定に大きく左右され，その値の設定が重要なポイントとなる．現状のパラメータの値設定が合理性を有するものの，最良であるとの証明はなされておらず，また問題の設定上，証明は困難である．今後はさらなる実験を通して，アルゴリズムの調整およびより良い結果をもたらすパラメータのチューニング作業をしていきたい．

7.2 同一言語ペア検出方法の見直し

本論文で提案した手法 II による実験結果の考察を通して，言語と方言の扱いの違いによる言語の同一性判定上の問題点が表面化してきた．このことを図 7.1 に示す．図 7.1 では，y は T_Y ではリーフノードの言語となるが，T_S では $y' = y$ である y'_1, y'_2, $\cdots$, のように，さらに細分化されており，リーフノードの言語ではなく，インナーノードの言語グループになっている．つまり，$y'_1, y'_2, \cdots,$ は T_S では言語として扱われているが，T_Y では方言として扱われているため，データとして編入されていない．

本研究では，問題が複雑化するのを避けるため，2 つの言語データにおいて同一言語が存在しているのであれば，必ず両方ともにリーフノードとして存在しており，かつ 1 対 1 の関係を持つという仮定の下で，同一言語ペアの検出処理を行なっている（6.4.2 を参照されたい）．つまり，本論文で提案した手法では，このようなケースは考慮に入れておらず，T_Y と T_S のリーフノードのみをマッチング処理の対象としている．よって，このような T_Y の言語 y に対し，T_S における同一言語である y' は検出されることはない．また，その逆のケース，つまり T_S では方言として扱われているが，T_Y では言語として扱われているデータの存在も否定できない．また，現状はリーフノードのみを対象としている同一言語ペアの検出方法を，インナーノードまで処理対象を広げる必要があると考える．この問題に対し解決方針を定め，本手法をさらに発展させていきたい．

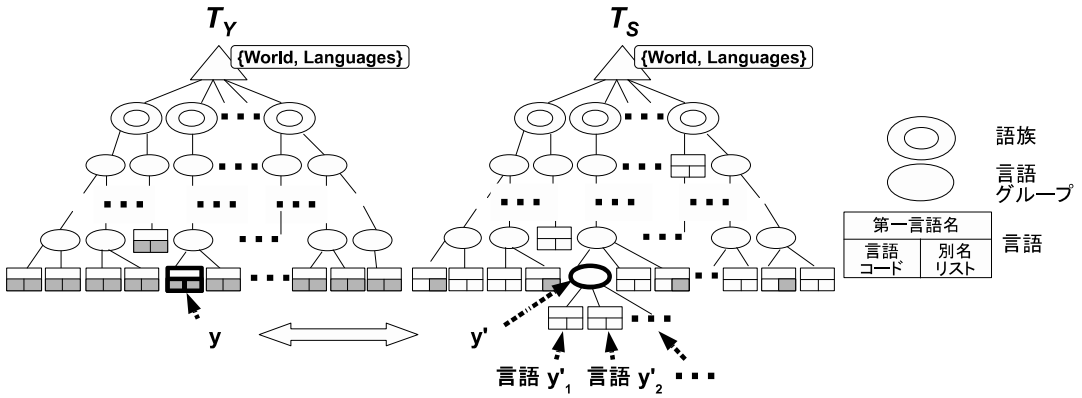


図 7.1: 言語と方言の扱いの違い

7.3 語類似度計算手法の検討

本論文の手法 I で提案した言語名の類似度は本研究におけるもっとも基本的，かつ核心的な類似度である．なぜならば，言語系統分類の類似度や言語総合類似度などもこの言語名の類似度に関連しており，それらの値の大きさがこの言語名の類似度の値によって影響されるためである．

本論文の言語名の類似度は Monge-Elkan 法の重畳構造に基いて設計されている（5.3.1 と 5.3.2 を参照されたい）．Monge-Elkan 法の式 (3.4) 中の $match(A_i, B_j)$ に相当する語類似度の計算法としては，編集距離を用いている．これを用いたのは，数多くある文字列類似度計算の基本的な手法のなかで，どの手法が本研究の言語名の類似度により適合しているかが未知であり，そのため，もっとも広く使われている編集距離に基づく方法を用いることにしたからである．

手法 I と手法 II のいずれも実験結果はまずまずの好結果を出していることから，編集距離に基づく語類似度の定義は見当はずれではないようである．しかし，これは言語名の類似度の計算に対し，より良い計算法がほかに存在することを否定するものではない．

Monge-Elkan 法は文字列類似度計算の重畳構造を用いている．これにより，$match(A_i, B_j)$ に相当する計算手法は，対応する問題の特徴に合わせて，変更することが可能になる．これが，Monge-Elkan 法を使用する利点である．実際に，人名マッチングなどの分野では，より高いマッチング率を実現するため，そのような実験が多く報告されている．

本研究において，今後の課題として，語類似度に用いている編集距離に基づく手法のかわりに，N-gram などの手法を取り入れ，言語名の類似度計算にとって親和性の高い文字列類似度の基本的計算手法について調査していきたい．

7.4 まとめ

本研究では，言語の同一性問題に対し，手法 I : 木構造と文字列類似度に基づく手法，手法 II : 言語名と言語系統分類の総合的尺度に基づく手法，の 2 つの手法を提案した．

手法 I では，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，言語系統木という木構造について定義し，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提案した．また，言語名と言語系統分類がいずれも曖昧な性質をもつことに対し，文字列類似度に基づく言語名の類似度と言語系統分類の類似度を定義した．これらの類似度に基づく言語の同一性判定のルールを定め，2 つの言語系統木に含まれる同一言語を見つけ出す手法を提案した．

手法 I は合わせて 88% の言語についての同一性判定ができた．そのうち，52% は言語名の類似度と言語系統分類の類似度の適用による結果であった．このことから，手法 I で提案した言語名の類似度と言語系統分類の類似度とゆれのある言語の検出手法は効果的である，といえる．

手法 II は，同一言語ペアの検出率を向上させるために，手法 I を発展させることを指針とし，第一に言語系統分類の類似性評価の改善に着目し，兄弟情報を考慮した新しい言語系統分類の類似度について定義した．第二に言語総合類似度という概念を導入した．言語の同一性判定においては，パラメータの導入により，言語名の

類似度と言語総合類似度の 2 つの指標を分離して用いるという手法 I の欠点を克服した新しい方法を考案した．その結果，T_Y の約 92%の言語に関し，T_S での同一言語が見つかった．これは，手法 I より検出率が 4%強向上している．このことから，兄弟情報を考慮した言語系統分類の類似度は，言語の系統分類の比較において，より良くその特徴を捉えており，有用である．また，言語名の類似度と言語総合類似度による同一言語ペアの検出方法も効果的である．

今後は，上で述べた課題について調査を行い，本論文で提案した手法をさらに発展させていきたい．

1.1 で述べたように，本研究に取り組むようになったきっかけは，2 つの表形式の言語属性データによって GIS 空間データを生成するということであった． 5.5.2 に示しているように，本研究の手法を使わない場合，すなわち完全一致の手法による場合は，36% の言語についてのみ同一性判定が可能であった．本研究で提案した手法によって同一性判定が可能な言語が 92% になり，大きく改善することができた．

GIS は多角的な時空間検索と分析の機能をもっているが，もっとも基本的な応用事例として，世界諸言語の言語特徴（例えば，語順）を地図化することができる．我々は以前，完全一致の手法によって同一性が判明した Yamamoto-Data の 36% の言語について，節語順タイプの地理的分布について語順地図を作成したことがある．今後は本研究の成果である 92% の言語について語順地図の作成を実施したい．GIS の多角的な時空間検索機能を活かした言語類型論研究における GIS のさらなる利用は，より多くの GIS 空間データの入手や整備など課題が山積みであるが，本研究により，その険しい長い道程の一歩を踏み出せたことには間違いないといえよう．

さらに，上で述べた課題のなかで，特に 7.3 で述べた語類似度計算手法の検討は，言語同一性判定の検出率のさらなる向上の可能性を探ることになるとともに，言語名の類似性の特徴の調査にもなる．言語名の類似性の特徴を判明させることにより，人名マッチングなどの分野で報告されている人名類似性の特徴との比較ができる．2.2 で述べたように，言語名の類似性（非類似性）は表記ゆれに起因するもので，人名も同様である．我々人間自身の名前である人名と我々人間が話す言葉の名前である言語名，この両者の表記ゆれの特徴に関する研究は，両者ともに我々と密接な関係にあるゆえ，大変興味深いテーマである．今後はこのテーマについても取り組込んでいきたい．

謝 辞

山口大学大学院理工学研究科教授 松野浩嗣先生には，博士後期課程の受け入れをご快諾いただき，研究の機会を賜りました．本研究を展開するにあたっては，ご多忙にもかかわらず，終始懇篤なるご指導ご鞭撻を賜り，研究者としての姿勢や心構えなど，身をもってご教示いただきました．先生には，懇切丁寧に論文作成のご指導や文章の添削などにあたっていただきました．複数回にわたり，国際学会や研究会などに参加する機会に恵まれ，研究発表の場における貴重な体験が得られ，研究者として自分なりの成長ができたことも，ひとえに先生の多方面にわたるご指導，ご支援の賜物と感謝しております．また，研究打ち合わせ時間の調整をしていただくなど，多分にご配慮をいただきました．公私にわたるご指導ご鞭撻，また惜しみなくご支援を賜り，研究者としての道を切り開いていただきましたことに，心より深く感謝し，この場をお借りして，厚く御礼を申し上げます．

山口大学人文学部准教授 乾秀行先生にも，研究の初期段階から多くのご指導とご教示を賜りました．人文学部開講の言語学関連授業の聴講をお許しいただき，言語学の知識をご教授いただきました．この度の学位取得に際しても，副査として本論文に対し多くの有益なご教示とご助言を賜りました．心より深く感謝し，厚く御礼を申し上げます．

本論文をまとめるに際し，学位取得審査の副査として本論文に対する真摯なるご検討と数多くの有益なご教示とご助言を賜りました，山口大学大学院理工学研究科教授 朝日孝尚先生ならびに菊政勲先生，ならびに同理工学研究科准教授 末竹規哲先生に厚く御礼申し上げます．末竹規哲先生には，博士後期課程の授業においてもお世話になりました．先生から公私にわたるご指導とご支援を賜りましたことに，心より深く感謝し，この場をお借りして，厚く御礼を申し上げます．

弘前大学人文学部教授 山本秀樹先生には，ご研究の成果である言語データの使用をご快諾いただき，また多くのご教示とご助言を賜りました．言語学関係の雑誌に投稿した際には，論文の作成に関連し，細部にわたる丁寧なご指導を賜りました．ま

た，研究会に参加した際にもお世話になりました．本研究は先生のご研究成果の下で進められてきており，いわば「巨人の肩の上に立つ」ような状況で安定した研究を遂行できました．心より深く感謝し，厚く御礼を申し上げます．

山口大学大学情報機構メディア基盤センター准教授 杉井学先生にも，研究の初期段階から多くのご指導とご助言を賜りました．また，研究会に参加した際にもお世話になりました．厚く御礼を申し上げます．

筑波大学大学院人文社会科学研究科教授 池田潤先生にも，研究会に参加した際にお世話になり，励ましの言葉をいただきました．御礼を申し上げます．

山口大学大学院理工学研究科博士後期課程の授業を受講させていただきました，山口大学時間学研究所教授 藤澤健太先生，同理工学研究科元准教授 松村澄子先生ならびに同理工学研究科准教授 浦上直人先生に，御礼を申し上げます．

本研究を遂行するにあたり，山口大学大学院理工学研究科自然科学基盤系学域情報科学分野ネットワーク科学研究室の皆様には多くのご協力をいただきました．共に研究活動をしました伊藤佳奈氏（平成19年度卒業生）と冨永理恵氏（平成21年度卒業生），学会発表のポスター作成および印刷などでご協力をいただきました三藤なつ美氏と宮川千種氏（両氏はともに博士前期課程平成19年度修了生），研究会参加の際にお世話になりました角朝香氏（博士前期課程平成21年度修了生），また本論文を完成させるにあたり，ご協力をいただきました森渉氏（博士前期課程在学），さらに，日ごろからお世話になり，研究会参加の際にもお世話になりました同研究室前技術補佐員 加藤玲子氏，同様に日ごろからお世話になり，ご協力いただきました同研究室技術補佐員　安永久美氏ならびに藤井好恵氏，の各氏へ御礼を申し上げます．

筆者の勤務先の山口短期大学では，所属する情報メディア学科の学科長兼学生募集委員長 佐藤和雅先生を始め，同学科教授兼学生部長 中村綱幸先生，同学科教授兼国際交流委員長 河村殖先生，同学科准教授 林孝哉先生，同学科講師 日置智子先生，また児童教育学科教授 藤澤初美先生の各氏には，本論文の作成にあたって，業務などでご迷惑をお掛けしたにもかかわらず，ご理解を示していただき，始終ご配慮と暖かい励ましをいただきました．厚く御礼を申し上げます．また，各委員会業務に関連し，事務の皆様方にも多くのご配慮をいただきました．厚く御礼を申し上げます．

また，筆者の良き理解者と相談相手で，筆者の話しに真摯に耳を傾け，いつも暖かい励ましをくださいました友人の李雪雲氏に，深く感謝し，御礼を申し上げます．

なお，本研究の一部は日本学術振興会科学研究科研費 挑戦的萌芽研究 23650129 の助成を受けたものです．ここに記して感謝の意を表します．

終わりに，筆者の健康を気遣い，協力を惜しみなくくれ，またいつも励ましてくれた実母ならびに義母，また叔母に，心から深く感謝しております．筆者の一人息子は，筆者が博士課程に入学した当初は高校生でした．肩を並べて勉強した日々を懐かしく思い出します．筆者の研究活動に理解を示し，励ましと協力をくれたことに，心から深く感謝しております．そして，筆者の教育や研究活動に理解を示し，多方面にわたり，協力と支援を惜しみなくくれ，また先輩研究者としても，日ごろから教示と励ましをくれた夫に，心から深く感謝しております．

最後に，筆者を支えて来られたすべての方々に，重ねて感謝を申し上げ，本研究の謝辞とさせていただきます．

平成 25 年 8 月

呉　靭

参考文献

[Akutsu 06] Akutsu, T.: A relation between edit distance for ordered trees and edit distance for Euler strings, *Information Processing Letters*, Vol. 100, pp. 105–109 (2006)

[Akutsu 10] Akutsu, T., Fukagawa, D., and Takasu, A.: Approximating tree edit distance through string edit distance, *Algorithmica*, Vol. 57, No. 2, pp. 325–348 (2010)

[Asher 07] Asher, R. E. and Moseley, C. eds.: *Atlas of the world's languages*, Routledge, New York, 2nd edition (2007)

[Bickel 07] Bickel, B.: Typology in the 21st century: major current developments, *Linguistic Typology*, Vol. 11, No. 1, pp. 239–251 (2007)

[Bickel 08a] Bickel, B.: A general method for the statistical evaluation of typological distributions, *Manuscript, Universität Leipzig* (2008)

[Bickel 08b] Bickel, B. and Witzlack-Makarevich, A.: Referential scales and case alignment: reviewing the typological evidence, *Scales*, pp. 1–37 (2008)

[Bille 05] Bille, P.: A survey on tree edit distance and related problems, *Theoretical Computer Science*, Vol. 337, pp. 217–239 (2005)

[Christen 06] Christen, P.: A comparison of personal name matching: Techniques and practical issues, in *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pp. 290–294 (2006)

[Cohen 03] Cohen, W., Ravikumar, P., and Fienberg, S.: A comparison of string distance metrics for name-matching tasks, in *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 73–78 (2003)

[Croft 08a] Croft, W. and Poole, K.: Multidimensional scaling and other techniques for uncovering universals, *Theoretical Linguistics*, Vol. 34, pp. 75–84 (2008)

[Croft 08b] Croft, W. and Poole, K. T.: Inferring universals from grammatical variation: Multidimensional scaling for typological analysis, *Theoretical Linguistics*, Vol. 34, pp. 1–37 (2008)

[Croft 09] Croft, W.: Methods for finding language universals in syntax, in *Universals of language today*, pp. 145–164, Springer (2009)

[Cysouw 07] Cysouw, M.: New approaches to cluster analysis of typological indices, *Exact methods in the study of language and text*, pp. 61–76 (2007)

[DeGraff 01] DeGraff, M.: *Language creation and language change: Creolization, diachrony, and development*, The MIT Press (2001)

[Donohue 11] Donohue, M., Musgrave, S., Whiting, B., and Wichmann, S.: Typological feature analysis models linguistic geography, *Language*, Vol. 87, No. 2, pp. 369–383 (2011)

[Dunn 05] Dunn, M., Terrill, A., Reesink, G., Foley, R. A., and Levinson, S. C.: Structural phylogenetics and the reconstruction of ancient language history, *Science*, Vol. 309, No. 5743, pp. 2072–2075 (2005)

[Euzenat 04a] Euzenat, J.: An API for Ontology Alignment, in *Proceedings of the third international semantic web conference*, Vol. 3298, pp. 698–712 (2004)

[Euzenat 04b] Euzenat, J. and Valtchev, P.: Similarity-based Ontology Alignment in OWL-Lite, in *Proceedings of the 16th European Conference on Artificial Intelligence*, pp. 323–327 (2004)

[Euzenat 07] Euzenat, J. and Shvaiko, P.: *Ontology Matching*, Springer-Verlag (2007)

[Galvez 07] Galvez, C. and Moya-Anegón, F.: Approximate personal name-matching through finite-state graphs, *Journal of the American Society for Information Science and Technology*, Vol. 58, No. 13, pp. 1960–1976 (2007)

[Gordon 05] Gordon, J. e., Raymond G. ed.: *Ethnologue: Languages of the World*, SIL International, Texas, fifteenth edition (2005)

[Gray 03] Gray, R. D. and Atkinson, Q. D.: Language-tree divergence times support the Anatolian theory of Indo-European origin, *Nature*, Vol. 426, No. 6965, pp. 435–439 (2003)

[Greenberg 63] Greenberg, J. H.: Some universals of grammar with particular reference to the order of meaningful elements, *Universals of language*, Vol. 2, pp. 73–113 (1963)

[Haspelmath 05] Haspelmath, M., Dryer, M. S., and Bernard Comrie (eds.), nd D. G.: *The world atlas of language structures*, Oxford University Press (2005)

[Horie 06] Horie, K.: The world atlas of language structures, 言語研究（Gengo Kenkyu）, Vol. 130, pp. 83–87 (2006)

[Ichise 08] Ichise, R.: Machine Learning Approach for Ontology Mapping using Multiple Concept Similarity Measures, in *Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science*, pp. 340–346 (2008)

[Islam 08] Islam, A. and Inkpen, D.: Semantic text similarity using corpus-based word similarity and string similarity, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 2, No. 2, p. 10 (2008)

[Islam 09] Islam, A. and Inkpen, D.: Semantic similarity of short texts, *Recent Advances in Natural Language Processing V*, Vol. 309, pp. 227–236 (2009)

[Jabbour 01] Jabbour, S. and Vercoustre, A.-M.: Wrapping Web Pages into XML Documents, Technical report, CMIS Technical Report 01 (2001)

[Janssen 06] Janssen, D., Bickel, B., and Zúñiga, F.: Randomization tests in language typology, *Linguistic Typology*, Vol. 10, No. 3, pp. 419–440 (2006)

[Jiang 95] Jiang, T., Wang, L., and Zhang, K.: Alignment of trees - an alternative to tree edit, *Theoretical Computer Science*, Vol. 143, pp. 137–148 (1995)

[Jimenez 09] Jimenez, S., Becerra, C., Gelbukh, A., and Gonzalez, F.: Generalized Mongue-Elkan Method for Approximate Text String Comparison, *CICLing 2009. LNCS*, Vol. 5449, pp. 559–570 (2009)

[Johnson 11] Johnson, K.: *Quantitative methods in linguistics*, Wiley-Blackwell (2011)

[Levenshtein 66] Levenshtein, V. I.: Binary codes capable of correcting deletions, insertions, and reversals, *Soviet Physics Doklady*, Vol. 10, No. 8, pp. 707–710 (1966)

[Lewis 09] Lewis, M. P. e. ed.: *Ethnologue: Languages of the World*, SIL International, Texas, sixteenth edition edition (2009)

[Monge 96] Monge, A. and Elkan, C.: The field-matching problem: algorithm and applications, in *Proceedings of the second international Conference on Knowledge Discovery and Data Mining*, pp. 267–270 (1996)

[Navarro 01] Navarro, G.: A guided tour to approximate string matching, *ACM Computing Surveys*, Vol. 33, No. 1, pp. 31–88 (2001)

[Neil C. Jones 07] Neil C. Jones 著, 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 (2007)

[Nicholls 08] Nicholls, G. K. and Gray, R. D.: Dated ancestral trees from binary trait data and their application to the diversification of languages, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 70, No. 3, pp. 545–566 (2008)

[Nichols 08] Nichols, J. and Warnow, T.: Tutorial on computational linguistic phylogeny, *Language and Linguistics Compass*, Vol. 2, No. 5, pp. 760–820 (2008)

[Piskorski 08] Piskorski, J., Wieloch, K., Pikula, M., and Sydow, M.: Towards person name matching for inflective languages, in *WWW 2008 Workshop NLP Challenges in the Information Explosion Era* (2008)

[R.M.W. ディクソン 01] R.M.W. ディクソン 著, 大角 翠 訳 : 言語の興亡, 岩波新書 (2001)

[Sellers 80] Sellers, P. H.: The theory and computation of evolutionary distances: Pattern recognition, *Journal of Algorithms*, Vol. 1, No. 4, pp. 359–373 (1980)

[Stoilos 05] Stoilos, G., Stamou, G., and Kollias, S.: A string metric for ontology alignment, in *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, pp. 624–637 (2005)

[Suzuki 03] Suzuki, T. and Tokuda, T.: Path set operations for clipping of parts of web pages and information extraction from web pages, in *Proceedings of the 15th International Conference on Software Engineering and Knowledge Engineering*, pp. 547–554 (2003)

[Suzuki 04] Suzuki, T. and Tokuda, T.: PSO: A language for Web information extraction and Web page clipping, in *Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 332–335 (2004)

[Tiedemann 99] Tiedemann, J.: Automatic Construction of Weighted String Similarity Measures, *In Proceedings of the Joint SIGDAT Conference on Empirical*

Methods in Natural Language Processing and Very Large Corpora, pp. 213–219 (1999)

[URLa] *Ethnologe* 第 15 版 Web サイト, http://archive.ethnologue.com/15/web.asp

[URLb] *Ethnologe* 第 16 版 Web サイト, http://www.ethnologue.com/

[URLc] *Ethnologe* Web サイト, http://www.ethnologue.com/

[URLd] ASP 技術に関する解説, https://www.microsoft.com/japan/msdn/web/server/asp/asptutorial.aspx

[URLe] 言語コードについて, http://www.ethnologue.com/codes/

[URLf] IT 用語辞典バイナリ, http://www.sophia-it.com/

[Valiente 01] Valiente, G.: An efficient bottom-up distance between trees, in *Proceedings of the 8th International Symposium on String Processing and Information Retrieval*, pp. 212–219 (2001)

[Wang 01] Wang, J. T. and Zhang, K.: Finding similar consensus between trees: an algorithm and a distance hierarchy, *Pattern Recognition*, Vol. 34, No. 1, pp. 127–137 (2001)

[呉 07] 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, 情報処理学会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)

[Yan 02] Yan, X. and Han, J.: gspan: Graph-based substructure pattern mining, in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pp. 721–724 (2002)

[Zaki 02] Zaki, M. J.: Efficiently mining frequent trees in a forest, in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 71–80 (2002)

[Zhang 93] Zhang, K.: A new editing based distance between unordered labeled trees, *Combinatorial Pattern Matching (Lecture Notes in Computer Science)*, Vol. 684, pp. 254–265 (1993)

[アン 02] アンク：HTML タグ辞典, 翔泳社 (2002)

[オフィス 01] オフィス エム:図解標準 最新XMLハンドブック, 秀和システム (2001)

[ノマド 00] ノマド ワークス：詳細 HTML 基本活用事典, 新星出版社 (2000)

[プロジェクト 03] プロジェクト A, 日本VBA 協会:VBA エキスパート教科書 Excel スタンダード, 翔泳社 (2003)

[伊藤 00] 伊藤 英毅：オントロジーを利用した知識の共有／再利用, in *UNISYS TECHNOLOGY REVIEW*, 第 64 巻, pp. 115–129 (2000)

[猿橋 08] 猿橋 大：詳解 HTML タグ辞典, 秀和システム (2008)

[乾 06] 乾 秀行：GIS を使ったクシ・オモ系言語研究, 一般言語学論叢, Vol. 9, pp. 47–58 (2006)

[韓 06] 韓 浩, 徳田 雄洋：Web 部分情報抽出システムとその応用, 日本ソフトウェア科学会第 23 回大会論文集, pp. 4B–1 (2006)

[亀井 96] 亀井 孝ほか編著：言語学大辞典, 三省堂 (1996)

[久保山 04] 久保山 哲二, 宮原 哲浩：木の編集距離を用いた Web ページからの情報抽出, in *The 18th Annual Conference of the Japanese Society for Artificial Intelligence*, pp. 3F2–05 (2004)

[久保山 05] 久保山 哲二, 申 吉浩, 宮原 哲浩 : 木の近似照合に基づく共通構造の発見, in *The 19th Annual Conference of the Japanese Society for Artificial Intelligence*, pp. 2F3–04 (2005)

[久保山 06] 久保山 哲二, 申 吉浩 : 効率的な無順序木の融合可能性判定アルゴリズム, 電子情報通信学会技術研究報告, 第 106 巻, pp. 49–56 (2006)

[後藤 07] 後藤 雅樹 : 言語資源のラッピング支援システムの開発 (2007), `http://www.ai.soc.i.kyoto-u.ac.jp/publications/thesis/B_H18_goto-masaki.pdf`

[江里口 96] 江里口 善生, 木谷 強 : パターンマッチング手法による名称特定処理の有効性の検討, 自然言語処理研究会報告, 第 96 巻, pp. 67–73 (1996)

[溝口 99a] 溝口 理一郎 : オントロジーと知識処理, *Bit* (1999)

[溝口 99b] 溝口 理一郎 : オントロジー研究の基礎と応用, 人工知能学会誌, Vol. 14, pp. 977–988 (1999)

[溝口 99c] 溝口 理一郎 : オントロジー工学基礎論, 人工知能学会誌, Vol. 14, pp. 1019–1032 (1999)

[高橋 02] 高橋 哲朗, 乾 健太郎, 松本 裕治 : テキストの構文的類似度の評価方法について, 自然言語処理研究会報告, Vol. 66, pp. 163–170 (2002)

[高橋 09] 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)

[高村 10] 高村 大也著, 奥村 学監修 : 言語処理のための機械学習入門, コロナ社 (2010)

[斎藤 98] 斎藤 信男, 西原 清一 : データ構造とアルゴリズム, コロナ社 (1998)

[斎藤 05] 斎藤 輪太郎著, 富田 勝監修 : バイオインフォマティクスの基礎 : ゲノム解析プログラミングを中心に, サイエンス社 (2005)

[山田 01] 山田 祥寛 : 10 日でおぼえる XML 入門教室, 翔泳社 (2001)

[山本 03] 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 (2003)

[山本 06] 山本 秀樹 : GIS と言語類型論-世界言語地図に基づく言語研究, 一般言語学論叢, Vol. 9, pp. 31–40 (2006)

[市瀬 02] 市瀬 龍太郎, 武田 英明, 本位田 真一 : 階層的知識間の調整規則の学習, 人工知能学会論文誌, Vol. 17, No. 3, pp. 230–238 (2002)

[市瀬 04] 市瀬 龍太郎, 濱崎 雅弘, 武田 英明 : 階層的分類データを統合するための規則学習機構, 人工知能学会論文誌, Vol. 19, No. 6, pp. 521–529 (2004)

[市瀬 07] 市瀬 龍太郎 : 情報の意味的な統合とオントロジー写像, 人工知能学会論文誌, Vol. 22, No. 6, pp. 818–825 (2007)

[市瀬 08] 市瀬 龍太郎 : オントロジーマッピングにおける有効な特徴の抽出, 人工知能学会全国大会 (第 22 回) 論文集, pp. 2E1–1 (2008)

[小西 07] 小西 いずみ, 三井 はるみ, 井上 文子, 岸江 信介, 大西 拓一郎, 半沢 康 : 方言学の技法, 方言学, 岩波書店 (2007)

[小嶋 93] 小嶋 秀樹, 古郡 延治 : 英語辞書を利用した単語の類似度の計算, 全国大会講演論文集, Vol. 46, No. 3, pp. 93–94 (1993)

[松本 06] 松本 克己 : 世界言語への視座 - 歴史言語学と言語類型論, 三省堂 (2006)

[深川 04] 深川 大路, 阿久津 達也 : 類似度の高い無順序木の比較に対する高速アルゴリズム, 電子情報通信学会技術研究報告, 第 104 巻, pp. 33–40 (2004)

[杉井 06] 杉井 学 : 言語分布地図を扱う GIS 構築, 一般言語学論叢, Vol. 9, pp. 41–46 (2006)

[星合 05] 星合 忠, 山根 康男, 津田 宏 : カテゴリマッチング技術に基づくオントロジーアラインメント問題への取り組み, 人工知能学会論文誌, Vol. 20, No. 6, pp. 437–447 (2005)

[西沢 07] 西沢 夢路：やさしくわかる Excel 関数・マクロ, Excel 徹底活用シリーズ, ソフトバンククリエイティブ, 改訂版 (2007)

[石畑 89] 石畑 清：アルゴリズムとデータ構造, 岩波講座 ソフトウェア科学 3, 岩波書店 (1989)

[川上 06] 川上 高志, 鈴木 寿：決定リストを利用した単語間の類似度計算法 (言語モデル・単語), 情報処理学会研究報告, 第 94 巻, pp. 85–90 (2006)

[浅井 04] 浅井 達哉, 有村 博紀：半構造データマイニングにおけるパターン発見技法, 電子情報通信学会論文誌, Vol. 87, No. 2, pp. 79–96 (2004)

[打越 08] 打越 浩幸：XML Notepad 2007 で XML ファイルを表示／編集する, http://www.atmarkit.co.jp/fwin2k/win2ktips/993xmlnotepad/xmlnotepad.html (2008)

[大西 10] 大西 拓一郎：方言学と GIS, 科学研究費補助金 (基盤研究 (B)) 研究成果報告書, 2006-2009 年度 (2010)

[池上 80] 池上 二良編：言語の変化, 講座言語, 第 2 巻, 大修館書店 (1980)

[池田 06] 池田 潤：GIS と言語研究, 一般言語学論叢, Vol. 9, pp. 1–10 (2006)

[中村 95] 中村 春木, 中井 謙太：バイオテクノロジーのためのコンピュータ入門, コロナ社 (1995)

[中島 05] 中島 平三編：言語の事典, 朝倉書店 (2005)

[田村 07] 田村 悟之, 清田 陽司, 増田 英孝, 中川 裕志：図書館における自動レファレンスサービスシステムの実現： Web 上の二次情報と図書館の一次情報の統合, 情報処理学会研究報告, 第 34 巻, pp. 1–8 (2007)

[土屋 06] 土屋 和人：Excel マクロ&VBA で業務を 3 倍スピードアップする技 70 選, ソシム (2006)

[箱田 06] 箱田 慶太, 市川 宙, 橋本 泰一, 徳永 健伸：構文的類似度を用いた文の検索, 言語処理学会第 12 回年次大会予稿集, pp. 1131–1134 (2006)

[風間 93] 風間 喜代三：印欧語の故郷を探る, 岩波新書 (1993)

[風間 04] 風間 喜代三, 松村 一登, 町田 健, 上野 善道：言語学, 東京大学出版会 (2004)

[北 97] 北 研ニ：確率的言語モデルに基づく多言語コーパスからの言語系統樹の再構築, 自然言語処理, Vol. 4, No. 3, pp. 71–82 (1997)

[落水 93] 落水 浩一郎：ソフトウェア工学実践の基礎ー分析・設計・プログラミング, 実践ソウトウェア開発工学シリーズ, 日科技連出版社 (1993)

[立川 04] 立川 敬行：XML 徹底入門, ねっとテクノロジー解体新書, 電波新聞社 (2004)

[廣安 11] 廣安 知之, 西井 琢真, 吉見 真聡：Smith Waterman 法のアルゴリズム, *IS Report System*, No. 2011021007 (2011)

[齋藤 06] 齋藤 裕明, 古賀 久志, 渡辺 俊典, 横山 貴紀：木編集距離を利用した木データの構造と内容の類似性を反映する手法, 電子情報通信学会技術研究報告, 第 106 巻, pp. 7–12 (2006)

[鍛治 07] 鍛治 優, 七條 達弘, 渡辺 健：やさしくわかる ExcelVBA プログラミング, Excel 徹底活用シリーズ, ソフトバンククリエイティブ, 第 3 版 (2007)