論文95-32B-1-15

일한 번역시스템을 위한 일본어 해석기 설계

(A Design of Japanese Analyzer for Japanese to Korean Translation System)

姜 哲 壎 *, 崔 炳 旭 *

(Seok Hoon Kang and Byung Uk Choi)

요 약

본 논문에서는 일한 기계 번역 시스템을 위한 일본어 형태소 해석 시스템을 설계한다. 제안하는 형태소 해석기는 입력된 일본어 문장을 문법적/사전적 정보를 포함하는 어절의 형태로 재구성한다. 이를 위해 형태소를 분리하고 그것을 다시 한국어의 어절에 대응하는 형태로 결합시키는 앨고리듬을 제안한다. 일본어 어절을 제어하기 위해 연결자를 정의하였으며, 이 연결자는 공백 없는 일본어 문장에서 어절의 시작과 끝을 조절하는 역할을 한다. 제안하는 일본어 해석기는 해석 사전과의 연계로 기존의 해석 방법보다 효율적인 해석을 수행하며 사전의 검색 시도 횟수도 감소하였다. 본 논문에서 설계한 일한 기계 번역 시스템을 위한 일본어 해석기는 기존의 방식과는 달리 문장 전체의 구조 생성에 중점을 두지 않고 한국어의 어절 단위로 해석을 수행하여 보다 정확한 한국어 표현을 생성할 수 있는 근거를 제시한다.

Abstract

In this paper, a Japanese morphological analyzer for Japanese to Korean Machine Translation System is designed. The analyzer reconstructs the Japanese input sentence into word phrases that include grammatical and dictionary informations. Thus we propose the algorithm to separate morphemes and then connect them by reference to a corresponding Korean word phrases. And we define the connector to control Japanese word phrases. It is used in controlling the start and the end point of the word phrase in the Japanese sentence which is without a space. The proposed analyzer uses the analysis dictionary to perform more efficient analysis than the existing analyzer. And we can decrease the number of its dictionary searches. Since the analyzer, proposed in this paper, for Japanese to Korean Machine Translation System processes each word phrase in consideration of the corresponding Korean word phrase, it can generate more accurate Korean expressions than the existing one which places great importance on the generation of the entire sentence structure.

*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 : 1994年 2月 3日

I. 서 론

기계번역은 1970 ~ 1980년대를 중심으로 문장단위

의 번역이 진행되었으며, 많은 번역 시스팀이 발표되었다[1][2]. 1990년대에 들어서는 조응구조와 생략 등 문장내 혹은 문장간의 제 언어 현상에 관한 문제를 해결하고자 하는 노력이 계속되고 있다[13][14]. 한국어와 관련되는 기계번역 시스팀은 주로 영어와 일본어를 중심으로 상호 번역하는 것이 주류였으며 이는 현재도 활발히 진행되고 있다[19]. 특히 한국어와 일본어 상호간의 번역에 관한 연구는 양국의 지리적, 문화적 특징으로 많은 관심의 대상이 되어왔다. 특히 일본어는 문장내에 공백을 가지지 않아서 일한 번역과 같이 입력문으로 일본어를 가지는 기계번역 시스팀에서는 항상 일본어 형태소 해석 시스팀이 난점으로 지적되어왔다. 그러나 이때까지 이에 관한 연구는 외국어에 관한 문제였으므로 국외에서 더욱 활발히 진행되어 온 것이 사실이다. 일영 번역 시스팀과 일한 번역 시스팀은 3개국의 언어 특성상 동일한 번역 구조를 가질 수 없으며, 특히 일한 번역의 경우는 양국의 언어가 많은 언어학적 유사성을 가지고 있으므로 일영 번역의 경우처럼 구문적 의미전달에 중점을 두기보다는 각 어절내의 부분번역에 더욱 중점을 두는 것이 정확한 번역결과를 가져 올 수 있을 것이다.

본 논문에서는 일한 기계 번역 시스팀의 전반부에 해당하는 일본어 해석기를 설계한다. 일한 기계번역 시스팀은 크게 일본어 입력문 처리부(입력부)와 번역의 결과가 만들어지는 한국어 생성부로 나누어 설계할 수 있다. 본 논문에서는 이 중 일본어 입력부를 대상으로 한다. 입력부에서는 일본어를 문장단위로 입력받아 전처리와 형태소 해석을 수행한 뒤 연속적으로 구문해석을 진행한다. 형태소 해석 부분에서는 분리된 연속하는 두 개의 형태소 후보로 2항간의 수식관계의 해석을 기본으로 하며, 모든 규칙은 단어 및 가능한 어절의 속성으로 기술한다. 입력문은 일본어의 특징을 살려 공백이 없는 형태를 대상으로 하였으며 전처리부에서 문장을 기본적으로 분리하여 처리의 효율을 도모하였다. 형태소 해석부의 처리가 끝나면 입력문은 문법정보를 포함한 형태로 분리되며, 이는 어절단위로 나타낼 수 있다. 형태소 해석이 끝난 입력문은 생성된 어절을 중심으로 구문해석을 수행하며 이를 위해 차트(Chart)를 이용하였다.

본 논문에서는 일한 번역 시스팀을 최종 목표로 하여 그 전단계로 위의 제 과정으로 구성한 일본어 해석 시스팀을 제안한다. 전체적인 번역 시스팀 중 본 논문에서는 그림 1의 형태소 분리와 형태소 재 결합부에 대해 주로 논한다. 이를 위해 형태소의 분리를 위한 앨고리듬을 제안하며 입력된 문자열을 한국어의 어절에 해당하는 문자열로 재구성하기 위한 앨고리듬도 제안한다. 구문해석은 기존의 차트를 이용한 방법을 동일한 구조로 이용하므로 구체적인 방법론은 본 논문의 대상에서 제외한다.

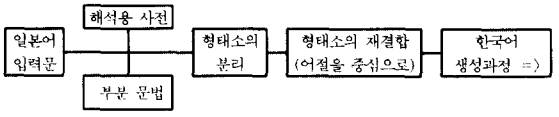


그림 1. 일한 기계 번역 시스팀의 구성도
Fig. 1. A Configuration of the Japanese to Korean machine translation system.

Ⅱ. 일한 번역

계산기에 의한 자연언어 처리의 입장에서 일본어를 고찰한다면 몇 가지의 특징을 정의할 수 있다. 그 첫째가 영어 계통의 언어 혹은 한국어와 달리 일본어에는 중국어와 마찬가지로 문장내에 공백이 존재하지 않는다는 것이다. 띄어쓰기를 하면 일반적으로 문장은 어절로 구분하고 다시 각 어절은 형태소 단위로 세분하는 일반적인 해석방법이 가능해지지만, 띄어쓰기를 하지 않는 일본어는 이런 제 과정이 상당히 난해하다고 볼 수 있다[6][7]. 따라서 일본어의 형태소 해석 과정은 형태소 분리와 이의 재결합으로도 정의할 수 있다[1][3][10]. 이와 같은 사실은 다음의 두 문장에서 알 수 있다.

He goes to the bank. (1)
私は本を讀む. (2)
("watasiha-hongwo-yomu", 나는 책을 읽는다.)

(1)의 예문의 경우 아래와 같이 "goes" 가 "go"라는 동사에 "-es"라는 변화가 발생했다고 분석하는 것을 형태소 해석이라고 정의할 수 있고(이에 따르는 세부적인 표현은 생략한다.), 이에 따라 문의 구조의 해석이 가능해 진다.

goes ⟶ go + es

그러나 (2)의 예문의 경우는 하나의 문장이 공백없이 결합되어 있기 때문에 형태소 해석의 기준은 문장전체를 제외하면 존재하지 않는다고 볼 수 있으며 다음과 같은 형태의 해석 결과를 얻기 위한 띄어쓰기 처리가(

형태소 분리 혹은 형태소 추출이라고 할 수 있으며 그 결과 구분된 각 단어를 문절이라고 한다[1].) 필요하다.

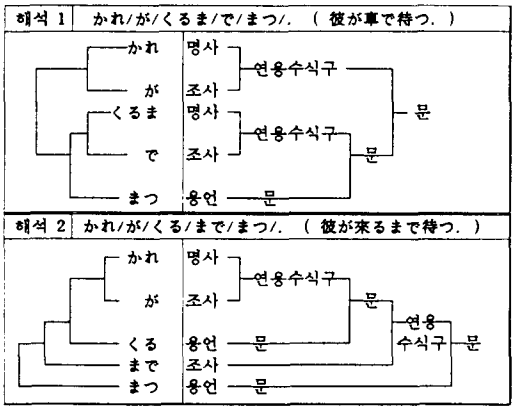
私は 本を 讀む. (3)

이 과정은 일본어의 번역을 위한 처리 시 반드시 필요한 부분이지만 모호성이 빈번히 발생하여 시스템의 처리효율을 저하시키는 가장 큰 원인이 된다. 다음의 예문을[9] 살펴보자.

かれがくるまでまつ. (4)
("karegakurumadematzu". 그가 차에서 기다린다.)

이 문장은 표 1에 나타낸 바와 같이 2가지로 해석할 수 있으며 이것은 모두 타당한 것들이다.

표 1. 예문 (4)의 해석의 일례.
Table. 1. An example of analysis of the sentence (4).



두 번째로 일본어는 교착어로 문절이라는 구문적 단위를 가지고 있고 문절내에서는 단어의 나열에 강한 제약이 가해지지만, 수식하는 말이 수식 받는 말의 앞에 위치한다는 것과 비교차 조건을 제외하면 문절 사이에는 구문상의 제약이 거의 없는 특징이 있다[10]. 이는 한국어 문법과도 유사한 것인데 문절의 수식관계는 어순과도 관련이 있으므로 기계번역이라는 측면에서는 중요한 사실이 아닐 수 없다. 더구나 일본어는 한국어와 어순이 기본적으로 거의 일치하는 것으로 알려져 있기 때문에, 일한 또는 한일 번역의 방식으로 직접번역 방식을 가능하게 하는 원인이 된다. 실제로 위의 예문 (3)은 "나는 책을 읽는다" 고 번역할 수 있을 것이고, 이것은 기본적으로 우리말과 똑 같은 어순을 가진다고 볼 수 있다. 그러나 모든 언어표현이 - 문절내 또는 문절간을 모두 포함하는 - 항상 똑같은 어순을 가질 수 있다고 보기는 어려우며, 따라서 구조적 친밀성에 바탕을 둔 직접번역 방식은 일한 또는 한일번역의 근간은 될 수 있지만 전체 시스템에 적용은 피해야 할 것으로 보인다.

결국 일한 기계번역 시스템은 형태소 분리에서 한국어 생성까지 제 번역 과정의 진행에 따라 발생하는 모호성의 억제가 관건이라 할 수 있으며 시스템은 모호성의 해소에 중점을 두어야 할 것이다. 특히 일본어는 한국어와 마찬가지로 조사와 용언의 활용범위가 다양하기 때문에 이에 대한 처리가 무엇보다 중요하다 하겠다.

본 논문에서는 일본어 문장의 형태소를 해석하고 이를 위한 새로운 앨고리듬을 제안한다. 해석된 결과는 연결자에 의해 크게 어절단위로 구분할 수 있으며 각 어절내는 하나의 자립어를 중심으로 서로 결합된 형태로 정의된다. 기존의 일한 번역을 위한 형태소 해석기는 문법적인 형태소의 분리에만 중점을 두었기 때문에 한국어 생성에 많은 난점이 발생하여 어색한 한국어 생성의 원인이 되었다. 본 논문에서는 연결자의 개념으로 한국어의 어절에 해당하는 부분을 생성하므로 이는 한국어 생성의 일차적인 근원이 될 수 있다.

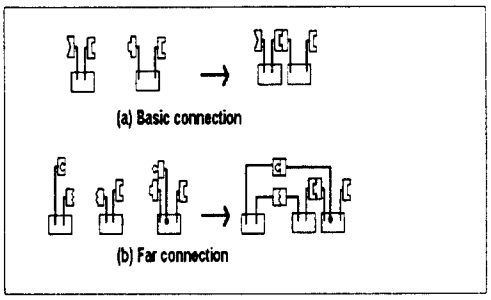
Ⅲ. 형태소 해석

1. 연결자에 의한 형태소 결합

형태소 해석부는 전처리 과정을 제외하면 일한 기계번역의 첫 단계이다. 형태소 해석부에서는 입력 문자열을 인접항들의 수식 관계에[15] 기반하여 접속 관계로 규정하며, 이 접속관계는 부분적인 문법으로 기술하였다. 인접항의 수식 관계에 관한 기본 개념은 그림 2와 같다.(그림2 등의 표현 방식은 Link Parser의 표현 방식이다.) 각 형태소들은 기본적으로 인접항을 위한 연결자들을 가지고 있으며 필요한 경우에 하나의 원거리 연결자를 추가할 수 있는 형태를 가지고 있다. 연결자는 기본적으로 그 개수에 제한이 없지만 하나의 해석에는 하나의 연결만 가능하며, 원거리 연결자는 해석된 중간결과에 의해 하나의 어절의 범위를 표시하는 기능을 가지게 된다.

그림 2의 (a)는 임의의 형태소 두개가 하나의 연결자로 연결이 가능함을 보이는 것이다. 사각형의 상자는 개개의 형태소를 가리키는 것이고, 각 형태소는 기술

된 문법에 의해 다각형으로 표시된 연결자들을 가지고 있다. 형태소별 연결자는 다시 필수 연결자와 선택 연결자로 나누어 정의하였다. 필수 연결자는 형태소의 품사에 의한 기본적인 연결자로 일반적인 결합규칙을 포함한다. 예를 들어 명사의 경우는 조사와의 결합에 일반적인 제약이 없으므로 명사와 조사의 결합에 해당하는 연결자를 모두 가질 수 있다.



(a) 기본연결 (b) 원거리 연결

그림 2. 수식관계의 기본 개념
Fig. 2. Basic conception of the connection.

선택 연결자는 필수 연결자와는 별도로 특정 단어에 의한 문법 규칙을 포함하는 경우로 이는 사전에 의해 별도로 기술한다. 예를 들어, 명사의 경우는 크게 단독명사와 동사형 명사로 구별하였으나[1], 단독명사가 위의 일반 규칙만을 포함한다면, 동사형 명사는 이외에도 동사어미 등 활용형과의 결합을 위한 선택 연결자도 가지게 된다. 그러므로 이 규칙에 의하면 단독명사는 용언의 활용형과 결합할 수 없는 것이다.

그림 2의 (b)는 원거리 연결자를 포함하는 형태소를 나타내는 것이다. 원거리 연산자는 그림 2의 (B)의 세 번째 상자에서 " ° "로 시작하는 연결이다. 일본어 문장이 공백을 가지지 않아서 어절의 구별이 난해하므로 형태소 해석과정에서 이것을 위한 처리가 필요하다. 원거리 연결자는 2개 이상의 형태소가 결합하여 하나의 어절을 생성할 수 있는 경우를 나타내기 위한 것으로, 그림2의 (b)는 3개의 형태소로 하나의 어절을 생성하는 것을 나타낸다. 다시 말해 최종의 노드를 포함하는 형태소를 제외하면, 원거리 연결자로 끝이 나야 어절을 표시할 수 있다. 기존의 해석 시스템은 이러한 어절의 구분없이 해석된 형태소의 의존관계나 구구조의 해석에 의한 것이 일반적인 것이었으나, 대상으로 하는 것이 일한 번역 시스템이고 대상 언어인 한국어는 문장이 어절로 구분되므로 이에 대한 정보를 부여하는 것이 타당할 것으로 보이며, 이에 따라 본 논문에서는 원거리 연결자로 어절의 구성을 표시하였다. 예를 들어 일본어의 경우 "彼には"이라는 어절을 가정한다면 이는 "彼 + に + は" 라는 형태소로 분리할 수 있을 것이다. 이 때는 3 가지의 형태소가 "명사 + 조사 + 조사"로 결합하여 하나의 어절을 이루는 경우로 한국어의 경우와 마찬가지의 해석결과를 얻을 수 있다.[2] 그러나 한국어는 띄어쓰기의 올바름을 전제로 한다면 어절의 단위가 명백하지만, 일본어의 경우는 3 개의 형태소로 분리가 되었다고 하더라도 구문해석의 입력이 되는 어절을 생성할 만한 정보가 존재하지 않는다.-이 경우 문법적 정보만으로 분리된 형태소를 이용하여 임의의 어절을 만들게 되므로 생성이라는 표현이 적당할 것이다.- 그러므로 형태소 결합을 위한 부분문법에서 이를 고려하여야 하는데, 이 때 어절의 끝을 나타낼 수 있는 것이 원거리 연결자이다. 앨고리듬 1 과 2는 전술한 방법에 의한 원거리 연결에 대한 규칙을 나타낸다.

앨고리듬 1과 2에서 NumberOf(n)은 사전에서 검색된 하나의 형태소를 하나의 노드로 정의했을 때, n 번째 노드의 순서를 의미하며, NodeOf(i)는 i번째의 노드를 의미한다.

기존의 일본어 해석법의 경우[3,4,5,6,7,9], 형태소간의 효율적인 결합에 중점을 두어 설계되었기 때문에 한국어로의 번역이라는 입장에서는 적절치 못한 것이었다. 그러므로 한국어로의 번역을 전제로 한다면 해석의 목표가 될 수 있는 것이 어절의 생성이다. 예를 들어 위의 "彼には"에 대한 경우, 이후에 분리 속성을 가지는 형태소가 이어짐을 가정하면(예를 들면 용언.), 명사 + 조사1 + 조사2"의 형태소 연결이 이루어지고, "명사 + 조사2"의 범위로 원거리 연결자가 결합하여 전체가 하나의 어절을 형성한다. 이를 위해서는 일본어 문법규칙에 따른 품사끼리의 결합뿐만 아니라 한국어의 어절에 해당하는 것을 생성하기 위해 품사끼리의 분리도 정의하여야 한다. "명사 + 조사"는 하나의 어절로 계속 이어지는 것으로 정의하며, "조사 + 명사"는 서로 분리되어 두 번째 명사는 새로운 어절의 시작점이 된

1. "학교"라는 명사는 오직 명사로밖에 쓰일 수 없지만 "사랑"이라는 명사는 고유의 명사 기능 이외에도 "사랑하다"라는 동사의 어간의 일부로도 쓰일 수 있다. 이때 "학교"를 단독명사로, "사랑"을 동사형 명사(상태성 명사)로 정의하였다.

2. 한국어는 "그에게는"으로 표현할 수 있으며 "그+에게+는"으로 형태소의 분리가 가능하고, 이는 위의 일본어와 같은 구조를 가질 수 있다.

다. 본 논문의 방식에 의하면 기존의 일반적인 규칙과 상이한 부분이 있는데 "NP + NP"의 경우가 그것이다. 일반적으로 문맥자유 문법으로 표현한 기존의 규칙은 NP -> NP NP의 형식으로 전체의 문장을 파악하려고 시도하지만 본 논문에서는 문장 전체가 파악이 대상이 되는 것이 아니라, 어절의 생성이 목적이므로 다른 결과가 발생된다. 즉 "명사 + 명사"는 결합 가능한 것이 아니라 분리되는 것으로 간주하며 두 번째 명사가 새로운 어절의 시작점이 되는 것이다. 이것은 본 논문의 해석방식이 전체문장이 아닌 어절을 파악하려고 하기 때문이다.

Algorithm 1. Make a Connection

```
Function MC( var n: NodeType, s: integer ) : Boolean;
type
    ConnectionType = ( simple, separation, notgood );
var
    a: NodeType;
    x: Connection;
    s, y: integer;
begin
  x <- CheckBackwardConnetion( n, n-1 );
  if x = simple then
    Connect( n, n-1 );
    return TRUE;
  else if x = notgood then
      return False;
    end { if }
  else
    y = MakeFarConnection( n-1, s );
    if y = NumberOf( n-1 ) then
      s <- NumberOf( n );
    else
      MakeMiddleSeparation( s, y, n-1 );
      s <- NumberOf( n );
    end { if }
  end { if }
  return TRUE;
end { MC }
```

Algorithm 2. Make a far Connection

```
Function MakeFarConnection
            ( var n: NodeType, s: integer ) : integer;
type
  ConnectionType = ( simple, separation, notgood );
var
  x: Connection;
  i: integer;
begin
  x <- notgood; { Initialize }
  i <- s;
  While x = simple do
    if i = NumberOf( n-1 ) then
        return NumberOf ( n-1 );
      end { if }
      x <- CheckBackwardConnection( NodeOf( i ),
      NodeOf(NumberOf(n)) );
      i <- i + 1;
  end { while }
  return i-1;
end { MakeFarConnection }
```

전체 문장에 대한 연결자 결합의 예를 들면 그림 3과 같다. 그림 3의 (a)는 (2)의 문장의 경우의 해석으로 부분 문법에 의한 형태소의 연결을 나타내고 있다. (b)의 경우는 3개의 형태소가 결합하여 하나의 어절을 이루는 것을 보이고 있는 것이다. 그림 3의 (b)에서 각 연결자는 어절내에서 문법적 정보에 의한 형태소의 연결만 나타내며, 각 어절을 원거리 연결자로 구분하여 이후의 구문해석의 입력 정보로 사용한다. 원거리 연결자는 이후의 형태소와 "분리" 관계에 의해서만 성립되며 어절을 대상으로 한 결합 범위를 결정짓는다.

2. 형태소 분리

3.1에서 형태소들의 결합으로 어절을 만들어 내기 위해 먼저 진행되어야 할 작업이 형태소 분리이다. 형태소 분리는 입력된 문장과 시스템이 보유한 사전과의 연관으로 이루어지는데, 이러한 관점에서는 기존의 한국어나 영어의 형태소 해석 부분에서 실시하는 형태소 분리와[3] 기본적으로 같은 것이다. 그러나 일본어는 이들과 근본적으로 다른 문제점을 가지고 있으며 분리의 범위가 그것이다. 해석할 문장이 어절단위로 띄어쓰기가 되어 있다면 형태소 분리의 범위는 어절이 될 것이지만, 일본어의 경우는 결국 문장 전체가 된다는 것이다. 이것은 다음과 같은 문제를 초래할 수 있다. 첫째, 한국어에서 하나의 어절이 사전에 완전한 형태로 등록되어 있을 수 있듯이(예를 들면 단어의 형태 변화가 거의 없는 부사의 경우가 그렇다.), 일본어에서는 입력된 문장(정확히는 하나의 토큰으로 인식되는 입력된 전체 문자열) 전체가 사전에 등록되었을 가능성을 배제할 수 없으며 이는 좌에서 우측으로 해석이 진행된다면 최악의 해석에 해당할 수 있는 경우이다. 우에서 좌측으로의 해석에 대해서도 최장 일치법의 경우는 첫 번째 시도로 해석을 진행하지만, 이것이 바로 적절한 해석이라고 할 수 없음은 이미 밝혀진 사실이다[2,3,6]. 둘째, 입력문 전체에 대해 여러 가지의 형태소 분리 결과가 발생할 수 있다는 것이다. 예문 (4)와 같이, 분리되는 범위에 따라 입력문은 전혀 다른 해석과 번역 결과를 가져올 수 있으며 이는 번역 대상 문장을 고려할 때 아주 빈번히 발생하는 문제이다. 일반의 텍스트가 아니라 담화문을 번역의 대상으로 한다면 고유명사 등 미등록어가 빈번히 발생할 수 있기 때문에 분리 범위의 설정을 더욱 곤란하게 한다. 결국 일본어의 해석은 기존의

3. 일본어의 경우 전체 문장이 하나로 어우러져 있기 때문에 형태소 해석과정 이전에 형태소 분리라는 표현을 사용하지만, 한국어나 영어의 경우는 이러한 표현을 사용하지 않는 것이 일반적일 것이다.

다른 언어의 해석에 형태소 분리라는 하나의 문제가 더 추가되는 형식이라고 할 수 있다. 형태소 분리는 별도의 작업으로 구성되는 것이 아니라 형태소 결합을 검사하기 위한 입력의 생성과정으로 진행되며 형태소 결합과 동시에 진행된다.

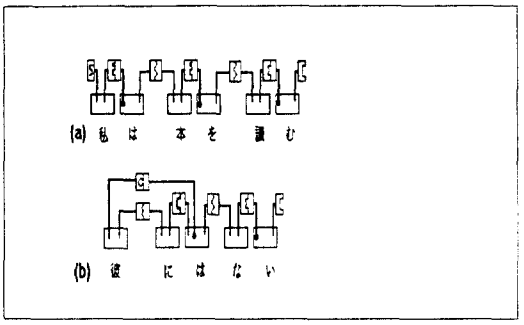


그림 3. 연결자에 의한 문장의 형태소 해석
Fig. 3. A morphological analysis using connector

본 논문에서는 위의 문제점을 해결해 나가고 불필요한 검색횟수를 줄이기 위해 형태소 해석사전과의 연계를 고려하여 시스템을 설계하였다. 해석될 대상의 문자열을 사전 시스템에 전송하면 사전 시스템은 검색한 최소범위의 등록어를 해석 시스템에 반환시켜준다. 해석 시스템은 반환된 등록어까지만 형태소 분리를 시도하므로 시스템의 입장에서는 효율적인 것이다. 이것은 실제로 시스템의 설계에서는 상당한 효율의 향상을 가져올 수 있는 것이다. 예를 들어 "ABCD…HIJ"와 같이 10개의 문자로 이루어진 입력문이 있다고 가정하면, 첫 번째 토큰이 되는 "A"부터 마지막 토큰인 "ABCD…HIJ"까지 모든 문자열이 하나의 단어가 될 가능성을 절대 배제할 수 없다. 그러므로 해석목의 생성시 매번 최장의 문자열에 대해 사전의 등록여부를 검사해야 하고, 이것은 시스템에서는 상당히 부담이 될 수 있다. 본 논문에서는 마지막 토큰까지 사전에 계속 등록 여부를 검색하는 것이 아니라 첫 번째 검색에서 입력문에 대한 검색의 범위를 사전에서 반환 받으므로 불필요한 부분에 대한 검색 시도를 방지할 수 있는 장점이 있다.

3. 형태소 해석 사전

형태소 해석 사전은 입력된 문자열을 분리, 결합하는 과정에 필요한 것으로 재 생성된 문자열에 문법적 정보를 부여하는 역할을 한다. 사전에는 문법적인 기본 정보에 대한 색인뿐만 아니라 단어 고유의 문법을 내장하고 있으며, 형태소 해석에 적용되는 사전의 문자열 정보도 유동적으로 포함하게 된다. 예를 들어 "かっこ"라는 명사는 다음과 같은 구조를 사전정보에 포함한다.

```
(かっこ  ((NOUN  +) (NVERB  +))                                    ①
         (Connector (for | ), (back | ), (far | ))                  ②
         (String  +)                                                ③
         ( | Partial grammar | ()))                                 ④
```

①과 ②는 단어의 문자열과 연결자에 대한 정보를 포함하고 있고, ④는 해당 단어에 대한 부분 문법을 표시하게 된다. ③은 일종의 인덱스에 해당하는 것으로 검색할 다음 단어가 자신을 포함하고 있는지를 나타낸다. 다음 색인에 문자열 정보가 다른 단어가 등록되어 있으면 "-" 값을, 문자열 정보에 자신을 포함한 단어가 등록되어 있으면 "+" 값을 가지며, 이것은 사전 검색과 형태소 분리의 효율 향상을 위한 것이다.

4. 형태소 해석

본 논문에서 제안하는 형태소 해석은 형태소 분리와 분리된 형태소의 재결합으로 이루어진다. 해석의 방향은 좌에서 우측으로 진행하며, DAG(Directed acyclic graph)를 구성하여 불필요한 반복을 줄여 효율을 상대적으로 높였다[11]. 기존의 일본어 해석 시스템에는 최장일치법에 의한 좌방향성 해석이 있었으나, 이는 유일의 해를 얻을 수 있는 반면, 최적의 해를 얻는다고 보장할 수는 없는 단점이 있었다[2,3,4,6,12]. 우방향의 해석 방식은 실시간 처리를 위한 On-Line Parsing에도 적용할 수 있으므로[17] 이후의 시스템 확장도 용이할 것으로 생각된다.

전술한 방식에 따른 형태소 해석 앨고리듬은 Algorithm 3과 같다.

Algorithm 3. Phrase Analysis

```
Function  PhraseAnalysis
          (var StartPoint: integer, EndPoint: integer ):Boolean
var
    x, DictionaryStartNumber, DictionaryEndNumber : integer;
    Word : Char;
    n: NodeType;
begin
    Word (- String( StartPoint, EndPoint );
    DictionaryStartNumber(-FindFirstDict( StartPoint, EndPoint );
    EndOfString (- CheckDict (DictionaryStartNumber, Dictionary
    EndNumber );
    if  DictionaryStartNumber = Fail  then
        return FAIL;
    end { if }
    While x < EndOfString do
        word (- String( StartPoint, x );
        if  AleadyFound( word ) = Fail  then
            MatchWord( word );
            n (- MakeANode( word );
            if  MC( n, CountNode( n )) = Fail  then
                return  FAIL;
            end { if }
            PhraseAnalysis( x, EndOfString );
            x (- x + 1;
    end { if }
  end { while }
  return TRUE;
end { PhraseAnalysis }
```

형태소 해석 앨고리듬은 형태소의 분리와 재결합의 반복에 의해 노드가 추가되고, 연결도에 의한 문법의 조사에 의해 DAG구조의 해석 그래프를 생성한다.

입력문자열은 그림 4와 같은 구조로 정의하였다. 문자열을 우선 하나의 문자단위로 구분하였으며, 이 때 문자와 문자 사이를 Point로 정의하고 이 Point에 정의되는 단어를 그래프에서 하나의 노드로 정의하였다. 최초 Point는 0이며 String(x, y)는 Point x에서 Point y까지의 문자열을 의미하며, 이 때 Connection 정보가 올바르다면 String(x, y)는 노드로 정의될 수 있다.

시작 포인트에서 하나의 문자를 기본으로 검색하며 첫 번째 검색에서 사전에서 검색할 범위와 여기에 맞는 문자열의 범위를 얻을 수 있다. 예를 들어 그림 4에서 첫 번째 문자 "う"가 입력문자열이 되면 첫 번째 사전 검색에서 "う"에 관계되는 사전의 범위를 돌려 받게 되며, 이에 따라 "うみがめ"를 검색의 최대 범위로 설정한다. 이로 인해 불필요한 검색을 줄일 수 있으며, 미정의어에 대해서도 능동적으로 대처할 수 있다. 또한 검색의 초기조건으로 현재 문자열에 대해 이미 검색이 이루어 졌는지를 검사하여 검색의 중복을 피할 수 있으므로 효율적인 그래프를 구성할 수도 있다. 앨고리듬 3의 초기 부분에 의해 문자열을 분리하면 연결자 검사로 노드간의 적합성을 조사하고, 검색에 성공하면 노드를 만들고 연결시킨다.

그림 5는 위에 서술한 형태소 해석과정을 영문자 열에 적용한 결과를 나타내고 있다. 최초 문자열을 "abcbabcd"라고 가정한다면, 먼저 해석 시스팀은 이 문자열을 사전 시스템에 전송하고 사전 시스템은 검색된 문자열의 최소 범위로 "abc"와 "abcb"를 반환한다. 해석 시스팀은 이들의 값을 접속관계를 기준으로 평가하여 해당 문자열을 선택하고, 이들을 제외시킨 "babcd"와 "abcd"를 다시 사전 시스팀으로 전송한다. 이와 같은 규칙을 문자열 검색이 종료되는 최종 순간까지 반복하여 그림 5의 ⑨와 같은 결과를 얻게 된다.

이와 같은 방식은 검색 대상이 되는 문자열의 길이에 영향을 크게 받지 않으므로 일본어 문과 같은 특징을 가진 문자열의 처리에 적합할 것으로 판단된다. 그림 4의 문자열을 동일한 검색조건을 사용하여 최장일치에 의한 해석법과 비교한 결과를 표 2에 나타내었다.

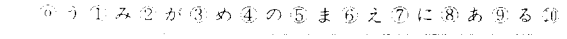
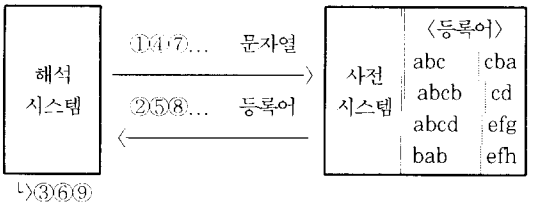


그림 4. 분리를 위한 입력문자열의 정의
Fig. 4. A definition of the input sentence.

본 논문에서 제안하는 방식은 적은 횟수의 검색으로 가능한 문자열을 분리해 내는 것을 알 수 있다.



① "abcbabcd"
② "abc", "abcb"
③ ("abc"), ("abcb")
④ "babcd", "abcd"
⑤ "bab", "abc", "abcd"
⑥ ("abc", "bab"), ("abcb", "abc"), ("abcb", "abcd", NIL)
⑦ "cd", "d"
⑧ "cd", NIL
⑨ ("abc", "bab", "cd", NIL), ("Abcb", "abcd", NIL)

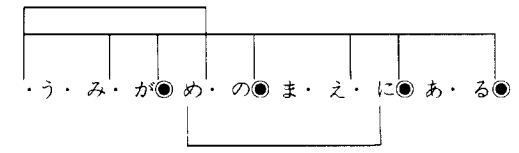
그림 5. 형태소 해석과정의 일례
Fig. 5. An example of the morphological analysis procedure

예문 "うみがめのまえにある(umigamenomaeniaru)"[19] 는 그림 6과 같은 해석 결과를 얻는다. 그림 6의 (a)는 해석된 결과를 입력의 형태로 노드와 포인트로 나타낸 것이다. 노드의 연결은 각 형태소를 일컫는 것이며 "◉"으로 표시된 포인트는 형태소 결합에 의해서 어절로 분리될 수 있는 것이다. (b)는 해석결과를 출력의 형태로 표시한 것이며 시스팀에 저장되는 형태이다. 너비우선의 최장 일치법에 근간한 방법에 비하면 문장의 저장 형태가 대폭 감소하였으며 이것은 문장간 번역을 고려한다면 큰 비중을 차지할 수 있는 문제이다.

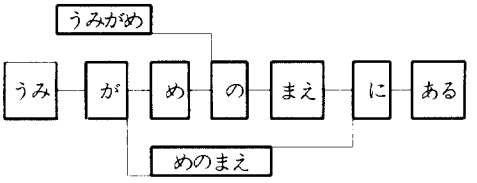
표 2. 해석결과의 비교
Table. 2. A comparison among analyzed results

	최종 분리된 문자열	검색시도 횟수
최장일치	"abcb" + "abcd"	17
최단일치	"abc" + "bab" + "cd"	23
본 논문의 방식	"abc" + "bab" + "cd" "abcb" + "abcd"	10

미정의어는 앨고리듬 초기에 실패하므로 각각의 문자열로 분리되어 개별 노드를 구성하게 된다. 형태소 해석 후처리에서는 이 미정의 노드를 연속된 것에 한해 하나의 노드로 결합하며 미정의어 노드는 원칙적으로 어느 곳이나 존재할 수 있게 하였다.



(a) 해석결과



(b) 해석결과의 저장 그래프

그림 6. 입력문의 해석결과 형태의 일례
Fig. 6. An example of the analyzed result about the input sentence.

5. 형태소 해석의 범위

일본어의 해석에 있어서, 특히 용언의 해석에 있어서 관건의 하나가 해석의 범위이다. 예를 들어 조동사 "ます"의 과거활용형은 "ました"이며, 이것은 "まし(연용형) + た(과거, 종지형)"로 볼 수 있다.[4] 이 때 활용형의 처리가 사전에 의존적인지 혹은 그렇지 않은 지의 여부가 문제가 된다. 즉, 기본형과 활용형으로 구분하여 모두 사전에 등록시키는 경우가 가능하고, 일부만 사전에 등록시키는 경우가 가능하다. 본 논문에서는 어절내의 충실한 해석에 목표를 두고 있기 때문에 후자의 경우를 대상으로 하여 일부만을 사전에 등록시키도록 하였다. 조동사의 변화형으로 본 "ました"의 경우, 기본형으로 "ま"만을 등록하고 "し"는 여기에서 단독활용하는 품사가 아니라 "ま"에 종속하는 연결형이므로 시스팀 내에서 처리시키며, "た"는 과거를 나타내는 활용형이므로 사전에 기술시킨다. 이 방식은 연결형을 먼저 시스팀 내에 기술함으로써 해석이 가능하며 사전의 구성 시에 다소의 부담을 줄 수 있다. 그러나 문법적 기능만을 가지고 의미정보를 지니지 않는 연결형의 종류가 많지 않으며 거의 모든 용언의 활용에 공통적으로 적용가능하므로 어절의 해석시 충분한 문법적, 의미적 정보를 부여할 수 있는 장점이 있다. 결국 조동사 활용형 "ました"는 "ま{し} + た"로 해석 가능하다. 여기에서 {}는 시스팀내에서 처리하며 사전에 등록하는 것이 아님을 의미한다.

본 논문에서는 전절까지 논술한 문자열의 분리와 생성을 근간으로 전술한 해석방식을 취하였다. 즉 "書きませんでした"라는 일본어 문자열은 "書き + ませ + ん + でし + た"로 분리하여 해석하며 전체가 하나의 어절을 이루게 하고, 사전에 등록하는 표제어는 "書, ま, ん, で, た"이다. "き, せ, し"등은 여기에서 의미적 정보를 갖지 않는 단순한 문법적 연결 기능만을 한다. 조동사는 부속어이지만 중심 문자열과 부속 문자열로 다시 구분하며, 사전에는 중심어만을 등록시킨다. 이것은 한국어 생성에서 충실한 번역어를 만드는 중요한 근거가 되며 각 부분은 기타의 용언에도 적용할 수 있으므로 효율적이다. 그러나 조사와 명사 등의 경우는 이와 달라서 더욱 복잡한 경우를 가진다. 일례로 일본어 조사 "でも"는 그 자체가 단독으로 조사의 기능을 수행하기도 하지만 "で + も"의 형태로 조사와 조사가 연속하는 형태로 해석할 수도 있다. 그러나 또 다른 조사 "とは"는 "と" 와 "は"가 각각의 조사 기능을 가지고 있기는 하지만 "と + は"의 결합 형태로 해석할 수는 없다. 두 가지 경우 모두 분리된 문자열이 분리되지 않은 문자열과 품사적으로 동일한 형태를 가지므로 분리되지 않은 문자열이 사전에 등록되어 있다는 것을 전제로 분리된 문자열을 포함하는 별도의 해석목을 생성하지 않는다. 이것은 최장일치와는 다른 것으로, 최장일치에 의한 방법은 최장의 문자열만 검색하지만 본 논문의 방식은 모든 문자열의 검출 이후에 재결합으로 문자열을 최종 분리시킨다.

6. 해석효율의 향상

일본어 문장에서 나타날 수 있는 문자열의 형태는 계산기용 코드를 기준으로 보통 히라가나, 가타카나, 한자, 기타 영문자와 숫자를 포함하는 특수 문자군들로 구분할 수 있다. 이들이 결합된 형태는 다양하며 [21]에 의하면 순수 문자가 아닌 혼종어(위의 여러 형태가 복합된 한 단어)가 전체 단어의 약 5% 정도를 차지한다고 한다. 본 논문에서는 이와 같은 문자열을 처리하고 검색의 효율을 높이기 위해 다음과 같은 휴리스틱을 두었다.

i. 숫자와 영문자, 기타 특수문자는 형태소 해석시 사

4. 이외에도 "ました"는 다른 해석이 가능하지만 본 절에서 주된 대상으로 하는 것은 용언의 활용이므로 조동사 "ます"의 과거형에만 국한하여 논하도록 한다.

전검색을 하지 않으며 명사 특징을 가지는 미정의 어로 분류하여 전처리한다. 이 단어는 이후의 한국어 생성시 역어 사전에서만 탐색이 이루어지게 하여 불필요한 검색을 줄이도록 한다.

ii. 접두어는 별도의 품사로 처리하여 접두어를 포함하는 다른 단어를 이중으로 사전에 등록시키지 않는다. 그러므로 "お宅"은 별도로 사전에 등록시키지 않고 "宅"이라는 명사에 '존칭'의 속성이 부가되는 [19] 형태로 해석한다.

예를 들어 "お客様の名字はKIMですか(손님의 성함은 KIM입니까?)"라는 문장은 "お/客様/の/名字/は/KIM/ですか"로 전처리하여 기본적인 사전 검색 기준을 제시한다. 이 휴리스틱은 미등록어 처리에서 불필요한 검색을 줄일 수 있는 효과가 있다.

Ⅳ. 구문해석

본 일본어 해석 시스템에서 구문 해석의 목적은 문의 격구조와 의미 관계의 추출에 있다. 일본어와 한국어는 다의어 문제 등을 제외하면 문 전체의 구조 해석이나 의미 해석을 하지 않아도 번역이 가능한 것으로 알려져 있다. 그러나, 문장간 번역을 고려한 번역 시스템을 최종 목표로 한다면 문장 단위의 의미 구조가 반드시 필요하며, 이를 위해 기존의 차트 파징 방법을 이용하였다. 파서는 병행하여 진행되고 있는 한국어 파서와 [19] [20] 기본적으로 동일한 구조를 가지고 있다. 그림 7에서 차트 파서의 입력은 전절까지 기술한 형태소 해석 결과로 생성된 각 어절이며, 최종 해석목과 그림 8과 같은 의미 표현식을 결과로 생성하였다.

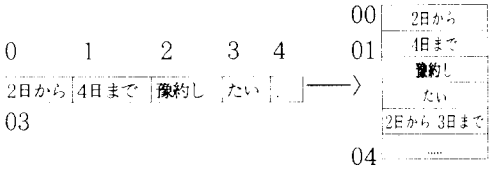


그림 7. 초기 차트와 Edge_queue
Fig. 7. An initial chart and the edge queue.

의미 표현식은 의미부와 문법부로 나누어 구성하였고 문장을 해석 결과와 비교할 때 간략화하여 저장하며, 특정 영역에 대해서는 TIME_SOURCE 등과 같이 별도의 처리 변수군을 두어 해당 영역에 관한 처리를 가능하게 하였다. 이 의미 표현식으로 연속적인 담화 형태의 문장 번역 시에 현 단계에서 기본적인 대화 진행에 관한 파악이 가능하며 [18] [20] 현재에는 대화 진행의 전 단계로 대화 저장을 수행하며 한국어 생성의 기본적인 의미 구조로 활용하고 있다. 구문 해석의 방법은 기존의 방법과 유사하며 본 논문이 구체적으로 논하고자 하는 대상이 아니라 문장 단위의 번역을 위한 응용이므로 세부 부분은 생략하도록 한다.

입력문 : 2日から4日まで豫約したい.
의미표현식:

```
(SEM (TIME_SOURCE      2日 〈TIDAT〉 )
     (TIME_DURATION    3日 〈TIDAT〉 )
     (PRED             豫約)
(SYN (VFORM            PREDICATIVE )
     (SAHEN            + )
     (HOPE             + ))
```

그림 8. 의미표현식의 일례
Fig. 8. An example of the semantic representation.

Ⅴ. 시스템 구성 및 실험결과

본 논문에서 제안한 일한 기계 번역을 위한 일본어 해석 시스템은 크게 문장 입력부, 형태소 해석부, 구문 해석부와, 관련되는 사전부로 세분하여 설계하였으며, 현 단계에서 가장 난점이 되고, 일한 번역을 전제로 했을 때 가장 많은 정보를 추출해야 하는 형태소 해석부를 중심으로 설계하였다. 시스템의 측면에서는 해석 그래프의 변형 형식으로 제 과정을 진행하여 요구 영역을 최소한으로 줄이도록 고려하고 차후의 시스템 확장에 대처할 수 있도록 하였다. 문장의 입력 과정에서는 효율의 향상을 위해 휴리스틱을 이용한 문자열의 전처리를 수행하여, 형태소 해석에 도움이 되는 형태로 문자열을 분리하였다.전처리에 의해 분리된 문자열에 해석적 정보는 없으며, 단지 히라가나와 가타카나로 구분한 일본 문자와 한자, 기타 영문자를 포함하는 특수 문자로 구분되어 있을 뿐이다. 이는 형태소 해석시 해석 범위의 기초를 제시하며 해석의 범위를 일차적으로 한정시키는 역할도 한다. 구문 해석은 별도로 연구하고 있는 한국어 구문 해석기와 동일한 구조를 사용하였으며 이로 인해 차후에 양방향 번역이 가능할 수도 있다. 의미 표현식에서는 한국어 생성을 위해 4 개 부류에 45 개의 속성을 설정하였다.

시스템은 워크스테이션 기종에서 C언어로 구현하였으며 단일 프로세스를 사용하였다. 입력은 ASCII 형태의 일본어 문장이며 최종 결과로는 그림 9와 같이 어절로 구분된 입력문이 출력된다. {}로 묶인 문인 분

자열은 사전에 등록되지 않은 부속어를 뜻하며 [ ]로 묶인 문자열은 형태소 해석시에는 미정의어로 처리되는 경우이다.

a. 입력문 :: うみがめのまえにある
해석결과 :: ("うみがめ+の", "まえ+に", "ある{る}")
—〉 바다거북의 앞에 있다.
("うみ+が", "めのまえ+に", "ある{る}")
—〉 바다가 눈앞에 있다.
("うみ+が", "め+の", "まえ+に", "ある{る}") —〉 바다가 눈 앞에 있다.

b. 입력문 :: お部屋の案内を聞きしたいですか
해석결과 :: ("お+部屋+の", "案内+を", "聞{き}+し+た{い}+で{す}+か")
—〉 방 안내를 듣고 싶습니까?

c. 입력문 :: 2日から4日まで豫約したい.
해석결과 :: ("[2]+日+から", "[4]+日+まで", "豫約+し+た{い}")

그림 9. 해석결과의 일례
Fig. 9. An example of the analyzed output.

내부 표현으로는 구문 해석된 결과와 그림 8과 같은 의미 표현식이 스택에 저장되어 이후의 한국어 생성의 입력이 된다. 이 때 일본어 문장의 형태소 해석 과정에서는 규칙에 어긋나지 않는 한 모호성을 처리할 수 없으므로 일의적인 결과를 출력할 수 없고 이에 따라 모든 해석 결과를 출력하도록 하였다. 이 복수개의 해석 결과는 시스템의 응용에 따라 한국어 생성시 의미 구분과 문장간의 대화 처리에 따라 단일 해석목에 가깝게 출력할 수 있도록 가지치기를 할 수 있을 것으로 예상된다.

VI. 결 론

본 논문에서는 일한 기계 번역을 위한 일본어 해석 시스템을 제안하였다. 기존의 해석 시스템과는 다르게 문의 문법적 연결을 주로 파악하려 하는 것을 지양하고, 한국어 생성 시에 유리하도록 한국어의 어절에 준하는 어절을 생성하고 각 어절 내에 필요한 정보를 최대한 추출할 수 있도록 설계하였다. 일본어 해석 시스템은 형태소간의 문법적 연결에 근거한 해석적 정보뿐만 아니라 구문과 의미적 정보를 부가시켜 한국어 생성시 형태소 단위에서 비교적 충실한 문장을 생성할 수 있는 정보를 제공한다. 제안한 시스템은 일한 기계 번역 시스템을 예상한 전단계로서 일본어 해석을 수행하며, 따라서 문장 단위에서 일의적인 해석을 시도하기보다는, 번역의 제 과정에서 얻을 수 있는 정보를 가능한 많이 부가시켜 차후의 문장간 번역에도 대비하였다. 일본어와 한국어간에는 전산 언어론적 입장에서 많은 유사점이 발견되어 품질을 따지지 않는다면 본 시스템의 기본적인 형태소 해석 결과에 간단한 한국어 생성처리로도 번역이 가능하리라고 보여진다.

본 논문에서는 또한 일본어 해석을 위해 연결자의 개념을 이용하여 일본어 형태소 분리를 실시하고 이를 위해 기본적인 연결자와 원거리 연결자의 기능과 역할을 정의하였다. 원거리 연결자로 띄어쓰기가 없는 일본어 문장에서 어절을 정의할 수 있었으며, 이로 인해 구문 해석을 실시할 수 있었다. 구문 해석은 차트 파징 기법을 이용하였으며, 한국어 해석을 위한 구문 해석기와 동일한 구조를 사용하여 해석의 규칙만 제공하면 일한, 한일의 양방향 번역이 하나의 시스템으로 가능하도록 구성하였다. 제안한 일본어 해석 시스템은 일한 기계 번역 시스템의 전단계로 의의가 있다고 보여지며 이를 위하여 시스템의 구조를 설계, 구현하였고 한국어 생성을 중심으로 필요한 제반 연구를 진행하고 있다.

참 고 문 헌

[1] 日本人工知能學會, 人工知能ハンドバック, オーム社, 1990
[2] 搏松明, "自動飜譯電話のための音聲處理と言語處理", 電子情報通信學會論文誌 D-II Vol.75, No.10, 1992
[3] 吉村賢治, 日高達, 吉田浮, "文節數最小法を用いたべた書き日本語文の形態素解析", 情報 處理學會論文誌, Vol.24, No.1, 1983
[4] 吉村賢治, 武內美津乃, 津田健藏, 首藤公昭, "未登錄語を含む日本語文の形態素解析", 情報處理學會論文誌, Vol.30, No.3, 1989
[5] 平井誠, 北橋忠安, "格の强度と述語の構文および意味屬性を用いた格構造の變換生成に ついて", 情報處理學會論文誌, Vol.28, No.3, 1987
[6] 奧雅博, "日本文解析における述語相當の慣用的表現の扱い", 情報處理學會論文誌, Vol.31, No.12, 1990
[7] 平井章博, 梶博行, 芦尺實, "機械飜譯向け前編集のための日本語係り受け構造の曖昧性 檢出方式", 情報處理學會論文誌, Vol.31, No.10, 1990
[8] Masahiro Oku, "A Method for Analyzing

Japanese Idioms"
[9] 元吉文男, 大場健司, etc,"未定義語を含む文の多段階構文解析法", 電子情報通信學會論文誌,D-II Vol .72, No.10, 1989
[10] 池田尙志, "語法規則方式による日本語文の構文意味解析", 情報處理學會論文誌, Vol.26, No.6, 1985
[11] Gosse Bouma, etc, "A Flexible graph-unification formalism and its application to natural-language processing", *Jounal of Research and Development*, 1988.
[12] N.Maruyama,M.Morohashi, etc, "A Japanese sentence analyzer", Jounal of Research and Development, 1988.
[13] Shalom Lappin,Michael McCord, "Anaphora Resolution in Slot Grammar", Computational Linguistics Vol.16,No.4 ,1990.
[14] Paola Velardi,Maria Teresa Pzienza, "How to Encode Semantic Knowledge", Computational Linguistics Vol.17,No.2, 1991.
[15] Daniel D.K.Sleator, Davy Temperley, "Parsing English with a Link Grammar",
[16] 강석훈,우요섭,김한우,최병욱,"다의어 처리를 고려한 일한번역 시스템의 구현에 관한 연구", 대한전자공학회 추계학술발표대회, 1990
[17] Seok-hoon Kang, Han-woo Kim, Byung-uk Choi, "On-Line Morphological Analyzer for Japanese to Korean Translation", JTC, 1992.
[18] Young-kil Kim, Seok-hoon Kang, Han-woo Kim, Byung-uk Choi, "Plan Recognotion for Q/A System Using Natural Language Processing", ICEIC, 1993.
[19] 한양대, 과학기술원, "대화체 기계번역에 관한 연구", 한국통신 장기기초연구과제 최종보고서, 1992
[20] 한양대, "자동통역전화를 위한 요소기술의 개발에 관한 연구", 전자통신연구소 최종보고서, 1992
[21] 佐治圭三 外, 日本語學의 理解

저 자 소 개

姜 晳 壎(正會員)
1966年 8月 31日生. 1989年 2月 한양대학교 전자통신공학과 졸업(공학사). 1992年 2月 ~ 현재 한양대학교 대학원 전자통신공학과 박사과정 재학중

崔 炳 旭(正會員) 제 31권 13편 제 3호 참조
현재 한양대학교 전자통신공학과 교수