2A1-03

論理プログラミングによる連濁現象の解析

Analysis of Rendaku Voicing in Japanese using Logic Programming

小林 郁夫[*1]
Ikuo Kobayashi

諏訪 正樹[*2]
Masaki Suwa

[*1] 慶応義塾大学 SFC 研究所
SFC Research Institute, Keio University

[*2] 慶応義塾大学 環境情報学部
Faculty of Environmental Information, Keio Univ.

In this article, we try to analyze *rendaku*, phonological phenomena where two words are compounded. We construct a small database containing positive and negative examples from a dictionary. Positive examples correspond to compounded words whose head clear sounds of the latter words become voiced in compounding, and the head clear sounds of the latter words of negative examples do not become voiced. The dictionary scopes on accent information, so we could consider including prosodic influence. Also we add information of part-of-speech. We show that such information is useful to mine some local rules, rather than comprehensive general explaining theory.

1. はじめに

連濁現象は日本語で複合語がつくられるときに起こる音韻現象のひとつである。発生に法則や例外があまりみつかっておらず、規則性を発見することが難しいとされている。この現象が完全に非規則的なものなのか、あるいは部分的に規則性を持ちえるものなのかを追求することが、本稿の研究の動機である。ここでは私たちは、論理プログラミングによるアプローチを試みる。

2. 連濁現象

日本語では複合語を構成して用いることができ、これが思考や伝達を効率化することに役立っている。たとえば、「包丁研ぎ」という名詞を作り用いることは、「包丁を研ぐこと」とか「包丁を研ぐ人」という句とほぼ同等の内容を概念操作上も言語の運用上も効率化的に表現することと考えることができる。

慣習的に用いられている在来の複合語を用いることはもちろん、日常の生活の中で用いられる語彙どうしを結び付けることによって、かなり柔軟に複合語を作り出すことができるため、日本語表現の中で頻繁に用いられている。

2.1 発生する条件

複合語は 2 以上の語が連結されて作られる。「包丁研ぎ/ほうちょうとぎ/」のような複合語には前の要素と後ろの要素が存在して、「包丁/ほうちょう/」が前、「研ぐ/とぐ/(研ぎ/とぎ/)」が後ろである。後ろの要素の語頭に、対応する濁音(半濁音)を持つ清音があるもののみが連濁するか否かの考慮の対象になりえる。「研ぐ」の語頭は清音「ト」であり、対応する濁音「ド」があるので「ト」から「ド」に変化をするかどうかを考慮することが可能である。後部要素御の語頭がもともと濁音であるか、清濁の対立のない音である場合は、この問題の分析の対象にならない。「空元気/からげんき/」の後部要素は「元気/げんき/」、「同居人/どうきょにん/」の後部要素は「人/にん/」で、今回の問題の対象から外れる例である。

3. データベースの構築

第 2 章で述べた問題は通常音韻論上の問題としてとらえられるため、音韻資料を用いたデータベースを構築することを試みた。[金田一 83] はアクセントを記録・一覧する観点から戦後の東京の発音を取材し、編纂された辞書である。この辞書を用いて、以下の要領でデータを構築した。

- 見出し語から 2.1 節の条件に適合する複合語を選ぶ。
- 選んだ複合語について、前部要素、後部要素を決定する。
- 複合語、前部要素、後部要素について、アクセント位置を辞書に基づいて決定する。
- 複合語、前部要素、後部要素について、品詞ないしそれに準ずるカテゴリーを決定する。

上記の情報を各複合語について図 1 のような形式で表現する。各述語の項にはそれぞれ型がある。rendaku/1 の項、cacc/2, cpos/2, facc/2, fpos/2, lacc/2, lpos/2 の各第 1 項、comp/3 のすべての項は「語・辞」の型、cacc/2, facc/2, lacc/2 の各第 2 項は「アクセント」の型で整数の定数、cpos/2, fpos/2, lpos/2 の第 2 項は「品詞および辞カテゴリー」を表わす文字列定数である。「品詞‥」を表わす文字列定数のリストを図 2 に示す。

```
rendaku(aikotoba).  %事例=複合語

cacc(aikotoba, 3).     %複合語のアクセント位置
cpos(aikotoba, n).     %複合語の品詞
comp(aikotoba, au, kotoba).
                    %複合語を構成する要素語・辞
facc(au, 1).           %前部要素のアクセント位置
fpos(au, v).           %前部要素の品詞
lacc(kotoba, 3).       %後部要素のアクセント位置
lpos(kotoba, n).       %後部要素の品詞
```

図 1. データベース中のレコードの例

n	名詞	adj	形容詞
nk	漢語名詞	adv	副詞
v	動詞	pf	接頭語
vn	体言化した動詞	sf	接尾語

図 2.「品詞および辞カテゴリー」のリスト

連絡先:小林郁夫, 慶応義塾大学 SFC 研究所 上席所員(訪問), ikuokoba@sfc.keio.ac.jp

4. 解析の方法

4.1 帰納論理プログラミング

帰納論理プログラミングでは、ある事態が成り立つ場合と成り立たない場合のそれぞれについて集められた事例を対象とし、成り立つ事例を説明でき、成り立たない事例を説明しないような説明を、より情報の利得が高い形で示そうとする。形式的には、正事例の集合 $E+$, 負事例の集合 $E-$,背景知識 B が与えられたとき、式の集合 P ={e <- top| $e \in E^+$}, N ={⊥ <- e | $e \in E^-$} について、式

$$B \cup H \models P$$

$$B \cup H \cup N \not\models \bot$$

を成り立たせる仮説 H を計算することが目標となる[Muggleton 95, Ray 09]。E+, E- は確定節として表わされた各事例 e の集合、B は節論理形式で表現されている必要がある。3 章で示したデータベースの表現は、論理プログラムにおける個々の確定節からなる論理積となっている。この表現は、論理プログラミングでの処理に適当な形式となっている。

4.2 帰納論理プログラミングシステム Aleph の利用

Aleph は、前節で述べた帰納仮説を prolog 上で計算し求めるシステムである。核となる逆汎囲法のアイディアは[Muggleton 95]に示されている。

図 1 で提示したレコードの表現のうち、述語 rendaku/1 を用いた確定節を正事例・負事例とし、他の述語を用いた節の集合全体が背景知識となる。図 1 で示したものが 1 件の負事例と背景知識のうちそれに関わる部分となる。正事例に関しても同じく rendaku/1 を用い、同様の述語を用いて背景知識を表現する。

5. 実験

3 章で述べたデータベースから 143 のレコードを使用し、4.2 節で述べた方法で解析を行った。143 事例のうち正事例が 77, 負事例が 66 である。

表 1 に帰納による説明の成績を示す。最上段の「許容ノイズ」は帰納的に導出される各ルールが誤って説明しても容認する負事例の個数を示している。4.1 節で示した標準的な目標においては、完全な説明を求めることを条件にしているが、元のデータを帰納表現によって効率化することによる表現上の利得とのトレードオフを考慮し、この値を正の整数値にすることができる。

許容ノイズ	0	1	2	3
総ルール数	59	47	33	30
複数の事例を説明するルール	5	9	9	8
単一の事例を説明するルール	54	38	24	22
誤って負事例とされた正事例	0	0	0	0
誤って正事例とされた負事例	0	5	11	11

表 1: Aleph による連濁データベースの帰納計算の結果

標準的な設定による計算が最左列の許容ノイズ 0 の場合にあたる。連濁をしめす複合語のうち 8 割近い事例について、その属性に関する共有部分を見いだせなかったことになる。一方で、5 件のルールで 18 の事例が説明されている。この計算で個別のルールとして最も成績のよかったものは 6 件の正事例を説明するもので、成立した複合語が動詞の体言化であるような語であった。許容ノイズを増やしてゆくことにより、個々のルールによって説明される事例が増えてゆく。1 つのルールが説明した最大の事例数は許容ノイズが 1 のとき 7, 2 のとき 11, 3 のとき 16 であった。

許容ノイズ 3 での計算で最も成績のよかったルールを図 3 に示す。複合語の品詞が体言化した動詞、前部要素のアクセントが頭部に来る場合には、濁る場合が多いというルールが抽出されたことになる。このルールは 16 件の正事例を説明する一方、3 件の負事例をも説明してしまう。同じ計算で 2 番目に多くの正事例を説明したルールは、正事例 11,負事例 2 を説明しているので、情報圧縮の利得を単純に(正事例-負事例)と考えることにするばあい、前者が最も成績のよいルールということになる。

```
rendaku(A) :-
  cpos(A, vn), comp(A, B, C), facc(B, 1).
```

図 3. 帰納されたルールの例

6. 考察

4 章で示した標準的な帰納論理プログラミングの目標から考えれば、仮説 H を構成するルールのうち 73-91%が単一の事例そのものであったことや、77 の事例を被覆するのに 30-59 ものルールを必要としたことからは、データ全体の説明理論を求める帰納問題はあまりよい設定ではないということがいえる。一方で、77 の事例のうち 16 もの正事例を被覆できるルールが存在したことは、部分的に規則性に近いものを発掘する一助としてこの手法を追求する意味も認められるものと考える。

また、[金田一 83]には直接とりあげられていない品詞の情報を付加したことで、ルールの発見が効果的に行われた。

7. 今後の展開

連濁は ad hoc な現象だというのが定説である。言語学が追求するような厳密な法則を見出すことはおそらく困難であろう。しかしながら、今回の分析における品詞情報による情報利得効果を無視することも難しい。単一の法則が支配していない理由として、複合語が生まれた時代や地域がまちまちであるためであることが考えられる。また、今回の分析に用いた背景情報は、連濁の問題を機械学習のパラダイムにあてはめるための最低限のものでしかない。この点を考慮すると、属性をより豊かにしていくことが最も重要な方針となる。とくに、認知言語学的なカテゴリーやより詳細な活用の分類は是非考慮にいれたい。

機械学習スキームとしては、属性構築や一貫性制約の導出も有用な可能性がある。

また、連濁を行わないものを正事例とすることで、導出されてくるルールの性質が変わってくる可能性も考慮してみたい。

参考文献

[金田一 83] 金田一春彦 著, 秋永一枝 編: 新明解日本語アクセント辞典 第 2 版, 三省堂 ,1981.

[Muggleton 95] Muggleton, S: *Inverse Entailment and Progol*, New Gen. Comput., 13:245-286.

[Ray 09] Ray, O: Nonmonotonic Abductive Inductive Learning, Journal of Applied Logic 7(3): 329-340, 2009.