

구문 분석의 결과로 나타나는 구조의 모호성을 해결하기 위한 방법 연구

박용욱*

¹울산과학기술대학교 컴퓨터정보학부

A Study of Disambiguation Method To Improve The Syntactic Analysis System

Yong Uk Park^{1*}

¹School of Information Technology, Ulsan College

요약 본 논문에서는 형태소 단위의 입력요소에 대해 의존규칙을 적용하여 가능한 모든 구문분석 구조를 생성할 수 있는 알고리즘을 적용한 구문분석기를 구현하였다. 따라서 형태소의 수가 증가함에 따라 생성되는 구문분석 트리의 개수가 기하급수적으로 증가할 수 있다. 이를 해결하기 위하여 본 논문에서는 주어진 문장에 대해 문장의 기본 구성요소들에 대하여 구간을 나누고, 나누어진 구간에서 최대연결단위를 구성하였다. 최대연결단위는 한 구간에 존재하는 모든 형태소가 결합된 것을 의미한다. 입력되는 문장에 따라서 한 구간에 하나 이상의 최대연결단위가 구성될 수 있다. 중학교 교과서에서 임의로 추출한 10어절 이내의 516문장에 대하여 실험한 결과 제안한 방법을 사용하지 않은 방법에 비해 약 28%의 개선효과가 있었다.

Abstract In this paper, we present a Korean syntactic analysis system which can generate all possible syntactic trees in a given sentence. Therefore, the number of syntactic trees by this syntactic analysis system can be increased exponentially. To solve this problem, we suggest a segmentation method and maximum connected unit in a segmentation. Maximum connected unit is a combined unit which contains all morphemes in a segmentation. According to the input sentence, it is possible one or more maximum connected unit in a segmentation. We extract 516 sentences to experiment randomly from the text book of Korean middle school. We could reduce about 28% of the number of syntactic trees.

Key Words : maximum connected unit, segmentation, syntactic tree

1. 서론

1980년대 말부터 한국어 구문분석에 대한 연구가 꾸준히 진행되어 오고 있고, 이를 통해 어느 정도의 성과를 거두고 있다. 영어의 경우 현재 구문분석이 많이 연구되어 영어를 다른 언어로 번역할 수 있는 시스템들이 비교적 많이 개발되어 있고, 만족할 만한 성과를 내고 있다. 그러나 한국어는 영어에 비해 구문분석

에 어려움을 주는 여러 가지 특징을 가지고 있다[1]. 한국어는 어순이 비교적 자유롭고 주어나 목적어와 같은 중요한 문장성분이 문맥상으로 이해할 수 있을 때 생략될 수 있기 때문에 문장 분석에 어려움을 겪어왔다. 한국어 구문분석에 대한 연구 초기에는 영어에서 사용되었던 구 구조문법(phrase structure grammar)을 많이 사용하였다. 구 구조문법은 영어와 같이 문장 성분의 기능이 위치에 따라 비교적 고정되어 있는 언어

*Corresponding Author : Yong Uk Park(Ulsan College)

Tel: +82-10-9663-2389 email: yupark@uc.ac.kr

Received December 5, 2014

Revised (1st February 2, 2015, 2nd March 12, 2015, 3rd March 26, 2015)

Accepted April 9, 2015

Published April 30, 2015

에는 잘 맞지만 한국어와 같이 조사(postposition)가 발달되어 있어서 문장성분의 위치가 비교적 자유로운 언어에는 다소 어려움이 따른다[1,2,3].

따라서 2000년대 초부터는 많은 한국어 처리 연구자들이 우리말의 특성을 비교적 잘 반영하여 구문분석을 할 수 있는 의존문법(dependency grammar)을 사용하고 있다[3,4,5]. 의존문법은 이웃하는 두 문법 요소에 대해 이진관계를 표현함으로써 어순이 자유로운 현상이나 문장요소의 생략 현상에 비교적 강건(robust)하다[3,5]. 이때 두 문법요소 사이에는 지배소(governor)와 의존소(dependent)라는 관계를 가진다. 예를 들어 “아름다운 동생”에서 “동생”이 지배소가 되고 “아름다운”은 의존소가 된다. 이러한 의존문법(dependency grammar)은 그 단순성으로 인하여 한국어의 특성을 잘 반영할 수 있다. 그러나 단순성 때문에 많은 중의성을 발생시킬 수 있다는 것이 문제이다[3,4,5].

한국어는 하나의 어절이 여러 개의 형태소로 분석이 될 수 있다. 지금까지 의존문법을 사용하여 개발한 한국어 구문분석기는 대부분 형태소 분석 후 품사태거를 통해 가장 정확하다고 판단된 한 개의 결과를 선택하여 이를 기본단위로 사용한다[4]. 그러나 본 논문에서는 품사태거를 사용하지 아니하고, 형태소 분석결과로 얻어진 모든 형태소 분석 후보에 대하여 통사적 제약규칙을 적용하여 모든 가능한 분석 구조를 찾아낸다. 따라서 품사태거를 사용하는 구문분석기보다 많은 구문분석 구조를 찾아내게 된다. 입력 문장의 길이에 따라서 하나의 문장이 수백 개의 분석구조로 분석되기도 한다. 이러한 중의성 문제를 해결하기 위하여 본 논문에서는 구간분할방법 및 구간내의 최대연결단위를 통하여 구문분석의 결과를 개선하는 방법을 소개한다.

본 논문의 구성은 다음과 같다. 2절에서 관련연구를 설명하고 3절에서는 시스템 구조 및 분석과정에 대하여 설명한다. 4절에서 실험 및 결과를 설명하고 5절에서 결론을 맺는다.

2. 관련 연구

1990년대 초에 의존문법을 이용한 한국어 구문분석에 대한 연구가 시작되었고[3], 2000년대 들어서는 여러 개의 구문분석 시스템이 개발되어 발표되었다[4,5]. 크게 두 가지의 방법이 주를 이루는데, 수작업에 의해

구축한 통사 규칙에 기초한 방법과 대규모 코퍼스를 이용하여 추출한 통계적 정보를 이용한 방법으로 나누어진다[1,2]. 통계적 정보를 사용하기 위해서는 정제된 대규모 말뭉치를 기반으로 구현되는데, 아직 국내에서는 대규모로 구축된 말뭉치에 대한 검증이나 보완에 대한 연구가 미흡한 실정이다. 통계적 정보를 이용한 방법에는 양[7]이 있다. 통사규칙에 기초한 연구에는 박[4], 김[8], 황[9]이 있다. 이들 연구에서는 구 묶음 기법(chunking)을 사용하였는데, 박[4]은 의존명사를 구 묶음으로 처리하여 구문분석의 결과로 나타나는 파스트리에 중의성을 제거하도록 하였다. 김[8]는 서술어절과 기능어절을 묶는 규칙을 사용했다. 황[9]은 본용언과 보조용언의 묶음 규칙을 사용했다. 박[11]은 구간분할 방법을 사용하였는데, 구간을 나누는 기준을 세 가지로 하고 있다. 관형형어미를 갖지 않은 용언 바로 다음에서 나누고, “~때”, “~인 이유로” 등 이유, 시간 등을 나타내는 구 바로 다음에서 나눈다. 그리고 “~한 김에”, “~할 시에” 등에 나타난 “김,시,양,지” 등과 같은 의존 명사 어절 바로 다음에서 구간을 분할하는 방법을 사용했다.

통사규칙에 기반을 둔 대부분의 한국어 구문분석 시스템들은 품사태거를 사용하고 있다. 또한 분석의 결과로 단 하나의 분석 구조를 나타내고 있다[4,10]. 자연언어에는 문법규칙에 의해 분석될 때 모호성(ambiguity)이 내재되어 있을 수 있기 때문에 기본적으로 모호성을 내재하고 있는 문장에 대해서는 모호성이 있는 그대로 분석결과로 나타내 주어야 한다[1,2].

본 논문에서는 품사태거를 사용하지 않고 전처리를 거친 모든 형태소 분석의 결과에 대하여 가능한 모든 분석 구조를 찾아준다. 따라서 주어진 문장에 내재된 중의성을 결과로 나타내어 준다. 의미적으로는 중의성이 없는데, 통사적으로 중의성이 존재하는 경우가 많기 때문에 이것을 제거할 수 있는 방법이 필요하다. 본 논문에서는 구간분할방법 및 구간내의 최대연결단위를 통하여 구문분석의 결과로 나타나는 많은 구문분석 트리의 수를 감소시킬 수 있는 방법을 제안한다.

3. 시스템 구성 및 분석과정

3.1 구문분석 시스템 구조

본 논문에서 사용하는 구문분석 시스템의 전체적인

구성은 Fig. 1와 같다. 어휘 분석기의 결과로 나타나는 여러 개의 형태소들에 대하여 형태소 후보 리스트가 만들어지고 이것을 구문분석기의 기본적인 입력요소로 사용한다. 하나의 어절이 여러 개의 형태소로 분석 가능하기에 한 문장에서 가능한 형태소들의 조합은 여러 개의 리스트로 생성될 수 있다. 예를 들어 “학교에 가는 영화”에서 어절 “가는”이 “가다(자동사)/가는다(형용사)/갈다(타동사)+-(관형형전성어미)”로 분석 가능하기 때문에 전체의 형태소분석 후보 리스트는 3개가 된다. 전처리단계에서는 문법적으로 결합해도 문제가 되지 않는 형태소들은 서로 결합하도록 한다. 본 논문에서 사용하는 구문분석기는 기본적으로 주어진 형태소들에 대해서 가능한 모든 구문분석 트리를 생성한다. 어떤 입력문장에 대해서는 수백 개의 완성된 구문분석 트리가 생성되기도 한다. 이것은 문법적으로는 분석이 맞지만 의미적으로는 틀린 분석이 포함되어 있다는 것을 의미이다. 잘못 결합되어 생성되는 구문트리의 수를 감소시키기 위하여 구간분할 방법 및 구간 내의 최대연결단위를 사용한다.

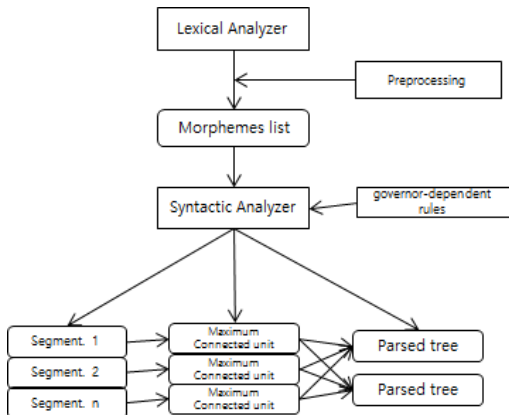


Fig. 1. The flow of Syntactic Analysis System

3.2 구문분석 알고리즘

본 논문에서는 사용하는 구문분석의 전체적인 알고리즘은 다음과 같다.

- 1) 어휘분석기를 통하여 형태소후보리스트 생성
- 2) 각 형태소후보리스트에 대하여 왼쪽에서 오른쪽으로 입력
 - 2-1) if(입력형태소가 주격조사/목적격조사/부사 성분에 속함)

{새로운 구간을 생성하고, 이전 사용 중인 구간에 대하여 최대연결단위를 찾아 구성한다. }

else if(입력형태소가 서술어에 속함)
 {서술어 형태소를 앞의 구간에 있는 최대 연결단위와 결합을 시도한다}

else {형태소간의 결합 rule에 의해 현재 구간 내에서 결합하여 결합된 형태소 생성}

- 3) 2)단계를 각 형태소 후보리스트에 대하여 형태소가 끝날 때 까지 반복함
- 4) 각 구간에 대하여 결합하여 완성된 트리를 생성함

구간분할의 기준으로 문장의 기본 구성요소를 사용한다. 문장의 기본 구성요소에는 주어, 목적어, 부사, 관형어, 술어 등이 있다. 그러나 본 논문에서는 술어에 직접적으로 의존관계를 맺을 수 있는 주어, 목적어, 부사어를 사용하여 문장 분할을 실시한다. 구문분석이 진행되는 동안 여러 개의 구간이 생성된다. 각 구간에는 형태소 후보 리스트에 있었던 원래 형태소와 이들이 결합 규칙에 의해 결합된 복합 노드들이 있게 된다. 이때 하나의 구간 내에 존재하는 모든 형태소들이 결합되어 복합 결합노드를 생성할 수 있는데 이것을 최대연결단위라 부른다. 이러한 최대연결단위가 최종적으로 완성트리를 구성하는데 사용된다.

3.3 구문분석 과정

본 논문은 형태소를 구문분석의 기본적인 단위로 사용하고 의존규칙에 의하여 주어진 문장에서 가능한 모든 분석 구조를 찾아내는 방법을 사용한다. 따라서 구문분석의 결과로 다수의 구문구조트리를 생성하게 된다. 다수로 출력되는 구문 구조트리는 구문의 모호성(ambiguity)를 의미하며, 이를 제거하기 위하여 본 논문에서는 구간분할과 분할된 구간 내에서 최대로 연결된 최대연결단위를 사용한다.

본 논문에서는 구문분석을 진행하면서 만나는 형태소가 주어조사, 목적어조사, 부사 성분에 속하면 현재까지 진행된 형태소까지 하나의 구간으로 처리하고, 그 이후 형태소에 대해서는 새로운 구간을 생성하여 구문분석을 계속하게 한다. 그리고 어떤 형태소가 서술어에 속하면 이전에 만들어져 있는 구간과 서술어 형태소의 결합을 시도한다. 예문(1)에 대하여 분석되는 과정을 설명하면 다음과 같다.

(예문1)철수가 예쁜 순이의 동생을 사랑한다.

먼저 형태소 분석을 하면, (철수/명사)+(가/주격조사)+(예쁘/형용사)+(ㄴ/관형형어미)+(순이/명사)+(의/관형격조사)+(동생/명사)+(을/목적격조사)+(사랑하/동사)+(ㄴ다/종결어미) 와 같이 분석된다. 이렇게 분석된 형태소리스트에 대하여 알고리즘을 적용하여 분석되는 과정을 보면 다음과 같다.

입력리스트 : (철수/명사),(가/주격조사),(예쁘/형용사),(ㄴ/관형형어미),(순이/명사),(의/관형격조사),(동생/명사),(을/목적격조사),(사랑하/동사)+(ㄴ다/종결어미)

- 1) [입력형태소] : (철수)
[구간1] : (철수)
- 2) 입력형태소 : (가)
[구간1]: (철수),(가),(철수(가))

여기서 (가)가 주격조사이므로 새로운 [구간2]을 생성하고, 기존 구간인 [구간1]은 모든 형태소가 연결된 의미 최대연결단위인 ((철수(가))만 남기고 나머진 제거한다.

- 3) [입력형태소] : (예쁘)
[구간2] : (예쁘)
- 4) [입력형태소] : (ㄴ)
[구간2] : (예쁘),(ㄴ),(예쁘(ㄴ))
- 5) [입력형태소] : (순이)
[구간2] : (예쁘),(ㄴ),(예쁘(ㄴ)),(순이),(((예쁘(ㄴ))(순이))
- 6) [입력형태소] : (의)
[구간2] : (예쁘),(ㄴ),(예쁘(ㄴ)),(순이),(((예쁘(ㄴ))(순이)),(의),((순이)(의),(((예쁘(ㄴ))(순이))(의))
- 7) [입력형태소] : (동생)
[구간2] : (예쁘),(ㄴ),(예쁘(ㄴ)),(순이),(((예쁘(ㄴ))(순이)),(의), ((순이)(의)),(((예쁘(ㄴ))(순이))(의)),(동생) (((순이)(의))(동생),(((예쁘(ㄴ))(순이))(의))(동생)),

((((예쁘(ㄴ))(순이))(의))(동생))

- 8) [입력형태소] : (을)
[구간2] : (예쁘),(ㄴ),(예쁘(ㄴ)),(순이),(((예쁘(ㄴ))(순이)),(의), ((순이)(의)),(((예쁘(ㄴ))(순이))(의)),(동생) (((순이)(의))(동생),(((예쁘(ㄴ))((순이)(의))(동생))),((((예쁘(ㄴ))(순이))(의))(동생) ((동생)(을),((((순이)(의))(동생)(을) (((예쁘(ㄴ))((순이)(의))(동생))(을),((((예쁘(ㄴ))(순이))(의))(동생)(을))

그리고 (을)이 목적격조사이므로 새로운 [구간3]을 생성하고, 기존 [구간2]는 최대연결단위인 “((((예쁘(ㄴ))(순이)(의))(동생))(을)”와 “((((예쁘(ㄴ))(순이)(의))(동생)(을))”만 남기고 나머지는 제거하여 정리한다.

- 9) [입력형태소] : (사랑하)
(사랑하)는 서술어에 속하므로 앞의 [구간2] 및 [구간1]의 최대연결단위와 결합을 시도한다. 최종적으로 다음과 같이 정리된다.

[구간1]:((((예쁘(ㄴ))(순이)(의))(동생))(을)(사랑하), [((((예쁘(ㄴ))(순이)(의))(동생)(을)(사랑하))

- 10) [입력형태소] : (ㄴ다)
[구간1]:
{((철수(가))
{((((예쁘(ㄴ))(순이)(의))(동생))(을)(사랑하))(ㄴ다)},
{((철수(가))
{((((예쁘(ㄴ))(순이)(의))(동생))(을)(사랑하))(ㄴ다)} }

최종적으로 [구간1]에 완성된 분석트리가 남게 된다. [구간1]을 보면 2개의 완성된 분석트리가 존재하고 있는데, 첫 번째 분석트리는 “순이의 동생이 예쁘다”는 의미로 분석된 구조이고 두 번째 분석트리는 “순이가 예쁘다”는 의미로 분석된 구조이다. Fig. 2는 (예문 1)의 처리 결과를 보여준다.

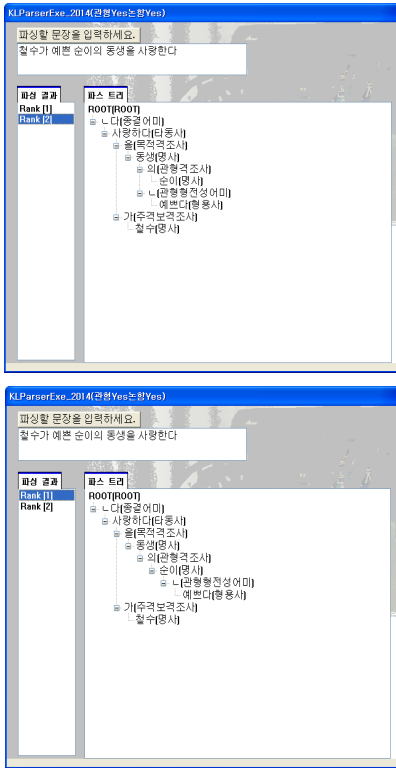


Fig. 2. The Parse Trees of Example 1.

4. 실험 및 결과

본 논문에서 제안한 시스템은 품사태거를 사용하지 않고 형태소 분석결과로 얻은 모든 형태소 분석후보에 대하여 통사적 제약규칙을 적용하여 가능한 모든 분석구조를 찾아낸다. 따라서 기존의 박[4], 김[8], 박[11] 등의 연구에서 제시한 정확도와 재현율을 구하여 단순히 비교하는 것은 무리가 있다. 본 논문에서는 구간 분할 및 구간 내에서의 최대연결단위가 구문분석에 얼마나 영향을 미치는지를 평가하기 위해 문장을 기준으로 평균 구문분석 트리의 수를 비교하고, 분석 결과트리에 정답트리가 얼마나 존재하는가를 실험하였다.

실험에 사용한 데이터는 중학교 교과서에서 추출한 10어절 이내의 516문장이다. 본 논문은 어절이 아니라 형태소를 기본단위로 사용하기 때문에 실제로 사용한 문장의 어절은 10어절 이내이지만 형태소 분석의 결과로 나타난 형태소의 수는 문장에 따라 차이가 있으나 대략 30개 정도이다. 본 논문의 시스템은 현재 약 207개의 의존규칙을 사용하고 있으며, 주어진 입력

문장에 존재하는 모든 의존구조를 출력할 수 있는 알고리즘을 사용하기 때문에 많은 분석결과 트리를 출력한다. 구문분석 결과로 출력되는 잘못된 분석구조들을 감소시키기 위하여 구문분석 과정 중에 구간을 나누고, 나누어진 구간에서 최대연결단위를 구성하여 최종 구문 분석트리 생성하도록 하였다. 결과를 평가하기 위하여 권[3]에서 적용한 입력문장에 존재하는 모든 구문구조를 찾아내 주는 방법과 구간분할 및 논항정보까지 적용한 박[6] 방법에 대하여 다음의 3가지 측면에서 비교 하였다. 첫째, 생성되는 평균 파서트리의 개수, 둘째, 생성된 분석트리 중에 정확한 분석트리를 얼마나 포함하고 있는지, 셋째, 가장 많은 분석트리의 수는 몇 개인지를 실험하였다. 실험에 대한 결과는 Table 1과 같다.

Table 1. Comparison of Experiment Results

	Kwon method	Park method	segmentation & maximum connected unit
average parsed trees	9.1	4.8	6.6
correctly parsed sentences	487/516	425/516	480/516
maximum parsed trees	368	287	310

실험결과에서 볼 수 있듯이 30형태소 이내의 516문장에 대한 평균 구문분석 결과 트리 수는 구간분할 및 최대연결단위를 사용하지 않은 권[3] 방법 9.1개이고 본 논문의 방법은 6.6개로 약 28% 감소하였다. 또 구문분석에 성공한 문장의 수는 구간분할 및 최대연결단위를 사용하지 않은 권[3] 방법이 516문장 중에서 487문장을 성공했고, 제안한 시스템은 480문장을 성공했다. 이는 서로 다른 구간에 있는 구성요소들 간의 결합의 제한으로 인하여 분석에 성공하지 못하는 경우가 발생함을 보여준다. 차후 이것에 대한 연구를 계속할 필요를 보여준다. 그리고 최대 분석트리 수는 기존 방식에서 368개이고, 제안한 시스템에서는 310으로 감소하여 제안한 시스템에서 개선되었음을 볼 수 있다. 박[6] 방법과 비교해 볼 때 평균 구문분석 트리 수는 1.8개 많게 나왔다. 구간분할방법과 동시에 논항구조 방법까지 사용하는 것이 평균 구문분석 트리 수를 더욱 감소시킬 수 있음을 볼 수 있고, 두 가지 중에서는 구간분할의 영향이 더 크게 작용함을 보여준다. 정확하게 분석된 트리를 포함하는 문장의 수에서는 박[6] 보다 제안한 방법이 더 좋게 나타났다.

5. 결론

본 논문에서는 어절이 아닌 형태소를 기본적인 입력 요소로 하고 의존문법에 근간한 의존 규칙을 적용한 구문분석기를 소개하였다. 본 논문에서 제안한 시스템은 의존규칙을 사용하여 가능한 모든 분석 구조를 찾아내는 알고리즘을 사용하였기에 형태소의 개수가 많아질수록 구문분석 트리의 수가 급격히 증가한다. 이를 해결하기 위하여 문장을 분석하면서 구간을 분할하여 구간 단위로 분석하고, 구간 내에서 모든 형태소가 연결된 최대연결단위를 찾아내고 이것을 이용하여 전체 문장의 구조를 분석하는 방법을 제안하였다. 실험결과 약 28%의 개선효과가 있음을 보였다. 하지만 서로 다른 구간간의 제한적인 결합에 의해 문장이 분석되지 못하는 경우가 발생하는 경우도 있었다. 차후 구간을 분할하고 구간간의 결합관계에 대해 보다 더 깊이 있는 연구가 필요하다. 또한 한 문장의 형태소수가 30개 이상 되는 문장에 대한 실험을 통해 문제점들을 발견하고 시스템의 성능을 개선할 필요가 있다.

References

- [1] D. R. Ra, "A Study about Korean Language Parsing", Journal of KIISE, vol.12, No.8, pp33-46, 1994.
- [2] Y. T. Kim "Natural Language Processing," SaengRung Press, 2001.
- [3] H. C. Kwon, J. Y. Choi, "A Korean Parser Using Dependency Grammar based on Unification," Journal of KIISE : Software and Application, vol.19, No.5, pp.467-476, 1992.
- [4] E. K. Park, D. Y. Ra, "Processing Dependent Nouns Based on Chunking for Korean Syntactic Analysisism." Korean Journal of Cognitive Science, vol.17, no.2, pp.119-138, 2006.
- [5] G. E. Im, Y. I. Jung, H. C. Kwon, "Implementation of Dependency Parser using Argument Information based on Korean WordNet," Proc. of the 19th Human and Cognitive Language Technology, vol.24, no.1, pp.158-164, 2007.
- [6] Y. U. Park, H. C. Kwon, "A Study of Paring System Implementation Using Segmentation and Argument Information", Journal of Korea Multimedia Society Vol. 16, No. 3, pp366-374, 2013.

DOI: <http://dx.doi.org/10.3346/jkms.2001.16.3.366>

- [7] J. H. Yang, Y. T. Kim, "Korean Analysis using Multiple Knowledge Sources," Journal of KIISE, vol.21, No.7, pp.1324-1332, 1994.
- [8] C. J. Kim, C. Y. Jung, Y. H. Seo, "An Efficient Korean Syntactic Analyzer Using Partial Combination of Words," Proc. of the KIISE Autumn Conference, vol.22, no.2(A), pp597-600, 1995.
- [9] Y. G. Hwang, H. Y. Lee, Y. S. Lee, "Using Syntactic Unit of Morpheme for Reducing Morphological and Syntactic Ambiguity", Journal of KIISE : Software and Application, vo.27, no.7, pp784-793, 2000.
- [10] Y. H. Lee, J. H. Lee, "Korean Dependency Parsing Using Online Learning," Proc. of the KIISE Korea Computer Congress 2010, vo.37, no.1(C), pp299-304, 2010.
- [11] E. K. Park, M. H. Cho, S. W. Kim, D. Y. Ra, "A Method for Extracting Dependency Relations Using Chunking and Segmentation", Proc. of the 16th Human & Cognitive Language Technology, Vol.16, no.1, pp131-137. 2004.

박 용 욱(Yong-Uk Park)

[정회원]



- 1991년 2월 : 부산대학교 전자계산학과 (이학석사)
- 1991년 3월 ~ 1997년 2월 : 전자부품연구원 연구원
- 2000년 2월 : 부산대학교 전자계산학과 (박사수료)
- 1998년 3월 ~ 현재 : 울산과학기술대학교 컴퓨터정보학부 교수

<관심분야>

자연언어처리, 정보검색, 문서처리