2단계 구문분석을 이용한 구문분석 말뭉치 구축도구

김혜겸[0] 박경미[+] 윤여찬[+] 임해창[+] 박소영[++]
[0,+]: 고려대학교 컴퓨터학과 자연어처리 연구실
{ hkkim, kmpark, ycyoon, rim }@nlp.korea.ac.kr
[++]: 상명대학교 소프트웨어학부
ssoya@smu.ac.kr

Tree Tagging Tool using Two-phrase Parsing

Hye-Kyum Kim[0] Kyung-Mi Park[+] Yeo-Chan Yoon[+] Hae-Chang Rim[+]
So-Young Park[++]
[0,+]: Dept. of Computer Science and Engineering, Korea University
[++]: College of Computer Software & Media Technology, SangMyung University

요 약

본 논문에서는 2단계 구문분석을 통한 구문분석 말뭉치 구축도구를 제안한다. 제안하는 방법은 대량의 구문분석 말뭉치를 수동으로 구축할 때 요구되는 작성자의 수작업을 줄이는 것을 목적으로 한다. 도구는 입력 문장을 문장 분할기준에 따라 분할하는 문장 분할 단계, 각 부분에 대해 자동 구문분석을 수행하는 부분 구문구조 생성 단계, 각 부분 구문구조를 통합하여 완전한 구문구조를 얻는 부분 통합 단계로 이루어져 있다. 자동 구문분석은 자질기반 한국어 구문분석모델을 이용하였고 문장을 부분으로 분할할 때는 문장 분할기준을 말뭉치에서 자동추출 하고 간단한 검증을 거쳐 적용하는 방법을 택하였다. 구문분석 말뭉치 구축의 각 단계에서 자동 구문 분석기가 출력한 결과를 작성자가 취소, 재구축 가능하게 하였다.

1. 서론

1990년대 이래 자연어처리 시스템의 실용화를 위하여 대량의 자료인 말뭉치를 기반으로 하는 자연어처리 방법이 주로 사용되고 있으며 문장의 구문 구조를 구문 트리의 형태로 표현하고 있는 구문분석 말뭉치는 여러 자연어 처리 분야와 언어학 연구에 유용하게 사용되고 있다.[1]

그러나 정확성이 높은 대량의 구문분석 말뭉치를 구축하는 것은 어려운 작업이다. 말뭉치에서 올바른 언어지식을 추출하기 위해서는 말뭉치가 충분한 양의 정확한 정보를 제공해 주어야 하는데 자동 구문분석기의 성능이 완벽한 결과를 보장 하지 못하므로 말뭉치 구축 시 자동 구문 분석 결과를 그대로 사용하지 못하기 때문이다. 그래서 작성자가 각 문장의 구문구조를 시간과 노력을 투자하여 수동으로 작성해야 하는 어려움이 존재한다. 이러한 이유로 작성자의 수작업을 줄여 주면서 정확한 말뭉치를 편리하게 작성 할 수 있는 구문분석 말뭉치 구축도구가 요구된다.

본 논문에서는 구문분석 말뭉치를 구축 할 때 작성자의 수작업을 줄이면서 편리한 작업을 가능하게 하는 '2단계 구문분석을 이용한 구문분석 말뭉치 구축 도구'를 제안한다.

논문의 구성은 다음과 같다. 2장에서는 기존연구에서 제안한 구문분석 말뭉치 도구의 장단점에 대해 비교하고 3장에서는 본 논문에서 제안하는 2단계 구문분석을 이용한 말뭉치 구축 도구에 대해 설명한다. 4장에서는 제안하는 방법이 말뭉치 구축에 얼마나 효율적으로 적용 되는지 실험을 통해 알아본다. 마지막 5장에서는 제안 하는 방법에 대해 결론을 내리고 향후 발전 방향에

대해 논한다.

2. 기존연구

기존연구에서는 도구를 이용한 구문분석 말뭉치 구축 시 수작업을 줄이기 위해 노력한 기존연구들을 비교한다.

영어 구문분석 말뭉치 PennTreeBank는 Fidditch[2]라는 도구를 이용하여 구축되었다. Fidditch는 입력 문장의 문장 성분을 보고 중의성이 없다고 판단되는 부분에 대해 부분 구문구조를 부착한다. 나머지 부분에 대해서는 사람의 입력에 따르는 방법을 사용하였다. 이 방법은 규칙을 이용하지만 입력 문장이 들어왔을 때 중의성이 없다고 판단되는 한 부분에 대해서만 부분 구조를 생성하고 나머지 작업은 사람의 수작업에 의존한다. 제안하는 방법에서는 문장을 여러 개 부분으로 나누어 부분에 대한 모든 구문구조를 제시한 후 부분을 통합해주어 사람은 잘못된 자동 분석 결과를 수정하는 작업만 하도록 하였다.

STEP2000 말뭉치[3]를 구축하기 위한 도구로 [4]가 있다. 이 도구는 '부분 구문구조 부착'과 '일관성 검사' 두 단계로 구문분석을 수행한다. 부분 구문구조 부착 단계에서는 부분 구문분석 규칙을 이용하여 입력 문장의 특정 부분을 구문구조로 변환하여 준다. 여기서 규칙은 전문가가 수동으로 작성한 것이며 규칙에 의해 만들어진 부분 구문분석 결과에 사람이 추가 구문구조를 작성하여 완전한 구문구조를 만든다. 한 문장의 구문구조가 완성된 이후에 일관성 검사를 통해 오류를 검출하고 오류 가능성이 있는 경우 사람에게 확인을 요구한다. 일관성 검사는 현재 작성된 구문구조와 기존에 구축된 말뭉치의 문장들 간의 비교를 통해 오류를 검출하는 과정이다. 이 도구는 위와 같은 과정을 통해 정확하다고 판단되는 특정 부분에 대해 부분 구문구조를 제시하고 일관성 검사를 통해 검증하는 방법으로 수작업을 감소시켰다. 그러나 이 방법은 규칙을 전문가가 수동으로 작성해야 하고 문장 특정 부분의 구문구조가 생성된 이후에는 사람이 수작업으로 남은 구문구조를 작성해 주어야 하는 단점이 있다. 본 연구에서는 자동으로 추출한 문장 분할 기준에 따라 분할 된 문장의 모든 부분에 대해 부분 구문구조를 제시하고 사용자의 검증을 거쳐 전체 구문구조를 얻는 방법을 취한다.

[5]는 구문분석 말뭉치 구축 시 수작업을 감소시키기 위하여 자동 추출한 구문패턴을 적용하는 도구를 제안하였다. 사용자가 구문 범주, 품사열, 어휘열과 같은 자질집합과 신뢰도를 선택하면 구문패턴 후보를 선택하고 신뢰도 검사를 통해 구문패턴을 확정한다. 문장이 입력되면 가능한 구문패턴을 적용하여 구문분석을 수행한 후 올바른 구문구조가 부착되면 끝낸다. 이 방법은 구문패턴을 자동으로 추출하고 패턴을 적용해 구문분석을 수행하는 방법으로 수작업을 줄이고 있다. 그러나 적용된 구문패턴이 옳은 지 아닌 지 매 단계마다 판단해야 하고 패턴이 맞는 경우에 사람이 묶기/이동 연산을 수행하여야 한다. 본 연구에서는 자동으로 분할된 문장의 각 부분이 최종 구문구조를 생성하는 데 오류가 있다면 사람이 수정하고 아니라면 다음단계로 바로 진행할 수 있게 하여 부분을 분할하는 데 있어 사람의 간섭을 줄였다.

3. 2단계 구문분석을 이용한 구문분석 말뭉치 구축도구

이 장에서는 제안하는 방법이 구문분석을 수행하여 말뭉치를 구축할 때 거쳐야 하는 각 단계에 대해 설명한다. 3.1 절에서는 구문구조 중의성을 줄이기 위한 '2단계 구문분석'에 대해 알아보고 3.2절부터 3.4절 까지는 실제 말뭉치를 구축할 때 수행해야 하는 각 단계에 대해 설명한다.

3.1 2단계 구문분석

도구를 이용한 구문 분석 시 작성자가 처음부터 수작업으로 모든 구문분석 과정을 수행하는 것 보다 컴퓨터에 의해 자동으로 얻은 구문분석 결과를 이용하는 것이 더 쉽지만 자동으로 얻은 구문분석 결과가 완벽하지 않으므로 많은 수정 과정을 거치게 된다.

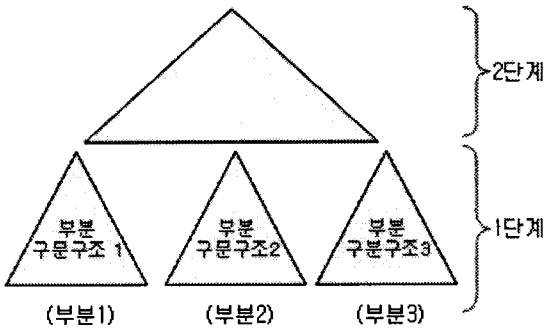
<표 1>은 테스트 문장을 길이 별로 나누어 각각의 정확율과 재현율을 구한 결과이다. <표 1>의 결과에서 알 수 있듯이 문장이 길어질수록 자동 구문분석의 성능이 떨어진다.

	입력 문장 수	정확율	재현율
5어절 이하	137	94.16	92.09
5어절-10어절	207	90.21	86.80
10어절-15어절	196	84.41	83.05
15어절-20어절	113	82.55	79.54
20어절-25어절	48	80.89	77.29
25어절 이상	40	79.66	75.35
전체	741	83.38	81.54

<표 1> 입력 문장의 길이 별 구문분석 성능

그러므로 문장이 길어질수록 말뭉치를 구축하기 위한 수작업 양이 늘어나게 될 것이다. 예를 들어 "승부에 연연하여 보다 빨리 달리게 하고자 선수들에게 약물을 먹이는 비인간적 수법이 아시아드에서도 검사되고 있다." 라는 14어절로 이루어진 문장 전체를 자동 구문분석 했을 때 제안하는 도구를 사용하여 이 문장을 옳은 구조로 수정하기 위해서는 생성된 구문구조를 취소하고 다시 생성하는 12번의 수작업이 필요하다. 그러나 동일한 입력문장을 '승부에 연연하여', '보다 빨리 달리게 하고자', '선수들에게 약물을 먹이는', '비인간적 수법이', '아시아드에서도 검사되고 있다.'와 같이 5부분으로 분할한 후 부분에 대한 정확한 구문구조를 얻고, 부분들을 통합하여 완전한 구문구조를 만든다면 4번의 수작업만으로도 같은 결과를 얻을 수 있다.

이와 같이 동일한 문장을 가지고 정답 구문구조를 만들 때 한 문장 전체를 한 번에 자동 구문분석 한 결과를 수정하는 것 보다 문장을 짧은 부분으로 분할하여 구문분석 후 통합하는 것이 작업자의 수작업을 줄일 수 있고 수정도 용이하다. 그러므로 본 논문에서는 '2단계 구문분석을 이용한 구문분석 말뭉치 구축도구'를 제안한다. 구문 구조 태깅에 걸리는 시간과 구문 구조의 재괄호 치기(re-blanketing)의 빈도수가 적을수록 작업량이 줄면서 말뭉치 구축 속도가 높아지기 때문이다.[2]



<그림 1> 2단계 구문분석

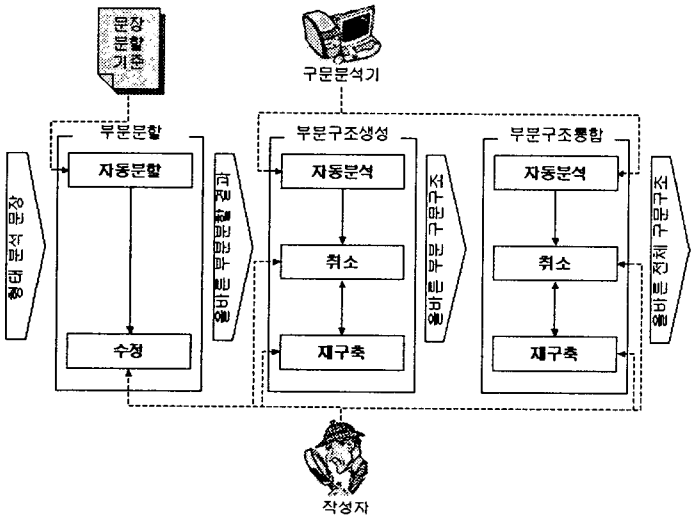
2단계 구문분석이란 하나의 최종 구문구조를 얻기 위해 단계적으로 구문분석을 수행하는 것을 말한다. <그림 1>에서 보여주는 것과 같이 첫 번째 단계에서는 형태소 분석 문장을 입력으로 받아 문장 분할 기준에 따라 부분으로 분할하고, 분할된 각 부분에 대해 구문 분석을 수행한다. 문장을 부분으로 분할하는 단계에서 가능한 구문구조 후보가 줄어들게 되므로 전체 구문구조를 생성하는 데 있어 중의성을 줄여줄 수 있다. 두 번째 단계에서는 부분에 대한 구문 분석 결과를 통합하여 완전한 결과를 출력한다. 구문분석은 자질기반 한국어 구문분석모델[6]을 사용하여 수행하였고 문장 분할 기준은 기존에 구축된 구문분석 말뭉치의 정보를 이용하여 추출한다.

<그림 2>는 2단계 구문분석을 이용한 구문분석 말뭉치 구축도구의 전체적인 과정을 보여준다. 문장이 입력되면 자동분할 단계에서 문장을 부분으로 분할한다. 문장 분할점은 작성자에 의해 수정 가능하다. 문장 분할점이 확정되면 각 부분에 대해 자동 구문분석을 수행하여 부분 구문구조를 얻는다. 작성자가 부분 구문구조를 수정하는 것이 가능하다. 각 부분에 대한 구문구조가 확정되면 부분 통합단계를 거쳐 하나의 완전한 구문구조를 얻는다. 전체 구문구조에 대한 수정도 가능하다. 제안하는 도구는 형태소 분석결과를 입력 받아 이진 구구조 형식으로 구문 분석 된 문장을 출력 한다.

; 40~50대가 대부분인 일꾼들은 하고 있는 일이 힘에 부친 듯 지친 표정들이었다.

```
(S      (NP_SBJ      (S_MOD  (NP_SBJ 40/SN + ~/SO + 50/SN + 대/NNB + 가/JKS)
                             (VNP_MOD 대부분/NNG + 이/VCP + ㄴ/ETM))
                             (NP_SBJ 일/NNG + 꾼/XSN + 들/XSN + 은/JX))
        (VNP         (NP_AJT      (S_MOD  (NP_SBJ      (VP_MOD      (VP 하/VV + 고/EC)
                                                                     (VP_MOD 있/VX + 는/ETM))
                                                       (NP_SBJ 일/NNG + 이/JKS))
                                             (VP_MOD     (NP_AJT 힘/NNG + 에/JKB)
                                                         (VP_MOD 부치/VV + ㄴ/ETM)))
                                  (NP_AJT 듯/NNB))
                     (VNP    (VP_MOD 지치/VV + ㄴ/ETM
                             (VNP 표정/NNG + 들/XSN + 이/VCP + 었/EP + 다/EF + ./SF))))
```

<그림 3> 문장 부분분할 기준 추출 예



<그림 2> 2단계 구문분석을 이용한 구문분석 말뭉치 구축도구

3.2 문장 부분 분할 단계

형태 분석된 문장이 입력되면 문장 분할 기준에 따라 문장이 부분으로 분할된다. 제안하는 방법에서는 기존에 구축된 구문분석 말뭉치에서 정보를 추출하여 문장 분할 기준을 마련하는데 구문분석 된 문장에서 구 묶음되는 부분의 어절과 앞 어절, 뒤 어절의 형태소 분석 정보를 이용한다.

예를 들어 <그림 3과> 같은 문장에서는 '일꾼들은' 과 '듯' 뒤에서 문장을 분할하는 것으로 보고 두 개의 분할 기준을 추출 할 수 있다. 분할 대상이 되는 어절과 하나 앞 어절, 하나 뒤 어절의 형태소 분석 결과를 이용하기 때문에 '일꾼들은' 에서의 분할 기준은 '/NNG /ETM # /NNG /JX # /VV /EC'가 된다. 이렇게 문장 분할 기준을 추출하여 일정 빈도 이상으로 나타난 것을 문장 분할 기준 리스트에 추가한다.

/NNG /JKO # /VV /EC # /NNG /JKS	
양성반응을	양성/NNG + 반응/NNG + 을/JKO
보였으며	*보이/VV + 었/EP + 으며/EC*
기록이	기록/NNG + 이/JKS
/VV /ETM # /NNB /JX # /NNG /JKO	
묶은	묶/VV + 은/ETM
것은	*것/NNB + 은/JX*
통증을	통증/NNG + 을/JKO
/NNG /JKS # /VV /ETM # /NNB /EF	
능력이	능력/NNG + 이/JKS
오른다는	*오르/VV + ㄴ다는/ETM*
것이다	것/NNB + 이/VCP + 다/EF + ./SF
/VV /EC # /VX /EC # /NNG /JX	
있게	있/VV + 게/EC
되자	*되/VX + 자/EC*
이번에는	이번/NNG + 에/JKB + 는/JX

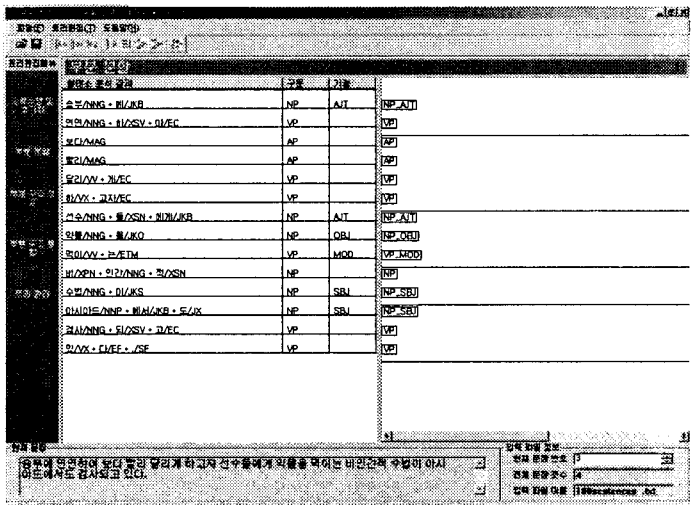
<표 2> 문장 부분분할 기준과 적용 예

위와 같이 추출된 문장 분할 기준의 정확성을 높이기 위해서 간단한 검증 단계를 거친다. 검증 단계에서는 문장 분할 기준을 추출하지 않은 말뭉치에 분할 기준을 적용시켜 오류를 많이 발생시키는 기준을 리스트에서 삭제한다. <표 2>는 문장 분할 기준의 일부와 적용된 예를 보여준다. 적용된 예에서 이탤릭체로 표시된 어절이 문장 분할되는 위치이다.

이렇게 마련된 문장 부분 분할 기준을 입력 문장 "*승부에 연연하여 보다 빨리 달리게 하고자 선수들에게 약*

물을 먹이는 비인간적 수법이 아시아드에서도 검사되고 있다."에 적용시키면 '*승부에 연연하여*', '*보다 빨리 달리게 하고자*', '*선수들에게 약물을 먹이는*', '*비인간적 수법이*', '*아시아드에서도 검사되고 있다.*'와 같이 다섯 부분으로 분할된다.(그림 4)

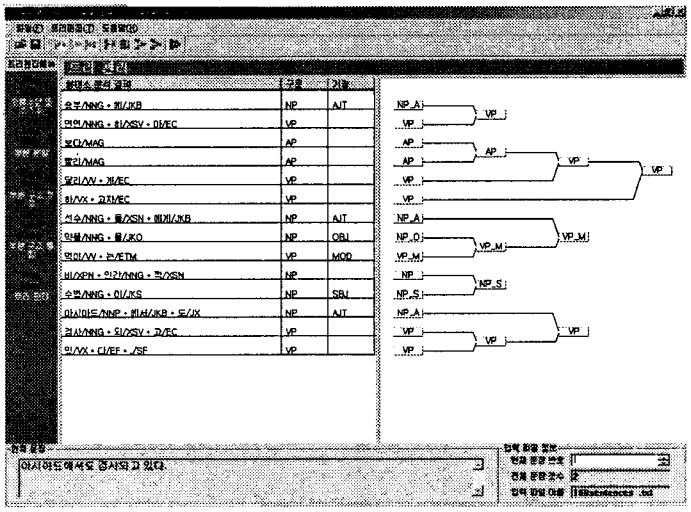
위와같이 자동으로 분할된 결과가 완전한 구문구조를 얻는데 오류가 없다고 판단이 되면 바로 다음 단계로 넘어갈 수 있고 아니라면 작성자가 분할 기준점을 추가, 삭제 하여 더 정확한 부분을 생성할 수 있다.



<그림 4> 자동 부분분할

3.3 부분 구문구조 생성 단계

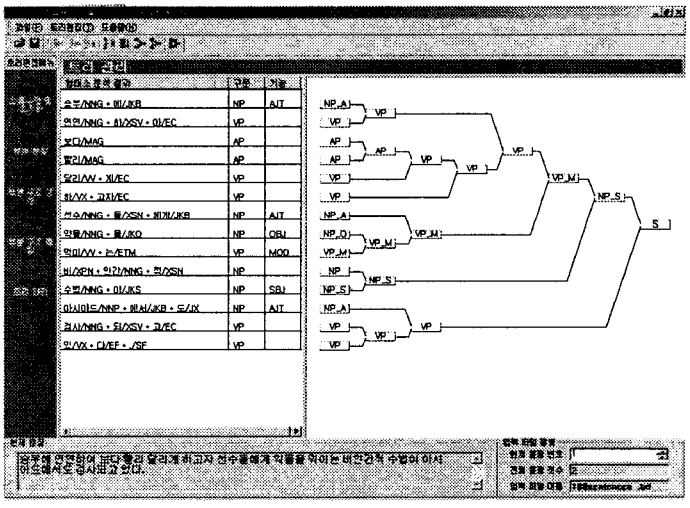
부분 분할에 의해 문장이 부분으로 나뉘어 지면 각 부분에 대해 부분 구문분석을 수행한다. 구문분석은 자동으로 수행되며 자질기반 한국어 구문분석모델[6]을 사용한다. 자동 구문분석에 의해 각 부분에 대한 구문구조가 얻어지면 작성자는 부분에 대한 구문구조가 옳은지 판단한다. 구문구조가 옳다고 판단되면 다음단계인 부분 구문구조 통합단계로 바로 진행 할 수 있고 자동 분석된 구문구조에 오류가 있으면 작성자가 부분 구문구조를 취소하고 재구축하여 수정 할 수 있다. <그림 5>는 5개의 부분 구문구조 생성이 완료된 것까지의 결과를 보여준다.



<그림 5> 부분 구문구조 생성

3.4 부분 구문구조 통합 단계

부분 구문구조 생성 단계에서 얻은 각 부분의 구문구조가 옳다고 판단되면 각 구조를 통합하여 완전한 하나의 구문구조를 생성한다. 통합 구문구조도 자질기반 한국어 구문분석모델[6]을 사용하여 자동으로 생성된다. 이렇게 얻은 전체 구문구조가 옳다고 판단되면 그대로 저장하고 아니라면 작성자가 수동으로 구문구조 취소, 재구축하는 과정을 통해 구문구조를 수정한다. <그림 6>은 통합된 구문구조를 보여준다.



<그림 6> 전체 구문구조 통합

4. 실험 및 평가

실험을 위해 세종계획 구문분석 말뭉치에서 약 6,700개 문장을 학습하였다. 실험은 테스트 집합에서 741개 문장을 대상으로 수행하였다. 도구를 정확하게 평가하기 위해서는 수작업이 감소된 정도를 정확한 수치로 제시해야 하지만 일일이 측정하기 힘들기 때문에 문장을 부분으로 나누는 정확도를 실험 결과로 제시하겠다. 자동으로 나누어진 부분이 정확해야 통합 결과의 성능도 올라가고 구문구조 수정에 필요한 수작업도 줄어들기 때문이다.

문장 부분분할 기준에 따라 전체 741개 테스트 문장이 1,555개의 부분으로 분할되었다. 한 문장이 평균 약 2.09개의 부분으로 분할되었고 한 부분 당 평균 어절 수는 약 5.6개 이다.

문장을 분할하여 통합하는 2단계 구문분석 과정은 사람이 관여하지 않고 모두 자동으로 수행하였다. 테스트 집합의 문장을 길이별로 나누어 부분 분할한 결과는 <표 3>과 같다. 부분 당 평균 어절 수가 문장 길이에 관계없이 5~6개 이므로 문장이 길어질수록 문장 당 평균 부분의 수가 증가하는 것을 볼 수 있다.

평가 방법으로는 부분 경계 정확도, 부분 구조 정확도, 상위 구조 정확도를 제시하였다. 부분 경계 정확도는 자동으로 분할 된 부분 중 부분의 경계가 정확한 것의 비율, 부분 구조 정확도는 자동으로 분할 된 부분 중 경계도 정확하고 부분의 구문구조도 정확한(exact match) 부분의 비율, 상위 구조 정확도는 부분을 통합할 때 부분 구조를 제외한 상위 구조가 정확한(exact match) 구문구조의 비율이다. 평가 방법에 대한 수식은 아래와 같다.

$$\text{부분 경계 정확도} = \frac{\text{부분분할의경계가맞은개수}}{\text{자동으로분할된부분의개수}} \times 100$$

$$\text{부분 구조 정확도} = \frac{\text{경계와구문구조모두맞은개수}}{\text{자동으로분할된부분의개수}} \times 100$$

$$\text{상위 구조 정확도} = \frac{\text{상위구조가맞은개수}}{\text{상위구조가생성된개수}} \times 100$$

실험 결과는 <표 4>와 같다. 전체 테스트 문장에서 부분 경계 정확도는 77.56%, 부분 구조 정확도는 60.51%이다. 부분 구조를 통합하여 전체 구문구조를 완성할 때 상위 구조의 정확도는 62.27%이다. 그러므로 문장의 경계가 정확히 주어진 다면 부분의 구조가 정확할 확률은 78.03%라 할 수 있다.

길이	문장 수	부분의 개수	전체 어절수	부분 당 평균 어절 수	문장 당 평균 부분 수
5어절 - 10어절	207	348	1659	4.7	1.68
10어절 -15어절	196	434	2507	5.7	2.21
15어절 -20어절	113	296	1977	6.6	2.61
20어절 -25어절	48	161	1101	6.8	3.35
25어절 이상	40	179	1163	6.4	4.47
전체	741	1555	8819	5.6	2.09

<표 3> 문장 길이 별 부분분할 결과

길이	부분 경계 정확도(%)	부분 구조 정확도(%)	상위 구조 정확도(%)
5어절 - 10어절	88.22	75.86	83.96
10어절 -15어절	74.42	55.76	64.00
15어절 -20어절	71.62	47.63	61.86
20어절 -25어절	71.43	48.45	33.33
25어절 -30어절	62.01	44.69	24.32
전체	77.56	60.51	62.27

<표 4> 문장 길이 별 2단계 구문분석 성능

문장 길이 별 성능 변화를 보면 문장 당 평균 부분의 수가 많아질수록 상위 구조 정확도가 떨어지는 경향을 보이는 것을 볼 수 있다. 문장의 길이가 길어지면 부분의 수도 많아지고, 다수의 부분 통합 시 발생하는 중의성 때문에 상위 구문구조의 성능이 떨어진다고 할 수 있겠다.

제시된 모든 부분 구문구조의 정확율은 90.41%(표 5)이다. 그러므로 올바른 경계가 주어진다면 자동 분석을 통한 결과가 높은 정확도를 보인다고 할 수 있다.

길이	정확율
5어절 -10어절	94.19
10어절 -15어절	91.07
15어절 -20어절	89.39
20어절 -25어절	87.47
25어절 이상	90.04
전체	90.41

<표 5> 부분 구문구조의 F-measure

길이	전체구문분석		부분구문분석	
	정확율	재현율	정확율	재현율
5어절 -10어절	90.21	86.80	90.51	86.51
10어절 -15어절	84.41	83.05	85.01	84.41
15어절 -20어절	82.55	79.54	84.44	78.55
20어절 -25어절	80.89	77.29	82.10	74.49
25어절 이상	79.66	75.35	80.99	72.86
전체	83.38	81.54	85.36	81.33

<표 6> 전체구문분석 결과와 부분구문분석 결과 비교

<표6>은 한 문장을 전체 구문분석 한 결과와 부분 구문분석 한 결과의 성능을 비교한 것이다. 정확율은 파서에 의해 생성된 구조 중 올바르게 생성된 구조의 비율이고 재현율은 정답 말뭉치에 있는 구조 중 파서에 의해 올바르게 생성된 구조의 비율이다. 전 단계를 자동으로 수행했지만 문장을 한 번에 구문분석 한 결과 보다 조금 향상된 성능을 얻었다.

실험결과에서 보여지 듯 2단계 구문분석에서는 부분의 경계를 제대로 설정해 주어 부분의 정확도를 높이는 것이 잘못된 구문구조를 수정하는 작업자의 수작업을 줄일 수 있는 방법이라고 할 수 있겠다.

5. 결론 및 향후연구

본 논문에서는 '2단계 구문분석을 통한 구문분석 말뭉치 구축도구'를 제안하였다. 제안하는 방법은 자동으로 추출된 부분 분할기준에 따라 입력 문장을 여러 개의 부분으로 분할하고 각 부분에 대한 구문분석 후 모든 부분이 옳다고 판단되면 부분을 통합하는 과정을 거쳐 전체 구문구조를 얻는다. 각 단계 마다 작업자가 개입하여 구문구조를 취소하고 재구축 하는 구문구조 수정을 가할 수 있다. 이 방법은 다음과 같은 특징을 가진다.

첫째, '2단계 구문분석을 통한 구문분석 말뭉치 구축도구'는 문장에 대한 올바른 부분 분할을 확정한 후에 구문분석을 수행하므로 구문구조 후보를 줄여준다. 그러므로 전체 구문구조 생성 시 구문구조에 대한 중의성을 줄일 수 있는 장점이 있다.

둘째, 문장 분할, 부분 구문분석, 부분 통합을 단계적으로 진행하기 때문에 처음부터 완전한 구문구조를 제시하고 오류부분을 수정하는 것 보다 작업자의 수작업을 줄여줄 수 있다.

2단계 구문분석에서는 부분의 정확한 경계를 알아내는 것이 정확한 구문구조를 생성하는 데 중요한 기초가 된다. 그러므로 현재 자동 추출하여 간단한 검증 후 적용하고 있는 부분 분할 기준을 더 정확한 기준이 되도록 검증하고 확대하는 일이 중요하다. 기준 추출 시 언어학적 지식을 추가하거나 작성자가 실제 작업 시 부분 분할점을 추가, 삭제 할 때의 정보를 이용하는 등 정확한 부분 분할 기준을 마련하는 것이 향후 연구 과제라 하겠다.

참고 문헌

[1] 김영택 외 공저, "자연언어처리", 생능출판사, 2001.

[2] Mitchell P. Marcus, B. Santorini , and M. A. Marcinkiewicz, "Building a large annotated corpus of English : the Penn TreeBank", Computational

Linguistics, Vol.19, No.2, pp.313-330, 1993
[3] 이공주, 김재훈, 장병규, 최기선, 김길창, "한국어 구문 트리태깅 코퍼스 작성을 위한 한국어 구문태그", CS/TR-96-102, KAIST, 1996.
[4] 장병규, 이공주, 김길창, "대량의 한국어 구문 트리 태깅 코퍼스 구축을 위한 구문 트리 태깅 워크 벤치의 설계 및 구현", 제 9회 한글 및 한국어 정보처리 학술 발표 논문집, pp.421-429, 1997.
[5] 임준호, 박소영, 곽용재, 임해창, 김의수, 강범모, "구문패턴을 이용한 반자동 구문분석 말뭉치 구축도구", 제 15회 한글 및 한국어 정보처리 학술 발표 논문집, pp. 343-350, 2003
[6] 박소영,"Probabilistic Feature-based Parsing Model for Korean Syntactic Analysis", 고려대학교 박사학위 논문, 2005.
[7] 임준호, 곽용재, 박소영, 임해창, "신경망을 이용한 반자동 구문분석 말뭉치 구축도구", 한국정보과학회 2003년 춘계학술대회, 2003.
[8] 김광백, 박의규, 나동렬, 윤준태, "구간 분할 기반 한국어 구문분석", 제 15회 한글 및 한국어 정보처리 학술 발표 논문집, pp.163-168, 2003.
[9] 장재철, 박의규, 나동렬, "구간분할 기반 한국어 대등접속 구문분석 기법", 제 15회 한글 및 한국어 정보처리 학술 발표 논문집, pp.139-146, 2003.