

5M-5

会話的日本語文による
ユーザ・インタフェースへ向けて

阿部 哲也 田中 英彦 元岡 達
(東京大学 工学部)

1. はじめに

近年、自然言語処理、とりわけ日本語処理の研究は盛んになる一方であり、機械翻訳などでは実用的なものも出始めている。従って、この日本語処理技術をユーザ・インタフェースに応用して、計算機と会話的日本語文でやりとりできるようになることも期待される。

しかし、現在の日本語インタフェースでは、どうしても定形的な日本語文のみを扱うものとなりがちであり、ユーザにとって余り使い易くない。従って、より使い易くするには、強力な日本語解析部が必要となるが、そのために負荷が大きくなって、応答が遅くなっては好ましくない。

そこで本発表では、ユーザ・インタフェースとして日本語を用いることを想定した場合の日本語文解析方針と、その実装例について述べる。この方針とは、リアルタイム応答を目指したもので、スタックを用いて単語ごとに逐次的に解析を進めるものである。

2. インタフェースとしての日本語文解析

ユーザ・インタフェースの充実を目指す立場で考えると、計算機と会話的な日本語でやりとりできることが望ましい。しかし、そのために応答が遅くなるのは好ましくない。

従って、ユーザ・インタフェースにおける日本語文解析には、機械翻訳などにおける日本語文解析とは異なる側面があると考えられる。そのうち大きな2つの側面を次に挙げる。

①リアルタイム性

ユーザ・インタフェースであるから、リアルタイムでの応答が望まれる。従って、解析もリアルタイム性が求められる。

②入力文の特徴

ユーザ・インタフェースとして、入力が予想される文は会話的な日本語なので、次の様な特徴があると考えられる。

- ・文章と比べて、文の長さは短い。
- ・埋め込みが何重もある、複雑な構造は少ない。
- ・省略や代名詞、語順変化などが多い。

よって、こうした側面を考慮に入れた解析機構であることが望まれるわけである。

3. 解析方針

3.1. 基本方針

本発表では、形態素解析は考えず、単語区切りしてある入力から格フレームの生成を目指すものとする。

従来より主流となっている日本語解析においては、図1に示すように、まづ、一文が入力され、続いて形態素解析・構文解析・意味解析と進んでいく。

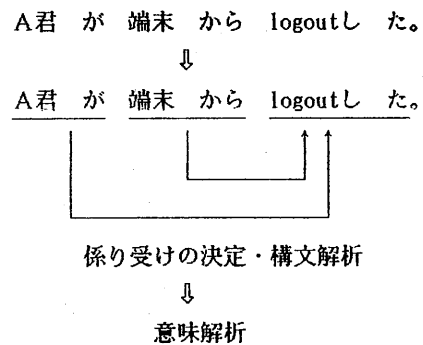


図1 現在の主流な解析方針

しかし、この方針では、一つの文全体の入力終了時から解析が始まるので、リアルタイム性を求めることは難しい。

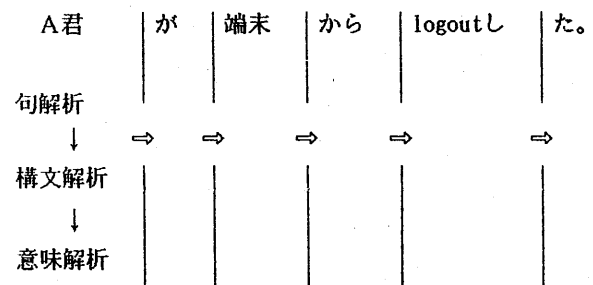


図2 ユーザ・インタフェース向き解析方針

そこで考えられるのが、図2に示すように、各単語

ごとに、入力され次第、逐次的に構文解析・意味解析を進めてしまう方針である。この方針では、文の入力中も単語と共に逐次的に解析が進んでいき、入力終了時には解析も終了へと向かう。従って、リアルタイム性を求めることが可能となりうる。

また、入力文の特徴として複雑な構造が少ない一方で多様な語順変化などが予想されるので、文法理論としては、表層で結合価文法、深層では格文法という枠組みが適していると考えられる。

以上の考察から、実際の解析では次の3種類のスタックを用意する。

①単語スタック

単語を積んでいき、格要素となりうる句ができたところで全体を句として格要素スタックへ送る。

②格要素スタック

格要素となりうる句を積んでいき、格フレームを生成しようところでスタック内容を出す。

③格フレーム・スタック

格要素スタックの中身から格フレームを生成した後、生成した格フレームを積む。

従って、入力された単語に応じて、単語スタックに積まれたり、単語スタックの中身を出して句として格要素スタックに積んだり、あるいは格要素スタックの中身を出して格フレームを生成するといった処理が行われて解析が進んでいくわけである。

3. 2. 単語ごとの処理

処理は、あくまで各単語に依存するが、品詞により、ある程度決められる。次に、その原則の例を示す。

名詞…単語スタックに積む。埋め込み文の後である

場合は、埋め込み文とのかかわりを調べる。

格助詞…単語スタックの中身をまとめ、句として格要素スタックに積む。この際、表層マーカとしての役割を果たす。

動詞…格要素スタックの中身から、結合価パターンを参照して格フレームを生成し、格フレーム・スタックに積む。

3. 3. 入力への制限

連用中止法などを扱うため、入力には、次のような制限を加える。

- ・連用中止法を用いる場合は“，”をつける。
- ・接続詞や接続助詞による、文の切れ目には“，”をつける。

- ・連体修飾や連用修飾であって、次の体言や用言を飛びこえて修飾する場合は“，”をつける。
- ・文の最後には“，”もしくは“？”をつける。

4. Prolog上への実装

以上の様な方針で、実験システムをC-Prolog上に実装している。入力方法は、キーボードからのローマ字入力であり、単語ごとにスペースをあけることとしている。Prologなので、解析の失敗によるバックトラックは自動的に行われる。図3にシステムのメイン・アルゴリズムを示し、図4に動作例を示す。

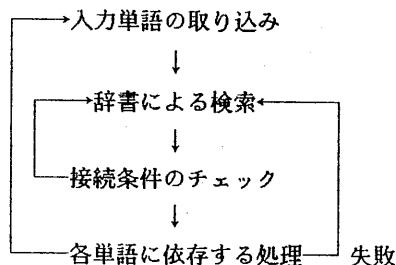


図3 メイン・アルゴリズム

```

?- start.
: taroh ha, jiroh ga logout_shi ta
tanmatsu ni login_shi ta.

[[tense,kako],[jutsu,[login_suru],1],
[loc,[u(tanmatsu)]],[theme,[taroh]]]
[[loc,[u(tanmatsu)]],[tense,kako],
[jutsu,[logout_suru],1],[agent,[jiroh]]]
  
```

図4 動作例

5. 検討課題

- ・処理単位(単語)の検討
- ・入力制限の緩和と、それに代る方法の検討
- ・会話文の特徴についての調査検討
- ・極端な省略文や代名詞、倒置文などの扱い

6. おわりに

ユーザ・インタフェースとして日本語を用いる場合を想定した日本語文解析の方針と、その実装例について述べた。単語逐次的でデータ駆動的な解析機構となっている。