方言コーパスの作成とその意義

沢　木　幹　栄

キーワード　琉球方言　徳之島　コンピューター　動詞活用　XML

0．はじめに

日本語の現代書き言葉でも話しことばでもあるいは古文でもコーパス作りが試みられ，実際に多くのコーパスが存在しているが，方言の本格的なコーパスはまだのようである。（注１）私は琉球方言に属する徳之島浅間方言のコーパスを作成することに成功した。本稿では，方言のコーパスのあるべき姿とその意義，作成の具体的方法について述べたい。

1．コーパス作成の効用

徳之島二千文コーパスは以下のような形をしている。

1 <Tok:m ps=noun id=874 >kara:zI</Tok>
<Tok:m ps=post_p id=33 type=1>kacI</Tok>
<Tok:m ps=noun id=1339 >tI:da</Tok>
<Tok:m ps=post_p id=42 type=1>nu</Tok>
<Tok:m ps=verb id=288 conjug=2 aux=00100000>tI:tuN</Tok>
<Tok:m ps=post_p id=5 type=1>da:</Tok>

表示の都合上改行を形態素ごとに入れた。実際は途中の改行はない。これのもとになったのは「徳之島二千文」というテキスト（注２）の１番目の文で

1　kara:zIkacItI:danutI:tuNda:

となっている。比較すれば分かるように，文を形態素に分解し，<Tok:m で始まる <> のなかでその形態素の情報付けを行っている。このようなデータ形式は XML と呼ばれ，コンピューターで利用することを前提としている。

ps は品詞，id は品詞別の id 番号である。それ以外の標識については以下で説明する。

このようなコーパスからは一般のコーパスと同じようにいろいろな情報を得ることができる。私が真っ先に行ったのは，助詞の連接の調査と，格助詞 ga と nu が接続する名詞にどんなものがあるかだった。

もちろん，これ以外にも有用な情報がこのコーパスから得られるが，コーパスを作ることそのものに効用があることがわかった。それは「数え尽くすこと」である。

方言の研究で一般的なことだと思われるが，コーパス作成にとりかかった時点で徳之島方言の文法や語彙について対象となるテキストをすべてカバーできるような知識を私は持っていなかった。むしろ，コーパスを作る過程で少しずつ知識を完全なものに近づけていったのである。そのことを念頭に置いて以下を読んでいただきたい。

さきほど id という標識は品詞別の id 番号を表すと述べた。このコーパスは品詞別の形態素リストを持っていて，コーパス内の形態素はすべて形態素リストに登録されている。これはテキスト処理でかならず生じる同語異語判別の問題を回避するためであった。たとえば，助詞で格助詞の ga と疑問詞疑問の終助詞 ga は全く同じ形だが，id 番号をつけることによって区別することができる。

コーパスを作るときにはもとのテキストを形態素に分解し，すべての形態素を形態素リストに帰属させることになる。この過程で「数え尽くし」が行われる。

たとえば，na:tI（だから）という助詞があるが，これは najui（成る）という動詞の活用形の一つと同じ形である。しかし，コーパスを作る場合になれば，そのつど動詞なのか助詞なのかを判断しなければならない。na:tI は動詞の活用形が文法化して助詞として使われると考えられるのだが，コーパスを作る過程で助詞として認定した。

このコーパスは作成にコンピューターを使うということが前提である。作業の段階で，助詞として分類された形態素を文脈と一緒にピックアップして表示させるプログラムを作っておいた。用法上全く異なっていて本来別語形とすべき助詞を区別せずに同じものとみなしていたことがこのプログラムで判明したこともある。

また，zIN（ばかり）という助詞はそれまでの調査では全く見過ごされていた。cIkkizIN（突きばかり）という形で出てきたのだが，「突く」を意味する動詞の一活用形ぐらいにしか考えていなかった。インフォーマントの内省にたよる対面調査では「だけ」「ばかり」を意味する助詞として別語形が回答されていたのである。

ところが，cIkkizIN を cIkjui（突く）の活用形と見なすことができないことがわかり，cIkki が名詞であることが明らかになって zIN を助詞のリストに加えた。テキストをすべて形態素に分解し，形態素をリストに帰属させるということがなければ助詞 zIN の発見はなかったと思われる。

上記のプログラムで品詞ごとの異なり形態素のリストを作ることもできる。コーパス作成の作業の要所要所でこれを何度も行った。情報の付け忘れも付け間違いもこれでチェックできる。コンピューターなくしては考えられないことである。

数え尽くす例をもう一つ挙げると，徳之島方言の共同研究者の福嶋が「徳之島二千文」を使ってテキストに現れるすべての動詞とその動詞がどのような活用を行うかのリストを作った。福嶋の動詞の記述は非常に綿密かつ詳細で正確なのだが，福嶋の動詞のリストから漏れている動詞がいくつかあることが判明した。これは人間の注意力には限界があることを示している。コンピューターを使えば情報付けがされていない動詞は簡単に拾い出すことができる。ほかの品詞と同じく「数え尽くす」ことができるのだ。

以上が「数え尽くす」例である。

2．動詞の活用をするプログラム

活用する形態素がどの活用形をとっているかの情報はコーパスに加えるべきであると考えた。活用情報があれば，コーパスから得られるものは大きい。そのために必須である自動的に動詞の活用を行うコンピューターのプログラム作りに取り組んだ。

私が考えるコーパスは，品詞別の形態素リストがあって，文法の機能を果たすコンピューターのプログラムがあり，そこに最初に述べたテキストの形態素に情報付けをしたコーパスがあるというものだ。

動詞の活用をプログラムで実行できれば，形態素に付加された情報から活用形を生成することができる。この活用形を情報が付加されている形態素と照合することで情報が正しいかどうか検証できるはずである。

わかりやすいように現代日本語を例にとると「本を読め」という文があったとき，これを

ホン
オ
ヨメ

と分解し，さらに「ヨメは動詞ヨムの命令形」とヨメに対して情報づけを行う。ヨムの命令形が本当にヨメになるかどうかプログラムでチェックするのである。

図示すると

ヨム　文法（活用・助動詞情報）→　ヨメ

となる。ヨムは動詞のリストのなかにあって（動詞リストの id 番号），活用・助動詞情報から実現形のヨメを作り出すのは文法ということになる。

今までどの方言でも語彙や文法が調べられているが，では実際に文法を適用して方言で使われている文節が作れるかということはやって見せた人がいない。これをコンピュータープログラムで実現しようということになる。今までは語彙，文法，方言のテキストがばらばらにあっただけだが，活用を実行するプログラムが三者を結びつけることになる。

動詞の活用のプログラムを作成する際に依拠したのは福嶋秩子による活用の記述（岡村ほか2009に所収）である。これは前述したように非常に綿密で詳細かつ正確なのだが，活用をプログラムとして表現しようとすると不十分な点があった。動詞は単独でテキストに現れることは少ない。多くは助動詞がついた形で出現する。しかもその助動詞が組み合わせで使われる。動詞が単独で活用するのではなく，動詞と助動詞が組み合わさったものが活用するのである。

福嶋の記述では珍しいことに複数の助動詞が組み合わさったときの形まで示されていた。

これは日本の方言研究では珍しいことで大きな助けになったが，これでも不足で，動詞と助動詞の組み合わせという見方をさらに徹底しないとプログラム化はできなかった。

たとえば，ヨムに対してヨマナイ，ヨンダという活用形を提示するのは方言の記述でごく普通のことだが，実際はナイ，ダは助動詞なのでこれが活用することも表現できなければならない。

また，助動詞は使役，受け身・可能，テイル，テオク，丁寧，否定，過去，疑問がこの順番に接続する。もちろん，これらの助動詞は使われることもあるし，使われないこともある。東京方言と違って，活用する付属形式はここに挙げたものだけである。東京方言には推量・様態などの助動詞があるが，徳之島方言では不変化詞がその機能を担っている。

ここで，助動詞と考えにくいものが挙げられているのに気づいた人もいるかもしれない。テイル，テオク，疑問は東京方言では助動詞ではない。しかし，テイル，テオクに対応する徳之島方言形は従属形式であり，活用を行うので助動詞と見なしうるし，単純疑問を表す助詞は動詞に接続すると全体の形を変えて融合する。前接する形態素との切れ目もわからなくなる。

疑問（単純疑問）の助詞は jI であると考えられる。たとえば「好きか」は sIki:jI となる。ところが，「した」sjI: に対して「したか？」は sjE: となる。また，「あります」are:juN に対して「ありますか？」は are:jumI である。

このように融合して形を変えるものは助動詞と同じ扱いにすると処理が簡単になるので助動詞と同列に扱った。今までの記述では，このような複雑な現れ方について考慮されていなかった。これは記述の穴と言うべきである。

このような動詞と助動詞をコーパスで情報付けをする際に 2 通りの方法が考えられる。

一つは動詞と助動詞に切り分けてそれぞれに対して情報付けをするというやり方，もう一つは動詞と助動詞が一つになったものを単語と考えてそれに対して情報付けをし，無理に切り分けたりしないというやり方である。

最初の動詞と助動詞を分けるやり方では分けようにも切れ目がわからないということが起きる。動詞と助動詞，あるいは助動詞同士で融合が起きて形態素の境目が分からなくなるからである。

それでもどうしても分けたい場合は，ヨマレル　を　ヨム　と　ラレルに分解するということになる。ヨマレルが実現形，ヨム　と　ラレル　は基底形となる。しかし，このコーパスではテキストの中で存在している形を形態素に分けて情報付けをしている。そこに基底形を持ってくるのはコーパスのなかに異質なものを持ち込むことになる。また，活用プログラムなしでコーパスからテキストを復元することができない。

第二の方法では動詞と助動詞が一体になったものはそれ以上切り分けることはない。したがって，テキストに使われた形がコーパスでそのまま使われる。助動詞がどう接続しているかは独立した情報となる。したがって情報として与えるのは動詞の識別番号（id），助動詞情報，活用情報ということになる。最初に示した XML の書式では助動詞情報は aux，活用情報は conjug である。

助動詞情報はさきほどの連接の順番に対応する 8 桁の数字で与える。一番左の桁は使役の助動詞の有無で 0 か 1 になる。次の桁は受け身・可能の有無でやはり 0 か 1 というようにす

ると 8 桁の数字でどの助動詞があるかが表せる。

このようにすれば，無理に切り分けなくても必要な情報をコーパスに与えることができる。したがって，私は第二の方法を選択した。

次に問題なのはヨムに可能の助動詞をつけてさらに否定の助動詞を接続したものという情報からヨマレナイを導き出すようなプログラムが実際に作れるかということである。作ってみるとかなり大規模なプログラムになった。使用言語は VBA である。これは，動詞の活用情報をエクセルで持っているためで，エクセルのデータを直接操作できるプログラミング言語である VBA が最良の選択だと考えた。

このプログラムは形態音韻論のルールをシミュレートするものではない。そのほうがプログラミングは容易だったと考えられるが，福嶋の記述が活用形のテーブル方式だったので，それを生かす方向で考えた。プログラムはインプットである語彙情報と活用に関する情報から正しいアウトプット（活用形）を生み出す装置である。装置の中身は問題ではないので形態音韻論のルールにはこだわらない。

このプログラムが正しければ，助動詞情報，活用情報から実際の語形が導き出せるはずである。逆に言うと助動詞情報，活用情報が間違っていれば実際の語形と違うものが出てくるので，こうした付加情報が正しいかどうかの検証ができる。今までであれば，人間が付加した情報を別の人間がチェックすることになるが，そうではなくコンピューターを使ったチェックが可能である。あくまでプログラムが文法の記述どおりの結果を出すという前提が成り立っている限りにおいてのことであるが。

このプログラムを動かしてテキストに現れた形（実現形）とプログラムが生成した形が違うものを発見した。「打たれる」'utattI である。食い違いを発見したあとで福嶋の記述の通りに自分の頭で考えたものはプログラムが生成した形（'uta:tI）と同じだった。ということは，プログラムは記述を忠実に再現しており，記述が修正を要するということになる。

コンピューターのプログラムによって記述の穴を見つけ，記述をより精密なものにすることができた。もちろん，記述を改良したあとはそれをプログラムに反映する。コーパス作りの意義に文法の記述を精密化することを加えたい。

今まで語彙と文法は別々に研究されてきた。ある動詞に文法を適用した結果どうなるか，助動詞を含めたすべてのケースについて検討するということはどの方言でもされていない。今回の徳之島方言では一つの動詞で助動詞の接続も考慮すれば活用形が400以上あるので，そんなことはできることではない。しかし，コーパスになっていれば，コーパス内の活用形はプログラムとの連携でチェックができる。

今，方言の動詞活用，あるいは形態論の研究発表を見ることはほとんどない。やるべきことはもうみんなされているという閉塞感のようなものが漂っている。しかし，動詞の活用のプログラムを作ってみて，まだ開拓されていない原野が広がっていると感じた。まだまだ分からないことがあり，それに今回はじめて思い至った。それは徳之島方言に限ったことではない。他の方言でもコーパスを作れば形態音韻論レベルで新しい発見があるだろう。

活用形生成プログラムに少しだけ手を加えると，考えられるすべての助動詞の組み合わせの活用形を生成することができる。全部で400以上ある活用形を，テキストに出てきた実際の形と順番に照合して一致したものがその形の助動詞情報，活用情報ということになる。

コーパスに情報付けは必須である。人間が人力のみで情報付けをした場合，方言のコーパスでは大変な作業量になる。今まで方言のコーパスというものを見たことがないのは，この作業量にみんな尻込みをしたからではないかと思われる。

しかし，上記のプログラムでコーパス作りの労力を最小にする見込みがある。助動詞情報，活用情報を入力するのは大変であるが，この作業を半自動化できる。実際は候補が複数ある可能性があるので，そのつど人間が判断しなければならないが，これだけで非常な省力化になる。

3．コーパス作成まで

ここでは，コーパス作成に至るまでの道のりを簡単に振り返りたい。

コーパスを作るまえに KWIC というものを作った（岡村ほか2009)。（注３）これは，方言文を文節で切り分け，それに対応するように標準語文も文節で切り分けて，文脈付きの文節リストを作るというものである。KWIC 自体は数多く作られていて何ら新しいことはないが，これの新味は方言と標準語訳を同時に提示することと，方言で KWIC を作ったということだった。

方言と標準語訳が同期するように文節切りしたデータをコンピューターで並べ替え（ソート）した。ソートは方言と標準語訳，順引きと逆引きで４通りのものを作った。KWIC はもとのテキストよりはるかに量が多くなるので，印刷データには収まらず，岡村ほか2009では付録の DVD に収録している。

方言の順引きでは一つの動詞の活用形が近いところに集まるので，動詞の活用を調べるのに便利である。方言の逆引きでは文節の最後に来る助詞が同様に集まるので助詞の調査ができる。このように KWIC によって徳之島方言の知識を深めたことと，文節切りをしたデータを持っていたことがコーパスを作るうえの下地となった。

KWIC は有用であったが，一方で限界を感じたのでコーパスを作ることを目指した。

まず，コーパスの原型を作成した。冒頭の文は

kara:zIkacI@tI:danu@tI:tuNda:

のように文節切りされていたが，そこから

kara:zI 名 @kacI 助 @tI:da 名 @nu 助 @tI:tuN 動 @da: 助

のように形態素で切り，品詞情報を加えた（名は名詞，動は動詞，助は助詞)。形態素で切り分けたのはその時点での私の徳之島方言の知識による。もちろん誤りが多かったが，大事なのは切り分けることであり，誤りはコーパス作成の過程で修正されていった。

形態素で切り分けたデータにプログラムをかけると

<Tok:m ps= noun id= >kara:zI</Tok>

<Tok:m ps=post_p id= type=>kacI</Tok>
<Tok:m ps=noun id= >tI:da</Tok>
<Tok:m ps=post_p id= type=>nu</Tok>
<Tok:m ps=verb id= conjug= aux=>tI:tuN</Tok>
<Tok:m ps=post_p id= type=>da:</Tok>

のようになる。コーパスのスケルトンだけで，品詞情報以外の情報がない。これを研究補助者（言語学者でもなく徳之島方言を知っているのでもない）に渡すと，その時点で分かっている情報を研究補助者が加える。ただし，最初のうちは不完全な助詞のリストと動詞のリストぐらいしかなかった。したがって付け加える情報は助詞と動詞の id 番号ぐらいであった。しかし，最後まで少しずつ進めるうちに助詞のリストは充実していった。

ここから先は私単独の仕事になる。品詞ごとの形態素リストを作成し，id を決定していく。これをコーパスに反映させる。情報を追加したり修正したりしながら形態素リストを作成する。この繰り返しを重ねるうちにコーパスが段々完成していく。全体の作業を通じて随所でコンピューターを使うことはいうまでもない。

ここまで手探りでコーパス作りを進めてきたが，決して能率よくできたわけではない。コーパスが完成に近づいた今になって最善と思われる方法が見えてきた。それは，形態素リストをうまく使った，コンピューターによる半自動化である。

4．他方言への応用

先に述べたように日本語で一番コーパス作りが遅れているのが方言研究の分野である。日本の方言を絶滅危機言語として保存する運動が行われている。何をもって保存と称するのか私にはよく分からないのだが，コーパス作りこそ最優先ですべきことだと考えている。

徳之島方言で見いだされたコーパス作成法は日本のほかの方言に応用できるだろうか。これについては私は楽観的に考えている。たとえば首里方言は『沖縄語辞典』にかなり詳しいレベルの高い記述がある。これによれば動詞の活用は徳之島のものにかなり似ていて，私のプログラムを根本的な部分はそのままに，残りを変えれば使えそうである。おそらく日本の方言のなかで動詞の活用の仕方が一番複雑なのは琉球方言なので，首里方言でも私の方法が使えるのであれば日本中の方言でコーパスを作ることができる。

沖縄の宮古方言では jumi（嫁）に対して jumja:（嫁は）のように名詞が一種の格変化を行う。似たような現象は鹿児島方言にもあり hai（針）に対して hari（針に）がある。これらは徳之島コーパスで動詞に助動詞が接続したものを動詞の活用形と見なしたのと同じように名詞から助詞を分離せずに名詞の格変化のようなものとして名詞の情報のなかに助詞を取り込めばよい。

全体の作業については，これはほかの方言でもそうだが，より強力に機械化を行えば能率よく進められるだろう。コンピューターによる自動化ができないのは文節切りぐらいで，ほかの作業は全自動化あるいは半自動化ができる。なによりも徳之島コーパスの経験から作業の最初から全体の仕事の内容が見えている。プログラムも全くゼロから作らなければならな

いものはほとんどないはずだ。

コーパス作りは莫大な人手をかければできないことはない。しかし，コンピューターでできることを大学院生の奴隷労働やアルバイターの人海作戦でしようとするのは私は美しくないと思う。コーパスはコンピューターで利用することを前提として作るものである。コーパスそのものにコンピューターを使う思想が入っている。それを人力だけで作るのは作成者がコンピューターを理解できないからという理由しか考えられない。

美的観点はともかくとして，第一に人間はミスをする。ミスを防ごうとすれば余計にお金がかかる。第二に人件費が莫大になる。人間はもっと知的な労働をしたほうがいい。第三にコーパスが完成するまでに時間がかかってしまう。

ここまで述べてきた方法でコーパス作りをする場合は，作成者が最低限のプログラム能力を持っている必要がある。私は自分が作った数十本のプログラムすべてを分かりやすく書き直し，ドキュメントも付けるつもりである。しかし，それらを全く修正なしで他の方言にも使えるとは思っていない。それぞれの方言に合わせた修正が必ず必要になる。特に動詞の活用のプログラムはほとんどの場合大きく書き換える必要がある。私の作ったプログラム群を参考にコーパス作りをする人はプログラムがどう働いているかを読解する程度の能力は持っていなければならない。

言語学者がプログラマーを兼ねるのは言語学者とプログラマーがチームを組むのとはまた違った良さがある。一人が両方を兼ねていれば，作業の途中で新しいプログラムが必要になったときその場で書くことができる。プログラマーが別人のときは言語学者からプログラマーにこんなことがしたいと話をしなければならない。どうしてもそこにタイムラグが生じる。また，言語学者がプログラミングを知っていれば，コンピューターを使ってできることの想像がつきやすい。そこから新たな発想が出てくることもある。

一つでも多くの方言でコーパスが作成されるように今後は助力を惜しまないつもりである。

この研究は平成18年度文部科学省科学研究費（基盤研究Ｂ）「徳之島方言辞典語彙編の作成のための研究」（課題番号18320066）ならびに平成23年度文部科学省科学研究費（基盤研究Ｂ）「奄美方言データベース作成のための研究」（課題番号23320095）の助成を受けた。

（注１） コーパスとはある程度以上の量のテキストに対し，形態素ごとに情報付けをしたものである。日本語の書き言葉においてはそれが作られるようになってから20年近くの歴史があるが，方言ではそのようなものは見たことがない。それは非常に奇妙なことだと思う。

国語研究所の木部暢子らが2017年東京で開催された国際学会の Methods in Dialectology 16 において方言コーパスについて発表を行った（Corpus based study of Japanese dialects: Regional differences in case marking system）が，これは方言テキストの標準語訳を材料にしたコーパスであって，方言テキストそのもののコーパスではない。

また近年になって方言文にグロスを付けることが行われるようになった。グロスは文法的意味の情報だが，これをテキスト全文に付加したものがあったとして（私は見たことがないが存在するらしい）それをコーパスと呼ぶのはちょっと抵抗がある。グロスは意味と形態の

うちどちらかと言えば意味に偏っているが，私のコーパスは意味と関係のない形式の記述だからである。おそらく，グロス付けされたテキストをプログラムなどでコーパスに変換することはできない。コーパスに必要な情報がグロス付けテキストでは不足している。逆もまた正しいであろう。また今までに書き言葉などで作られたコーパスが文法的意味に重きをおいたものでないことも言っておいたほうがいいだろう。

語彙リスト，文法として機能するプログラムそして情報付けがされたテキスト本文が有機的につながり，テキストのなかであればすべての形態素がなぜその形なのか説明できるというのが私のコーパスである。テキストという狭い世界のなかでは，すべてが数え尽くされている。説明されないで残っているものは一つもない。正確に言えば説明がつかなかったものはそのものとしてすべて記録に残されている。あくまで形式的な面だけのことだが。

正直な話，私はこの数年コーパス完成というゴールを目指して誰かと競争しているつもりでいた。いま私はまわりを見回して誰もいない状況に当惑している。一番乗りをしたつもりが実はそんな競争など最初からなかったらしい。誰かに先を越されるのもいやだが，ひとりぼっちもさびしいものだ。

（注２）「徳之島二千文」はアンリ・フレ1971をもとに徳之島方言の話者である岡村隆博が作成したテキストである。これはアンリ・フレの作った日本語の2000文をできるだけ忠実に徳之島浅間方言に翻訳したものである。これは岡村ほか2009の KWIC などのもとになった。ここで述べているコーパスもこの「徳之島二千文」をそのまま使っている。なお，「徳之島二千文」という名前の出版物は存在しない。

本論での徳之島方言の表記は岡村ほか2009でも用いられている中舌母音を大文字で表記する方式に従っている。

（注３）沢木2003で「コーパス」と言っているのは実は KWIC である。2009年には DVD で KWIC を公開したが，2001年には KWIC を作っていた。

【参考文献】

国立国語研究所（1963）『沖縄語辞典』 大蔵省印刷局
アンリ・フレ（1971）『日本語二千文』 早稲田大学語学研究所
沢木幹栄（2003）「方言コーパスによる徳之島方言の研究」 『人文科学論集第37号＜人間情報学科編＞（信州大学人文学部）』
岡村隆博，沢木幹栄，中島由美，福嶋秩子，菊池聡（2009） 『徳之島方言二千文辞典　改訂版』 徳之島方言の会（科学研究費報告書）

（2017年11月６日受理，11月21日掲載承認）