因果関係を用いたかな漢字変換アルゴリズム*

3L－1

山下 浩一†　吉田 敬一‡

静岡大学大学院理工学研究科§

1 はじめに

かな漢字変換システムは日本語入力の基礎に位置するものであり、その重要性は極めて高い。既存のかな漢字変換システムは共起 (co-occurrence) の概念を導入することにより、その変換精度を飛躍的に向上させた。しかし、共起の概念をもってしても完全な変換までには及んでいない。本稿では共起の概念だけでは解決できない問題を解消するために、共起にはない時間的な流れを導入した概念、因果関係を提案する。また、簡単なプロトタイプの処理結果から因果関係の有効性について報告する。

2 因果関係の定義

共起による変換では、例えば「夜が明ける」、「窓を開ける」、「グラスを空ける」という各文の「あける」をそれぞれ「夜」、「窓」、「グラス」と結びつけて一意に変換する。しかし、共起だけでは一意な変換が不可能な場合もしばしば生じる。例えば、「ビールをあける」の「あける」は「ビール」と結びつけただけでは「開ける」なのか「空ける」なのかが決定できない。ほかにも「金をかける」の「かける」、「頭をうつ」の「うつ」、「友人をたずねる」の「たずねる」などが同様の例として挙げられる。このような問題を解決するために、本稿では次のような概念を提案する。

定義 1 任意の名詞 n と共起し得る後続の動詞のひらがな表現を v とし、その v に対応する漢字を $v_1, v_2, \cdots, v_m$ とする。今、n と共起できる漢字が $v_1, \cdots, v_m$ の中に二つ以上存在する時、v は n に対して多重共起可能であると言い、n に対して多重共起可能である動詞を n－多重共起動詞と言う。

定義 2 n－多重共起動詞 v に対応する漢字が、同文中の他の動詞 v' を考慮することによってのみ決定可能となる時、v と v' の間には因果関係があると言い、v' を v の因果動詞と言う。また、v よりも文頭方向に v' が存在した場合、v' は因の性質を持つと言い、v よりも文末方向に v' が存在した場合 v' は果の性質を持つと言う。

以上のように共起だけでは一意に変換できない動詞を、それと因果関係を持つ因果動詞を用いて変換させる。また、例えば「ビールを開けて飲む」という文の「あける」は因果動詞「飲む」を用いて「開ける」と変換できるが、「ビールを飲んで空ける」という文では同じ因果動詞「飲む」を用いるにも関わらず「空ける」と異なった漢字に変換される。v の前方に現れた v' と v の後方に現れた v' とを別のものとして取り扱っているのはこのような場合の問題を解決するためである。

3 アルゴリズム

まず最初に本稿で表現される「文」、「節」について、以下の定義をする。

定義 3 句点で区切られたひと区切りのことを文と言い、文相当の語句でより大きな文の直接成分となっているものを節と言う [1]。

3.1 文の解析

因果関係の概念を導入するには、因果関係にある二つの動詞の間、あるいは共起関係にある二つの語の間に、任意に語が挿入され得るという問題点をまず解決しなければならない。例えば現在最も普及していると思われる某社のかな漢字変換システムでは、「きょりをはかる」は「距離を測る」と正しく変換されるが、「きょりをものさしをつかってはかる」は「距離をものさしを使って図る」と誤って変換されてしまう。これは「ものさしをつかって」が挿入されたことで「距離」と「測る」の共起関係が抽出できなくなってしまったためであると考えられる。因果関係は共起関係のほ

---

*An Algorithm for Kana-Kanji Conversion Using the Relation between Cause and Effect

†Kouichi Yamashita

‡Keiichi Yoshida

§Graduate School of Science and Engineering, Shizuoka University

かにもう一つ動詞を用いるため、この問題は上の例よりもさらに深刻である。この問題を解決するために、まず以下の仮定をおくこととする。

仮定 二つ以上の節は次の 1, 2 を任意に組み合わせて文を構成するものとする。

1. 各節が並列的に現れるもの
   例：ものさしを使って、距離を測る
2. 一つの節に一つ以上の節が包含されて現れるもの
   例：距離を、ものさしを使って、測る

この仮定を利用すると、全ての文はリスト構造で表すことができるようになる。さらに因果関係と共起関係は複数節に及ばないと仮定すれば、ある語のかな漢字変換に必要な情報はその語が存在する節の中にのみ存在することとなり、それ以外の節に含まれる語はかな漢字変換に曖昧さをもたらすと考えられる。従って節のリスト構造を明らかにすれば、共起関係、因果関係の抽出失敗に関する曖昧さを除いてその抽出力を高めることができ、前述の語の挿入の問題の解決策になると考察される。以上のことから本稿のかな漢字変換アルゴリズムは文の解析を含んだ形となる。ここで言う文の解析とは、節のリスト構造を明らかにするだけのものであり、比較的軽い処理で済む。

3.2 アルゴリズム

本アルゴリズムに入力される文はすでに形態素解析され、文頭の形態素から順に配列 $w[1], w[2], \cdots$ に格納されているものとする。以下にアルゴリズムを示す。

```
while (w[i] is not the end of sentence)
begin
  while (w[i] is not a verb)
    i = i + 1;
  j := i − 1; ok := false;
  while (j > 1 ∧ ¬ ok)
  begin
    if w[j] is an element of a clause
    then {if w[j] and w[i] hold the
            relation between cause and effect
      then {include all elements up to w[i]
        into a new clause; ok := true;
        convert Kana into the corresponding Kanji;}}
    else {if w[j] and w[i] co-occur
      then {construct a clause consisting of all
        elements between w[j] and w[i]; ok := true;
        convert Kana into the corresponding Kanji;}}
    j := j − 1
  end
end
```

ビールを開けて飲む	(本アルゴリズム)
ビールをあけて飲む	(某社システム)
ビールを飲んで空ける	(本アルゴリズム)
ビールを飲んであける	(某社システム)
金を懸けて争う	(本アルゴリズム)
金を掛けて争う	(某社システム)
頭を撃って殺す	(本アルゴリズム)
頭を打って殺す	(某社システム)
友人を尋ねて旅発つ	(本アルゴリズム)
友人を訪ねて旅発つ	(某社システム)
彼に会ったらあれでぴったり合ったと言っていた	(本アルゴリズム)
彼にあったらあれでぴったり合ったと逝っていた	(某社システム)

表 1: プロトタイプの処理結果

このアルゴリズムから簡単なプロトタイプを構築した。その処理結果を表に示す。また、比較のために同じ文を前出の某社システムで変換させた結果も付け加えておく。

4 おわりに

本稿では、共起の概念だけでは解決できない問題を解決するものとして、因果関係の概念を提案した。また、因果関係を導入するにあたり、因果関係にある二つの動詞の間、あるいは共起関係にある二つの語の間に任意に語が挿入され得るという問題点を、文の解析を用いて解決する方法を示した。プロトタイプの処理結果から、以上の手法を導入することにより、既存のかな漢字変換システムにおいては正しい変換が得られない文に対して正しい変換を得ることが可能となる。さらに、かな漢字変換において共起関係、因果関係による意味的な処理と、文の解析による構文的な処理の組み合わせが漢字候補の決定に有益な情報を与え、変換精度を向上させることが明らかになった。このことは今後のかな漢字変換システムにさらなる変換精度の向上が期待できることを示唆するものである。

参考文献

[1] 金田一春彦他：日本語百科大事典,p168, 大修館書店,1988.

[2] 橋川潤：同音・同訓ハンドブック,pp9-10,p44,pp155-156, 池田書店,1994.