

# 語彙の意味認識による形態素解析・文章認識を使用した情報提供

Information offer which used the morphological analysis  
and text recognition by semantic recognition of a vocabulary.

菲塚 昇  
Noboru NIRAZUKA

株式会社 システム イグゼ  
(〒285-0846 千葉県佐倉市上志津 1005-1-1-512 E-mail: system@olive.ocn.ne.jp )

**ABSTRACT.** Analysis of language has so far been carried out based on pattern recognition. However, by this technique, it cannot respond to the conversation using free grammar. This analysis technique makes a self-judgment of whether words and phrases are materialized as a member in the text. Autonomous processing of the words and phrases which analyze the whole text synthetically has realized by judging mutually whether it is right as a member of a text to other words and phrases.

## 1. 背景

インターネットの普及により、多くの人々がコンピュータを使用する環境が整ってきた。しかし、コンピュータの機能が向上するにつれ、専門家でも自由に使いこなすのが困難になってきている。ソフトウェアは多くの機能を盛り込む事で差別化を図り、利用者に利便性を訴えている。現在、ソフトウェアのユーザーインターフェースは、GUIにより行われ、視覚によるコンピュータ操作を実現している。しかし、機能が多く存在し、プッシュボタンが羅列しているプログラムは、利用者に多くの混乱を与える皮肉な結果をもたらしている。視覚によって簡単な操作を実現する事を目的に開発されたGUIが、多くの機能を許容する事ができる為に、かえって操作を複雑にしてしまう結果となっていると考えられる。

このような状況は、GUIによるインターフェースの限界を示し、これまでの開発手法では解決できない事を示している。これを解決する為には、「コンピュータは操作するもの」といった固定概念を取り除く必要がある。操作は、利用者が結果を得る為の手段であり、結果を直接取り出す事ができれば、操作は不要となる。文章(言葉)によって目的を入力する「意思入力」を行う事で、中間操作を省く事ができる。

これまで開発されてきた形態素解析手法では、解析する事のできない、自由な文章表現に対応し、文章が意味する動作や条件を認識する必要がある。本プロジェクトは、これまで開発してきた形態素解析システムを基盤に、意味データを登録する事で、ユーザーインターフェースとして使用できる形態素解析システムを構築する事にある。

## 2. 目的

これまで語句や文章の意味を把握する事はできていない。コンピュータに対して「意思」を入力する場合、

言葉の持つ意味を把握できなければ、状況に応じた解析を実現する事ができない。言葉の意味は、時間、空間などの全てを含んでいる為、コンピュータに認識させる事は難しい。しかし、コンピュータへの指示を行うような文章を考えた場合、意味の一部を理解する事で多くの解析が実現できる。文章には1つ、または、1つ以上の意味があり、文章を構成する語句の意味を利用して、文章に含まれる動作や条件を個別に認識する事で、どのような表現に対しても解析し、動作を決定させる事ができる。このような解析は、現在、行われているプログラミング技術では実現する事はできない。文章は、無限の表現が可能である。普段 会話で使用されるような文章は、文法的に誤っていても表現としては正しい場合が殆どである為、このような無限の文法パターンを従来の手法を使用したプログラムやデータとして登録する事はできない。

本解析手法は、語句が文章の一員として成立するか自己判断する。他の語句に対して、文章の一員として正しいかを互いに判断する事で、文章全体を総合的に解析する事を特徴とした語句の自律処理によって実現している。自由な文章表現を解析し、文章に含まれる条件や動作を認識する事で、コンピュータを利用する者が、事前に操作方法を覚える必要のないユーザーインターフェースを実現する事を目的とする。

## 3. 自律処理による形態素解析

開発した形態素解析システムは、「ひらがな」「漢字」「英数字」「記号」の何れか、又は、その組み合わせによる文章を解析し、文章全体の意味に従って同音異義語の判断を行い、文章が示す動作を認識する事が可能である。また、会話表現のような文法にとらわれない文章の解析も可能である。本形態素解析は語句の自律処理によって実現している。

プログラムは、条件に従って処理手続きを実行するものである。この処理手続きの実行により、プログラムは特定の処理対象を解析する事ができる。逆に言えば、

この処理手続きから外れた処理対象は解析できない。しかし、処理手続きを生成する事ができれば、プログラムはあらゆる処理対象に対して解析できる事になる。このような観点から開発したのが自律処理である。

自律処理による形態素解析では、下記の機能を有している。

- 品詞による語句の分離
- 語句の合成による品詞の推定
- 文章の意味から推定する同音異義語の判定
- 文章に定義されている動作や条件の情報抽出
- 文章に定義された情報の蓄積
- 文章による情報検索
- 文章に定義された複数の動作を実行

### 3.1 自律処理

プログラムは、処理対象を分析し、その処理要素と条件の集合として定義できる。

図1は、3個の処理単位と2個の条件分岐を持つ簡単なプログラムのフローチャートを示したものである。

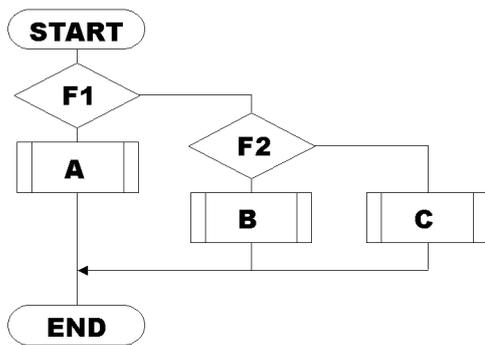


図1:一般的なロジック構造例

処理 A, B, C は、条件分岐 F1, F2 により選択される。条件 F1, F2 は、

- 条件 F1 が(真)の場合、処理 A を実行
- 条件 F2 が(真)の場合、処理 B を実行
- 条件 F2 が(偽)の場合、処理 C を実行

と定義した場合、処理 A, B, C から処理全体を考えると、実行できる条件は固定されている。この処理 A, B, C が実行できる条件は、

- 処理 A:  $F1$
- 処理 B:  $\bar{F1}$  and  $F2$
- 処理 C:  $F1$  and  $\bar{F2}$

$\bar{F1}$ ,  $\bar{F2}$ は Not F1, Not F2 を示す

と記述できる。これを

- $f_A = F1$
- $f_B = \bar{F1}$  and  $F2$
- $f_C = F1$  and  $\bar{F2}$

と記述する。  $f_A, f_B, f_C$  は、処理 A, B, C の実行条件となる。処理 A, B, C からこの実行条件を考えた場合、処

理 A は  $f_A$  が成立すれば実行できる必要十分条件を満足する事になる。処理 B, C についても同様である。この事は、処理 A, B, C は他の処理に影響されない独立の関係にあると言う事ができる。図1では簡単な処理について記述しているが、より複雑な処理を考えた場合でも、この関係は成立する。

処理 A, B, C の内部処理として、 $f_A, f_B, f_C$  の判定を記述すると、処理 A, B, C は独立の関係があるので、図1は、処理 A, B, C を線形接続した関係として記述できる。処理 A, B, C を線形接続した処理に対して、 $f_A, f_B, f_C$  の何れかの条件を与えた場合、図1のロジックが成立する。処理 A, B, C の線形接続を独立させ、処理 A, B, C へ動作条件を渡す管理を処理 F として記述すれば、図1のフローチャートは、図2のように表現する事ができる。

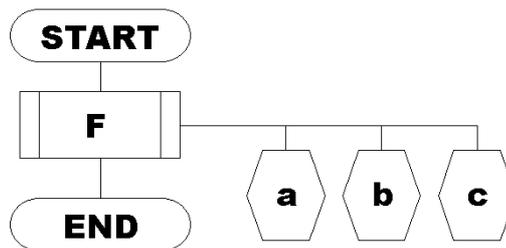


図2:処理条件を包含した処理構造

図2の処理 a, b, c は、処理 A, B, C へ動作条件  $f_A, f_B, f_C$  の判定を内部記述している事を示す。

処理 a, b, c は、出現順序に意味はなく、処理 F から渡される条件のみに依存し、処理 F は、F に接続されている処理の識別を行う必要がない。処理 F へ接続された処理 a, b, c から見た場合、処理 F にのみ依存している。処理 a, b, c の動作は、内部記述した動作条件を満足するかのみによって決定され、独立した存在である。このような自己に内在する判断により、動作を決定する単位処理を自律処理と定義する。

自律処理は、独立した存在であり、階層構造を持たない線形構造により表される。一般的に処理 a,b,c,...n から成るプログラムは、図3のように記述できる。

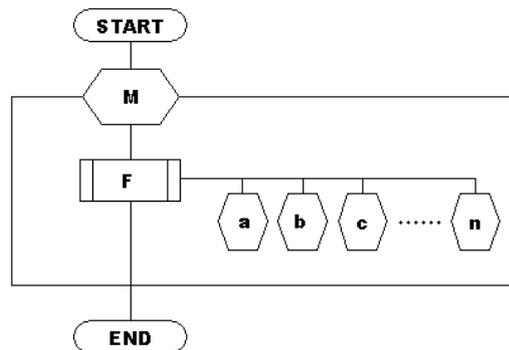


図3:自律処理構造

図3は、処理 F から渡される条件により自律処理要素 a,b,c,...n の判断により実行され、処理 F の条件が変更される。自律処理要素は条件を満足する全ての要素が対象となる。判定 M により、入力した条件が平衡状態になるまで繰返される。従って、図3は、自律処理要素の数と平衡状態になるまでの回数数によって、プログラムの動作が決定できる事を示している。

処理対象となる事象を解析する為の要素に分割された自律処理要素の線形接続構造によって、自律処理構造を持つプログラムは、自律処理要素を組み合わせて定義できる全ての処理手続きを生成可能である事を示している。

### 3.2 語句の意味定義

従来の言語解析では、品詞、文法、用法などを利用し、カテゴリー別に分類された語句によって意味を推定している。このような解析方法は、決められた文法や数値化された意味データを用いたパターン認識を基本としている。また、語句の意味を TREE 構造やネットワーク構造により定義する試みが行われている。このような手法は、語句の意味を分類し、事前に分類した意味の組み合わせを固定した構造により定義している。しかし、通常、語句は複数の意味があり、どのような語句と組み合わせられて使用されるかによって意味が決定する為、固定構造を持つ意味定義では、構造自体が成立しない欠点を持っている。

本解析方法では、語句の意味を次のように定義している。

語句の概念と属性  
選択条件による判断  
情報分割の判断  
動作決定の判断  
回答文生成の判断

～ の判断は、スクリプト言語によって、小さなプログラム（数ステップによるマイクロプログラム）として定義し、各プログラムは独立した判断を実行するように作成されている。

#### 語句の概念と属性

本解析方法で使用する語句の概念は、形態素解析で利用する為に、カテゴリー別に分類する事なく、本来、語句が持っている集団、どのような事に使用されるかといった内容を言葉によって定義した語句群で表される。複数の意味を持つ語句に対しては、その全ての意味に対して概念語句群を分類する事なく定義する。

本解析方法による語句の概念とは、

「語句から想像する事のできる他の語句」と定義している。

属性は、振り仮名、品詞、活用などの固定データを定義する。

概念語句群は、語句が使用される条件を示しています。このような概念と属性を使用して常識的判断による解析が行われる。

#### 選択条件による判断

語句は、一般的に、複数の意味がある。日本語の場合、同音異義語が多くある為、語句を選択する判断が必要になる。同音異義語の判断は、語句毎に異なり、語句の意味形成の一部として定義する。これは、常識的判断による Default 語句を設定しておき、文章を構成する他の語句に、この Default 語句を排除する情報があ

るか判断する事で、他の語句から影響される語句によって置き替える手続きによって同音異義語の判定を行っている。

語句の合成は、語句が他のどのような語句と接続されていて、どのような位置に置かれているかの判断を手続きによって定義する。

#### 情報分割の判断

語句の修飾や条件定義などにより文章が切断できる範囲を特定し、その範囲を抽出する判断と手続きを定義する。範囲の確定は、離れた語句に対して修飾している状態や、1つの語句が複数の語句へ修飾している場合などについても判断を行うように定義する。このような複雑な判断は、言葉と言葉の関係から、語句の特徴として定義する意味の一部となる。

#### 動作決定の判断

文章が解析された結果、その文章がどのような動作を内装しているかを判断する手続きを定義する。解析した文章を構成する語句の組み合わせとして、動作定義する条件が満足しているかを判断し、条件が満足している場合、語句によって定義されている個別動作の組み合わせによって全体の動作を決定する。複数の動作が含まれている場合は、情報分割判断によって切断・抽出された手順に従って動作を実行する。

#### 回答文生成の判断

は動作による回答であり、は、言葉による回答を生成する判断と手続きを記述する。

本形態素解析システムでは、文章によって入力された文章を条件や情報毎に抽出し、その関係を維持した状態で情報蓄積する事ができる。この蓄積した情報を質問を受け付ける事で、質問に含まれる検索条件を解析し、回答を生成する手続きを定義する。

### 3.3 語句の自律処理による形態素解析例

語句は、概念語句と形態素解析の為の自律処理が定義されている。

本形態素解析では、

「かきのみをたべた」を入力した場合、

```
Input(1)=>かきのみをたべた
Anal=>>柿 の 実 を 食べた
Parts=>>柿 の 実 を 食べた
Kana=>>かき の み を たべた
情報(1)かきのみをたべた =>> 柿 の 実 を 食べた
```

と解析し、「かいのかきをたべた」を入力した場合、

```
Input(2)=>かいのかきをたべた
Anal=>>貝 の 牡蠣 を 食べた
Parts=>>貝 の 牡蠣 を 食べた
Kana=>>かい の かきを たべた
情報(1)かいのかきをたべた=>>貝 の 牡蠣 を 食べた
```

と解析する。このような解析は、文法パターンに当てはめて解析しているのではなく、語句と語句の関係により成り立っている。「かきのみをたべた」は、

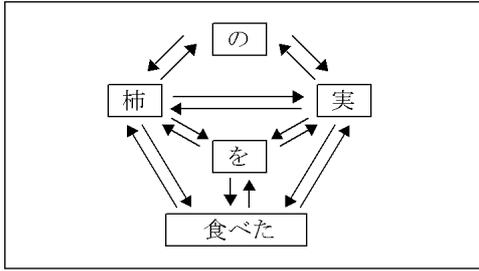


図4:語句と語句の関係

最終的に、図4に示す関係が成立している。

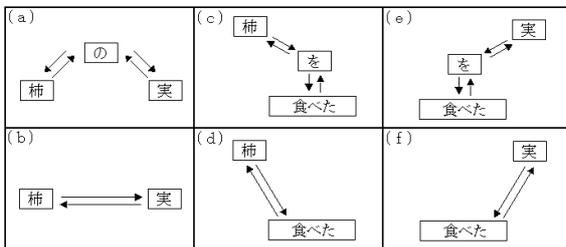


図5:個別語句の関係要素

図5の(a)から(f)は、図4の語句と語句の関係を要素に分解したものである。分解された要素は、図4の関係に依存しない独立した関係として成立している。(a)は、格助詞『の』を中心に名詞と名詞を接続する関係を持つ。『の』の後に動詞が来るような場合、名詞句となり、『の』の前後には必ず、名詞が成立する。どのような名詞が来るかは、(b)から(f)の関係によって定義されている。(a)の関係から、(b)の関係として成立するのは、同類の名詞であり、(c)から(f)の関係から「食べ物」を表す名詞である事が推定できる。何故なら、目的語としての名詞の意味は動詞の影響を受けて意味を推定できる関係があり、『食べる』の目的語として「食べ物」を表す名詞が推定される。

(a), (c), (e)のような助詞などによって接続される関係は、助詞によって判断する事ができ、助詞に接続される語句は、(b), (d), (f)のように、接続語句同士、及び、動詞との関係によって定義できる。

「かきのみをたべた」に対して、組み合わせ可能な語句は、「か」から始まる、以下のものが存在する。

[柿](かき)	[書](か)<き>
[花期](か き)	[欠](か)<き>
[描](か)<き>	[火気](か き)
[牡蠣](か き)	[夏季](か き)
[花卉](か き)	[夏期](か き)
[花器](か き)	[下記](か き)
[課](か)	[化](か)
[科](か)	[か]
[垣野](かき の)	[個](か)
[掻](か)<き>	

同じように、「き」から始まる語句、「の」から始まる語句、... 「た」から始まる語句が存在する。各語句

は図5に示すような個々の語句要素の関係を持っている為、文章を構成する要素として判断した場合、他の構成要素との関係から、図4に示すような矛盾しない関係を成立させる必要がある。この関係が、語句の絞り込みを行い、「花器の実を食べた」のような矛盾した解析結果を成立させない判断が実行されるのである。

語句と語句の関係は、語句の意味に影響を与え、逆に、この関係から語句の絞り込みを可能にしている。この関係を語句の意味データとして登録したのが、自律処理と語句の概念データである。自律処理による語句解析は、使用文法に関係なく、語句と語句の関係のみによって、語句の意味を絞り込む事で、同音異義語の判定や修飾語句の判定を行い、結果として文法を生み出している。

<実際の自律処理の一部>

```

Input->[かきのみをたべた]

Default Script Exec(0x00000001)

Exec Script ID= 7323 語句:[柿][かき] Pos = 0 Span = 4
Parts: ANAL(0x0000
0001) DIC(0x00000001)
Line .....1.....2.....3.....4.....5.....6....
+
0001: 00 if(動詞スル(CUR, 0, NONE))
0002: 00 {
0003: 01 if(!IsConnectParts(全品詞, NEXT)) 動詞スル
(CUR, 0, CHANGE);
0004: 01 }
0005: 00 if(IsSelectScript()) return(NULL);
0006: 00 if(!IsConnectParts(全品詞, NEXT))
ChangeCurParts(全動詞);
0007: 00 return(NULL);
Line .....1.....2.....3.....4.....5.....6....
+
< Select Script 実行結果 >
Line -----
0001: if(偽) = FALSE
0005: if(偽) = FALSE
0006: if(偽) = TRUE => ! => FALSE
0007: return(NULL);

Default Script Exec(0x00000001)

Exec Script ID= 5980 語句:[花期][かき] Pos = 0 Span = 4
Parts: ANAL(0x00
000001) DIC(0x00000001)
Line .....1.....2.....3.....4.....5.....6....
+
0002: 00 {
0003: 01 if(!IsConnectParts(全品詞, NEXT)) 動詞スル
(CUR, 0, CHANGE);
0004: 01 }
0005: 00 if(IsSelectScript()) return(NULL);
0006: 00 if(!IsConnectParts(全品詞, NEXT))
ChangeCurParts(全動詞);
0007: 00 return(NULL);
Line .....1.....2.....3.....4.....5.....6....
+
< Select Script 実行結果 >
Line -----
0001: if(偽) = FALSE
0005: if(偽) = FALSE
0006: if(偽) = TRUE => ! => FALSE
0007: return(NULL);
:
:

```

```

:
Exec Script ID= 44968  語句:[田][た] Pos = 14  Span = 2
Parts: ANAL(0x000
00001) DIC(0x00000001)
Line  ....+....1....+....2....+....3....+....4....+....5....+....6....
+
0001: 00 if(動詞スル(CUR, 0, NONE))
0002: 00 {
0003: 01  if(!IsConnectParts(全品詞, NEXT))  動詞スル
(CUR, 0, CHANGE);
0004: 01  }
0005: 00 if(IsSelectScript())  return(NULL);
0006: 00 if(!IsConnectParts(全品詞, NEXT))
ChangeCurParts(全動詞);
0007: 00 return(NULL);
Line  ....+....1....+....2....+....3....+....4....+....5....+....6....
+
      < Select Script 実行結果 >
Line  -----
0001: if(偽) = FALSE
0005: if(偽) = FALSE
0006: if(真) = FALSE => ! => TRUE
0007: return(NULL);

```

このスクリプトによる判断は、形態素解析を行なう処理として定義されている。同じように、語句と語句の関係から、語句の合成、文章は示す複数の動作や、質問に対する回答文を生成させる事が可能となっている。

#### 4 . 結論

本形態素解析は、語句と語句の関係、及び概念語句によって個々の語句が文章として意味を成す事が出来るかを語句に登録された自律処理によって判断している。また、語句の合成や、文章が示す動作、質問に対する回答文の生成を同じように自律処理によって生成している。この解析方法によって、文法パターンを持つ必要が無く、最小限の辞書構成で語句の合成を行う事が出来る為、これまで不可能であった日常会話や専門領域での言語解析を詳細に解析する事が可能になった。文法にとらわれない自由な文章表現に対応した解析が行える事で、コンピュータに対して、利用者は自由な文章で命令する事が可能になる。現在までに開発した内容は、2年計画の開発の初年度分に相当し、形態素解析を行なう段階にある。2002年度では、実際に、言葉のみによる命令(意思入力)によってコンピュータをコントロールし、インターネットのホームページにアクセスし、利用者が知りたい情報を言葉によってやり取りできるシステムを構築する計画である。その後、音声入力を付け加える事で、コンピュータに話し掛けるだけで、必要な情報を取り出せるシステムを構築する計画である。

このようなシステムを開発する事で、専門知識が必要となるコンピュータの操作自体を省略でき、コンピュータを使用して情報を取り出す事が可能になる。また、操作が言葉によって実施される為、コンピュータ画面に依存した操作が不要になる事で、コンピュータの形状が自由に設計でき、これまでとは異なったコンピュータの使用法が生まれる可能性を含んでいる。更に、携帯電話に代表されるモバイル端末において、これまでとは違ったサービスを提供できる可能性がある。

#### 5 . 参加企業及び機関

株式会社 システム イグゼ

#### 6 . 参考文献

なし。