自然対話文の認識・理解に関する研究（第 1 報）

－日本語文の形態素解析、構文解析－

山本　　寧、橋場　参生

Natural Language Dialogue Understanding（Part I）

-Morphological and Syntax Analysis of Japanese Sentence-

Yasushi YAMAMOTO , Mitsuo HASHIBA

抄　録

本研究はキーボードから入力された日本語文解析を行うコンパクトなプログラムの開発を目的とする。文解析の対象は対話内容を限定したものであり、比較的短い文である。18 種の品詞に分類した辞書を作成し、日本語の規則を取り入れて、入力した文字列を単語に切り分け、品詞の確定を行う「形態素解析」のプログラムを作成した。また、5W1H のテンプレートへの当てはめを前提として、構文（文法）を設定して、単語（品詞）の並びを文法と照合する「構文解析」のプログラムを作成した。

1．はじめに

人とコンピュータのコミュニュケーションを図る上で、文の理解に関する研究が注目され、数多くの報告が出ている。現在、日本語文の理解を行うプログラムでは、汎用的なものは出現していない。しかし、対話内容を限定して、実用となりつつある文解析プログラムはいくつか出始めている。対話の内容が多岐にわたる日本語文の理解を行うためには、文解析に必要とする時間も多くなり、また、参照する辞書も大きくなる。しかし、応用ソフトウェアにおいて、文解析が必要とされる場合､対話内容は限定されることが多い。そのため、文解析に必要とする時間（応答時間）や辞書のサイズもそれほど大きくはならないと考える。

文解析は一般に、入力された文字列を単語に切り分け、品詞を確定する「形態素解析（字句解析)」、単語に分解した文が文法に適合するかどうかを検証する「構文解析」、文字の並びが具体的に何を意味するかを解析する「意味解析」の順に行う（図 1)。本研究においてもこの手順に従った。解析対象はコンピュータからの質問に対してキーボードから入力（応答）する、比較的短い日本語文とした。その内容は、業務予定の問い合わせ、趣味の問い合わせなどに限定したものであり、入力する文字の種類は、“読み”と対応するひらがな（全角）とした。プログラムの作成言語はパソコン用の C 言語を用いた。意味解析のプログラムも一部作成したが、これについては適用分野毎に異なるため、本報では日本語対話文の形態素解析、構文解析についてのみ報告する。

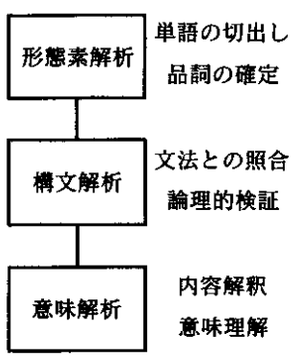

図1　文の解析手順

2．辞書の作成

文解析では、入力した文字列と単語を照合するための辞書を必要とする。単語の分類方法に関しては、国語辞典や国文法を基本としたが、文解析を容易にするため、一般的な分類方法とは若干異なる。

作成した辞書は単語を 18 の品詞に分類し、ひとつの単語に対して“読み”と“表記”および“属性”の構造体で構成した（表 1 、図 2 、図 3)。属性値は意味合いを表す 32 ビット

表1　品詞の分類（18種類）

文節先頭可（自立語）		文節先頭不可（付属語）	
品詞名	例	品詞名	例
動詞	いく（行く）	助詞	わたしは（私は）
名詞	おうふく（往復）	助動詞	します（します）
固有名詞	やまだ（山田）	形容動詞	しずかだ（静かだ）
時間名詞	さんがつ（3月）	接尾辞	ごさい（5才）
代名詞	かれ（彼）	その他１	－
数詞	いち（1）		
接頭辞	おれい（お礼）		
連体詞	あのひと（あの人）		
形容詞	はやい（早い）		
副詞	しばしば		
感動詞	いいえ		
接続詞	しかし		
その他	えーと		

（ｕｎｓｉｇｎｅｄ　ｌｏｎｇ）のコードである。この値に対して、活用形等品詞的な意味と言葉自体から連想する意味（行くという動詞に対して、意志、移動等）を持たせた。属性値は主として意味解析の際に参照する情報であるが、構文解析において構文パターンを決定する際にもこの値を参照した（文種の識別など）。

```
#typedef ULONG unsigned long

#define   Syushi      0x08000000
#define   Rentai      0x04000000
#define   Utikeshi    0x00200000
#define   Ishi        0x00100000
#define   Dou_idou    0x00000800
#define   Dou_koumu   0x00000200
  ・・・・・・・・・・・・・・・・・・・・

struct KOTOBA {
 char   *YOMI   ; /*  読み  */
 char   *HYOKI  ; /*  表記  */
 ULONG   ATR    ; /*  属性  */
 }

struct KOTOBA DOUSHI [] = {
                 /*動詞*/
  ・・・・・・・・・・・・・・・・・
  "しゅっちょうする"  ,  /*  読み  */
  "出張する"          ,  /*  表記  */
  Syushi              /*  終止  */
 + Rentai             /*  連体  */
 + Ishi               /*  意志  */
 + Dou_idou           /*  移動  */
 + Dou_koumu          /*  公務  */
          ,
  ・・・・・・・・・・・・・・・・・・
}
  **  １８種の品詞毎に登録
```

図2　辞書の構成と登録例

＜文節の頭にはこない品詞：付属語＞

31	30	29	28
27	26	25	24
23	22	21	20
			その他1
19	18	17	16
接尾辞	形容動詞	助動詞	助詞

＜文節の頭にくる可能性のある品詞：自立語＞

15	14	13	12
			その他
11	10	9	8
接続詞	感動詞	副詞	形容詞
7	6	5	4
連体詞	接頭辞	数詞	代名詞
3	2	1	0
時間名詞	名詞	固有名詞	動詞

＊＊　数字はｂｉｔ番号

```
unsigned long HINSHI [] ;
                ／*品詞*／
```

図3　品詞のコード割当

3．形態素解析の方法

コンピュータのキーボードから入力した文字列を切り分け、品詞を確定する「形態素解析」は、左最長一致法を基本とした。左最長一致法は、入力した文について、文の頭から、辞書と一致した単語のうち、最も長さの大きいものを選び（切り出し）、残った文字列に対して同様の操作を繰り返す手

入力文：

とうきょうはにほんのしゅとである

最初の一致：　とうきょう
残　　り：　はにほんのしゅとである

（繰り返し）

結果：

とうきょう　は　にほん　の

しゅと　で　ある

図4　左最長一致法による形態素解析の例

順である（図4）。 しかし、この方法だけでは期待した通り切り分けができない場合が生じた。

例えば、“はつか”と単語が登録された辞書を用いて「ちとせはつかなざわいき」という文の切り出しを左最長一致法に対応したプログラムを作成して字句の切り出しを行った時に「ちとせ／はつか／な／ざ／は／いき」という結果が得られた。

人の対話は、通常、一方の人が質問し、その相手が答える（応答する）という形で展開していく。対話の内容（分野）は多岐にわたる場合もあるが、場面を限定した場合、使用頻度の多い言葉が存在する。また、質問に対する応答文では期待（想定）される単語や句が存在する。そこで、場面に対応する“予約語”という辞書を別に作成し、最初の段階で、入力された文字列の中に予約語があるかどうかを調べ、予約語の前後で文を区切り、この方法で区切った文字列をさらに左最長一致法によって区切ることとした。

こうした考えを適用し、“ちとせ’、“かなざわ”、“とうきよう”などの単語を予約語とし、予約語のマッチングを先に行うことによって、「ちとせ／はつ／かなざわ／いき」という意図した字句の切り出しが行えた。

図5に予約語を用いた文の切り出しの例を示し、さらに左最長一致法を適用した結果を表2に示す。

形態素解析では字句の切り出しの他に、品詞の確定を行う必要がある。作成した辞書照合のプログラムでは、単語の照合を品詞毎に行ない、読みが一致すると図3の品詞位置のビットを立てる。単語の読みが複数の品詞の辞書にマッチングする場合、図3示す品詞の配列の値は複数のビットが立つ。

文例：

らいしゅうのきんようにとうきょうにしゅっちょうするよていです

予約語の例：

とうきょうほんしゃ	しゅっちょうよてい	しゅっちょうする
しゅっちょう	しっちょう	らいしゅう
とうきょう	おおさか	らいげつ
ほんしゃ	よてい	あした

予約語との照合による切り出し結果：

らいしゅうのきんように

とうきょうに

しゅっちょうする

よていです

図5 予約語を用いた文の切り出し

“さん”という読みの単語では数詞、接尾辞に相当するビットが立ち、“に”という読みの単語では数詞、助詞に相当するビットが立つ。そこで、前後の品詞の並びから対象とする単語の品詞を確定（選択）するためのプログラムを作成した。読みが同じ単語の品詞の確定手順の例（数詞の“に”と助詞の“に”）を表3に示す。

表2 応答文の形態素解析結果の例

読み 表記	品詞コード 品詞名	文字位置	文字長さ	属性コード値
らいしゅう 来週	8 時間名詞	1	10	40
の の	10000 助詞	11	2	88010000
きんよう 金曜	8 時間名詞	13	8	40
に に	10000 助詞	21	2	814a0000
とうきょう 東京	4 固有名詞	23	10	21800000
に に	10000 助詞	33	2	814a0000
しゅっちょうする 出張する	1 動詞	35	16	c100a00
よてい 予定	2 普通名詞	51	6	80000
です です	20000 助動詞	57	4	2000

表3 品詞選択の確定手順例

単語の位置と選択基準	選択品詞	文例
文節の始め	数詞	にじゅうのはる
文節の途中		
次が接尾辞の場合	数詞	にさい
次が数詞の場合	数詞	としはにじゅう
次が動詞	助詞	さんじゅうになる
前が名詞系	助詞	こうえんにいく
その他の場合	助詞	−
文節の終わり	数詞	としはさんじゅうに

4．構文解析

文法との照合を行う構文解析の方法はいくつか提案されているが、文の理解では、最終的には意味を解析することが必要となる。そこで、入力された文を5W1H（WHEN、WHERE、WHO、WHAT、WHY、HOW）の句（テンプレート）にまとめることを前提として、構文パターンを設定し、構文をコード化した。実際には、形態素解析で品詞に区切ったものを一致した構文パターンと照合し、5W1Hの各パターンに当てはめた。

構文解析の手順を以下に示す。

1）5W1Hの各句（テンプレート）に対応した構文パターンを予め作成する（文型を用意する）。

2）キーボードから入力した文をWHEN句、WHERE句、WHO句、WHAT句、WHY句、HOW句の構文と照合し、どの句かを判定する。

3）品詞の並びと品詞の属性を主体として各句の構文コードを得る。

4）照合した文字列を相当するテンプレートに当てはめる。

5）照合が済んでいない文字列に関して2）～4）を繰り返す。

6）各句毎に、一致した構文パターンのコード値を与える。

なお、構文パターンのコード値は品詞の並びを基準とし、以下のように設定した。

構文パターンコード＝
（第1単語の品詞番号）＊（2の24乗）＋（第2単語の品詞番号）＊（2の16乗）
＋（第3単語の品詞番号）＊（2の8乗）＋（8ビットの識別番号）

品詞は18種類であるため、品詞番号は8ビットの値として（5ビットで間に合うが8ビットとした）、上位24ビットに品詞の並びの情報を与え、最下位の8ビットの値に意味的あるいは補助的な情報を与えた。意味的な情報として、否定的か肯定的あるいは疑問的な構文かの判定を補助する情報を付与した（後段の意味解析を容易化する）。補助的な情報は品詞の並びが同じ構文に対して識別するための情報である。

構文パターンの例を図6に示す。コンピュータからの質問に対し、キーボードから入力した文を形態素解析のプログラムで字句を切りわけた後、構文解析のプログラムで5W1Hのテンプレート（構文）に当てはめた例を図7に示す。

```
WHEN句：
    時間名詞    [助詞]  時間名詞      [接尾辞] [助詞]
例："らいしゅう" ["の"]  "きんよう"    ["び"]   ["に"]

    時間名詞    ["の"]  数詞          [接尾辞] ["に"]
例："らいげつ"  ["の"]  "にじゅうご"  ["にち"] ["に"]
  ・・・・・・・・・・・・・・・・・・・・

WHERE句：
    固有名詞    [ 助詞 ]
例："とうきょう"  ["へ"]
    固有名詞    名詞          (助詞)
例："とうきょう" "ほんしゃ"   ("に"|"へ")
  ・・・・・・・・・・・・・・・・・・・・

・・・・・・・・・・・・・・・・・・・・

WHAT句：
    動詞
例："しゅっちょうする"              ：／＊  肯定  ＊／

    動詞              助動詞
例："しゅっちょうし"    "ます"      ：／＊  肯定  ＊／

例："しゅっちょうし"    "ない"      ：／＊  否定  ＊／

  ・・・・・・・・・・・・・・・・・・・・・・・・

＊＊  [ ] 内は省略化、（A | B）はAまたはBを意味する
```

図6　構文パターンの例

```
質問：コンピュータ
  らいげつしゅっちょうのよていはありますか
応答：ユーザ
  らいしゅうのきんようにとうきょうにしゅっちょうするよていです

WHEN：
  らいしゅう  の  きんよう  に    構文コード  040e0401
WHERE：
  とうきょう  に                  構文コード  030e0001
WHAT：
  しゅっちょうする  よてい  です  構文コード  01020f02
```

図7　構文（5W1Hのテンプレート）への当てはめ

5．おわりに

辞書の作成を行い、形態素解析、構文解析のプログラムを作成して、複数の文例を用いて検証した。設定した文法にかなった入力に対して、良好な構文コードが得られることを確認した。結果として、対話内容を限定した場合において、日本語文の形態素解析、構文解析を行う基本的な部分のプログラムが得られた。形態素解析においては、期待される入力の単語（予約語）の辞書を作成することによって、解析の速度や精度の向上を図った。また構文解析においては 5W1H のテンプレートへの当てはめを前提として、構文を設定しているため、意味の解析が容易化すると考える。しかし、人のキー入力や話言葉の単語の並びは必ずしも期待通りにならないこともあり、特にキー入力においては誤入力も少なくはない。入力全てが設定した構文と合わなければ再入力してもらうというのでは、文解析プログラムとして実用とはならないという指摘もあり、いわゆる、「頭を働かせる」といった推論機能を付加することが今後の大きな課題と考える。

また、コンピュータの性能向上にともない、音声認識・理解の研究も活発化してきている。音声認識・理解は文字列の認識・理解よりも難しいが、対話内容を限定し、さらに、用いる言葉も限定した場合には音声と文字とのパターンマッチングも十分に行い得る。音声の場合においても「形態素解析」、「構文解析」ということが必要であり、研究要素としては共通的な部分も多いため、本研究の結果をこうした分野にも活用していく。

参考文献

1）多近洵一　：くわしい国文法（1993）
2）金田一京助：明解国語辞典（1970）