

# 개념 및 구문 정보를 이용한 한국어 대화체 분석시스템

왕지현, 서영훈

충북대학교 컴퓨터공학과

## Korean Spoken Language Analysis System Using Concept and Syntactic Information

Ji Hyun Wang, Young-Hoon Seo

Dept. of Computer Engineering, Chungbuk National University

### 요 약

개념기반 분석방법은 발화문에서 발화자가 전달하고자 하는 중요한 부분만을 추출하여 개념어로 대표하여 분석하기 때문에 문장에서 발생하는 불필요한 여러 언어현상을 무시하고 주요 의미만 추출할 수 있는 강건함을 가장 큰 장점으로 갖는다. 한국어는 영어권 언어와는 달리 교착어와 부분 자유 어순의 특징을 가지기 때문에 구문정보를 이용하지 않는 순수 개념 기반의 분석기법을 한국어에 그대로 적용하면 문법의 복잡도가 증가하여 시스템 성능이 크게 저하된다.

본 논문에서 제시하는 구문정보를 이용한 개념기반의 분석방법은 순수 개념 기반의 분석기법이나 구문정보만을 사용하는 방법보다 모호성이 적고, 문법의 기술이 용이하며, 대화체 처리의 어려운 점들을 상당수 극복할 수 있다. 또한 분석루틴의 skip기능은 자연 발화문의 분석률을 높여 주며, 어근으로부터 분리한 어미를 일정한 개념으로 분류함으로써 교착어의 특성으로 인한 문법의 복잡도를 해소하였고, 분석문법으로 부분 자유 어순에 따른 다양한 문장들을 수용할 수 있다.

### 1. 서론

인간이 구사하는 자연언어는 크게 대화체와 문어체로 나누어 볼 수 있으며 그 간 국내외에서 수행된 자연언어 처리에 관한 대부분의 연구들은 문체의 범위를 문어체를 중심으로 진행되어 왔다. 그러나 자연언어처리라함은 사람들이 일상에서 사용하는 말들도 처리가 이루어져야 한다. 사람들의 일상적인 대화에는 문법적으로 처리할 수 없는 많은 잡음 요소가 섞여 있다. 사람들 특유의 버릇, 간투어, 특정 부분의 반복 등 문어체 문장과는 다른 많은 문제점이 있다. 또 사람이 말할 때는 문법적인 것을 고려하기보다는 의미를 전달하는데 중점을 두어 말하기 때문에 더욱 처리가 힘들다.[2,3] 대화체를 전사한 문장들의 특징들을 요약하면 다음과 같다.

1. 음운 축약이나 탈락현상이 빈번히 일어난다. :

재밋는(재미있는), 땀에(때문에), 그믐(그러면),...

2. 양성음을 음성음으로 발화한다. : 하구요(하교요),...

3. 어미 '요'가 체언, 용언, 부사, 조사, 어미 등의 뒤에 자주 붙는다. : 나는요, 빨리요, 세시요, 서울요, 가요, 제가요,...

4. 방언이나 은어등의 비표준어를 사용한다. : 어저께(어제), 갈애(갈아), 뵈께요(뵈께요),...

5. 간투어의 사용이 잦다. : 음, 저, 어, 그,...

6. 수정 또는 반복발화 : 세시 아니 네시 네시가 좋겠네요, 한 시 십일 정도,...

7. 문장성분의 생략, 필수격의 생략, 한 단어가 문장전체를 대표함 : 네시요, 언제요, 그러죠 뭐,...

8. 문장의 경계를 찾기 어렵다.

9. 존칭어의 사용

10. 잦은 조사의 생략 : 너 나 좋아하니,...

11. 문장부호가 나타나지 않는다.
12. 아라비아 숫자나 외국어 문자가 나타나지 않는다.
13. 격조사와 보조사의 결합이 빈번하다. : 께서가, 같이도,...

이외에도 기침소리나 호흡소리, 목 가다듬는 소리, 입맛 다시는 소리 등의 각종 잡음이 습관적으로 끼어들기도 한다.[1,3,4] 이러한 대화체의 특징들로 인해 현재까지 연구되고 개발되어 온 순수한 문어체 위주의 형태소분석이나 구문분석 시스템은 위와 같은 특징들을 해소하기 어렵다.

국내에는 아직까지 대화체에 관한 연구가 활성화되어 있지 않지만 국외의 대화체 연구는 CMU(Carnegie Mellon Univ)의 JANUS연구가 대표적이라 할 수 있다. JANUS project는 형태소 분석을 거치지 않고 단어 중심의 문법기술을 통해 의미 중심으로 파싱을 수행하는 Phoenix시스템에 기반한 번역과 구문과 의미정보를 이용한 Tomita 파싱기법을 확장한 GLR파싱시스템에 기반한 번역으로 연구가 이루어지고 있다.[1]

본 논문은 Phoenix시스템에서 사용하는 개념에 기반한 접근방법을 통해 대화체를 효율적으로 분석하고 영어권 언어와는 다른 특징을 갖는 한국어의 교착어 성질과 부분 자유 어순에 따른 성질을 해소하는 방안을 제시하고자 한다. 그리고 본 시스템의 대상 영역은 '여행안내'영역에 한정하여 설계하였다.

## 2. 개념기반 분석

순수 개념 기반 분석[Phoenix시스템]은 형태소분석이나 구문분석을 전혀 수행하지 않고 단어중심으로 문법을 기술한다. 이것은 한국어와 같이 어미를 여러가지 형태로 변화시켜 문장의 구성성분을 이루는 교착어인 경우에는 문법기술상의 단어 갯수에 따른 문법의 복잡도가 증가하게 된다. 따라서 형태소분석에 의한 어근과 어미를 분리하여 종결어미에 대해, '평서', '의문', '청유'로 나누고, 비종결어미는 '관형격', '일반부사격', '연결', '역접'등과 같이 문법적인 의미를 부여하며, '의지', '상태', '추측', '가능', '당위', '원인', '희망', '요구'등과 같이 양상정보를 추가하여 어미를 세분류하였다. 이와 같은 처

리는 무수히 많은 단어들에 대해 어근과 결합할 수 있는 어미의 갯수를 일정하게 제한하여 문법의 복잡도를 줄인다. 또한 어미의 세분류는 파싱루틴의 부담을 줄여주며, 생성의 구문구조를 결정하고 얼마나 잘 세분화되어있는지에 따라 생성의 질을 결정할 수 있는 요소가 된다. 따라서 분석시스템의 전단계에서는 형태소분석 시스템으로 입력 단어들의 형태소를 분석하여 체언과 용언에 대해 어근과 어미로 분리하고 분리된 어근과 어미에 개념을 부여하여 분석을 하게된다.

대상 영역에 자주 등장하는 어근들에 대해 85개의 최하위레벨 개념들로 나누어 코퍼스를 분석한 결과, 'year', 'month', 'day', 'hour', 'minute'등과 같은 시간개념과 'nation', 'city', 'island', 'ocean', 'park'등과 같은 장소개념이 매우 자주 사용되었으며 화폐단위나, 예약에 관련한 단어들도 자주 사용되었다. 여기서 사용된 어근들은 '호텔예약'영역 1575개의 발화문에서 중요하고도 자주 등장하는 어근들을 선택하여 개념들을 선정하였다. 이러한 정의된 개념들을 토큰이라 부르기로 한다.

어근과 어미에 대한 개념분류는 최하위레벨의 기본개념들을 나타내는데, 이와같이 최하위레벨 토큰들을 사용하여 구성된 상위레벨 토큰으로 입력문장의 부분개념에서 전체 개념을 구성해 나간다. 이에 따라 대상영역의 분석을 통하여 필요시되는 개념들을 아래와 같은 최상위 개념들을 포함하여 145개의 개념들을 작성하였다.

[give\_info] - 청자에게 발화자가 정보를 주는 문장에 대한 토큰.

[i\_want] - 발화자의 희망을 나타내는 문장에 대한 토큰.

[i\_will] - 발화자의 의지를 나타내는 문장에 대한 토큰.

[nicety] - 인사말이나 헤어질 때의 인사를 위한 토큰.

[query] - 발화자의 질의를 나타내는 문장에 대한 토큰.

[request] - 발화자의 요구를 나타내는 문장에 대한 토큰.

[respond] - 짧은 응답을 나타내는 문장에 대한 토큰.

위의 최상위 개념 토큰들은 발화자가 정보를 주거나 동의(agreement)나 거절(rejection)과 같은 문장의 틀을 제공하는 토큰들이다.

아래에 보이는 하위레벨 토큰들은 주로 명사와

동사를 위주로 구성되었으며, 이는 가능한 한 대상 영역이 바뀌더라도 쉽게 적용될 수 있도록 하기 위한 목적이다.

[applied], [call], [cancel], [confirm], [go], [know], [my\_availability], [reserve] 등은 동사를 중심으로 구성된 토큰들로서 발화자나 어떠한 객체의 행위를 나타내는 토큰들이다.

[airline], [alphabets], [arrival\_time], [car], [card], [cities], [currency], [local], [temporal], [time] 등은 명사나 구를 중심으로 구성된 토큰들로서 동사를 중심으로 구성된 토큰들에 의해 사용되는 토큰들이다.

분석문법은 하위레벨의 토큰과, 문법을 기술할 때만 사용되고 파싱결과에는 아무런 영향을 주지 않는 비단말(nonterminal)과, 하위레벨 토큰들과 비단말들로 구성된 파일 단위의 상위레벨 토큰을 사용하여 기술한다.

아래 그림은 시간에 관련한 토큰 '[point]'의 작성 예인데 토큰 '[point]'에서 볼 수 있듯이 년, 월, 일에 관련한 하위레벨 토큰들이 기술되어 있고 요일에 관련한 토큰 '[d\_o\_w]'과 시간관련 토큰 '[time]'이 기술되어 있다.

```
[point]
(*<year> *<month> *<day> *<noun 날> *TIME *AUX_NOUNS)
([d_o_w])

TIME
([time] **ANDTIME)

ANDTIME
(*<conj> [time])

AUX_NOUNS
(<noun 정도>)
```

'<'와 '>'로 둘러싸인 부분은 최하위레벨 토큰이며, 대문자로 구성된 문자열은 비단말이고 '['와 ']'로 둘러싸인 부분은 상위레벨 토큰이다. 하위레벨 토큰은 입력 토큰열과 비교대상이 되는 단말 패턴이 되고 '('와 ')'사이의 비단말 패턴은 비단말의 호출을 나타내며 비단말의 정의부는 호출의 이후에 기술되어 있다. 비단말의 사용은 파싱결과를 변화

시킬 수는 없지만 토큰열을 비단말로 나타냄으로써 복잡한 문법의 기술을 막을 수 있으며, 재사용률을 높여준다. 상위레벨 토큰은 '('와 ')'사이의 하위레벨 토큰과 비단말, 그리고 상위레벨 토큰의 비교가 모두 성공해야만 완성이 된다. '('와 ')'사이의 상위레벨 토큰의 호출은 또다른 문법을 하나의 문법이 포함함으로써 개념에 기반한 모듈별 작성을 가능하게 하며 이와 같은 방법을 통해 문법의 전체 크기를 줄일 수 있다. 또한 분석문법은 문법작성의 효율을 위해 토큰이나 비단말의 앞에 선택적(optional) 또는 반복적 성분으로 기술할 수 있다. 선택적 성분은 비교되지 않거나 한번 비교될 수 있음(zero or one)을 나타내는 의미로 '\*'기호를 사용하고 반복적 성분은 한번 비교되거나 한번이상 비교됨(one or more)을 나타내는 의미로 '+'기호를 사용하며 두가지 성분을 함께 나타내는 반복 선택적 성분은 한번도 비교되지 않거나 한번이상 비교됨(zero or more)을 나타내는 의미로 '\*\*'기호를 사용한다. 이러한 반복적 성분과 선택적 성분을 문법패턴과 함께 기술하면 한국어의 부분 자유 어순의 특징에 따른 발화문의 여러 형태를 수용할 수 있다.

입력 예1: 제가 구월 심요일날 뉴욕행으로 예약되어 있는지  
 입력 예2: 구월 심요일날 제가 뉴욕행으로 예약되어 있는지  
 입력 예3: 구월 심요일날 뉴욕행으로 제가 예약되어 있는지  
 입력 예4: 제가 뉴욕행으로 구월 심요일날 예약되어 있는지  
 입력 예5: 뉴욕행으로 제가 구월 심요일날 예약되어 있는지  
 입력 예6: 뉴욕행으로 구월 심요일날 제가 예약되어 있는지

```
분석문법:
[reserve]
(*SUBJ *TEMP *SUBJ *LOCAL *SUBJ VERB)
(*SUBJ *LOCAL *SUBJ *TEMP *SUBJ VERB)

SUBJ
({human})

TEMP
({temporal}) ; 시간관련토큰

LOCAL
({local}) ; 장소관련토큰

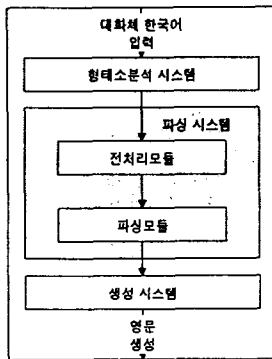
VERB
({reserve})
```

분석시스템은 대화체를 전사한 문장을 토큰들의 열로 입력을 받아 분석문법을 기반으로 분석을 수행한다. 대화체는 문어체와 달리 한번 발화한 말이 고칠 수가 없기 때문에 중복발화 현상이 잦고 간

투어가 자주 등장한다. 따라서 분석문법 안에 발화문의 전반적인 내용을 기술하였다 할지라도 불필요한 말이나 어구의 반복으로 인해 문법이 적용되지 못하는 경우가 있다. 이에 따라 분석시스템은 분석을 수행하다가 문법패턴이 입력에 맞지 않더라도 토큰을 일정한 갯수만큼 건너뛰어 볼 수 있는 기능을 가지고 있다. 이와 같은 특징은 대화체 처리에 필수적인 요소로 시스템의 강건함(robustness)을 제공한다. 분석시스템의 skip능력은 대화체 문장의 불필요한 요소를 제거하는 방법이지만 바꾸어 말하면, 입력 토큰열이 분석문법 어디에도 정확히 분석되지 않을 때 최소한의 분석된 정보라도 얻어 낼 수 있는 장점을 가지고 있다. 그러나 너무 많은 수의 토큰들을 건너뛰게 되면 문장의 의미를 제대로 살릴 수가 없으므로 일반적으로 하나 또는 두개 정도로 제한한다.

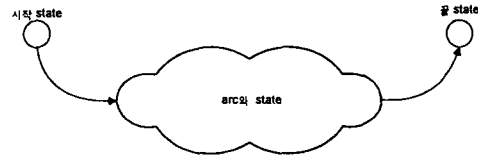
### 3. 시스템 구현

전체 파싱시스템은 형태소분석을 거친 입력을 대화체 처리에 맞게 변형하는 전처리 모듈과 전처리된 입력을 개념에 기반하여 분석을 수행하는 파싱모듈로 구성된다. 이와 같이 전처리 모듈을 따르는 이유는 문어체기반의 형태소분석 시스템을 그대로 사용하기 때문이다.



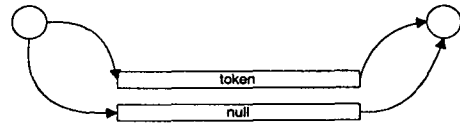
파싱모듈은 파일단위로 작성되는 상위레벨 토큰을 RTN(Recursive Transition Network)으로 해석한다. RTN은 상위레벨 토큰 단위로 만들어지며 문법과일에 작성된 문법은 유한 오토마타로 바뀌어 link로 연결된 state와 arc들로 구성되게 된다. RTN단위로 하나의 시작state와 끝state가 있고 그 사이에 문법패턴들이 비단말을 중심으로 rewrite된

다. 따라서 최상위레벨 RTN은 시작 state, 끝state와 함께 토큰들이 arc의 형태로 표현되어 있고, arc들간의 경로를 state들이 연결하고 있는 형태를 갖는다. 아래 그림들에서 둥근 원은 state를 나타내며, 직사각형은 arc를, 타원은 arc와 state의 집합을 나타낸다.



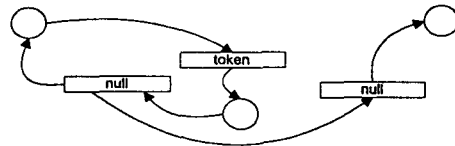
위 그림의 구름은 선택적 성분과 반복적 성분으로 표시되어 있거나 또는 어떠한 성분 표시도 되어 있지 않은 토큰이나 비단말에 의해 생성된 Sub RTN들의 모임으로 이루어져 있다.

선택적 성분

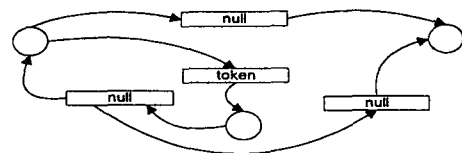


null로 표기된 arc는 다음 state로 그냥 전이할 수 있음을 나타낸다.

반복적 성분



반복 선택적 성분



반복 선택적 성분은 반복적 성분의 RTN에 null인 arc를 추가한 형태이다. 위의 세가지 성분에 따른 부분 RTN들을 조합하여 하나의 상위레벨 토큰에 대한 RTN을 구성한다.

기본 파싱절차는 RTN을 따라가며 하위레벨 토

큰을 만나면 입력 토큰열과 비교를 통한 pattern match를 수행하며 상위레벨 토큰을 만나면 그에 해당하는 RTN을 호출하여 match된 결과를 요구하는 동작을 반복한다. 파싱모듈은 전형적인 차트 파서로서, 차트에 기록하는 단위는 하나의 RTN단위로 기록을 한다.

간단한 파싱 수행 결과를 표현하면 아래와 같다.

한국어입력: 오월 팔일부터 오월 십오일까지....

```

          1         2         3         4
입력토큰열:<month 오> <day 팔> <from 부터> <month 오>
          5         6
        <day 십> <to 까지>

```

분석문법:

```

/* interval.gra */
[interval]
({begin_point| {end_point|})

/* begin_point.gra */
[begin_point]
({point| <from>})

/* end_point.gra */
[end_point]
({point| *<to>})

/* point.gra */
[point]
(*<year> *<month> *<day> *<noun 날> *TIME
*AUX_NOUNS)

```

분석결과:

```

[0]interval(1, 6)
[1]begin_point(1, 3)
[2]point(1, 2)
[3]end_point(4, 6)
[4]point(4, 5)

```

최종결과:

```

[interval]({begin_point}({point}(<month 오><day 팔>)<from
부터>)[end_point]({point}(<month 오><day 십>)<to 까지>))

```

위의 분석문법에서 [interval]이나 [begin\_point] 등과 같은 상위레벨 토큰들과 그것의 rule들은 따로 각각의 파일에 저장되어 있으며, 분석결과와 '('와 ')'사이의 숫자들은 입력 토큰의 번호를 나타낸 것이다.

## 4. 실험 및 고찰

개념기반의 분석은 생성과 매우 밀접하게 연관되어 있다. 개념을 설계할 때 생성을 고려하여 설계해야만 올바른 생성을 할 수 있기 때문이다. 따라서 생성에 관한 내용을 간략히 살펴보면, 분석문법과 같이 생성문법도 하나의 문법이 다른 문법을 포함하는 형식을 취한다. 'noun'이나 'month', 'day' 등과 같은 하위레벨 토큰들은 간단히 대역어들을 참조함으로써 생성을 한다. 상위레벨 토큰들에 대한 생성문법은 하위토큰(sub-token)들의 생성결과가 삽입될 수 있는 일련의 생성 틀로 기술된다.

실제 작성 예는 다음과 같다.

```

[there_is]
there are [many] places such as [local] <sublocal> there
there are <times> times <everyday> [*local]
we have [classes]
there are {room_class} in {local}

```

위의 생성문법은 토큰 '[there\_is]'에 대한 문법이다. 하나의 행이 하나의 생성 틀을 나타내는데 분석시스템의 분석 결과중 토큰 '[there\_is]'안에 포함되어 있는 토큰들이 가장 많이 생성문법에 기술되어 있는 생성 틀이 선택이 된다. 생성모듈은 선택된 생성 틀을 왼쪽에서 오른쪽으로 생성을 해가면서 토큰이 아닌 단어들은 그대로 생성을 하고 토큰을 만나면 그에 따른 생성 틀이 넘겨주는 생성결과를 그 위치에 삽입하여 생성을 한다.

아래에 대화체 입력에 대한 구문분석 결과와 영문생성의 예를 보인다.

1.대화체용 전사한 문장:

몇시에 출발하는 비행기요.

2.전처리를 거친 토큰열:

```

<what_time 몇시><verb 출발하><comi adj><airplain 비행기
>><comi quest>

```

3.분석결과:

```

[query]([question])(<what_time 몇시>[airline]([depart](<verb 출발하
>><comi adj><airplain 비행기>)<comi quest>))

```

4. 생성된 영문:

```

what time does the airplain departs

```

1. 대화체를 전사한 문장:

여보세요 제가 오월 팔일부터 오월 십오일까지 영국의 런던과 에딘버러를 여행하려고 하는데요 몇가지 물어볼 사항이 있어서 전화 드렸습니다.

2. 전처리를 거친 토큰열:

<hello 여보세요><month 오><day 팔><from 부터><month 오><day 십오><to 까지><nation 영국><city 런던><conj 과><city 에딘버러><verb 여행하><comi want><some 몇가지><verb 물><comi adj><noun 사항><verb 있><comi adj><verb 전화하><comi decl>

3. 분석결과:

```
[nicity]([greeting](<hello 여보세요>))
[i_want]([tour]([temporal]([interval]([begin_point]([point](<month 오><day 팔>>[from 부터]>[end_point]([point](<month 오><day 십오>>[to 까지]>))))[local]([local_name](<nation 영국>[cities](<city 런던><conj 과><city 에딘버러>))))[verb 여행하])<comi want>])
[query]([question](<some 몇가지><verb 물><comi adj><noun 사항><verb 있><comi adv>[call](<verb 전화하>)<comi decl>))
```

4. 생성된 영문:

Hello  
I'd like to tour London and Edinburgh in England from May eighth to May fifteen  
I have some questions

위의 두 예를 보면 알 수 있듯이 대상영역이 '호텔예약' 영역이기 때문에 장소와 시간에 관련한 토큰이 많이 사용되는 것을 볼 수 있다. 상위레벨 토큰 중에는 '[i\_want]'나 '[query]'와 같은 문장의 틀을 제공하는 분석문법이 있고 장소토큰 '[local]'이나 시간토큰 '[temporal]'과 같이 문장 틀에 의해 자주 사용되는 분석문법이 있다. 분석문법을 추가하다보면 주로 시간이나 장소토큰과 같은 부속토큰보다는 문장 틀에 관련한 토큰이 복잡해지는 경향이 있다. 그리고 발화자의 억양이나 강세(accent)는 문장안에 표현되지 않으므로 '몇시에 출발하는 비행기죠.'와 같은 문장이 의문문인지 평서문인지를 종결어미만을 통해 구별할 수는 없다. 따라서 파싱 루틴이 주변 토큰을 살펴서 '몇시에'와 같은 단서를 얻어내서 추측하는 방법밖에 없다.

또한 본 시스템은 발화자의 희망이나 감정의 '정도'를 인식하지는 않는다. 다만 정보를 주고 받거나, 동의또는 거절과 같은 기본적인 행위만을 인식한다. 하지만 더 많은 토큰을 사용하여 개념을 늘리고 각 개념에 해당하는 표현의 종류를 제한함으로써 더 넓은 표현들을 만들어 낼 수 있다.

## 5. 결론

대화체 처리의 개념에 의한 접근방법은 발화자의 순수 입력문장에서 구체적으로 전하고자하는 의미만을 추출하여 개념 단위로 묶어주는 구문분석을 수행한다. 구문분석 과정이 정형화된 통사정보를 이용하는 것이 아니라 필요에 의한 주관적인 의도에 의해 생성에 맞게 기술된 문법을 이용하므로 인식하려는 문장자체를 문장단위로 그대로 문법안에 표현하기 때문에 문법이 복잡해질 수 있는 경향이 있지만 대화체의 모호성을 줄일 수 있는 장점을 갖는다. 또한 개념기반 시스템의 한국어 특성에 관한 해결책과 함께 시스템의 skip능력에 따른 강건한 특성을 제시하였다.

현재 코퍼스의 분석과 함께 분석문법의 training 작업을 하고 있는 중이다. 분석문법의 확장으로 인해 문법이 방대해지다보면 이미 train된 문법들이 서로 영향을 받는 경우가 간혹 있다. 이와 같은 현상은 개념의 정확한 정의와 세분류로 해결을 할 수 있다. 따라서 개념기반 분석시스템이 분석문법에 크게 의존하는 만큼, 시스템의 성능은 잘 정의된 분석문법에 좌우한다.

### 참고문헌

- [1] '음성 언어 번역을 위한 개념 기반의 한국어 분석 및 생성', 서영훈, 정보과학회논문지(96.11)
- [2] '개념 구조를 이용한 한국어 대화체 구문 분석', 김영훈, 석사학위논문(97.2)
- [3] '대화체 및 문어체 기계번역을 위한 한국어 구문/의미 해석시스템 개발', 연구보고서, 한국전자통신연구소
- [4] '대화분석에 있어서의 몇가지 문제: 호텔 예약 전화 대화를 중심으로', 최재웅, 고려대, 한글 및 한국어정보처리(96.10)
- [5] 'PARSING REAL INPUT IN JANUS:A CONCEPT-BASED APPROACH TO SPOKEN LANGUAGE TRANSLATION', L.Mayfield, M.Gavaldà, Y-H.Seo. Interactive System Laboratories, Carnegie Mellon University.
- [6] 'JANUS:TOWARDS MULTILINGUAL SPOKEN LANGUAGE TRANSLATION', B.Suhm, p.Geutner, A.Lavie, L.Mayfield. Interactive System Laboratories, Carnegie Mellon University.
- [7] 'Recent Advances in JANUS:A Speech Translation System', A.Waibel, M.Woszczyna.