구묶음과 구간분할을 이용한 의존 관계 추출 기법

박의규[0] 조민희 김성원 나동열
연세대학교 정보기술학부
ekpark63@dragon.yonsei.ac.kr

A Method for Extracting Dependency Relations Using Chunking and Segmentation

Eui-Kyu Park[0], Min-Hee Cho, Seong-Won Kim, Dong-Yul Ra
Div. of Information Technology, Yonsei University

요 약

본 논문에서는 구묶음과 구간분할에 기반하여 한국어 문장에서 명백한 의존관계를 파악 추출하는 기법에 대해 다룬다. 구묶음 기법은 문장에 나타나는 복합 명사, 본용언/보조용언, 의존 명사 등을 묶어서 문장의 구조를 단순하게 만든다. 특히 문장에 다양한 형태로 나타나는 의존 명사의 처리를 강화하여 구묶음을 효과적으로 할 수 있도록 하였다. 구간분할 기법은 긴 문장을 여러 개의 구간으로 나누어 각 구간을 구문분석 한다. 각 구간은 분할 이전보다는 단순화된 형태이기 때문에 긴 문장보다는 중의성이 줄어들어 견고한 구문분석을 할 수 있게 된다. 본 논문에서는 한국어 구문분석 시스템 개발의 1 단계 과정으로써 일단 중의성이 있는 상황이 아닌 명백한 의존관계를 수집하는 것을 목표로 한다. 본 논문에서는 실험을 통하여 구묶음과 구간분할 기법이 문장의 구조 중의성을 줄여 줌으로써 보다 많은 명백한 의존관계를 정확하게 추출할 수 있음을 보였다.

1. 서 론

한국어는 비교적 어순이 자유롭고 격 조사 및 어미의 사용이 매우 발전되어 있는 언어이다. 또한 한국어는 문맥으로 파악할 수 있으면 주어나 목적어 등과 같은 필수적인 문장 요소까지도 생략할 수 있다[3]. 이러한 특징으로 인해 구 구조 문법으로 한국어를 표현하면 생성 규칙의 수가 지나치게 많아지므로, 구 구조 문법은 한국어 통사 구조의 분석에는 적합하지 않다고 주장하는 사람이 많다[6]. 따라서, 한국어 구문 분석에서는 의존 문법을 많이 이용한다.

의존 문법은 단어 사이의 의존 관계에 중심을 두는 문법이다[8,9]. 주어진 문장 안의 단어 사이의 의존 관계를 파악하는 작업이 의존 문법에 의한 언어 분석의 중요한 작업이다[7]. 의존 관계의 파악은 한국어가 가진 수식 관계의 특성에 기반을 두고 있다.

어순의 자유성, 생략 등 한국어의 특성을 잘 처리할 수 있다고 생각되어 의존 문법은 지금까지 한국어의 분석에 가장 많이 이용되어 왔다[4]. 이러한 점을 감안하여 우리는 한국어 구문분석 시스템으로서 의존관계를 파악하는 의존 문법 기반 시스템을 개발하고자 한다.

구문분석에 있어서 문제가 되는 것은 중의성의 해소이다. 그러나 중의성의 해소를 위해서는 많은 지식이 필요하다. 예를 들면 용언에 대한 하위범주화 정보 같은 것들이다. 우리는 각 용언에 대하여 하위범주화 정보를 구축하는 대신 많은 가능한 의존관계를 수집하여 이를 이용하고 자 한다. 많은 통계적인 의존관계 정보는 중의성의 해소에 매우 주요한 역할을 한다[7]. 이러한 정보의 수집은 대량의 트리-부착 말뭉치(tree-tagged corpus)를 필요로 한다. 그러나 이의 구축은 많은 인력과 시간이 필요하여 단시간 내에 달성하기 어렵다.

이 문제를 해결하기 위해서 우리는 얕은 파싱(shallow parsing) 기법을 이용하여 원시말뭉치에서 자동으로 많은 의존관계를 추출하는 것을 시도하고 자 한다. 만약 말뭉치에 들어 있는 의존관계를 어느 정도 이상 상당한 정확도로 추출할 수 있다면 대량의 의존관계 구축이 가능하게 되고 이를 이용하여 구문분석의 가장 핵심적인 문제인 중의성 해소를 달성할 수 있다.

이러한 관점에서 본 논문에서는 원시 말뭉치에서 명백한 것으로 판단되는 의존관계를 자동으로 추출하는 시스템을 개발하는 문제에 대하여 설명한다. 우리는 견고한 얕은 파싱 기법을 이용하여 구문적으로 전혀 중의

성이 개입되지 않은 명백한 의존관계를 가능하면 많이 그리고 정확히 추출하고 자 한다. 그 기술은 차후에 대량의 의존관계를 수집하여 구문분석 시스템의 핵심단계인 중의성 해소 모듈에서 유용하게 사용하기 위함이다.

본 논문에서 제시하는 주요 기술은 문장의 복잡성을 감소시킴으로써 가능하면 더 많은 올바른 의존 관계를 파악하는 것이다. 이것은 구문 분석 시스템의 견고성과 정확성을 높이게 된다. 이를 위하여 본 연구에서는 두 가지 기법을 도입한다.

첫째는 구간 분할 기법이다. 구간 분할이란 문장을 구간이라고 부르는 절 단위로 분리하는 것이다[1]. 구간 분할을 통하여 구문 분석의 대상이 되는 구간의 단어의 수를 줄여서 생성 가능한 구문 분석 트리의 수를 줄일 수 있다. 이는 문장의 복잡성을 감소시킨다 할 수 있다.

둘째는 구묶음(chunking) 기법이다. 구묶음이란 문장을 구성하는 단어들 중에서 서로 연관된 단어들을 합쳐서 한 덩어리로 만드는 것이다[2,5,10]. 구묶음 기법은 구문 분석의 대상이 되는 단어의 수 및 구문 패턴의 다양성을 줄여 주어 문장의 복잡성을 감소시키는 역할을 한다.

실험을 통하여 구묶음과 구간분할 기법이 문장의 복잡성을 감소시켜 많은 정확한 구문 구조(의존관계)를 추출하는 것을 관찰하였다.

2. 파서의 동작 원리

2.1 기본 알고리즘

문장에서 나누어진 각 구간에 대하여 CYK 파싱 기법을 이용하여 모든 가능한 구문 구조를 구하고 그 중에서 가장 적합한 구문 구조를 선택한다. 이것은 기존의 전체 문장에 대하여 구문분석을 행하던 것을 그보다 길이가 작은 각 구간에 대하여 구문분석을 시행하게 되므로 구간마다에 대한 구문분석 작업의 복잡도가 많이 감소하게 된다.

최적의 구문 구조는 통계적인 정보를 이용하여 구한다. 이용되는 통계정보는 의존 관계를 상호정보 형태로 구하여 이용한다. 이러한 의존 관계를 구축하기 위해서는 의존 관계가 부착된 말뭉치가 필요하다. 그러나 현재로선 이러한 말뭉치의 구축이 미미한 상태이다. 따라서, 본 연구에서는 대량의 원시말뭉치(raw corpus)로부터 중의성이 없는 의존 관계를 추출하고 이를 많이 수집하여 나중에 최적의 구문 구조를 구하는 중의성 해소 작업에 이용하고자 한다.

본 논문의 핵심 목표인 중의성이 없는 의존 관계의 파악 및 추출을 위해 우리는 다음과 같은 기법을 사용한다. 주어진 구간에 대하여 구한 모든 가능한 구문구조 즉 구문 트리 안에 어느 의존 관계가 공통으로 나타나는가 안 나타나는가에 의해서 판단한다. 특정 의존관계가 가능한 모든 구문 트리에 나타나면 이 의존관계는 중의성이 없다라고 판단할 수 있다. 이러한 기법으로 원시말뭉치로부터 중의성이 없는 의존 관계를 추출한다.

2.2 구문 구조의 양면성을 이용한 분석

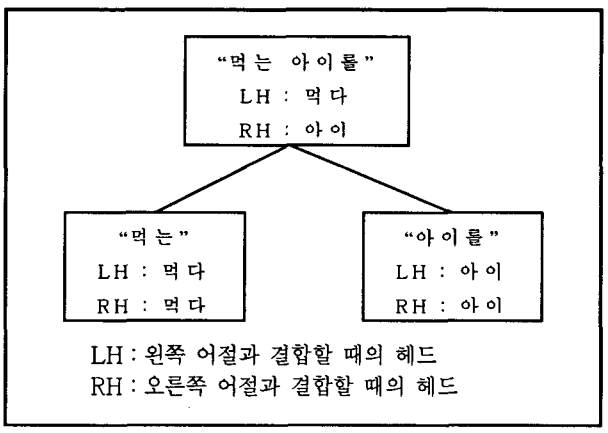
한국어에서는 어떤 어절이 두 가지의 성격으로 사용되는 경우가 있다. 이 기법을 사용함으로써 분석이 어려운 구문을 우리는 효과적으로 처리할 수 있다.

2.2.1 구문 구조가 동시에 용언과 체언으로 사용되는 경우

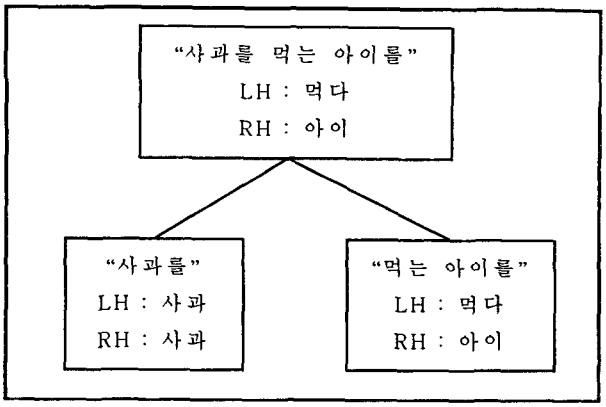
용언의 관형형이 체언을 수식하면 이러한 현상이 나타난다. 용언어절과 체언어절의 결합은 두 성격을 띠는 새로운 노드를 생성한다. 이 노드는 왼쪽의 어절과 결합할 시에는 용언처럼 동작하고, 오른쪽의 어절과 결합할 시에는 체언처럼 동작한다. 단, 관형형 어미를 갖는 용언은 반드시 먼저 오른쪽에 인접한 헤드가 명사인 구문 구조와 결합을 한다.

"철수가 사과를 먹는 아이를 보았다"

위의 예문에서 "먹는"이라는 어절은 관형형어미를 갖는 용언이다. 이 어절은 먼저 오른쪽의 체언어절 "아이를"과 결합한다. 결합한 구문 구조 "먹는 아이를"은 용언과 체언의 성격을 모두 갖게 된다. 용언으로 사용될 경우에는 "먹다"가 헤드가 되며, 체언으로 사용될 경우에는 "아이"가 헤드가 된다. "먹는 아이를"이 어절"보았다"와 결합할 때는 체언으로서 "아이"가 헤드가 되어 "아이"와 "보다"의 의존 관계가 생성된다. "먹는 아이를"이 어절 "사과"와 결합할 때는 용언으로서 "먹다"가 헤드가 되어 "사과"와 "먹다"의 의존 관계가 생성된다.



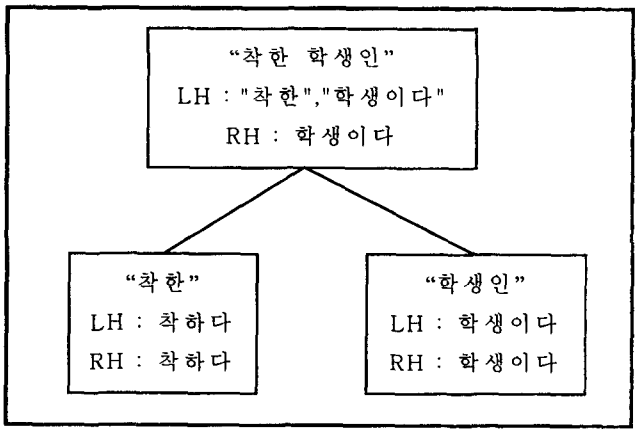
[그림 1] "먹는"과 "아이를"의 결합



[그림 2] "사과를"과 "먹는 아이를"의 결합

2.2.2 구문 구조가 동시에 두 용언에 의해 대표되는 경우

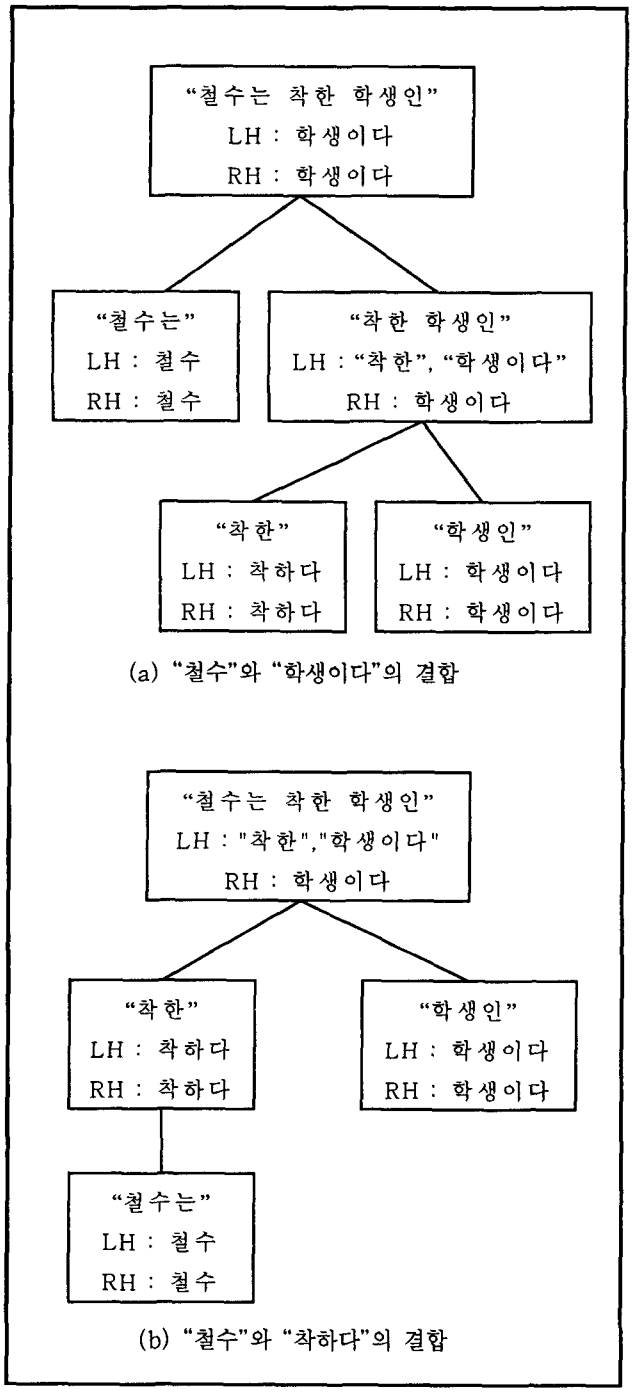
용언의 관형형어절이 오른쪽의 "체언+지정사어절"을 수식하면 이러한 현상이 나타난다. 이 두 어절은 체언+지정사어절이 용언의 관형형어절의 격을 채우면서 결합한다. 이 결합된 노드는 용언1과 용언2의 성격을 갖는다. 용언1은 관형형어절의 용언, 용언2는 체언+지정사어절의 체언+지정사이다. 이 노드가 왼쪽의 체언과 결합할 시에는(즉 이 노드의 LH 가 이용되는 경우에는) 용언1과 용언2 어느 것과도 결합이 가능하다. 이 노드가 오른쪽의 체언과 결합할 시에는(즉 RH 가 이용되는 경우) 용언2의 성격으로만 이용된다.



[그림 3] "착한"과 "학생인"의 결합

"철수는 착한 학생인 길수를 칭찬했다"

위의 예문에서 관형형어미를 갖는 어절 "착한"과 체언+지정사어절 "학생인"이 결합하면, "착하다"와 "학생이다"의 두 가지 성격을 갖는 노드가 생성된다. 이 노드가 어절 "철수는"과 결합할 시에는 "착하다", "학생이다"의 두 가지 성격으로 모두 동작이 가능하다. "착하다"가 헤드로 간주되어 만들어진 "철수,가,착하다"의 의존 관계와 "학생이다"가 헤드로 간주되어 "철수,가,학생이다"의 의존 관계가 생성된다. "착한 학생인" 에 대한 노드(구문구조)가 다음 어절 "길수를"과 결합할 시에는 "학생이다"의 한 가지 성격으로만 동작한다. 즉 "학생이다"가 "길수"와 관계를 맺어 "길수,가,학생이다"의 의존 관계만 생성된다.



[그림 4] "철수는"과 "착한 학생인"의 결합

3. 의존관계 추출의 정확성 향상 기법

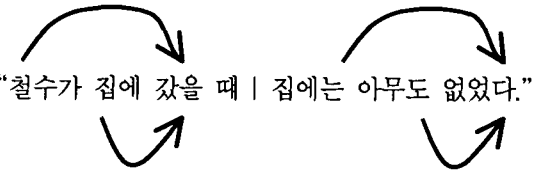
3.1 구간 분할

한국어 문장에서 구문분석이 쉽지 않은 이유는 긴 문장과 한국어의 자유로운 어순에 있다. 따라서 본 논문에서는 문장을 여러 개의 구간으로 나눈 후에 각 구간 단위의 구문분석을 수행하고 구간에 대한 구문분석 결과를 통합하여 전체에 대한 구문분석 결과를 도출하는 기법을 사용한다.

구간은 절과 같은 의미로 사용되며 관형형어미를 갖지 않은 용언 바로 다음에서 구간이 나누어 지게 된다. 또한, "~때", "~인 이유로" 등 이유, 시간 등을 나타내는 구 바로 다음에서 구간분할을 수행한다. 그리고, "~한 김에", "~ 할 시에", "~할 양으로", "~할 지" 등에 나타난 "김, 시, 양, 지" 등과 같은 의존 명사 어절 바로 다음에서 구간 분할한다.

관형형어미를 갖는 용언에서 구간을 나누지 않는 이유는 관형형어미를 갖는 용언 앞의 명사구가 이 관형형어미를 갖는 용언의 뒤에 나타나는 용언에 연결되는 경우가 많기 때문에 관형형 용언에서 나눌 경우 구간 내에서 의존관계를 찾을 수 없는 경우가 많이 발생하기 때문이다.

아래 예문에서 문장은 | 에 의하여 두 구간으로 분할되어 있다. 각 구간마다 그 안의 명사구는 자신의 구간 안의 용언에만 연결되는 것을 볼 수 있다. 따라서 전체를 한번에 구문 분석하는 대신에 각 구간을 구문 분석한 뒤에 그 결과를 통합하여 전체 문장에 대한 구문분석 결과를 얻을 수 있다.



그러나 위 방법으로 구간분할을 할 경우에 어떤 구간의 명사구가 다른 구간의 용언에 연결되어야만 하는 경우들이 발생한다. 다음 예문을 보면,

"철수의 아버지가 철수를 꾸짖은 선생님이 지나가시자 | 인사를 드렸다."

"아버지가"는 첫 구간에 속해 있다. 그러나 이 명사구는 두 번째 구간의 용언과 연결되어야 한다. 첫 구간의 용언들과는 결합되지 않는다. 결국 구간 구문분석에서 이 명사구는 홀로 동떨어지고 구간 내의 구문 구조의 일부가 되지 못한다. 구간 분석에서 구간 내의 용언과 결합되지 못한 명사구들은 구간 통합에 의해서 다른 구간의 용언과 연결하게 된다.

3.2 구묶음

기존의 구묶음 처리 방식은 다음과 같다.

• 복합명사 처리

복합명사를 구성하는 단어들을 합하여 하나의 단위로 만든다. 단위의 마지막 명사가 헤드가 되어 복합명사를 대표하게 된다.

• 본용언/보조용언 처리

보조용언은 자체적으로 독립적인 의미를 갖지 않고 본용언의 의미에 추가적인 성질을 부여하는 역할만 하므로 본용언에 붙인다.

• 의존 명사 처리

"먹을 수 있는 사과"에서 구묶음을 통하여 "먹을+수+있는 사과"가 된다. 여기에서 "먹을+수+있는"은 구문분석 시스템이 볼 때에는 "먹+은"처럼 보이도록 한다.

위는 과거의 연구에서 이미 제안된 구묶음 기술이다. 위와 같은 기존의 구묶음 처리 방식에서 의존 명사 처리를 보면 의존 명사를 무조건 앞의 용언에 붙이도록 한다. 이는 의존 명사에 대한 처리를 매우 단순하게 처리한 것이다. 그러나 의존 명사는 문장 내에서 매우 다양한 문형을 일으킨다. 따라서 이러한 다양한 경우에 대한 고찰이 없이 일률적인 처리를 시도하면 올바른 구문 구조를 생성할 수 없게 된다. 본 논문에서는 다양한 형태로 나타나는 의존 명사들에 대한 처리 기법을 제안하고자 한다.

의존 명사란 자립성이 없는 특수한 명사를 일컫는다. 곧 그 앞에 어떤 한정 성분이 나타나지 않으면 홀로 쓰일 수 없는 비자립적인 명사라는 것이다. 그 앞에 나타나는 성분은 관형어 곧 관형사나 관형사 기능을 지니는 한정어이다.

본 논문에서는 의존 명사는 자립적으로 사용할 수 없다라는 사실에 주목한다. 이 사실은 구문 분석에 있어서 의존 명사 어절을 그대로 두어서 그 패턴이 너무 다양하여 견고한 구문분석을 할 수 없다는 것을 의미한다. 의존 명사를 일반 명사처럼 처리하면 많은 중의성과 다양한 문형을 유발하게 된다. 그렇다고 해서 기존의 단

순한 방식의 의존명사 처리 기법을 이용하면 다양한 형태로 나타나는 의존 명사에 대한 처리가 불완전하게 된다. 따라서 견고하고 정확한 구문 분석을 위해서는 의존 명사에 대한 세부적인 특수 처리가 필요하다.

의존 명사는 그 앞에 나타나는 단어 즉 관형어에 어떠한 미세한 의미를 더하지만 전체적인 관점에서는 얕은 파싱에서는 무시할 수 있는 것이라 생각된다. 본 논문에서는 의존 명사와 이 의존 명사를 한정하는 관형어 사이의 관계와 의존 명사가 관형어에 부여하는 의미를 파악하여 이에 맞는 처리 방법을 각 의존 명사에 대해서 정의한다.

의존 명사는 크게 단위 의존 명사와 비단위 의존 명사로 나뉜다. 단위 의존 명사는 선행 관형어인 수사와 어울려 수량을 나타내는 구실을 하는 의존 명사이다. 본 시스템에서 사용하는 의존 명사는 다음과 같다.

- 단위 의존 명사(646개) : 자, 치, 푼, 마, 리, 간, 평, 마지기, 섬, 가마니, 푸대, 말, 되, 홉, 통, 잔, 병, 개, 가지, 그루, 포기, 자루, …

- 비단위 의존 명사(144개) : 노릇, 것, 놈, 만큼, 대로, 분, 자, 치, 지경, 따위, 무렵, 뿐, 때문, 즈음, 데, 터, 바, 듯, 나름, …

위의 144개 각각의 의존명사를 살펴보고 이에 합당한 처리 작업을 준비하였다. 이들을 대략 크게 분류한다면 의존 명사에 대한 처리 방법은 크게 다음과 같이 나눌 수 있다.

■ 단위 의존 명사의 처리

- 체언 수사+단위의존명사+(조사) (체언+(조사)

수사+단위의존명사를 앞의 체언과 구묶음한다. 만일 단위 의존 명사에 조사가 붙어 있고 체언에 조사가 없으면 단위 의존 명사의 조사를 체언에 붙인다.

"그는 사과 열개를 먹었다."
→ "그는 사과(열+개)를 먹었다."

■ 비단위 의존 명사의 처리 :

- 앞 체언에 구묶음 되는 경우

비단위 의존 명사를 앞의 체언과 구묶음한다.

"쌀, 보리, 밀 따위가 곡식이다."
→ "쌀, 보리, 밀(따위)이 곡식이다."

- 앞 용언에 구묶음 되는 경우

비단위 의존 명사를 앞의 용언과 구묶음한다.

"결코 실패할 리가 없다."
→ " 결코 실패하(리+없)다."

- 구간 분할을 일으키는 경우

비단위 의존 명사 뒤에서 구간 분할을 한다.

"급히 달려오는 바람에 서류를 놓고 왔다."
→ "급히 달려오는 바람에 | 서류를 놓고 왔다."

위의 예문에서 의존 명사"바람"은 이유를 나타내는 경우로서 구간 분할을 할 수 있다.

- 무격 연결의 경우

비단위 의존 명사가 앞의 용언과 무격으로 연결된다. 무격이란 체언이 용언과 연결될 때 특정 격으로 연결되지 않는다는 것을 나타낸다. 특정 격이란 주격, 목적격, 부사격을 말한다.

"그곳을 떠난 지 10년이 지났다."

X(무격)

위의 예문에서 "떠난"과 "지"는 연결이 되지만 특정 격으로 연결이 되는 것은 아니다. 다만 "떠난"에 미세한 의미를 더하여 주는 역할을 한다. 따라서 이러한 연결(의존 관계)을 무격 연결이라 한다.

- 위 작업의 조합이 필요한 경우

비단위 의존 명사의 처리는 하나의 작업만이 필요한 경우도 있지만 두 가지 이상의 작업을 해야 하는 경우도 있다. 위의 무격 연결의 예에서 의존 명사 "지" 다음에 구간을 분할할 필요가 있다.

4. 실험 및 결과

4.1 실험 데이터

국내에서 공식적으로 개발된 한국어 구문트리 부착 말뭉치로 세종계획에서 구축된 것이 있다. 이는 완전한 구문분석을 나타낸 것으로서 우리의 얕은 파싱 시스템의 목적에 잘 맞지 않으며 품사 태그 셋의 차이점 등이 있다. 따라서 본 연구에서 개발하는 시스템의 실험 및 평가에 적용하기 어려운 점이 많다. 따라서 우리는 얕은 파싱 기법을 이용하는 시스템에 적용하기 적합한 구문 트리 말뭉치를 구축하고 있다. 우리가 사용한 실험 데이터 즉 말뭉치는 동아일보 2002년 기사이다[1]. 현재까지 100 문장에 대해서 의존 관계 태깅을 수행하였다. 의존 관계 태깅 문서는 다음과 같은 형식으로 되어 있다.

```
; 철수는 천벌을 받을 놈이다.
55 4
1 4 S
2 3 O
3 4 S
; 저기 뛰어오는 놈이 제 막내아들입니다.
56 5
1 2 A
...
```

위의 그림에서 ';'로 시작하는 행은 실험 문장이다. 그 다음 행의 두 개의 숫자는 문장번호와 어절 수를 나타낸다. 그 다음 행부터는 실험 문장에 나타난 의존 관계를 나타낸다. 의존 관계의 첫번째 숫자는 수식을 하는 어절의 번호이고, 두번째는 숫자는 수식을 받는 어절의 번호이다. 세번째에 있는 문자는 어절 간의 격 관계를 나타낸다. 격 관계로는 S(주격), O(목적격), A(부사격), M(관형격), X(무격) 이 있다.

전혀 구문 관계 태깅이 되지 않은 원시 말뭉치에 대해서도 구묶음과 구간분할을 이용하여 의존 관계 추출 실험을 하였다. 여기에 사용된 실험 데이터는 2000년도 세종 말뭉치이다.

4.2 실험 결과

실험은 다음과 같은 4 가지 방식을 실험하였다.

- ¬C, ¬S : 구묶음 하지 않고, 구간 분할 하지 않음.
- C, ¬S : 구묶음 하고, 구간 분할 하지 않음.
- ¬C, S : 구묶음 하지 않고, 구간 분할 함.
- C, S : 구묶음 하고, 구간 분할 함.

$$\text{재현율} = \frac{\text{제안한 의존 관계중 올바른 것의 수}}{\text{정답 문서가 제안한 의존 관계수}}$$

$$\text{정확률} = \frac{\text{제안한 의존 관계중 올바른 것의수}}{\text{파서가 제안한 의존 관계수}}$$

[표 1] 방식에 따른 의존 관계 추출 재현율 및 정확률

방식	재현율	정확률
¬C, ¬S	8.91%(54개/606개)	65.85%(54개/82개)
C, ¬S	22.28%(135개/606개)	93.75%(135개/144개)
¬C, S	37.13%(225개/606개)	75.50%(225개/298개)
C, S	61.06%(370개/606개)	98.93%(370개/374개)

표1 에서 보면 의존 관계 추출 재현율과 정확률은 구묶음과 구간 분할을 모두 사용한 방식이 가장 좋은 성능을 나타내고 있다.

[표 2] 세종 말뭉치에 대한 처리 결과

총 문장 수	137,153 개
총 어절 수	1,082,717 개
의존 관계 추출 개수	609,530 개
의존 관계 추출율	56.3%

표2 에서 의존 관계 추출율은 총 어절 중에서 의존 관계가 추출된 어절의 비율을 나타낸다. 이 실험은 구묶음과 구간 분할을 다 적용한 시스템으로 하였다. 추출된 의존 관계의 수는 약 61만개이다. 본 연구에서 개발한 시스템의 정확률을 99%라고 했을 때 약 60만개의 정확한 의존 관계가 추출되었을 것이라 기대된다.

4.3 검토

위의 실험 결과에서 알 수 있듯이 본 시스템은 구묶음과 구간 분할을 통해서 중의성이 없는 많은 의존 관계를 정확하게 추출할 수 있었다. 이는 구묶음과 구간 분할을 통하여 문장의 구조를 단순하게 할 수 있었기 때문이다. 특히 문장에 다양한 형태로 나타나는 의존 명사에 대한 구묶음 처리는 의존 명사로 인해 발생하는 많은 중의성을 줄여 주어서 중의성이 없는 의존 관계 추출에 많은 도움을 주었다.

구간 분할이 문장의 구조를 단순하게 만들어 중의성을 줄여 주나 다음과 같은 문제점이 실험을 통해서 관찰되었다.

1) 부산대 한국어정보처리 연구실에서 제공된 띄어쓰기 말뭉치의 일부임.

"여론의 바탕 위에서 서두르거나 쉬지 않고 남북 관계를 발전시켜야 한다."

위의 예문은 다음과 같이 구간 분할이 된다.

"여론의 바탕 위에서 서두르거나 | 쉬지 않고 |남북 관계를 발전시켜야 한다."

첫번째 구간의 "위에서"는 마지막 구간의 "발전시켜야"를 수식해야 한다. 그러나 구간이 분할되어 있기 때문에 같은 구간에 있는 "서두르거나"를 수식한다. 따라서 잘못된 의존 관계가 추출된다. 이것이 표1 의 C,S 의 정확률에서 볼 수 있는 오류를 이루고 있다. 향후에는 이러한 문제를 해결하는 기술에 대한 연구가 필요하다.

이번 실험에서 사용한 말뭉치는 실험을 위해 직접 구축한 구문 태깅 말뭉치로서 그 양이 적다. 세종계획에서 구축한 구문 트리 부착 말뭉치가 있으나 우리 시스템에서 쓰기에는 품사 태그 셋, 의존 관계 태그 셋, 의존 관계의 종류나 성격 등이 달라 바로 실험에 사용하기에 어려움이 있었다. 향후 세종계획에서 구축한 말뭉치를 이용하는 방안을 모색할 예정이다.

5. 결 론

길이가 긴 한국어 문장의 구문 분석은 매우 어렵다. 계산량이 증가할 뿐만 아니라 중의성이 폭발적으로 증가하여 올바른 구문 구조의 결정이 매우 어렵다. 또한 한국어 문장에서 다양하게 나타나는 의존 명사도 올바른 구문 구조의 결정을 어렵게 한다.

본 논문에서는 이러한 문제점들을 극복하기 위하여 긴 문장을 여러 개의 구간으로 나누어 처리하는 구간 분할 구문 분석 기법과 문장의 구조를 복잡하게 만드는 의존 명사를 처리하여 문장의 구조를 단순하게 만드는 구묶음 기법을 제안하였다.

이 기법을 이용하여 중의성이 없는 명백한 의존 관계를 추출하는 시스템을 개발하고 실험하였다. 실험 결과에서 알 수 있듯이 구묶음과 구간 분할 기법을 이용함으로써 중의성이 없는 의존 관계를 좀 더 정확하게 많이 추출할 수 있었다.

참고 문헌

[1] 김광백, 박의규, 나동렬, 윤준태, "구간 분할 기반 한국어 구문분석", 제14회 한글 및 한국어 정보처리 학술대회, pp.163~168, 2002.

[2] 김미영, 강신재, 이종혁, "규칙과 어휘정보를 이용한 한국어 구문 모호성 해결", 제12회 한글 및 한국어 정보처리 학술대회, pp.103~109, 2000.

[3] 나동열, "한국어 구문 분석에 대한 고찰", 정보과학회지, 12(8), pp.33~46, 1994.

[4] 류범모, 이태승, 이종혁, 이근배, "술어중심 제약전파를 이용한 2단계 한국어 의존파서", 한국정보과학회 1996 봄 학술발표논문집, pp.923~926, 1996.

[5] 윤덕호, "숙어 정보를 활용한 한국어 파싱", 서울대학교 박사학위 논문, 1993.

[6] 홍영국, 이종혁, 이근배, "의존문법에 기반을 둔 한국어 구문 분석기", 정보과학회 1993 봄 학술발표 논문집, pp.781~784, 1993.

[7] M. Collins, "A new statistical parser based on bigram lexical dependencies", Proc. ACL'96, pp.184~191, 1996.

[8] P. Hellwig, "Dependency Unification Grammar", Coling 86, pp.195~198, 1996.

[9] I. A. Mel'cuk, Dependency Syntax : Theory and Practice, State Univ. of New York Press, 1988.

[10] Juntae Yoon, Efficient Dependency Analysis for Korean Sentence Based on Lexical Association and Multi-layered Chunking, Literary and Linguistic Computing, vol 16(3), pp.265~285, 2001.