한국산업정보학회논문지 제9권 제4호(2004. 12)

문장 형태 정보를 이용한 조건단일화 기반 한국어 파싱

(A Conditional Unification Based Parsing for Korean Using Sentence-Type Information)

양 승 원*
(Seungweon Yang)

요 약 본 논문에서는 한국어 문장의 구조를 파악하는 데에 결정적인 역할을 하는 조사에 대한 정보를 보다 적극적으로 이용하는 파싱 방법을 기술한다. 이러한 방법을 실현하기 위하여 용언을 세밀하게 분류하였으며, 이를 바탕으로 문장의 형태를 분류하고 이 문장 형태에 필수적으로 나타나는 명사구의 문법적 역할을 파악하는 방법을 이용한 파싱을 시도하였다. 또한, 내부적으로는 언어학적인 틀에서 크게 벗어나지 않는 범위 내에서 약간의 경험지식을 동원하였다. 이를 구현함에 있어서 문법의 수준에서 파싱을 직접 제어하기 위하여 조건단일화 파싱을 사용하였다. 본 논문에서 제시한 파싱 방법은 분석의 초기에 불필요한 가지를 전지함으로써 결과 트리가 간략해져 복잡한 문장의 모호성을 상당부분 제거할 수 있게 해준다.

핵심주제어 : 파싱, 문장형태 정보, 조건단일화

Abstract In this thesis, we introduce a parsing method which use information of the post position in Korean to get the exact parsing tree. In order to implement this method we classified categories of the predicates, and defined sentence-types based on these categories. We tried to make parsing using the method grasping the grammatical role of the noun phrase that have to exist in each sentence-type. In parser control mechanism, we use some heuristics based on linguistic frame. We use conditional unification to implement analysis. It is possible to reduce ambiguous because the parsing method suggested helps to prune the branches which are unnecessary.

Key Words : Parsing, Sentence-type information, Conditional Unification

1. 서 론

자연언어 파싱은 문장에서 용언과 명사구 사이의 표층적 문법 관계를 밝히는 작업이다. 이러한 작업은 언어적 현상으로 파악되고 있으며 파싱을 기계화하기 위해서는 정리된 언어적 현상을 규격화하는 작업이 필요하다. 대부분의 한국어의 기계적 파싱에 있어서는 언어적 현상을 규격화하는 작업 보다는 자동화에 더욱 주안점을 두었다[1,2,3,4]. 따라서, 파싱에 의해 분석되는 문법적 관계가 한정되어 있었다. 즉, 주격이나 목적격을 표층적으로 나타나는 격 관계로 설정하고 나머지 명사구는 단지 보조적 명사구로 문장의 양태만을 나타내는 수의적 성분으로 분류하였다. 이러한 연구들은 대부분 5가지의 명사-동사간 문법 관계(subject(주격),

* 우석대학교 컴퓨터공학과

object(목적격), location(처소격), instrument(도구격), else(기타))를 설정하여 구문 분석을 시도하였다. 이 방법에서의 문제점은 5가지 문법적 관계 이외에도 특별한 격을 문장에 반드시 수반하는 용언이 무수히 많이 존재하는 데에도 불구하고, 이러한 용언들을 바라볼 때 5가지 문법적 관계 이외의 격을 보조적인 의미로(수의격으로) 파악하기 때문에 문장의 올바른 의미를 파악하기 힘들거나 국소적인 모호성 발생의 원인이 되었다. 즉, 한국어의 구문분석에서는 서술어를 단지 자릿수 서술어나 형용사, 자동사, 타동사 등으로 분류하므로 필수격 이외의 명사구들은 신경을 쓰지 못하기 때문에 틀린 문장이 옳은 문장으로 인식되는 경우가 빈번하게 발생된다. 국소적으로 틀린 문장이 옳은 문장으로 인식되는 것은 의미없는 부트리(sub-tree)를 전지(prunning)하지 못하고 파싱의 다음 단계에 제공하게 되므로 전체 문장의 모호성을 가중시킨다[5]. 따라서 용언을 보다 세밀하게 분류하여 이와 어우러지는 명사구를 가능한 한 분석의 초기에 찾아내는 시도가 중요하다.

본 논문에서는 한국어 기계번역을 위한 문법체계에 적합한 모델로 한국어에서 문장의 구조를 파악하는 데에 결정적인 역할을 하는 조사에 대한 정보를 보다 적극적으로 이용하는 구문 분석 방법을 기술한다. 이를 위하여 국어학에서 연구된 문장 형태 분류를 바탕으로 용언 자체를 세분화하고 용언이 요구하는 격 체계를 이용한 파싱을 수행하였다. 또, 제안된 방법을 구현함에 있어서는 문법의 수준에서 파싱을 직접 제어할 수 있는 CFG 기반의 조건단일화 파싱[1]을 사용하였다.

2. 연구에 도입된 개념

2.1 경험적 정보를 포함한 한국어의 문장 형태 분류

문형이란 문장의 구조적 유형으로, 수많은 개개의 구체적인 문장을 그 구조적 형식의 공통성에 따라 공식화한 틀로 정의 할 수 있다. 지금까지는 서구어의 문형을 한국어에 적용함으로써 여러 가지 무리가 따랐다. 여기에서 파생되는 문제점들을 근본적으로 해결하기 위해서는 한국어 문장의 기본 구조를 용언의 특성에 따라 다시 분류해야만 한다. 이러한 분류는 결국 용언이 어떠한 구문적 특성을 가지고 문장 요소를 필수적으로 동반하는지에 대한 광범위한 조사가 필요하다. [6]에서는 광범위한 조사를 통하여 다양한 문장의 구성 요건들을 기준으로 41가지의 기본 문장 형태를 제시하였다. 이는 언어 현상을 규격화한 것으로 상당히 세밀하게 분류되었으나 이를 곧바로 기계적인 파싱에 사용하는 데에는 많은 무리가 있다. [7]에서는 [6]에서의 분류를 바탕으로 기계적인 파싱을 시도하였다. 그러나, 이 연구에서는 이와 같은 분석 방법의 가능성만을 보여 주었을 뿐, 구문 분석에 필요한 충분한 정보를 이용하지는 못하였다. 본 논문에서는 이 기본 문장 형태를 기본으로 하되 다량의 코퍼스 분석을 통하여 얻은 경험적 지식을 첨가하여 한국어 분석의 틀을 결정하였다.

2.1.1 기본 문장 형태

문장은 아래(a-e)와 같이 5가지의 기본형으로 분류되고

a. NP[이] + V
b. NP[이] + NP[을] + V
c. NP[이] + {NP[을]} + V ;; {}은 생략 가능
d. NP[이] + A
e. NP[이] + N이다

문장의 기본 틀을 이루는 데에 사용되는 표층 문법형태소는 다음과 같다.

이(subj,comp), 을(obj), 에(loc), 에서(sour), 로(tar), 와(and,with), 게(AD), 를 위해(for), 에 의해(by), 에 대해(to), 에 비해(then), 보다(than), 라고(call)

여기에서 주목할 점은 본 논문에서는 '에 대하', '를 위하', '에 의하' 등도 하나의 필수격 조사처럼 취급한다는 것이다. 언어학적으로는 매우 어색할지 모르지만 전자계산학적인 견지에서 보면 이러한 경험지식은 상당히 유용하다. 이들을 하나의 격조사처럼 취급할 경우, 문장의 분석의 결과가 엄청나게 간결해 질 수 있다. 다음을 통하여 이 둘 사이의 차이를 비교하여 보자.

예문 2-1) 그는 정의를 위해 싸운다.

a. NP[는] NP[를] Sen[여] Sen[ㄴ다]
Sen -> Sen[suffix]Sen
(그는 (그는 정의를 위해) 싸운다)

b. NP[는] NP[를 위해] Sen[ㄴ다]
Sen -> NP[case] Sen[suffix]
(그는 정의를 위해 싸운다)

<그림 2.1> '을 위해'를 위한 문법 규칙

<그림 2.1> a.에서 보는 바와 같이 '을 위해 싸운다'를 언어학적인 견지에서는 내포문을 포함하는 복합문의 구조로 분석된다. 즉, 용언이 두 개이므로 주절의 용언 '싸우다'에 대한 문장소들과 부사절의 용언 '위하다'에 대한 문장소들이 필요하므로 두 개의 문장이 계층을 이루는 구문트리를 생성한다. 반면에 b.에서는 정의를 위해가 용언 '싸우다'의 하나의 문장소이므로 단순히 편평한 단문으로 분석된다.

2.1.2 문장 형태 확장

앞 절에 소개한 기본 문장 형태와 13개의 표층 문법 형태소의 결합으로 41개의 기본적인 문장구조를 형성한다. 이러한 문장 구조는 동아일보 사설, 조선일보 사설, 고등학교 국사교과서 등으로부터 무작위로 발췌한 25만 여 문장을 분석하여 얻은 정보를 문장 형태 확장에 사용하였다. 확장된 문장 형태를 몇 가지 살펴보면 다음과 같다.

NP[이] + AD[게] + V
NP[이] + NP[에] + V
NP[이] + NP[이] + A
NP[이] + NP[보다] + A
NP[이] + NP[에게] + NP[을] + V
NP[이] + NP[와] + NP[에 대해] + NP[을] + V
NP[이] + NP[와] + NP[에 대해] + NP[을] + V

이러한 정보를 통해 술어를 중심으로 단어들이 가지는 구조적 특성으로 분류하면 문법 구조가 명확해 질뿐만 아니라, 구문적으로는 가능하나 의미적으로 불가능한 구조를 제거할 수 있다. 예를 들면, '논다'라는 용언이 자동사이기 때문에 기존의 방법에 의하면 주어와 용언(자동사)만이 문장을 이루는 필수적 성분이다. 그렇지만 완전한 문장 "철수가 재미있게 논다"에서 하나의 문장소를 생략하고 보면 "재미있게 논다" 보다 오히려 "철수가 논다"가 더 어색한 문장이다. 즉 '논다'의 경우 반드시 필수적인 문장 요소로 "귀엽게"나 "재미있게" 같은 상태를 나타내는 부사가 존재해야 한다. 따라서 이러한 종류의 용언에서는 상태 형용사에서 부사로 전성한 것들을 필수적 문장 성분으로 인정해 주어야만 한다.

그리고 용언을 좀더 자세히 분류할 경우 구문적 반복 패턴이 존재한다. 예를 들면, "기원한다", "우러나온다"와 같은 용언(자동사)의 경우를 살펴보면 다음과 같다.

예문2-2) 양서류는 어류에서 기원한다.
예문2-3) 이런 감정은 민족적 자부심에서 우러나온다.

위의 예에서 "어류에서", "민족적 자부심에서" 등이 문장에서 생략된다면 자동사 문장에서 문장의 구조가 비록 옳바를 지라도 문장의 의미는 전혀 통하지 않는다. 그러나 이런 용언류는 반드시 "어디에서부터" 라는 뜻을 나타내는 명사구가 "에서"류의 격조사를 매개로 반드시 나타나야 한다. 본 논문에서는 이러한 경험정보들을 문법 규칙에 직접 추가하여 문장형태를 확장하였다.

2.2 조건단일화

단일화 문법을 이용한 파싱에 있어서 문법 수행의 선택 조건을 문법자체에 부착(attatch)시킴으로써 단일화를 제어[8]할 수 있는데 이를 조건단일화[1,9]라 한다. 이 방법의 특징은 문장의 파싱 시에 파서의 내부를 제어하는 방법을 탈피하여 문법을 기술하는 수준에서 직접 파싱 조건을 기술할 수 있다[1]는 것이다. 이 방법은 문법이 간단하고 각 문장소들 간의 자유도가 매우 높은 한국어의 파싱에 적합한 것으로 보인다.

```
S->NP V                                  문법래이블
    <X0 SUBJ> = <X1>   )
    <X0> = <X2>        )  명세(단일화)
```

<그림 2.2> 단일화 문법 기술의 예

단일화 문법 기술 도구인 PATRII에서는 문법이 <그림 2.2>에서 문법라벨의 아래에 명세부분을 부착시킨 형태로 기술되는데 명세의 X0는 S, X1은 NP, 그리고 X2는 V와 각각 대응된다. 또한 이곳에서 사용한 "="기호는 좌우항의 단일화를 의미한다. [1]에서는 완벽한 조건단일화를 실현하기 위하여 PATRII를 다음과 같은 기능을 추가하여 확장하였다.

① 명세의 이접(disjunction)기술 기능이다. 본 분석기에서는 이접기술의 기호로 '*OR*'를 쓴다. '*OR*'로 시작되는 명세에는 대등한 위치를 점하는 동일문의 집합을 나열할 수 있다. 이 기능을 통하여 규칙밖에 별도의 제어 기구를 두지 않아도 'IF(조건식) THEN S1 ELSE S2'의 기능을 수행하도록 명세를 기술할 수 있다.

② 동일성 검사기능 '=c'이다. '=c'는 LFG의 검사장치로 단일화 연산인 '='과는 달리 좌우항 자질구조 간의 완전한 단일화는 일어나지 않으며 단지 좌우항의 동일성만 검사해서 같으면 참, 다르면 거짓이 된다. 이것은 무조건적인 단일화에 의한 오류를 막기 위해 사용한다.

③ 존재여부 검사기능이다. 이 기능은 특정 속성의 값이 이미 존재하는 지의 여부를 판단하기 위해 사용된다. 이 기능은 특정 자질구조 안의 임의의 속성의 존재 여부를 동적으로 검사하기 위하여 도입하였다. 이 기능을 위한 기호는 '*EXIST*'와 이것의 부정인 '*NONEXIST*'를 사용한다.

④ 부가기능이다. 등위문의 경우 두개의 대등한 문장을 표현할 때 단일화가 아닌 부가 연산을 통해 해결한다. 이 기능을 위한 연산자로 '>'를 쓴다.

3. 문장 형태 정보를 이용한 문장 분석

다음은 문장 형태 정보를 이용한 구문 분석틀에서 어떠한 문법 규칙이 필요한 지를 살펴보자.

;; 단문 처리 규칙(개념)

```
Sen2[suffix] -> NP[문법형태소] Sen1[suffix]
      Sen1이 어떤 문형 범주에 포함되는가?
      NP의 문법 형태소가 어떤 형태인가?
```

;;단문 처리규칙(알고리즘)

```
Sen2 -> NP Sen1
        If Sen1 cat = V1
        If NP case = 이/가
            Sen2.subj = NP, Sen2 = Sen1
        If Sen1 cat = V2
        If NP case = 이/가
            Sen2.subj = NP, Sen2 = Sen1
        If NP case = 에
            Sen2.loc = NP , Sen2 = Sen1
        If Sen1 cat = V3
            ...
```

<그림 3.1> 한국어 단문 처리용 기본 문법의 개념과 알고리즘

<그림 3.1>에서는 한국어의 단문을 파싱해 내기 위한 문법 규칙과 이를 제어하기 위한 조건들이 표시되어 있다. 이 그림의 상반부에는 한국어 단문에 대한 문법 규칙의 개념을 보여주고 하반부에는 이를 규격화한 형태를 보여준다. 우리의 시스템에서는 이러한 개념과 대응하는 실제 문법 규칙을 <그림 3.2>와 같은 형태의 규칙을 기술하여 사용하였다.

```
(<s><==>(<np><s>)
  ;NO1 + V
  ; 새가운다
  (
     (*OR*(((X2 cat) =c v1)
              (((x1 p-part root) = c (*or*이 가 은
                                      는 에서 께서))
                    (xo = x2)
                    ((x0 subj) = x1))))
  ;N이 + N에 + V
  ;그는 무명도에 정착했다.
          (((x2 cat) = c v2)
  (*or*
  (((x1 p-part root)=c(*or* 이 가 은 는 에서 께서))
  (x0=x1)
  ((x1 subj)=x1)))
```

```
(((x1 p-part root)=c(*or*에 에게 께))
(x0=x2)
((x0 loc)=x1))))
;N 이 +N로 + V
;그는 반장으로 선출되었다
(((x2 cat) =c v3)
(*or*
(((x1 p-part root)=c(*or* 이 가 은 는 에서 께서))
(x0 = x2)
((x1 p-part root)=c(*or*로 으로))
(x0 = x2)
((x0 tar) = x1))))

;N이 + N와 + v
;나는 영수와 다투었다
(((x2 cat) =c v4)
(*or*
(((x1 p-part root) = c(*or* 이 가 은 는 에서 께서))
(x0 = x2)
((x0 subj)= x1))
(((x1 p-part root = c (*or* 와 과))
(x0 = x2)
((x0 with) = x1))))

;N이 + N에서 +v
;그는 학회에서 탈퇴하였다
(((x2 cat) = c v5)
(*or*
(((x1 p-part root)=c(*or*이 가 은 는 께서))
(x0 = x2)
((x0 subj)=x1))
(((x1 p-part root) = c (*or*에서 를 을))
(x0=x2)
((x0 sour)=x1))))
;....
)))
```

<그림 3.2> 한국어 단문 파싱을 위한 문법 규칙

<그림 3.2>에 기술된 하나의 문법 규칙 아래에 부착되는 명세에는 총 41개의 조건이 부가되어있다. 이렇게 기술된 문법이 실제의 입력 문장에 대해 문형 기반의 구문 분석을 행하는 과정을 살펴보자.

예문3-1)

a. [그는(subj)] [서점에서(loc)] [책을(obj)] 구입했다(V15).
(He bought a book at the book store.)

b. [호랑이는(subj)] [사자와(with)] 비교된다(V10).
(Tiger is compared with lion.)

분석결과 트리

```
a. ((subj (n-part ((root 나) (cat N)))
          (p-part ((root 는) (cat p))))
    (loc (n-part ((root 서점) (cat N)))
         (p-part ((root 에서) (cat p))))
    (obj (n-part ((root 책) (cat N)))
         (p-part ((root 을) (cat p))))
    (cat V15 )
    (root 구입하 )
    (tense past)
    (voice passive)
    (mood dec))

b. ((subj (n-part ((root 호랑이) (cat N)))
          (p-part ((root 는) (cat p))))
    (with (n-part ((root 사자) (cat N)))
          (p-part ((root 과) (cat p))))
    (cat V10 )
    (root 비교되)
    (tense present)
    (mood dec))
```

<그림 3.3> 예문3-1)의 파싱 결과

위의 파싱 결과를 살펴보면, 문장 a.는 <그림 3.2>의 문법에서 서술어의 범주(V15)가 적용되어 <그림 3.3>의 a.와 같이 평면 구조의 단문으로 분석됨을 볼 수 있다. b.도 서술어의 범주(V10)이 적용되어 그림 3.3의 b.와 같은 구문 트리를 생성한다. 특히 b.의 예에서 "비교되다"는 자신의 자질에 이미 수동형(passive) 값을 가지고 있다. 이 예문을 구문 분석한 결과에서 볼 수 있듯이 문형 정보를 포함한 문법 체계는 비교적 간단하다. 왜냐하면, 분석의 주체인 서술어 자체가 어떠한 문법 구조를 요구하는 지에 대한 정보를 이미 가지고 있기 때문이다. 또한, 예문 3-1)의 분석 결과에서 보는 바와 같이 이제까지 수의격으로 분리되어 처리외의 대상으로 취급하던 자질 값(loc)을 정확하고 쉽게 얻어 올 수 있으므로 사전의 용언에 대한 하위 범주화만 완전히 이루어진다면 파싱의 결과가 의미

구조로 직접 사상될 수 있다.

4. 실험 평가

본 논문에서 소개한 파싱 방법은 [7]의 방법을 확장하여 수정한 것이며, Windows ME하에서 Visual C++ 6.0을 사용하여 구현하였다. 실험을 위한 말뭉치는 23,730 텍스트 문장과 여행 정보 제공 영역의 예문을 토대로 발화된 문장을 대상으로 음성인식기에서 출력된 527발화, 10,200문장을 사용하였다. 특히 후자는 간투사나 연속되어 나오는 의미있는 단어들을 제거한 후에 사용하였다. 분석결과 트리는 리스트 구조를 이용하여 표현하였는데 이는 일반 트리와는 달리 매우 복잡한 구조를 갖는 파싱 트리를 제한된 화면상에 일목요연하게 표시하는 가장 이상적인 방법이기 때문이다. 이 실험의 결과는 <표 4.1>과 같다.

<표 4.1> 실험 결과

비 고	문장수	분석오류	평균파스트리수
텍스트	23,730	9(6)	8.6(9.4)
음성인식결과	10,200	210(469)	5.1(10.1)

()안은 [1]방법을 이용한 결과

실험의 결과에서 보는 바와 같이 정형화된 문장(텍스트)에 대해서 분석오류가 기존의 분석 방법을 사용했을 대 보다 다소 많아진 것을 볼 수 있었다. 또한, 모호성의 개수인 평균 파스 트리 수도 음성인식 결과 예문의 결과와 비교했을 때 기존의 방법에 비해 상대적으로 많은 향상 효과를 보이지 못했다. 그 이유는 본 논문에서 시도한 분석이 단지 조사에서 얻을 수 있는 정보만을 가지고 용언과 명사의 관계 분석에 주안점을 두었기 때문이다. 이에 반해서 대화의 음성인식의 결과를 대상으로 한 실험에서는 분석오류나 평균 파스트리수가 상당히 낮아진 것을 볼 수 있었다. 대화환경에서는 발화자가 완전한 문장을 구사하기 위하여 문법에서 요구하는 모든 문장소를 사용하여 발화하지는 않는다. 따라서 대화체의 문장은 문어체 텍스트 문장에 비하여 매우 비정형적일 수 밖에 없다. 이러한 요인에 의하여 대화체 파싱의 결과가 산출된 것이다. 이를 토대로 생각해 볼 때, 문장 형태를 이용한 파싱은 잘 정제되지 않은(문법에서 요구하는 구문적인 요소를 완벽하게 갖추지 않은) 대화체의 문장에서 더욱 좋은 성과를 낼 수 있음을 알 수 있다.

5. 결론 및 향후 연구 방향

본 논문에서는 분석의 초기에 한국어의 문장 형태 특성 정보를 적극적으로 활용함으로써 보다 정확하고 간단한 결과를 산출하는 파싱 방법을 기술하였다. 이를 위해서 경험적 지식을 가미해 41개의 문장 형태를 정하였다. 문장 형태를 결정하는 데에 추가된 경험적 지식은 각 일간지 사설 등에서 무작위로 발췌한 25만 여 문장을 분석한 결과에서 얻어진 것이다.

실험을 통하여 얻어진 결론은 다음과 같다. 1)문장 형태를 이용한 파싱을 통해 간단한 결과 트리를 얻을 수 있으며 이는 다음 단계의 파싱에 영향을 미쳐 구문적 모호성을 많이 배제할 있다. 2)한국어에서 조사의 역할이 상대적으로 커서 문형 분류가 의미론적 특성을 상당 부분 포함할 수 있으므로 용언 중심의 하위 범주화 사전이 완벽하게 구비된다면 의미 구조로의 직접 사상이 가능하다. 3)문법이 잘 정리된 형태의 파싱 틀을 제공하고 있으므로 정제되지 않은 입력 문장에 대해서도 강건한 파싱을 수행할 수 있다. 4)부사의 의미를 내포하고 있는 문장소들에 대한 정확한 분석은 문맥의 유지에 매우 지대한 영향을 미칠 수 있으므로 대화체 번역에서의 발화 매니저에게도 보다 풍부한 정보를 제공할 수 있다.

이러한 장점을 갖는 문장 형태 기반 파싱을 위해서는 사전의 엔트리 중 용언의 범주가 세분화되어야할 것이다. 또한, 본 논문에서 분류하여 사용한 문장 형태의 분류가 보다 세분화되어야 할 것이다. 현재 우리는 사전의 용언 엔트리의 자질들의 변경을 90% 이상 완료하였고 문장의 형태에 보다 많은 경험 정보를 포함시키기 위해서 명사의 의미 분류를 계속하고 있다.

참 고 문 헌

[1] 양승원, 조건 단일화 기반 PATRII를 이용한 한국어 구문분석, 전북대학교 박사학위논문, 1995.

[2] 서영훈, 의미 정보를 이용하는 중심어 주도의 한국어 파싱, 서울대학교 박사학위 논문, 1991.

[3] 서광준, 최기선, "어절사이의 펴지 의존 관계를 이용한 한국어 파서에 관한 연구," 한국 정보과학회학술 발표 논문집, pp.1151-1154, 1993.

[4] 양재형, 김영택, "다중 지식원을 이용한 한국어 분석," 한국정보과학회 논문지, Vol. 21, No. 7, 1994.

[5] Tomita M., Efficient Parsing for Natural Language, Kluwer Academic Publishers, 1986.

[6] 강은국, 조선어 문형 연구, 도서출판 박이정, 1994.

[7] 양승원, 이종석, "문형정보를 이용한 한국어 분석," 우석대학교 자연과학 연구지, pp.12-22, 1996.

[8] Shieber, S. M.,lntroduction to Unification-Based Approaches to Grammer, CSLI Lecture Notes, 1986.

[9] Koiti Hasida., "Conditioned Unification for NLP," Coling 86, pp.85-87, 1986.

양 승 원 (Seungweon Yang)

우석대학교 컴퓨터공학과 부교수
한국전자통신 연구원 초빙연구원
University of Guelph 방문교수
(관심분야 : 자연언어처리, 바이오 인포믹스)