論文 88-25-10-6

PC를 이용한 일·한 번역 시스템 ATOM의 개발에 관한 연구(I)
-해석 사전 구성과 형태소 해석을 중심으로-

(Development of Japanese to Korean Machine Translation System ATOM Using Personal Computer I - Dictionary Construction and Morphological Analysis -)

金 榮 選* 金 漢 宇* 崔 炳 旭*

(Young Sum Kim, Han Woo Kim and Byung Uk Choi)

要 約

정확한 형태소 해석과정의 구현을 위해 heuristic 정보를 부가한 형태소 사전과 접속정보 테이블을 구성하고 문절수최소법을 근간으로 하여 자동 띄어쓰기 과정을 구현한다. 또한 독립적인 활용어 테이블을 구성하고, 접속정보 테이블과 상호 연계시켜 적용함으로 해서 접속정보와 활용어 정보의 구성을 간단하게 하였으며, 시스템의 검증과 확장 효율을 제고하였다.

번역 사전은 해석 사전과 생성 사전으로 구성하며, 해석과정의 효율과 보다 자연스런 역어의 생성을 위해 통계적으로 추출한 고빈도의 종결구를 관용어로 기술하고, 사전상에 직접 프로시쥬어를 기술하여 시스템의 적응성을 증대시켰다.

Abstract

In this paper, we describe heuristic information-added morphological dictionary and connection table, and automatic MUNJEUL separation process on the basis of least cost method for efficient morphological analysis.

It is simplified the composition of connection and inflective word information by mutually interconnect conjugation table with connection table. As a result, the applicability of system is increased.

Translation dictionary consists of analysis and generation part and, increase the applicability by describing frequently using termination phrase which is extracted statistically as idiom and the procedure directly on the dictionary for the efficiency of analysis process and more natural generation of translation sentence.

*正會員, 漢陽大學校 電子通信工學科
(Dept. of Elec. Comm. Eng., Hanyang Univ.)
接受日字 : 1988年 5月 31日

I. 서 론

일·한 번역 및 한·일 번역 시스템의 구현은 언어 지역적 특성상 우리와 일본 양국의 공동 과제이나,

현재의 실정은 필요성의 증가에도 불구하고 소수의 연구자에 의한 선험적인 연구 성과에 의존하고 있으며 개발중이거나 상용화된 시스템도 극 소수이고 시스템이 갖는 번역 대상의 범위나 역문의 품질도 아직은 상당히 제한적인 상태에 머무르고 있다. 또한, 번역 시스템의 개발지원 환경의 정비도 미비하며, 대상 시스템도 중형 이상이어서 시스템의 범용화에는 상당한 제약이 있다.

일·한 번역 시스템을 구현하는 경우에 가장 난점으로 지적되는 사항은 일본어 입력문이 문절 단위로 분절되지 않은 연속된 문자열로 구성되어 있기 때문에 문절 분리를 포함하는 형태소 해석과정의 구현이 난해하다는 것이다. 즉, 입력문 열이 연속되어 있기 때문에 단어의 접속 가능성이 점차 누증되므로, 해석결과의 일의성을 보장할 수 없는 난점이 존재한다. 결국 양질의 일·한 번역 결과를 얻기 위해서는 형태소 해석과정의 효율성을 제고해야 하며, 이는 이 후 번역과정의 품질을 예상할 수 있는 척도라고 볼 수 있다.

한편 번역과정의 입력 데이타로 되는 번역 사전의 구성은 실제 시스템의 구현 과정에서 가장 핵심적 테마라고 할 수 있으며, 번역 프로세스 기술의 난점을 제외한다면 전체 작업의 8내지 9할 정도의 개발 노력이 요구되는 분야이다. 즉, 사전 데이타 자체가 번역 프로세스의 입력으로 되기 때문에 시스템 전체의 효율과 번역의 품질을 좌우한다고 할 수 있다.

대표적인 문절 분리(자동띄어쓰기) 알고리듬에는 최장일치법과 문절수 최소법이 있으며, 시스템의 구성 측면에서 볼 때 필요 기억량과 일반해를 구하는 시간에서는 최장일치법이 우수하지만, 적격한 해석을 구하는 시간과 최초에 얻어진 해석의 적격도에서는 문절수 최소법이 상대적인 우위를 갖는다. 그러나 실제 시스템을 구축 할 때는 알고리듬의 상대적인 비교 우위보다는 해석 사전의 효율적인 구성과 접속정보 데이타와 활용어정보 데이타등의 효율적이고 정확한 기술이 시스템의 능력을 좌우하는 척도로 된다. 즉, 효과적인 형태소해석과 구문해석 과정의 실현은 단어간의 접속 가능 여부의 엄밀한 정의와, 다수의 출력 결과가 예견될 때 이의 해소 방안을 설정하는 문제의 연구가 선행되어야 한다.[1,7,8,11,12]

본 논문에서는 범용성을 갖는 일·한 번역 시스템의 실용화를 목표로 PC를 이용한 번역 시스템 ATOM (jApanese to kOrean machine translation)의 제과정을 구축하고, 시스템을 개략적으로 이분하여 번역 환경을 지원하는 번역지원 S/W의 개발과 heuristic과 프로시쥬어를 부가한 해석 사전의 구성, 그리고 최장일치법을 근간으로 하는 접속정보 테이블과 활용어 테이블의 연계에 의한 문절분리 과정을 포함하는 형태소해석에 대하여 논한다. 그림 1은 본 연구에서 구축한 시스템의 개략도이며, 본 논문에서는 시스템의 전반부 즉, system(1)의 부분을 중심으로 기술한다.

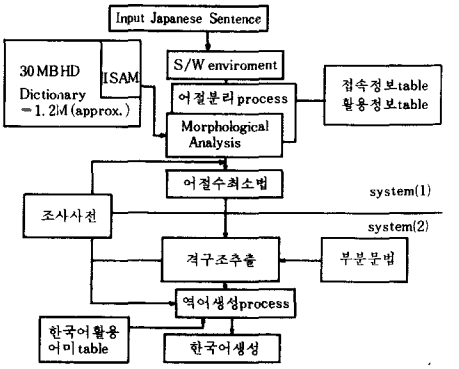


그림 1. ATOM 시스템의 구성도
Fig. 1. ATOM system configuration.

Ⅱ. 기계번역을 위한 지원 환경과 사전 구성

1. 입출력 인터페이스의 설계

번역지원 환경으로서 입출력 인터페이스는 기계번역 시스템과 입력·출력문과의 매개 역할을 수행한다 즉, 번역 시스템의 입력으로 되는 문장의 유형(일반문과 도표의 혼용 관계 등) 또는 번역의 입력과 출력을 대상으로 하는 전편집과 후편집 개념을 고려한 범용적인 인터페이스의 설계는 번역과정의 실제적 구현에 효과적인 지원 수단으로서 중요한 의의를 갖는다.

실제, 지원 환경으로서 입출력 인터페이스의 구축은 번역 시스템의 연구 자체와는 독립적인 작업을 요구하지만, 번역 시스템 실용화의 선결 과제로서 중요한 의의를 갖으며, 구축 방식의 용이함에 비해서는 상당한 작업량이 요구되는 분야이다.

본 연구에서 설계한 인터페이스는 일·한 번역과정을 지원하기 위하여 영어, 일어, 한자, 한글 입력과 문장의 도표화를 위한 기능을 부가하여 즉, PC상에 워드프로세서를 설계하고 이를 번역 시스템에 연계시켜, 번역과정의 입출력 과정과 전, 후편집의 수단으로 사용하며, 시스템 유지, 보완과 확장성을 고려하여 사전과 문법 기술의 보완과 수정을 위한 interactive (대화적) 한 사용자 인터페이스의 역할을 수행

한다.

2. 사전 구성

기계번역 시스템의 구현에 있어 사전의 구성은 시스템의 성능에 가장 밀접한 관계 요인이며, 효율적인 사전 구성 그 자체는 번역과정 전반에 성패 여부를 좌우한다고 해도 과언이 아니다. 그러므로, 실용화를 목표로하는 번역 시스템의 설계에서 효율적인 사전의 구성은 최우선의 과제로 간주될 수 있다.

실제, 사전 기술의 일관성에 주의하면 연구용이 아닌 실용 시스템을 목적으로 번역과정을 구현하는 경우에 요구되는 방대한 사전 기술에서 일관된 기술 기준의 설정 문제는 많은 난점을 수반한다. 즉, 단위 단어의 결정, 다품사어에 있어서 사전 기술의 지표 항목으로 되는 품사의 설정 문제, 그리고 의미 기술과 격 관계, 또한 각각의 단어에 대한 의미표지의 부여는 일관된 기준을 설정하기가 곤란한 문제이다.[8,9]

현시점에서 사전 기술 전반을 객관적인 언어 이론적 배경하에서 그 기준을 부여 한다는 것은 불가능하며, 단어 개개의 특수한 용례를 고려해야할 경우가 다수 존재 한다. 결국, 번역 시스템의 실용화에는 언어 이론적 기술과 실제적인 시스템의 처리과정과의 사이에 존재하는 trade-off 관계에 유의한 기술적인 절충을 도모하여야 한다.

본 연구에서 작성한 사전 시스템은 일본어의 해석관계 사전과 일본어와 한국어간의 대응 변환 사전, 그리고 한국어 형태소 생성 사전으로 구성된다. 한편, 번역 과정을 처리 시스템의 견지에서 볼 때 사전 구성은 선언적인 기술만으로는 충분한 효율을 얻을 수 없기 때문에 사전 기술에 프로시쥬어를 데이터로 도입하여 처리 효율을 증대시켰다.

그림 2에 각 단어(표제어)에 할당된 해석 사전의 기본적인 구성례를 보인다. 단, 자립어와 부속어 사전을 혼용해서 보인 것이다.

사전은 자립어 사전과 부속어 사전으로 개략적 구분을 할 수 있으며, 부가적으로 일반적인 관용구와 통계적 작업에 의해 추출된 문의 종결구(주로 동사, 조동사의 복합 관계)를 관용구에 포함하여 처리한다. 그림 3은 고빈도 종결구의 일례이다.

해석 사전은 형태소 해석 테이블(접속정보 테이블을 참조하는 접속정보와 다품사어의 가중치, 그리고 활용어 테이블의 배열 참조치와 さ변 명사등 단어의 특수 용례를 처리하기 위한 프로시쥬어 명과 그 인수값이 기술되어 있다. 한편, 단일의 사전 항목을 갖더라도 접속정보가 다르거나, 전후의 문절에 대한 접속 여부에 따라 그 생성 의미가 특수한 용례를 갖

```
struct DICT    {                  /* ── dictionary information ──    */
        char     word[21];        /* word                           */
        char     lexicat;         /* lexical category               */
        int      back[5];         /* backward connection            */
        int      forw;            /* forward connection             */
        int      weight;          /* homonym weight                 */
        int      gen[5];          /* generation entry               */
        int      act;
        /* action table ref.information or sahen procedure */
        int      actback;         /* action table array ref.        */
        int      actforw;         /* default f_connect.informat.    */
};
```

```
==================================================================
強 a 88 2 1 1 15          關係 n 1 1 1 1 99 11 2    |   ar io po
起動 n 1 1 1 1 99 11 2    あわせ v 99 2 1 3 8       |
存在 n 1 1 1 1 99 11 2    高速 n 1 1 1 1            |   om dm ce
い v 99 4 1 3 9           な x 66 6 1 1 30          |
な a 99 2 1 1 15          境遇 n 1 1 1 1     ===>   |   po do to
もしくは c 5 2 1 2        その n 1 1 1 1            |
毎回 n 1 1 1 1            れ x 66 8 1 1 20          |   cp it ic
年 u 1 5 1 1              出版 n 1 1 1 1 99 11 2    |
本 n 1 1 1 1              はじま v 99 2 1 2 8       |
はじまり n 1 1 1 1        すべて b 3 2 1 2          |
==================================================================
に h 24 28 1 5            は q 0 35 1 1
と h 25 32 1 5            と h 26 32 1 5
と h 24 33 1 5            と p 19 24 1 3
の h 0 31 1 1             を h 23 27 1 2
て p 20 22 1 4            ======================================
ポインタ n 1 1 1 1        シソーラス n 1 1 1 1
システム n 1 1 1 1        ディスク n 1 1 1 1
==================================================================
config.sys n 1 1 1 1      MS-DOS n 1 1 1 1

關係〔かんけい〕==>, oh/of do, io/io, が に
あわせ〔合わせ〕==>, of oh//as ce/, of oh/io co/io co/, が を に
                    :of oh//of oh, が に が
```

그림 2. 일본어 해석사전의 구성과 그 일례
〔사전 상단의 점선 우단은 명사의 의미 표지이며, 하단은 의미 표지에 관계된 동사의 격 프레임의 일례이다.〕

Fig. 2. An example of Japanese dictionary.

52 行ないます	53 つくりなさい
54 代入します	54 になります
55 説明します	56 示しています
58 なつています	59 思います
59 必要です	60 行われます
61 例である	63 となります
63 なっています	64 用います
64 指定します	65 とおりである
69 參照してください	74 表示します
83 可能です	83 行います
89.實行します	107 入力します
114 あります	114 します
120 なければならない	123 なります
153 なっなている	195 示します

/* 慣用 終結局の 抽出例 但, ある, いる等は 除外 */

그림 3. 컴퓨터 매뉴얼(약 32,000문)을 대상으로 추출한 고빈도 종결구의 일례

Fig. 3. An example of high frequency phrase.

을 때는 사전 항목에 복수로 단어를 입력하여 접속 정보에 따라 그 항목의 선택 여부를 결정할 수 있도록 구성한다. 동사의 격할당은 표층격을 중심으로 기술하며, 심층격의 할당은 복수의 생성정보를 갖는 단어로 제약 한다. 또한 명사의 의미표지 할당은 각 단어당 3개 이내로 제약하여 기술하며, 그 기준은 빈도수에 의하여 선택하고, 의미표지의 기술 체계는 ICOT 분류를 근간으로 하였다.[8,12]

동음이의어의 경우에는 사전상에 빈도수를 기준으로 하는 가중치를 기술하여 형태소 해석시 우선 적용되도록 하며, 부가적인 다의어의 애매성 해소 프로시쥬어를 최우선의 가중치를 갖는 단어의 사전 항목에 기술하여, 구조 해석시에 이를 결정하도록 한다. 단, 해소 프로시쥬어의 결과가 0으로 리턴되면 최우선의 가중치를 갖는 단어를 일의적(一意的)으로 선택한다.

일본어와 같은 첨가어의 해석을 수행하는 경우에 자립어에 부가되는 조사의 해석이 매우 중요한 의미를 갖는다. 해석과정에서 조사가 갖는 다의성과 자립어와의 대립에 의한 애매성등은 구문 의미 해석의 결과를 좌우하는 커다란 요인으로 작용한다. 본 시스템에서 구성한 일본어 조사 사전의 구성은 격조사(15개), 부조사(38개), 계조사(12개), 접속조사(45개), 종조사(25개), 그리고 격조사 상당 표현(8개)으로 구성된다. 사전 구성에서 고어, 회화체, 또는 속어적 표현에 이용되는 조사는 제외하였으며, 부가적으로 조사 파생의 언어를 사전상에 기술한다. 그림 4(a)는 격조사 상당 표현이며, (b)는 연어로서 사용되는 관용구 표현의 일례이다.

を用いて	0	をよいて	0
たあ	0	によつて	0
に對して	0	にたいして	0
とともに	0	について	0

(a) addtional case

かどうか	といわすば	かも知れません
とくると	かもしれません	ときたら
からとて	ときては	からに
との	からには	ともなく
からは	ともなしに	てから
にしるろ	といった	にせよ
といって	といっても	につき
といったらない	につけ	というものの
とはいうものの	ものかは	をして

(b) compound expression

그림 4. 격조사 상당표현과 조사파생 관용구의 일례
Fig. 4. An example of addtional case particle and idiomatic phrase.

3. 일본어문의 해석

1. 형태소 해석

형태소 해석은 선형적인 입력 문자열을 대상으로 사전을 참조하면서, 각 단어와 문절을 분할하여 형태소정보를 부여한 단어와 문절의 출력을 얻는 과정이다. 형태소 해석과정은 기계 번역 과정의 첫 단계로 표층의 언어 현상을 직접 처리하기 때문에 일본어와 같이 띄어쓰기를 하지 않는 언어에서는 전처리 단계로서 매우 중요한 의미를 갖는다.[3,4,8,11]

1) 접속 테이블

일본어의 형태소 해석은 자립어, 부속어, 관용어 등을 포함한 해석 사전과 접속 테이블 및 활용어의 활용정보를 기술한 활용 테이블을 이용하여 행한다.[10,12]

본 시스템에서 작성한 접속 테이블은 42*39 배열로 구성된 매트릭스로 연속하는 단어 간의 접속 여부를 판단할 수 있는 정보가 기술되어 있다. 접속정보의 구성은

가) null (0) → 접속 불가.

나) 1 → 접속 가능이나 문절을 분리하지 않음.

다) 2 → 접속 가능하며 문절을 분리한다.

라는 3개의 정보로 구성된다. 접속 테이블의 행(row) 정보는 해석 사전에 기술된 후접속 정보에 대응하며 해석 사전의 후접속 정보 결정 프로시쥬어와 활용 테이블에 의해서도 대응된다. 그리고, 열(column) 정보는 해석 사전의 후접속 정보에 대응 되어 있다. 그림 5(a),(b)에 접속 테이블의 일례와 그 할당 정보를 보인다.

2) 활용 테이블

활용정보 테이블은 동사, 형용사, 형용동사 등의 자립 활용어와 부속어인 조동사의 활용정보가 기술된다. 각 활용어의 활용정보는 종지, 연체, 미연, 연용, 가정, 명령의 순으로 열 지표에 기술되어 있으며, 행 정보는 각 활용어의 활용형이나 단어 자체의 고유한 활용형을 의미한다.

해석 사전에 각 활용어의 후접속 정보는 결정되어 있지 않고 프로시쥬어 〔ex : 동사→99, 조동사→66〕로 기술되어 있기 때문에 사전 할당 프로시쥬어는 접속정보를 검색하기 전에 활용 테이블을 참조하여 후접속정보를 결정한다. 이때, 부가적으로 할당 활용형의 열 지표에 해당하는 정보를 형태소 정보에 추가하여, 생성시 활용 어미 테이블의 지표 정보로 사용할 수 있도록 한다. 그림6, 7은 활용단어의 어미 검색 테이블과 이에 연계된 접속정보 생성 테이블이다.

2. 일본어 형태소 해석 시스템의 구성

일본어 입력문은 단어와 단어 사이에 분리 기호가

```
char    connect〔42〕 〔39〕= {                  /* word connectivity data */
 /*          2          1111111111222…33333333 */
 /* (dymmy0)1234567890123456789012… 12345678 */
          dy
 /*  0 */ 0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,…,0,0,0,0,0,0,0,0,
 /* 1  */ 0,1,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,…,2,2,2,2,2,2,0,1,
 /*  2 */ 0,2,2,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,…,2,2,2,2,2,2,0,0,
 /*  3 */ 0,2,2,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,…,2,0,2,2,2,2,0,0,
                 :                         :
 /*  7 */ 0,2,2,0,0,0,0,0,0,0,2,2,2,0,0,0,0,0,0,2,…,0,0,0,0,2,0,0,0,
 /*  8 */ 0,2,2,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,0,0,…,0,0,0,0,2,0,0,0,
 /*  9 */ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,2,2,2,2,…,0,0,2,0,0,0,0,0,
 /* 10 */ 0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,…,0,0,0,2,0,2,0,0,
 /* 11 */ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,…,0,0,2,0,0,0,0,0,
                 :                         :
                 :                         :
}
```

(a)

〔row ; backward connection info.〕
1 : noun
2 : adverval noun
3 : adverb
　: 　:
7 : 오단[さ행], 상일단, 하일단, か변, さ변 연용형
8 : 오단동사[さ행 이외] 연용형
9 : 동사 종지형
10 : 동사, 형용사 연체형
11 : 동사 명령형
　: 　:

〔column ; forward connection info.〕
1 : noun
2 : 명사이외의 자립어, 접두사
3 : 형식명사
　: 　:
7 : 조동사 ぬ
8 : 조동사 れる, せる
9 : 조동사 られる, させる
10 : 조동사 しめる
11 : 조동사 う
　: 　:

(b)

그림 5. (a) 접속 테이블의 일례
(b) 접속 테이블의 행(row)과 열(column) 에 할당된 형태소 범주

Fig. 5. (a) An example of connection table.
(b) Morphological category for connection table.

```
char     act_1〔36〕〔12〕〔5〕 = {                 /*        conjugation array  string       */
/*          1     2     3     4       5     6     7       8     9      10     11     12        */
/*  0 */  "う",  " ",  "う",  "わ",    "お",  " ",  "つ",    "い",  " ",    "え",   "え",   " ",
/*  1 */  "く",  " ",  "ぬ",  "か",    "こ",  " ",  "い",    "き",  " ",    "け",   "け",   " ",
/*  2 */  "ぐ",  " ",  "ぐ",  "が",    "ご",  " ",  "い",    "ぎ",  " ",    "げ",   "げ",   " ",
/*  3 */  "す",  " ",  "す",  "さ",    "そ",  " ",  "し",    " ",   " ",    "せ",   "せ",   " ",
                 :                       :                          :
/*  8 */  "る",  " ",  "る",  "ら",    "ろ",  " ",  "つ",    "り",  " ",    "れ",   "れ",   " ",
/*  9 */  "る",  " ",  "る",  "*",     " ",   " ",  "*",     " ",   " ",    "れ",   "ろ",   "よ",
/* 10 */  "る",  " ",  "る",  "*",     " ",   " ",  "*",     " ",   " ",    "れ",   "ろ",   "よ",

/* 15 */  "い",  " ",  "い",  "かろ",  " ",   " ",  "かつ",  "く",  "う",   "けれ", " ",    " ",
/* 16 */  "だ",  " ",  "な",  "だろ",  " ",   " ",  "だつ",  "で",  "に",   "なら", " ",    " ",
                 :                       :                          :
} :
```

그림 6. 활용 테이블의 일례
Fig. 6. An example of conjugation table.

```
int      act_2 [36] [12] = {            /* conjugation array backward               */
     /*                                                          1    1    1      */
     /*        1    2    3    4    5    6    7    8    9    0    1    2      */

     /*  0 */ 9 ,  0 ,  10,  28,  29,  0 ,  30,  8 ,  0 ,  32,  11,  0 ,
     /*  1 */ 9 ,  0 ,  10,  28,  29,  0 ,  30,  8 ,  0 ,  32,  11,  0 ,
     /*  2 */ 9 ,  0 ,  10,  28,  29,  0 ,  31,  8 ,  0 ,  32,  11,  0 ,
     /*  3 */ 9 ,  0 ,  10,  28,  29,  0 ,  8 ,  0 ,  0 ,  32,  11,  0 ,
                        :                        :
     /*  8 */ 9 ,  0 ,  10,  28,  29,  0 ,  30,  8 ,  0 ,  32,  11,  0 ,
     /*  9 */ 9 ,  0 ,  10,  33,  0 ,  0 ,  7 ,  0 ,  0 ,  32,  11,  11,
     /* 10 */ 9 ,  0 ,  10,  12,  0 ,  0 ,  12,  0 ,  0 ,  32,  11,  11,
     /* 11 */ 9 ,  13,  10,  35,  36,  0 ,  7 ,  0 ,  0 ,  32,  11,  11,
                        :                        :
     /* 15 */ 15,  0 ,  10,  37,  0 ,  0 ,  38,  14,  38,  32,  0 ,  0 ,
     /* 16 */ 17,  0 ,  18,  37,  0 ,  0 ,  38,  16,  38,  32,  0 ,  0 ,
                        :                        :
                        :                        :
} ;
```

그림 7. 활용 테이블에 연계된 접속 정보 결정 테이블
Fig. 7. An example of connection information table related to conjugation table.

개입되지 않은 연속된 문자열로 구성되어 있기 때문에 형태소 해석에 상당한 난점이 있다. 자동 띄어쓰기 과정은 Lingol에서와 같이 구문 해석과 병렬로 처리하는 방법도 고려되고 있지만, 본 시스템에서는 형태소 해석과 구문해석을 분리하여 실현한다.[6]

본 연구에서 개발한 시스템은 기본적으로 문절수최소법을 근간으로 형태소 해석을 진행한다.[8,10] 입력문의 사전 참조 과정에서 출력된 모든 접속 가능한 문자열은 최장일치에 의해 구해진 최초 단어를 기점(root node)으로 하여, 하위 절점에는 이후의 해석 결과가 부가되는 연계 리스트형으로 구성된다. 그림8은 형태소 해석 과정의 개념적인 흐름도이다.

한편, 입력 문자열에 대하여 접속정보를 검색하여 구성한 연계 리스트는 다품사어등의 존재로 인하여 하나의 상위(father) 노드에 복수의 프로세스 결과가 하위(son) 노드에 연결되어 있는 경우, 복수개의 가능한 출력이 나타날 수 있다. 실제 이러한 경우는 문절수최소법을 적용하여도 적격한 해를 구할 수 없다. 그러므로, 이러한 경우 형태소 해석과정에서는 그 처리를 보류하고, 복수의 결과를 격구조 추출 과정으로 전달한다.[2,4,7]

격구조 추출 과정은 동사의 격을 중심으로 형태소 해석 결과 얻어진 리스트를 재구성한다. 이때, 동사등의 다품사어 결정이 이루어지게 되며, 결정되지않은 복수의 프로세스는 사전상에 기술된 해당 단어의 동음이의어(homonym)에 대한 가중치를 참조하여 일의적(一意的)으로 결정한다. 그림9는 부정의 의미를 갖는 "ない"가 조동사와 형용사 두개의 문법 범주를 갖는 경우의 출력 예이다. 그러나 "ない"와 같이 문법적인 접속정보의 할당은 상이하더라도, 실제적인 의미의 기술이 (즉, 생성 과정의 견지에서) 동일한 경우는 임의로 프로세스를 결정한다.

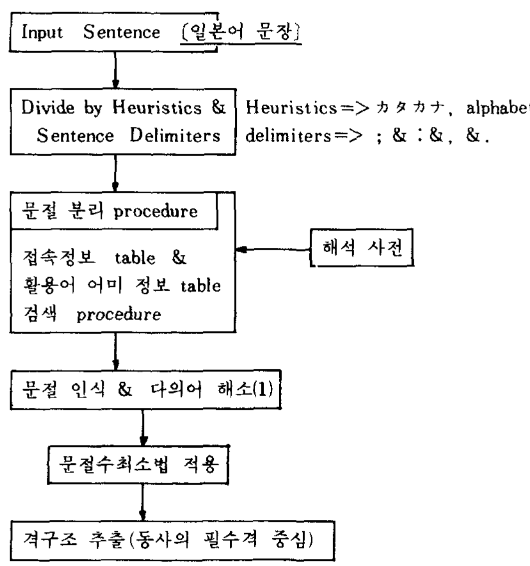


그림 8. 일본어 해석 과정의 개요
Fig. 8. Outline process of japanese analysis.

```
input → config . sys ファイルが存在していない 場合,……

〈config . sys   n (0 1)  ( 1) - ( 1  0  0 0 0) - (0 0 0 0 0) - (0 0 0 0 0)  (1 1)
〈ファイル       n (0 0)  ( 1) - ( 1  0  0 0 0) - (0 0 0 0 0) - (0 0 0 0 0)  (1 1)
〈が             h (0 0)  (27) - ( 0  0  0 0 0) - (0 0 0 0 0) - (0 0 0 0 0)  (1 4)
〈存在し         y (0 0)  ( 2) - (99 37  7 0 0) - (0 6 7 0 0) - (0 0 0 0 0)  (1 1)
〈て             p (0 0)  (22) - (20  0  0 0 0) - (0 0 0 0 0) - (0 0 1 0 0)  (1 4)
〈い             v (0 0)  ( 4) - (99 34  7 0 0) - (0 4 7 0 0) - (0 0 0 0 0)  (1 3)
〈ない           x (0 0)  ( 6) - (66  9 10 0 0) - (0 1 3 0 0) - (0 1 0 0 0)  (1 1)
〈場合           n (2 0)  ( 1) - ( 1  0  0 0 0) - (0 0 0 0 0) - (0 0 1 0 0)  (1 1)
〈,              n (1 0)  ( 1) - ( 1  0  0 0 0) - (0 0 0 0 0) - (0 0 0 0 0)  (0 0)

〈config . sys   n (0 1)  ( 1) - ( 1  0  0 0 0) - (0 0 0 0 0) - (0 0 0 0 0)  (1 1)
〈ファイル       n (0 0)  ( 1) - ( 1  0  0 0 0) - (0 0 0 0 0) - (0 0 0 0 0)  (1 1)
〈が             h (0 0)  (27) - ( 0  0  0 0 0) - (0 0 0 0 0) - (0 0 0 0 0)  (1 4)
〈存在し         y (0 0)  ( 2) - (99 37  7 0 0) - (0 6 7 0 0) - (0 0 0 0 0)  (1 1)
〈て             p (0 0)  (22) - (20  0  0 0 0) - (0 0 0 0 0) - (0 0 1 0 0)  (1 4)
〈い             v (0 0)  ( 4) - (99 34  7 0 0) - (0 4 7 0 0) - (0 0 0 0 0)  (1 3)
〈ない           a (0 0)  ( 2) - (99 16 10 0 0) - (0 1 3 0 0) - (0 0 1 0 0)  (1 1)
〈場合           n (2 0)  ( 1) - ( 1  0  0 0 0) - (0 0 0 0 0) - (0 0 1 0 0)  (1 1)
〈,              n (1 0)  ( 1) - ( 1  0  0 0 0) - (0 0 0 0 0) - (0 0 0 0 0)  (0 0)
```

그림 9. 형태소 해석 결과의 일례

Fig. 9. An example of morphological analysis.

3. 형태소 해석과정의 정비

접속 정보 테이블과 활용 테이블에 의하여 적격한 접속으로 인정된 형태소 해석 결과는 일반적으로 복수개의 출력을 갖는다. 접속 과정의 검색 즉, 자동 띄어쓰기 과정의 출력은 근사적 해석 방법인 문절수 최소법에 의해서 일의적(一意的)으로 결정되기 때문에 적격한 출력은 보장되지 않는다. 그러므로 가능한 모든 구문정보를 이용하여 형태소 해석과정을 정비하여 최선의 해석 결과를 얻어야 한다. 또한 문절수 최소법이 자립어를 기준으로 하는 문절을 단위로 하기 때문에 자동 띄어쓰기 과정에서 출력된 복수개의 출력의 해석 여부에 따라 문절수에 직접적인 영향을 받는다. 그러므로, 문절수최소법의 전처리 과정으로서 어떠한 형태로든 형태소 해석 결과를 정비하는 프로세스를 기술할 필요가 있다. 실제 애매성의 많은 부분이 의미적인 요인에 의해 발생한다고 보여지지만 형태적, 구문적인 정보에 의해 많은 부분이 해소될 수 있다.

본 시스템에서 구성한 전처리 과정은 조사의 형태적 애매성 해소에 중점을 두어 프로세스를 구성하며, 다음과 같은 프로세스 모듈로 구성된다.

(1) 접속 과정의 출력이 단어 단위로 구성되어 있기 때문에 자립어와 부속어의 조합에 의해 문절을 인식한다. 이때 "に對して, によって"의 조사적 표현과 "における, かもしれません"등의 관용 표현을 문절로 인식하는 확장 문절의 개념을 이용한다.

(2) 조사 사전에 부가된 애매성 해소 프로시쥬어를 실행하여 해석 결과의 애매성을 제거한다. 조사 사전의 부가 프로세스는 그림10과 같은 정보를 갖으며 시스템상의 대역적(global) 함수의 입력으로 사용 된다.

4. 시스템 구현 및 고찰

본 연구에서 구축한 일본어 해석 시스템의 제과정은 IBM PC/AT상에서 Microsoft C(V. 5. 0)을 이용하여 구현하였다. 시스템의 개략적인 규모는 번역 환경 지원 S/W로 개발된 interface가 약 400kbyte, 그리고 사전을 제외한 주 프로그램이 약 100kbyte 정도이며, 사전은 대략 20,000 단어 규모로 1.2Mbyte 정도의 크기로 구성되어 있다.

실제 시스템의 검증 과정에서 사용한 데이타는 문제의 다양성을 고려하여 임의로 선정한 일본어 컴퓨터 매뉴얼 10권에서 임의로 (=1206문장)추출하여 사용하였고, 문절 분리 과정의 효율을 높이기 위해 빈도수 높은 종결구를 관용어 형으로 분류하여 사용하였다.

번역 과정의 구현에 장애 요인으로 되는 다품사어의 처리는 접속정보상에서 가급적 검색할 수 있도록 엄밀한 접속 테이블의 구성을 행하였다. 현재의 시스템은 확장성을 고려하여 가능한한 모듈화의 개념

/* index table for multivocal resolution (work sheet 1) */

a	b	c	d	e				f										
/* lex . cat	cat	fw	bw	act.	array	ref.		special char. (approx.)										
/*		1 2 3 4 5 6 7 8 9 0 1	12	1	2	3	4	f	backward									
/*				000	000	0000	000	1	1		2		3		4		5	
10 から	h	n																
	p	vajgx		01	01	01	01											
		x		03	03	03	03		&	28	&	25						
	n	m (sky)																
	n	m (shell)																
22 か	k	nfdemvajgx		01	01	root	01											
	q	nfdemvajgx		01	01	root	01											
	s	v		08														
24 が	h	nfdemvajgx		03	03	03	03											
	p	nfdem							c	28	p	28	c	25	p	25		
		J				root			c	28	p	28	c	25	p	25		
		vagx		03	03	03	03											
	k	vajgx		01	01	01	01		#	ed								
	c	", ", ", "							#	ft								
	n																	

a : 어휘 항목, b : 품사 정보, c : 전 접속 품사 정보, d : 후 접속 품사 정보, e : 전 접속의 품사가 활용어일 때〔1 : 동사 2 : 형용사 3 : 형용동사 4 : 조동사〕의 각각에 대한 활용 정보, f : 전 접속 단어가 특정의 단어일 때의 정보 기술〔ex : & 28& 25는 から가 접속 조사로 사용될 때 조동사에 후속할 때는 28, 25의 조동사 즉, "だ, です"에 접속하는 것을 의미한다.〕

그림10. 조사의 다의성 해소 프로시쥬어의 입력 정보 테이블
Fig. 10. Input data table of ambiguity resolution for particle.

에서 설계하였으며, 해석 결과의 출력 시간도 상당한 수준에 이르고 있어, 시스템의 효율적인 정비가 이루어진다면 실용화에의 접근도 충분히 가능하다고 사료된다.

그러나 다품사어의 처리 문제와 접속정보 검색시 어미활용 테이블의 불필요한 검색 방지가 요구되며, 사전 구성의 효율성을 제고하는 문제도 연구를 계속해야 할 것이다.

다음은 해석 과정 출력의 일례이며, 출력시간은 HD access 시간까지를 합한 것이다.

example : 1
/* 해석 결과의 정보 : a ; 어휘항목(lexicon), b;어휘범주(lexical category),
c;접속형분류→ 1 ;미정의어, 2 ;분류기호,
d;접속형→ 0 ;단어분리, 1 ;접속가능이나 분리하지 않음,
e;전 접속 정보, f;활용 테이블에서 리턴된 접속 정보의 열(최대 5 개),
g;활용 테이블에서 리턴된 활용형 정보,
h;homonym weight, i;generation entry No. */

a	b	c	d	e	f	g	h	i
〈thesaurus	n	(0	0)	(1)	-(1	0 0 0 0) -(0 0 0 0 0)	(1	1)
〈は	q	(2	0)	(35)	-(0	0 0 0 0) -(0 0 0 0 0)	(1	1)
〈roget	n	(1	0)	(1)	-(1	0 0 0 0) -(0 0 0 0 0)	(0	0)
〈が	h	(2	0)	(27)	-(0	0 0 0 0) -(0 0 0 0 0)	(1	4)
〈1852	m	(1	0)	(2)	-(6	0 0 0 0) -(0 0 0 0 0)	(0	0)
〈年	u	(0	0)	(5)	-(6	0 0 0 0) -(0 0 0 0 0)	(1	1)
〈に	h	(0	0)	(28)	-(24	0 0 0 0) -(0 0 0 0 0)	(1	5)

a	b	c	d	e	f	g	h	i
〈出版し	y	(0	0)	(2)	-(99 0 7 0 0)	-(0 0 7 0 0)	(1	1)
〈た	x	(0	0)	(15)	-(66 0 10 0 0)	-(0 0 3 0 0)	(1	2)
〈本	n	(0	0)	(1)	-(1 0 0 0 0)	-(0 0 0 0 0)	(1	1)
〈が	h	(0	0)	(27)	-(0 0 0 0 0)	-(0 0 0 0 0)	(1	4)
〈はじまり	n	(0	0)	(2)	-(88 0 8 0 0)	-(0 0 8 0 0)	(1	2)
〈であり	x	(2	0)	(17)	-(66 8 0 0 0)	-(0 8 0 0 0)	(1	2)
〈,	n	(1	0)	(1)	-(1 0 0 0 0)	-(0 0 0 0 0)	(0	0)
〈すべて	b	(0	0)	(2)	-(3 0 0 0 0)	-(0 0 0 0 0)	(1	2)
〈の	h	(0	0)	(31)	-(0 0 0 0 0)	-(0 0 0 0 0)	(1	1)
〈単語	n	(0	1)	(1)	-(1 0 0 0 0)	-(0 0 0 0 0)	(1	1)
〈を	h	(0	0)	(27)	-(23 0 0 0 0)	-(0 0 0 0 0)	(1	2)
〈體係	n	(0	1)	(1)	-(1 0 0 0 0)	-(0 0 0 0 0)	(1	1)
〈的	s	(0	0)	(38)	-(1 0 0 0 0)	-(0 0 0 0 0)	(1	1)
〈に	h	(0	0)	(28)	-(24 0 0 0 0)	-(0 0 0 0 0)	(1	5)
〈分類し	y	(0	0)	(2)	-(99 37 0 0 0)	-(0 5 0 0 0)	(1	1)
〈よう	x	(0	0)	(12)	-(22 0 0 0 0)	-(0 0 0 0 0)	(1	1)
〈と	h	(0	0)	(33)	-(25 0 0 0 0)	-(0 0 0 0 0)	(1	5)
〈する	v	(0	0)	(2)	-(10 0 0 0 0)	-(0 0 0 0 0)	(1	2)
〈もの	n	(0	0)	(1)	-(1 0 0 0 0)	-(0 0 0 0 0)	(1	1)
〈である	x	(2	0)	(17)	-(66 0 10 0 0)	-(0 0 3 0 0)	(1	2)
〈.	n	(1	0)	(1)	-(1 0 0 0 0)	-(0 0 0 0 0)	(0	0)

The total time is 00 : 09 : 66
thesaurusは rogetが 1852年 に 出版 した本がはじまりであり, (00 : 04 : 95)
すべての単語を體係的に分類しようとするものである. (00 : 04 : 71)
ex : 2
MS-DOSを起動したシステムデイスクに, (00 : 01 : 87)
config. sysつァイルが在存していない場合, (00 : 03 : 73)
もしくは存在してもその内容にBUFFERS＝xxがない場合は, (00 : 06 : 21)
DIRのキーインのたびにデイスクがアクセスされるはずです. (00 : 03 : 82)
The total time is 00 : 15 : 63

V. 결　　론

본 논문에서는 실용화를 목표로 하는 일·한 번역 시스템의 구현을 위한 일본어 입력문의 형태소 해석 시스템과 사전 구성에 대하여 기술하였다.

접속 테이블의 구성은 일본어 입력문의 유형과 문체를 고려하여 구성하였으며, 사전 기술도 통계적인 방법을 도입하여 문법 범주의 할당을 시스템에 효율적인 면을 고려하여 기술하였다. 또한 접속 테이블 구성시 활용어의 접속 정보를 효과적으로 제어하기 위하여, 어미 활용정보와 접속 정보를 갖는 2차원 활용 테이블을 구성하고, 형태소 해석과정에서 접속 테이블과 상호 연계시켜 해석을 수행하였다. 그 결과 접속 테이블이 간략하게 구성되어, 시스템의 검증 과정과 개선 과정의 작업을 한층 용이하게 할 수 있었다. 또한 형태소 해석과정에서 얻어진 활용어 테이블의 출력정보는 구문/의미 해석 과정에서 동사의 필수격을 중심으로한 격 리스트를 구성할 때, 복수개의 형태소 출력 해소 과정의 보조 정보로서 이용되며, 동시에 한국어 생성 과정으로 변환할 때에 부가정보를 제공한다.

한편, 본 논문에서 구현한 시스템은 출력의 품질과 시간이 어느 정도의 수준에 도달하였다고 볼 수 있으므로, 사전 구성의 효율성 제고와 다의어의 애매성 해소를 위한 부가 프로시쥬어의 연구를 진행한다면 상용 시스템의 구축에도 응용될 수 있다고 생각된다.

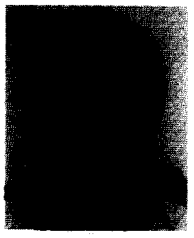
본 시스템의 후반 부분인 구문 해석과 생성 과정의 구성은 문헌[13]에 기술한다.

參 考 文 獻

[1] Ullman, A.: "The Theory of Parsing, Translation and Compiling," Pretice Hall, 1975.

[2] Fillmore, C.: "The Case for Case," in Bach& Harms (eds.), Universals in Linguistic Theorys Holt Renehart&Winston, New York, 1968.
[3] 村木一至："知識Baseと，言語に獨立の中間表現とを用いた日英機械翻譯システム," Nikkei Electronics, Dec. 17, 1984, pp. 195-220.
[4] 内田 裕士："言語に依存しない概念構造を中間表現の基本とし，常識を使う多言語向き機械翻譯システム," Nikkei Electronics, Dec. 17, 1984, pp. 221-240.
[5] 内田，增山："機械翻譯における機念變換について," 情報處理學會者然言語處理技術シンホジウム資料, Jun. 1983.
[6] 田中："自然言語處理のためのプログラミングシステム－擴張LINGOLについて－," 電子通信學會論文誌D, vol. 60-D, no. 12, 1977.
[7] 長尾眞："計算機による日本語文章の解析に關する研究," 文部省科學研究費特定研究研究報告書, 1978.
[8] 長尾眞："國語辭書の記憶と日本語の自動分割," 情報處理, vol. 19, no. 6, 1978.
[9] 長尾眞："言語の機械處理," 三省堂, 1984.
[10] 吉村，武内，聿田，首藤："コスト最小法を用いた日本語文の形態素解析," 自然言語處理研究會報告, no. 60, 1987.
[11] 坂本："日本語形態素解析の基本設計," 自然言語處理研究會, no. 38, 1983.
[12] 長尾眞(eds.)："機械翻譯システムの調査研究," 日本電子工業振興協會, 1984.
[13] 김영섭, 김한우, 최병욱：PC를 이용한 일·한 번역 시스템 ATOM의 개발에 관한 연구 (Ⅱ), 대한 전자공학회 논문지, vol. 25, no. 10, 1988. *

著 者 紹 介



金 榮 暹(正會員)
1959年 10月 15日生. 1983年 2月 한양대 전자통신공학과 졸업. 1985年 2月 한양대학교 전자통신공학과 대학원 졸업. 1985年 3月～현재 한양대학교 전자통신공학과 박사과정. 주관심분야는 Natural Language Understanding, Knowledge Engineering, Computational Linguistics.

●

金 漢 宇(正會員)
1952年 8月 13日生. 1978年 2月 한양대학교 대학원 전자공학과 졸업. 1978年 9月 한양대학교 대학원 박사과정. 1980年 3月～1981年 9月 일본 Kyoto대 체재. 1981年 9月～현재 한양대학교 전산과 재직. 주관심분야는 Machine Translation, Text Understanding, 계산언어학등임.

崔 炳 旭(正會員)
1949年 10月 2日生. 1973年 2月 한양대학교 전자공학과 졸업. 1981年 3月 일본 KEIO 대학원 공학박사학위 취득. 1986年 8月～1987 8月 University of Maryland 교환교수. 1981年 9月～현재 한양대학교 전자통신공학과 교수. 주관심분야는 Computer Vision, Natural Language Processing 등임.