無辞書言語における自動処理: 現代ナワトル語の対話型タグ生成

佐々木 充文 （東京大学大学院・学術振興会特別研究員 DC1）

フィールド言語学では, ときに, 話者数・研究者数ともに少なく, 電子的に利用可能な辞書や言語処理パッケージはおろか, 表記法すら存在しないような言語を記録・保存・分析する必要が生じる. 本稿では, こうした少数言語のうち, 語形変化が比較的複雑な言語のドキュメンテーションを, わずかな労力で自動化する手段として, (i) 多層式の活用形総生成型辞書, (ii) 対話型の半自動スクリプトの利用を検討し, その利用例として, メキシコの複統合的な先住民語であるナワトル語 (Nahuatl) の未記述の一方言におけるグロス生成の例を報告する.

Auto-processing an underdocumented language: Interactive glossing of modern Nahuatl texts

Mitsuya SASAKI (University of Tokyo, JSPS Research Fellow)

Field linguistics often involves the archiving and analysis of minority languages that have very few people speaking and working on them. Many such languages even lack an orthography, let alone digitized dictionaries or program modules ready for computer processing. In order to automate with minimal effort the documentation of such languages, especially those with a relatively rich morphology, this paper considers the use of: (i) a multi-layered dynamic dictionary which pre-generates all conjugated forms, and (ii) a semi-automated interactive script. The case study deals with the automated glossing in an understudied dialect of Nahuatl, a polysynthetic indigenous language from Mexico.

1. まえがき

本稿では, フィールド言語学で通常行われている, 書き起こしテキストの注解作業 (annotation, glossing) を自動化する手段について論じる.

語形変化が複雑で, 1 つの語根に対し無数の活用形が存在する言語では, テキストに現れる語形を単に機械的に辞書と照合するだけでは注解を行うことができない. 以下では, こうした言語の注解作業を自動化しようとする際の問題点と, その解決策を検討し, 例として, ナワトル語 (Nahuatl) の未記述の方言であるプエブラ州イシュキワカン (Ixquihuacan, Puebla) 方言のテキストの自動グロス生成を試みる.

2. 言語記述とその自動化

2.1. フィールド言語学におけるテキスト

世界各地で衰退の一途をたどる少数言語を次代に伝え, 消滅の前に少しでも多くの資料を後世に残すため, フィールド言語学の分野では, 近年, 単なる言語特徴の記述にとどまらない包括的な言語保存活動のありかたとして,「言語ドキュメンテーション」の必要性が叫ばれている. 言語ドキュメンテーションとは, ある言語を, 話者集団や研究者をはじめとする各関係者の協働のもと, 文化的背景や使用場面を含めて体系的に記録し, 将来の研究や復活運動における利用を期して, 容易に検索・共有・再利用できる形で保存するものである [1, 2].

従来より, フィールド言語学者はテキスト（談話, モノローグ, 口承文芸など）の収集・保存を精力的に行ってきたが, 言語ドキュメンテーションにおいては, テキストは, 言語の実際の姿の記録として, 従来以上に重要な位置を占める. また, 従来の記述言語学では, テキスト収集の規模は調査者の個人的リソースに依存することが多く, ほとんどの場合, せいぜい数時間が記録され, そのさらに一部が注解・整形・出版されるにすぎなかったが, 現在では, 記録装置の発達やコンピューターの普及, インターネット上での公開の一般化も手伝って, 複数人の話者を雇用しての体系的な

転写が行われるようになり, 少数言語においても比較的大量のテキストが生成されるようになってきた.

テキストは, 書き起こしただけでは言語資料として利用できず, 検索性と再利用性を確保するためにタグづけなどの注解をほどこす必要がある. コーパスにはコーパス用のタグが必要になるほか, 一般に「○○語のテキスト」として公開される比較的小規模の資料でも, 言語学的な用途に用いるには, 例 (1) のような転写文 (transcription) に対し, (2) のような逐語訳（グロス glosses）を施すのが普通である. したがって, 大量のテキストを効率的に整形する方法が必要になる.

(1) *wan PARA āmo monēxtīs n xīkol...*

(2) *wan PARA āmo Ø-mo-nēxtī-s n xīkol...*
and for NEG 3S-REFL-SHOW-FUT ART devil
'it in order that the devil would not appear...'

2.2. 自動化における問題点

利用者が 1 名ないし数名しかいない少数言語で上記の作業を効率的に行うために, SIL International [3] の配布している *Language Explorer* (*FLEx*) [4] のような言語分析ソフトウェアがしばしば用いられる. これらのソフトウェアでは, 形態素辞書に加え, 各要素の活用クラスや形態音韻規則, 異形態などを自由に定義することができ, 各機能に習熟すれば, 注解作業をある程度自動化することができる.

しかし, 地域や語族によっては, 後述のナワトル語のように, 1 つの語に多くの形態素が含まれる統合的 (synthetic) な言語が多く, こうした言語では, 辞書に登録できる抽象的な語彙形と実際の語形が大きく乖離するため [5, 6], *FLEx* などの形態素辞書ベースの自動処理機能の恩恵を受けにくい. 変化が複雑で不透明になればその分設定すべき項目は増え, 労力に比して精度が上がりにくくなるためである. また, 設定内容がソフトウェアに依存するため, 研究者間での共有は容易でも, 他環境への流用は比較的難しい.

3. ナワトル語の自動グロス生成

統合的な言語のテキストの注解作業を自動化する手段として, 本稿では, (i) 多層式の活用形総生成型辞書, (ii) 対話型の半自動辞書作成スクリプトの利用を提案する. 例として, メキシコの先住民語ナワトル語の一方言であるプエブラ州イシュキワカン (Ixquihuacan) 方言の転写文からグロスつきテキストを生成するスクリプトを作成し, 報告する.[1]

3.1. 言語の概要と自動化状況

ナワトル語はメキシコで話される先住民語の一派で, アメリカ先住民語としては研究が進んでいる部類に属するが, 地域ごとに多様な方言が発達しており, 方言によっては未記述に等しい.[2] いわゆる複統合的言語で, 動詞が主語・目的語の人称にしたがって活用するほか, 名詞抱合を含む各種の語形変化をもち, 注解に際しては, 語形（活用形）からいくつもの規則を用いて語幹や語構成を分析する必要がある [6, 8].

諸方言のうち, もっとも記述が進んでいるのが, 大航海時代の古方言である文献言語の古典ナワトル語 (Classical Nahuatl) で, 電子化された辞書や検索可能なテキスト [9] のほか, 形態素解析器 *Chachalaca* [10] が開発されている. このほか, ゲレロ州オアパン (Oapan) 方言で, *Xerox* 有限状態変換器 (*XFST*) [11] を用いた電子化辞書・解析器の開発プロジェクトがある [6, 12].

一方, 本稿で扱うプエブラ州イシュキワカン方言は, 上記の方言とは語形や文法が多少異なり, 上記のリソースは流用できない. 電子処理に使える辞書やデータベースは当然未整備で, 書き起こしたテキストと収集した語彙集があるのみである.

3.2. 要件

本稿では, 下記の例 (3) のようなナワトル語イシュキワカン方言の書き起こしテキストを入力とし, (4) のように言語学の慣習に従ったグロスつきテキストを出力する場合について論じる.[3]

(3) `wan PARA aamo moneextiis n xiikol ookitioochiiwkeh n tlamatkeh.`

(4) *wan PARA āmo monēxtīs n xīkol*
wan PARA āmo Ø-mo-nēxtī-s n xīkol
and for NEG 3S-REFL-SHOW-FUT ART devil
ōkitiōchīwkeh n tlamatkeh.
ō-Ø-k(i)-tiō-chīw-(k)eh n tlamatkeh
PST-3S-3SGO-god-make-PRET.PLS ART sorcerer.PL

[1] スクリプトの試験には Python 2.7 を用いた.

[2] 世界の言語のデータベースを提供している *Ethnologue* [7] では, 29 のナワトル語方言を異なる言語として登録している.

[3] 今回, 実際には, グロス整形用の gb4e.sty マクロに対応した LaTeX ソースを出力としている.

(3)–(4) の例が示すように, また他方言について再三指摘されているように [6, 8, 9], この言語では複数の接辞が付加された複雑語が多く, 語形が千変万化するので, 語を逐次辞書ファイルに手動で登録することは現実的ではない.

たとえば, この言語では, 他動詞は語根単独で現れることはなく, *palēwia*「助ける」という動詞であれば, *ni-mits-palēwia*「私はおまえを助ける」, *non-ki-palēwia-h*「おまえたちは彼を助ける」のように, 必ず人称接辞を伴う. このように, 1 つの動詞に対して多くの語形が登場するうえ, 人称変化した形からその構造と動詞語幹を分析できなければならない. 人称変化だけでなく, *tē-palēwia*「だれかを助ける」, *mo-palēwia*「自分を助ける」のように自動詞化接辞がつくなどして動詞が拡張されることもある. さらに敬語活用も存在するため, *non-ki-palēwia-h*「おまえたちは彼を助ける」に対して *non-k-on-palēwia-h*「あなたがたは彼を助ける（敬語）」といった語形もカバーできなければならず, 可能な語形の数は膨大である.

さらに, 単純な分析が困難な不透明な語形変化もある. たとえば, 三人称目的語接辞 *k-/ki-* は, 音韻的条件で音形を変える. また, この言語では, 過去形や進行形など, 時制・アスペクトや法性にしたがって語幹交替が起こり, *palēwia* という動詞（第 3 活用動詞, *ia* タイプ）であれば, 完了語幹 *palēwih*, 未来・希求法語幹 *palēwī*, 未完了語幹 *palēwā* といったように, 語幹内部の音形も変わる. このほか, 完了語幹 *palēwih* や未来・希求法語幹 *palēwī* は語末では音韻規則によりそれぞれ *palēweh*, *palēwi* に変わる. テキストにはこれらの交替が起こった形で登場するので, プログラムはこうした変化をすべて逆算できなければならない.

また, 少数ながら不規則動詞も存在する. 英語のように活用形の数が比較的少ない言語であれば, 不規則形は個別に辞書に登録しておけば十分と考えられるが, ナワトル語のような言語では, 不規則動詞についても人称変化があり, 自動詞化形や敬語形も存在するため, 不規則形だからといって全ての語形をあらかじめ手動で網羅するのは効率的でない.

加えて, ナワトル語では名詞にも人称変化があり, *itskwintli*「犬」に対して *notskwī*「私の犬」のような所有人称標示が起こるほか, *siwātl* 「女」→ *nimosiwah*/*nimosiwāw*「私はおまえの妻だ」のように, 名詞が自動詞と同様に人称変化して単独で名詞文を作る場合もある.

3.3. 活用形総生成型辞書

こうした形態論的分析を少ない労力で自動化するために, 本稿では活用形総生成型辞書を利用する. 活用形総生成型とは, 日本語における *ChaSen* のように, 限られた語彙情報からテキストに現れうる語形をあらかじめ生成してリスト化しておく方式を指す.

Chachalaca [10] やオアパン方言のプロジェクト [12] で行っているように, 活用形を解析して語構成を特定することも可能であり, 理論的には望ましくもあるが, 考慮すべき可能性が膨大になり, 再帰的な処理も多く必要になるため, 長いテキストの分析には向かない. *Chachalaca* では, 1 語を分析するのに, 2015 年現在の通常のコンピューターで十数秒を要する. したがって, テキストの注解を目的とするなら, あらかじめ全活用形を生成する方式が現実的である.

この方式では, 登録された辞書（手動登録用辞書）の項目に, スクリプトが文法規則や形態音韻規則を適用することで, 可能な活用形をあらかじめ生成し, 参照用辞書に格納する. テキストに現れる語形はこの参照用辞書と照合される.

例として, *palēwia*「助ける」という動詞の活用形を生成する場合を考える. 動詞はさまざまに語形変化するが, 辞書には *palēwia*[4] という抽象形と, グロス作成に用いる ‘help’ という逐語訳をセットにして手動で登録しておき,[5] メタデータとして, 価数（自動詞か他動詞か）や活用クラスなど, 抽象形から規則活用形を生成するのに必要な語彙情報も合わせて登録しておく（図 1）.

POS	RAW	ANALIZED	TRANSLATION	VALENCY	CLASS	PLURAL
m	tonyoh	ti-on-yoh	2¥sgs-¥textsc{hon}-go			
m	xonyoh	xi-on-yoh	2¥textsc{opt}-¥textsc{hon}			
m	chuweh	chuweh	go.1¥pls{}			
v	<<0>>ta	(i)ta	see	t	1	
v	chiya	chiya	wait.for	t	2	
v	tlachiya	tlachiya	see	i	2	
v	kuui	kuui	buy-¥textsc{appl}	t	3	
v	iixtlamachili	iix-tla-machi	face-¥textsc{uno}-know-¥t	i	3	
v	maka	maka	give	d	1	
v	<<0>>hto	(i)hto	say	t	3	
v	<<0>>lpi	(i)lpi	tie	t	3	
v	chihchiiwa	chih~chiiwa	¥textsc{dist}~make	t	2	
v	paleewi	paleewi	help	t	3	
vs	its	its	see	t	pg	0
vs	ii	ii	<<cop>>	i	ft	0
vs	iito	ii-to	<<cop>>-¥textsc{prog}	i	op	0

図 1: 辞書の登録項目の例

[4]実際には *palēwi-* という部分形で登録するほうが効率的である.

[5]実際には, 後述のとおり, 形態素境界を加えた分析表記も合わせて登録している.

人称接辞や時制接辞といった要素も同様に, どういう要素に付加されるか, 活用形生成のいつの段階で付加されるかといった情報とともに, それぞれ辞書に登録しておく. この抽象形情報が, 図 2 のような活用形生成スクリプトに渡され, 個々の活用形が生成される.

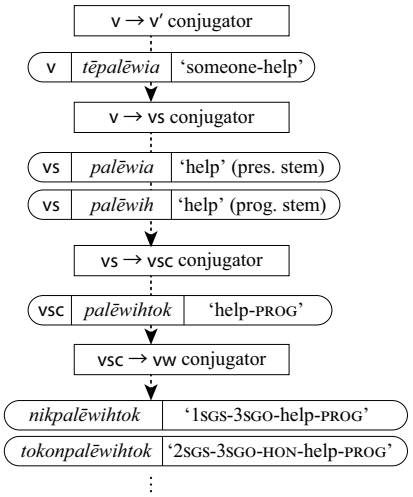

図 2: 動詞抽象形 *palēwia* からの活用形生成

以下, 図 2 のスクリプトが行う処理を概観する. まず, 3.2 節で触れたように, *palēwia*「助ける」のような他動詞には, *tē-palēwia*「だれかを助ける」, *mo-palēwia*「自分を助ける」のように自動詞化接辞がついた形が存在する. こうした形を再現するため, 辞書に登録された抽象形情報 (*V*) から, 抽象形拡張器 (*V* → *V′*) が *mopalēwia*「自分を助ける」, *tēpalēwia*「誰かを助ける」といった新たな抽象形 (*V*) を自動生成し, 辞書に戻す. この段階で, 辞書には, 手動登録した *palēwia*「助ける」という他動詞抽象形と, *mopalēwia*「誰かを助ける」などの自動生成された自動詞抽象形が両方存在することになる.

それぞれの動詞抽象形は, 音形以外にも, 上記の活用情報, 形態素分析を加えた分析表記, 逐語訳などの情報がセットになったリスト型データの形をしている. *mopalēwia* という抽象形であれば, *mopalēwia* という文字列そのもの（音形）の他に, 形態素境界を加えた *mo-palēwia* という分析表記, 'REFL-help' というグロス文字列, 各種の活用情報が, 1 つのリスト型データにセットとして格納されている.

つづいて, これらの抽象形 (*V*) から, 語幹交替器 (*V* → *VS*) が各語幹（*VS*; 現在語幹 *palēwia*, *mopalēwia*, 現在進行形語幹 *palēwih*, *mopalēwih* など）を生成する. それぞれの語幹は, どの時制に用いられるかというメタデータとともに, 次の段階である時制活用器 (*VS* → *VSC*) に渡され, 時制活用器はそのデータと接辞辞書ファイルをもとに各時制形 (*VSC*) を生成する.

たとえば, 現在進行形の活用形は現在進行形語幹に現在進行形接辞を付加することで作られるので, 現在進行形語幹 *palēwih* と進行形接辞 *-tok*（単数形）, *-tok-eh*（複数形） からそれぞれ *palēwihtok*, *palēwihtokeh* という時制活用形 (*VSC*) が作られる. それぞれの接辞も動詞抽象形と同様にリスト型データなので, 生成された時制活用形も, *palēwihtokeh* のような音形以外に, 形態素境界を含む *palēwih-tok-eh* という分析形や, 'help-PROG-PLS' というグロス文字列をもっている.

これらの情報はさらに人称活用器 (*VSC* → *VW*) に送られ, 人称・敬語接辞を付加され, 各種音韻規則の適用を受けて, 実際にテキストに現れるのと同一の語形 (*VW*) となり, 参照用の語形辞書に登録される.

こうして, 参照用辞書は, *nimitspalēwia*「私はおまえを助ける」, *nonkonpalēwiah*「あなたがたは彼を助ける（敬語）」といった人称変化・敬語活用や, *nimitspalēwihtok*「私はおまえを助けている」, *ōnimitspalēweh*「私はおまえを助けた」, *nimitspalēwītih*「私はおまえを助けにいく」のような時制変化・助動詞的活用をすべて網羅したリストとなる. グロスを生成する際には, テキストに含まれる形と参照用辞書を照合し, テキスト中の語形に対して分析形やグロス文字列を返す. プログラムがそれを組み合わせ, 整形されたグロスとして出力する.

当然, 参照用辞書に登録される語形の数は膨大になるが（単純他動詞 1 抽象形につき活用形約 2000 語）, 各語に対して形態素解析を行うよりも参照は高速になり, 長いテキストの注解も可能である.

この方式の問題点として, 単一の音形が複数の解釈を持ちうる例が考えられる. たとえば, この方言の場合, *xikalaki* という語形には「おまえが入れ」（*xi-kalaki* < *kalaki*「入る」）・「おまえがそれを入れろ」*xi-k-kalaki*（*xi-k-kalakī* < *kalakia*「入れる」）という 2 通りの解釈が可能であり, どちらを用いるべきかスクリプトが判定することはできない. 両方の動詞抽象形が辞書に登録されていれば, 活用器がすでに存在する語形と同音の形を生成した時になんらかの例外処理を行うこともできるので, 対策として, 3.5 節で行っているように, ユーザーに確認を求めることはできる.

また, 上記のスクリプトには, 不規則動詞に対して存在しない規則形を生成してしまうという問題もあ

るが, 本スクリプトの目的がテキストに出てくる語形を解釈にある以上, テキストに現れない形を過剰生成しても, 精度上の問題は生じないと考えられる.

3.4. 多層式辞書

3.3 節で見た活用形生成プログラムは, 多層式の辞書を採用している. ここでいう多層式とは, 抽象形 (*V*), 語幹 (*VS*), 時制活用した語幹 (*VSC*), 語形 (*VW*) といった複数のレベルを設定し, その全てに同様のモデルを適用することで, あらゆるレベルの不規則性を必要最小限の指定で再現する仕組みをいう.

すでに 3.3 節で見たように, 語形変化には, 自動詞化接辞の付加, 語幹交替, 時制活用, 人称変化など, いくつもの段階が存在する. この煩雑な多層性を逆に利用することで, 不規則形をその不規則さの度合いに応じて規則動詞と同様に登録し, 1 つの辞書ファイルに保存しておくことができる.

たとえば, 動詞 *chiya* 「待つ」の現在進行形は, イシュキワカン方言では, 規則形 **chixtok* ではなく, 不規則形 *chīxtok* になるが, これは語幹交替レベルでの不規則形（抽象形 *chiya* → 現在進行形語幹 *chīx*）に由来するものであり, 人称活用や複数活用の語形変化は規則動詞と変わらない. したがって, 不規則形だからといって, *chīx* という語幹をもつ活用形全てを列挙し登録するより, 語幹レベルの不規則性を辞書に登録しておくほうが効率的である.

そこで, *chiya* のような動詞抽象形に「*V* レベル」, *chīx* のような語幹交替の起こった語幹に「*VS* レベル」とラベルを付与し, 辞書ファイルには両方を一緒に登録しておく. 語幹交替器は, *V* レベルの要素から *VS* レベルの要素を一定の規則にしたがって生成するモジュールとする.

3.3 節および図 2 で見たように, 全ての活用形が規則的な *palēwia*「助ける」(3.3 節参照）のような動詞であれば, 辞書に *palēwia* という抽象形[6] だけを *V* レベルとして登録しておけば, そこから語幹交替器が規則的な交替語幹を生成し, 生成された全ての語幹を *VS* レベルとして辞書に出力する. 一方, 抽象形 *chiya*「待つ」に対する *chīx*（現在進行形語幹）のような不規則な語幹交替は, あらかじめ *VS* レベルとして辞書に登録しておけば, 規則語幹と不規則語幹がひとしなみに *VS* レベルの要素として辞書に登録されている状態となる. これをそのまま次のレベル（語幹 *VS* から時制形 *VSC* を生成する時制活用器）の入力とすれば, それ以降の活用（時制接辞の付加, 敬語活用, 人称標示など）は規則動詞も不規則動詞も同様に行われる.

したがって, 不規則な語幹交替のある動詞でも, その不規則な部分だけを適切なレベルで辞書に登録しておけば, 規則動詞と同じように活用形を生成することができる. すなわち, 動詞抽象形と不規則語幹の差は, 辞書の読み込みのタイミングを変えるだけで再現することができる（図 3）.

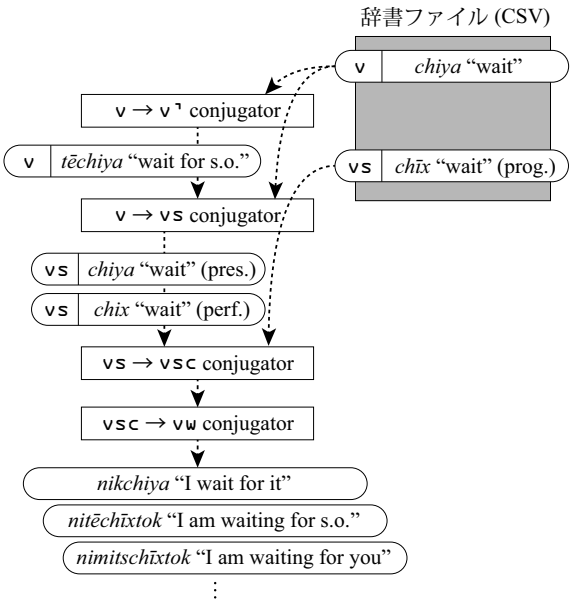

図 3: 多層式の辞書・活用器と不規則形の処理

人称に応じて現れる完全な不規則形や縮約形であれば, 同様により後の段階に割り込ませればよく, あらゆる不規則性を最小の指定で再現することができる.

3.5. 対話型辞書スクリプト

次に, 辞書整備の手間を最小にするため, あらかじめ辞書を用意するのではなく, 新しい語が現れた際に手動で辞書に新項目を追加する方式を提案する. 本スクリプトは, 前ページ (3) のような表層語形のテキストを入力とし, テキスト内の語を逐次読み込んで 3.3 節で生成した全活用形の辞書と照合していくが, 辞書にない語形が現れた場合, 処理を中断し, 簡易 UI で使用者に新語の登録を促す. この際, 新語の登録は抽象形単位で行い, 入力が行われた時点で, スクリプトはその抽象形から生成可能な全活用形を作成し, 再照合する. この際に登録された語は抽象形辞書ファイル

[6]注 4 参照.

に追加され, 次に同じ抽象形から作られた語形が現れた場合, 登録済みの語として分析することができる.

自然言語のトークンの頻度分布は Zipf 分布に似ることが知られているので, 現れた語をその都度辞書に登録していくことで, 少ない語彙項目で高いカバレッジを得ることができる.

4. 応用と今後の展望

以上で論じた方式は, グロス作成のみならず, コーパス用の品詞情報登録など, 様々な形のタグづけに利用できる. 形式を統一し, 本稿の例でスクリプト本体に組み込んでいた形態音韻規則を外部スクリプトに分離すれば, 単一のプログラムで多くの言語に対応できる汎用のタグづけソフトフェアを作ることもできる.

また, 実際の言語の活用規則にのっとって語形をシミュレートする方式なので, アサバスカ諸語のように音形から基底形が分析しにくい言語や, オトマンゲ諸語のように声調が複雑に相互作用する言語でも分析することができる.

一方, 本発表で報告した方式は, 日本語の受動接辞や使役接辞のように, 再帰的な性質をもつ形態法の分析には, 生成する語数が指数関数的に増えるため不適である. ナワトル語の分析に際しても, 使役形・充当形・名詞抱合など, これに近い性質をもつ活用は, 個別に登録する必要がある. 本発表の方式をチュルク諸語やエスキモー諸語のような入れ子式 (scope-ordered) の語構成を主とする言語の処理に応用するためには, これらの問題を解決する必要がある.

引用文献

1) Austin, P. K.: Introduction, Austin, P. K. and McGill, S. (eds.): *Language Documentation and Description*, vol. 1, pp. 6–14, SOAS (2003).

2) Himmelmann, N. P.: Language documentation: What is it and what is it good for?, Bisang, W., Hock, H. H. and Winter, W. (eds.): *Essentials of Language Documentation*, pp. 1–30, Mouton de Gruyter (2006).

3) SIL International, available from <http://www.sil.org/>.

4) SIL International: Language Explorer (FLEx), available from <http://fieldworks.sil.org/flex/>.

5) Munro, P.: Entries for verbs in American Indian language dictionaries, Frawley et al. (eds.) [13], pp. 86–107 (2002).

6) Amith, J. D.: What's in a word?: The *whys* and *what fors* of a Nahuatl dictionary, Frawley et al. (eds.) [13], pp. 219–258 (2002).

7) SIL International: Ethnologue, available from <https://www.ethnologue.com/>.

8) Karttunen, F. and Amsler, R. A.: Computer-assisted compilation of a Nahuatl dictionary, *Computers and the Humanities*, vol. 17, pp. 175–184 (1983).

9) Canger, U.: An interactive dictionary and text corpus for sixteenth- and seventeenth-century Nahuatl, Frawley et al. (eds.) [13], pp. 195–218 (2002).

10) SUP-INFOR: Programmes > Chachalaca, analyseur morphologique du nahuatl, available from <http://www.sup-infor.com/program/program.htm#CHACHALACA>.

11) Karttunen, L., Gáal, T. and Kempe, A.: Xerox finite-state tool, available from <http://www.cis.upenn.edu/~cis639/docs/xfst.html>.

12) Maxwell, M. and Amith, J. D.: Language documentation: The Nahuatl grammar, Proc. *Computational Linguistics and Intelligent Text Processing: 4th International Conference, CICLing, 2005, Mexico City, Mexico, February 13–19, 2005*, pp. 474–485 (2005).

13) Frawley, W., Hill, K. C. and Munro, P. (eds.): *Making Dictionaries: Preserving Indigenous Languages of the Americas*, University of California Press (2002).