SVM을 사용한 한국어 종속절의 의존관계 분석*

김상수 박성배 이상조
경북대학교 컴퓨터공학
{skim, sbpark, sjlee}@sejong.knu.ac.kr

Analyzing Dependency of Korean Subordinate Clauses Using Support Vector Machine

Sang-Soo Kim, Seong-Bae Park, Sang-Jo Lee
Dept. of Computational Engineering, Kyungpook National University

요 약

한국어 구문 분석에서 가장 어려운 작업들 중에 하나는 종속절의 의존관계 파악이다. 본 논문에서는 이를 해결 하기 위해서 종속절의 의존관계를 절을 구성하는 서술어부(동사와 어미)의 관련 정보의 유무에 따라 의존관계가 성립한다고 가정했다. 즉 각각의 절들의 서술부의 관련 정보의 유무로 보고, 이진 분류 문제로 이 문제를 해결하였다.

사용한 자질은 정적 자질(static feature)와 동적 자질(dynamic feature)를 구성되어 있다. 정적 자질은 동사와 어미에서 표면적인 어휘 정보이고 이는 단어, POS 테그 및 위치 정보들이다. 동적 자질은 문장에서 절이 가지는 문법적인 형태를 의미하고, 이를 추출하기 위해 간단한 규칙을 만들고 이를 바탕으로 CKY 차트 파서를 통하여 추출하였다. 기계학습 방법으로는 이진 분류 문제에서 널리 사용되는 SVM을 사용하였다.

실험 결과 어휘 정보들 중에서 어미의 정보만 사용하였을 경우는 64.4%의 정확도를 보였고 문법적인 정보인 동적 자질을 사용한 경우는 73.5%로 어휘 정보만을 사용한 경우 보다 9.1%의 성능 향상됨을 보였다

1. 서론

구문분석(parsing)은 자연 언어처리에서 가장 핵심이 되는 기술이고 여러 응용 기술의 발전에 반드시 필요하지만, 현재 구문 분석 기술은 아직 실용적인 단계에 있지 못하다. 구문분석에 사용된 방법은 예전에 널리 사용된 규칙 기반(rule based) 처리 방법과 휴리스틱(heuristic) 처리 방법 등과 기계 학습(machine learning) 방법으로 간략히 나누어 볼 수 있다. 규칙 및 휴리스틱 기반의 방법들은 규칙을 추출하기 쉬운 어순을 가진 유럽 언어권에서 어느 정도의 성능을 보여주고 있다. 요즈음은 규칙 기반 방법과 휴리스틱 방법에는 많은 문제점을 내포하고 있어서 기계학습(machine learning) 방법을 이용한 구문 분석 방법이 주류를 이루고 있다.

기존의 구문 분석 과정(full parsing)은 형태소 분석 과정, POS(part of speech) 태깅 및 단어들 간의 의존관계를 분석하는 것으로 수행된다. 이런 과정은 전체 처리 대상이 문장에 속한 모든 단어가 됨으로써, 복잡도가 매우 많이 증가하여 전반적인 구문분석 성능의 저하를 가져왔다. 따라서 최근의 연구 방향은 구문분석의 처리단위를 단어에서 벗어나 더 큰 단위로 확대하여 구문분석의 복잡도를 줄여가는 부분 구문분석(partial parsing)에 관한 연구들이 활발히 진행되고 있다.

한국어에서는 구문 분석의 단위를 단어에서 구 단위로 확대하는 구 단위화(chunking)에 관한 연구가 활발히 진행되어 왔고, 이들 연구의 성능이 어느 정도 안정된 결과를 보여주고 있다[1]. 구 단위화는 처리대상을 명사구에만 그치지 않고, 동사구까지 확대되어 동사구를 인식하는 연구까지 확대되어 왔다. 최근에는 구문분석의 처리단위를 구에서 만족하지 않고 하나의 완성된 문장 성분을 가지는 절 단위까지 확대되고 있다[].

구문분석 대상을 절까지 확대하기 위해서는 절을 구성하는 각각의 성분들의 의존관계 분석과 더불어 절들 간의 의존관계를 분석해야 한다. 절을 구성하는 성분들은 구 단위화 등의 방법들을 통하여 표면적인 정보, 즉 문법적 정보(syntactic Information)만으로 만족할 만

*본 논문은 정통부 및 정보통신 연구진흥원의 정보통신선도기반기술개발사업의 연구결과로 수행되었습니다.

한 의존 관계를 밝힐 수 있었다. 그러나 절들간의 의존관계의 파악은 절을 구성하는데 필수 요소인 서술부를 이루는 동사의 의미적 정보(semantic information) 등을 사용하여 의존 정보를 밝혀야 하는데 이런 의미정보를 획득하고 처리하는데 많은 비용과 어려움이 있어서 의존관계를 밝히는데 상당한 어려움이 있었다.

본 논문에서는 의미적 정보를 배제하고 손쉽게 획득 가능한 문법적 정보만을 가지고 종속절의 의존관계를 밝히는 방법을 제안한다. 이를 처리하기 위해서는 먼저 절들간의 의존관계를 각각의 절의 서술부(동사와 어미)들의 서로 관련성 유무로 보았다. 이 작업은 이진 분류 작업(binary classification task)으로 볼 수 있고, 이를 기계학습 방법을 사용하여 의존관계를 밝혔다. 기계학습 방법으로는 이진 분류 문제에서 가장 좋은 성능을 발휘한다고 알려진 Support Vector Machine(SVM)을 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 절 인식 및 이들 간의 의존관계를 밝히는 기존 연구를 살펴보고, 3장에서는 본 논문에서 사용하는 SVM에 대해서 간략히 설명하고, 4장에서는 SVM 모델을 사용하여 절들의 의존관계를 밝히는 알고리즘에 대해서 설명한다. 5장에서는 본 연구에 사용된 말뭉치(corpus)에 대해서 설명하고, 절들 사이의 의존관계 인식에 대한 실험 결과를 보인다. 마지막으로 6장에서는 결론과 향후 연구에 관해 논하는 것으로 구성되어 있다.

2. 관련연구

문장(sentence)은 하나로 통합된 사상이나 느낌을 글자로 기록하여 나타내는 단어의 집합으로 형태론적으로 하나의 서술어와 서술의 대상으로 구성된다. 그러나 문장을 구성하는 최소의 요건은 하나의 종결을 의미하는 서술어만 존재하면 됨으로 한 문장에는 여러 개의 서술어가 나타날 수 있다. 실제로 사람들이 사용하는 문장은 서술어가 하나인 단문 보다 서술어가 여러 번 나타나는 복문을 많이 사용한다. 이것이 구문 분석과정을 복잡하게 만드는 요소로 크게 작용함으로 하나의 문장에서 여러 번 나타나는 서술어들간 즉 종속절을 인식하고 이들의 의존관계를 분석함으로써 구문 분석의 어려움을 현저하게 줄일 수 있다.

종속절의 의존관계를 인식하는 연구는 종속절을 인식(clause identification)하는 연구와 문장의 의존구조(dependency structure)를 밝히는 연구에서 주로 있어 왔다. 종속절을 인식하는 연구는 각각의 종속절의 시작과 끝 지점을 인식하기 위해 절들의 포함관계를 고려해서 처리하는 수준에서 진행되고 있다. 그리고 문장의 의존 구조를 밝히는 연구에서는 처리 대상이 어절 단위로 각각의 어절들의 의존관계를 밝혔다. 의존 관계를 밝히는 단계로는 먼저 각 어절을 명사구와 동사구의 의존관계를 밝히고 이들 구사이의 의존관계를 밝혀 절을 구성하였다. 마지막으로 절들간의 의존관계를 밝히는 단계로 진행되어 왔다.

종속절을 인식하는 연구에서 절들간의 의존관계는 주된 연구 대상이 아니라 절의 경계 즉 시작점과 끝지점을 인식는 과정에서 절들의 의존 관계 정보를 사용하였다. 특히 2001년 CoNLL(Conference on Computational Language Learning)에서는 shard task로 종속절을 인식하는 다양한 연구들이 있었다. Carreas는 Boosting Tree를 사용하였고[2], Molina는 HMM(Hidden Markov Model)을 사용하여 종속절을 인식하였다[3].

문장의 의존구조에 관한 연구는 예전부터 다양하게 있어 왔으나 최근 들어 다양한 기계 학습 방법을 사용하여 진행되어 왔다. 유럽 언어권에서는 Klein은 코퍼스(Corpus)에서 문법 구조(Syntactic Structure)를 유도함으로써 의존성을 파악했다[4]. 특히 일본어에서는 Uchimoto가 Maximum Entropy Model을 사용하여 다양한 자질(Feature)를 사용하여 의존 구조를 밝혔고, 특히 각각의 자질들이 의존구조에 어떤 영향을 주는지 각각 실험하였다[5]. Kudo는 의존 구조를 이진 분류 문제로 정의하고 Support Vector Machine을 사용하였다. 기계학습에 사용된 자질은 정적인 자질(Static feature)와 동적인 자질(Dynamic feature)로 나누어 처리하였다. 정적인 자질은 의존관계에 있는 각각의 단어와 POS Tag 등의 문법적 자질 뿐만 아니라 의미역(Semantic Role) 같은 의미정보을 사용하였고, 동적인 자질은 각 체언의 조사(Functional word)와 서술어의 어미의 활용(inflection)을 사용했다[6]. Gao는 의존 구조를 사용하여 의존 언어 모델(dependency language model)을 비지도 학습(unsupervised learning)을 사용하여 만들고, 이를 사용하여 의존 관계를 밝혔다[7]. Utsuro는 구 묶음(chunk)에 소속된 마지막 단어의 기능어휘(functional word)에 따라 몇 개의 종류(type)로 나누고 이를 이용하여 의존관계를 밝혔다[8].

한국어를 대상으로 한 연구는 의존 문법을 기반으로 조사와 어미를 구분하여 각각의 어절을 지배소와 의존소로 나누고 이들간의 의존관계를 구분하는 구문 분석이 있었을 뿐이고, 이들은 규칙 기반으로 구문분석을 수행했다[9]. 그리고 종속절을 인식하거나 의존관계를 분석하는 연구는 없었고 주로 규칙 기반으로 복합문에서 단문을 인식하고, 생략된 문장 성분을 복원하는 연구들이었다. 김광진은 내포문을 대상으로 의미 표지 테이블과 용언의 하위범주 정보를 사용하여 생략된 성분을 복원하고 대용어를 처리하였고[10], 김미진은 한국어 복합문에서 영대용어를 처리하기 이전 단계로 문장의 유형에 따른 규칙을 이용하여 종속절을 인식하였다[11]. 기계학습방법으로 이현주는 SVM을 사용하여 문장에서 종속절의 시작과 끝 지점을 인식하는 절 인식 관해 연구하였다[12]. 그러나 이들 논문은 절 인식 및 구의 단위화에 관한 연구들이고 절들간의 의존관계를 밝히는 연구는 없었다. 한국어에서 기계학습 방법을 사용한 다양한 연구가 존재하지 않는 것은 기계학습에 필요한 말뭉치가 없다는 것이다.

앞에서 살펴본 연구에서는 종속절들만의 의존관계를 파악하는 것이 아니라, 각각의 어절 및 구 단위에서 의존관계를 인식하거나, 종속절의 경계를 인식하는 수준에서 연구가 진행되고 있다. 본 논문에서는 먼저 절을 구성하는 요소들을 구로 보고, 이들을 구성하고 있는 절의 경계가 인식되어 있는 상황에서 절들의 서술부의 표면적인 정보를 바탕으로 절들간의 의존관계를 파악하였다. 따라서 학습 데이터로는 구문 분석 말뭉치에서 절들의 시작과 끝을 표시하고 절들간의 의존 관계를 밝힌 형태로 변환하여 학습 데이터로 사용하였다.

3. Support Vector Machine

Support Vector Machine(SVM)은 Vladimir Vapnik이 제안한 기계 학습 방법으로 이진 분류 문제(Binary class classification Problem)를 해결하는데 좋은 성능을 발휘한다고 널리 알려져 있다. 자연어 처리 분야에서는 스팸 메일 필터링(Spam Mail Filtering), 구묶음(Chunking), 문서 분류(Text Categorization) 등의 다양한 분류 문제에 널리 적용되어 왔다.

SVM을 사용하여 분류 작업을 수행하기 위해서는 먼저학습 데이터를 구성해야 한다. 학습 데이터는 학습 데이터를 +1의 값을 가지는 긍정 부류(Positive Class)와 -1의 값을 가지는 부정 부류(Negative Class)로 구성되고 아래와 가정할 수 있다

$$(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_i, y_i), \cdots, (\mathbf{x}_l, y_l)$$
$$\mathbf{x}_i \in \mathbf{R}^n, \quad y_i \in \{+1, -1\}$$

$\mathbf{x}_i$는 n차원 벡터($\mathbf{x}_i = (f_1, \ldots, f_n) \in \mathbf{R}^n$)로 표현된 i번째 데이터이고 y_i는 i번째 데이터의 레이블이다. 이와 같이 학습 데이터가 결정되면 이 문제는 패턴 인식 문제로 볼 수 있고, 결정 함수(Decision Function)는 $f: \mathbf{R}^n \rightarrow \{+1, -1\}$과 같은 형태가 된다.

SVM의 기본 구조를 살펴보면, SVM은 학습 데이터에서 긍정례(Positive Example)와 부정례(Negative Example)를 둘로 나누는 선형적인 초평면(Linear hyperplane)을 찾는다. 이때 초평면은 아래 수식 (1)과 같이 정의된다.

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0 \qquad \mathbf{w} \in \mathbf{R}^n, \; b \in \mathbf{R} \tag{1}$$

위에 식에 따라 긍정례를 구분하는 초평면과 부정례를 구분하는 초평면을 각각 정의 하면 다음 식 (2,3)과 같다.

$$(\mathbf{w} \cdot \mathbf{x}_i) + b \geq +1 \qquad if\,(y_i = +1) \tag{2}$$
$$(\mathbf{w} \cdot \mathbf{x}_i) + b \geq -1 \qquad if\,(y_i = -1) \tag{3}$$

위의 식에서 (2)는 긍정례, (3)은 부정례를 나누는 초평면이고, 다시 이들 식들을 하나의 식으로 다시 쓰면 식(4)와 같다.

$$y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1 \qquad (i = 1, \cdots, l) \tag{4}$$

위의 수식들에 따르면 부정례와 긍정례를 구분하는 초평면은 무수히 많이 존재하는데, 이들 초평면들 중에서 최적의 초평면(Optimal hyperplane)은 긍정례와 부정례를 구분하는 거리(margin)가 최대가 되는 초평면으로 정의할 수 있다. 그림 1은 긍정례와 부정례를 나누는 초평면들 중에서 초평면들 사이의 거리(d)가 최대인 최적의 초평면을 보여주고 있다.

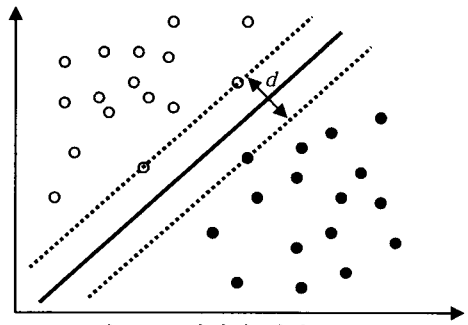

그림 1. 초평면의 거리(Margin)

최적의 초평면은 각 분류의 거리를 사용하여 결정하는데, 특정 지점 $\mathbf{x}_i$에서 거리는 아래의 수식과 같이 정의된다.

$$d(\mathbf{w}, b; \mathbf{x}_i) = \frac{|\mathbf{w} \cdot \mathbf{x}_i + b|}{\|\mathbf{w}\|}$$

특정 지점에서의 거리를 포함하는 초평면의 거리는 위의 식을 사용하여 구할 수 있는데, 긍정례와 부정례를 구분하는 최적의 초평면이 가지는 거리는 다음과 같이 표현된다.

$$\min_{\mathbf{x}_i; y_i = 1} d(\mathbf{w}, b; \mathbf{x}_i) + \min_{\mathbf{x}_i; y_i = -1} d(\mathbf{w}, b; \mathbf{x}_i)$$
$$= \min_{\mathbf{x}_i; y_i = 1} \frac{|\mathbf{w} \cdot \mathbf{x}_i + b|}{\|\mathbf{w}\|} + \min_{\mathbf{x}_i; y_i = -1} \frac{|\mathbf{w} \cdot \mathbf{x}_i + b|}{\|\mathbf{w}\|}$$
$$= \frac{2}{\|\mathbf{w}\|}$$

위의 식에서 거리를 최대가 되기 위해서는 $\|\mathbf{w}\|$가 최소가 되어야 하는 것을 알 수 있다. 즉, 분모가 최소가 되어야 최대가 되어야 하고, 이는 어떻게 분모를 최소화 할 수 있는가 하는 최적화 문제(Optimization

problem)를 푸는 것과 같은 것이 된다 즉 긍정례와 부정례를 나누는 초평면-식(4)-에서 다음 식을 최소화 하는 최적화 문제가 된다.

$$L(w)=\frac{1}{2}\|\mathrm{w}\|^2$$

이 최적화 문제를 라그랑제 멀티플라이어(Lagrange Multiplier)로 정리하면 아래의 수식(5)를 최대화 하는 것으로 변환되고, 이때 $\alpha_i \neq 0$ 인 x_i를 Support Vector(SV)라고 한다.

$$L(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\alpha_i\alpha_j y_i y_j(\mathrm{x}_i,\mathrm{x}_j) \qquad (5)$$

SV를 사용하여 긍정례와 부정례를 구분하는 최적의 초평면을 구성하는 w와 b를 다시 표현하면 아래의 수식과 같다.

$$\mathrm{w}=\sum_{i,\mathrm{x}_i\in SVs}\alpha_i y_i\,\mathrm{x}_i \qquad b=\mathrm{w}\cdot\mathrm{x}_i-y_i$$

마지막으로 긍정례와 부정례를 나누는 초평면에 SV가 놓여져 있다면 결정함수(decision function)는 다음과 같이 정리된다.

$$\begin{aligned} f(x)&=\mathrm{sgn}\Big(\sum_{i;\mathrm{x}_i\in SVs}\alpha_i y_i(\mathrm{x}_i\cdot\mathrm{x})+b\Big)\\ &=\mathrm{sgn}(\mathrm{w}_i\cdot\mathrm{x}+b)\end{aligned} \qquad (6)$$

4. 한국어 종속절의 의존관계 분석

이 장에서 한국어 종속절의 의존관계를 밝히기 위한 확률 모델을 일반화하고 의존관계를 분석하기 위한 방법을 소개한다. 그리고 학습 데이터의 구성 및 실험에 사용한 각종 자질에 관해서 설명하고 전반적인 시스템에 관해서 설명한다.

4.1. 확률 모델(Probability Model) 및 학습 데이터

문장에 속한 종속절들의 집합인 C는 $\{c_1, c_2, ..., c_n\}$으로 구성되고, 의존 관계의 패턴 집합 D는 {Dep(1), Dep(2), ..., Dep(n-1)}로 정의된다. 이때 Dep(i)=j는 절 C_i가 C_j와 의존관계에 있다는 것을 의미하고, 이 것은 D가 가장 마지막 절을 제외한 모든 절의 의존관계가 형성되는 절은 항상 오른쪽에 나타나는 것을 제약(Constraint)로 삼고 있다.

입력으로 C가 주어졌을 때 의존 관계 패턴인 D를 결정하는 문제는 P(D|C)가 최대되는 D를 찾는 문제로 볼 수 있고 이는 아래의 수식과 같이 정의할 수 있다

$$D_{best}=\arg\max_{D}P(D\mid C)$$

만약 의존 관계 패턴인 D가 독립이라고 가정하면 **P(D|C)**는 아래의 식과 같아진다. P(***Dep***(i)=j|f_{ij})의 의미는 f_{ij}가 주어졌을 때 c_i와 c_j사이의 의존 관계가 존재하는 확률을 의미한다. 여기서 f_{ij}는 *n*차원 벡터로서 c_i와 c_j사이의 의존관계를 판단하는데 사용되는 다양한 언어적 및 표면적인 자질(feature)를 의미한다.

$$P(D\mid C)=\prod_{i=1}^{m-1}P(Dep(i)=j\mid \mathrm{f}_{ij}\})$$
$$\mathrm{f}_{ij}=\{f_1,...,f_n\}\in\mathrm{R}^n$$

SVM에서 분류 작업을 수행하기 위해서는 긍정례와 부정례의 학습 데이터가 필요하다. 학습 데이터의 구성은 아래의 식과 같이 왼쪽에서 오른쪽으로 진행하면서 모든 절들간의 의존 관계를 살펴가면서 의존관계에 있는 절은 긍정례로 그렇지 않은 관계는 모두 부정례로 삼아서 학습 데이터로 삼았다.

$$\bigcup_{\substack{1\le i\le m-1\\ i+1\le j\le m}}(\mathrm{f}_{ij},y_{ij})=\{(\mathrm{f}_{12},y_{12}),(\mathrm{f}_{23},y_{23}),...,(\mathrm{f}_{m-1\,m},y_{m-1\,m})\}$$

$$\mathrm{f}_{ij}=\{f_1,...,f_n\}\in\mathrm{R}^n$$
$$y_{ij}\in\{Dep(+1),\ Not-Dep(-1)\}$$

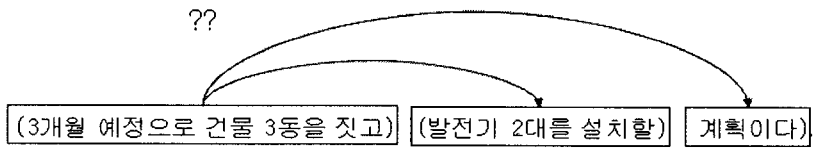

그림 2. 종속절의 의존관계의 예

위의 수식에 따른 예를 아래 그림 2에서 살펴보면, 가장 왼쪽에 있는 "~동을 짓고" 라는 절은 " 발전기 2대를 설치할" 과 "계획이 있었다"라는 절과 의존관계를 각각 검사해서 의존관계가 있는 절을 긍정례로 아닌 경우로 부정례로 삼았다. 좀더 구체적으로 예를 들어 보면 표 2와 같다.

"3개월 예정으로 건물 3동을 짓고" 인 절의 의존 관계는 아래 표 1과 같이 하나의 쌍들로 표현 가능하고, 이들 쌍 중에서 의존관계가 발생하는 하나의 절을 결정하는 것이다. 이때 왼쪽에 존재하는 절을 의존절이라 하고 의존절을 지배하는 절을 지배절이라고 부르기로 한다.

번호	의존절	지배절	구분
1	3개월 예정으로 건물 3동을 짓고	발전기 2대를 설치할	부정
2	3개월 예정으로 건물 3동을 짓고	계획이다	긍정

표 1. 학습 데이터 생성의 예

4.2. 문제 정의 및 자질 선택

절(Clause)은 서술어와 서술대상으로 구성되고, 이들 중에서 절과 절을 이어주는 역할을 하는 연결 어미들은 용언과 결합되어 서술어를 이룬다. 절은 어미의 활용과 사용에 따라 다양한 종류로 나누어 진다.

본 논문에서는 다양한 종류들 절들 중에서 연결어미를 가지는 절들만을 의존 관계의 대상으로 삼았고, 절과 절들의 의존관계는 서술어와 서술어의 연결관계로 가정했다. 따라서 절들의 의존관계는 다음의 제약(Constraints)을 가진다.

1. 가장 오른쪽 절을 제외한 절들 중에서 의존절은 반드시 어미가 연결어미로 구성된 절이 된다.
2. 의존절은 반드시 하나의 지배절을 가진다.
3. 지배절은 가장 왼쪽의 절을 제외한 모든 절이 지배절이 될 수 있다.
4. 지배절은 여러 개의 의존절을 가질 수 있다.

위의 제약 1과 2는 의존절은 연결어미로 연결되고 연결되는 절이 하나라는 것을 의미하고 3은 지배절은 연결어미를 가지는 절과 전성어미 및 종결어미를 가지는 절들도 지배절이 가능하다는 것을 의미한다. 제약 4는 여러 개의 의존절을 가질 수 있다는 것을 의미한다.

본 논문에서 사용한 자질은 크게 정적 자질(static feature)와 동적 자질(dynamic feature)로 나누어 진다. 정적인 자질은 표1 에서 1번과 같은 지배절과 의존절이 있을 때 의존절의 동사와 어미와 그리고 지배절의 동사와 어미를 각각 추출하고 이들의 어휘, POS 태그, 문장에서 용언의 위치 및 거리를 말한다. 이들을 정적 자질로 삼은 이유는 상황에 따라 변하지 않고 고정적으로 나타나기 때문이다.

<table>
<tr><td rowspan="2">정적자질
(Static
Feature)</td><td>의존절
및
지배절</td><td>의존절
동사 어휘
동사 태그
어미 어휘
어미 태그
지배절
동사 어휘
동사 태그
어미 어휘
어미 태그</td></tr>
<tr><td>절들간
거리</td><td>의존절
절의 위치(Position)
지배절
절의 위치</td></tr>
<tr><td colspan="2">동적자질
(Dynamic Feature)</td><td>절들간의
문법적 구성관계</td></tr>
</table>

표 2. 자질의 구성

표 2는 본 논문에서 사용한 자질을 간단하게 보여주고 있는데 여기서 동적 자질은 절들의 의존 관계가 형성되는 상황에 따라 변화하는 자질을 의미한다. 동적으로 변화하는 문법적 상황의 구현은 표 3에서와 같은 간단한 문법 규칙을 기반으로 CKY 차트 파서로 구현하였다. 여기서 ET는 명사형 전성어미 및 관형형 전성어미로 끝나는 절들을 의미하고 EF는 종결 어미를 포함하는 절을 의미한다. 그리고 VP는 연결 어미가 포함된 절을 뜻한다.

ET → ET ET
VP → ET VP
VP → VP VP
ET → VP ET
VP → VP EF
VP → ET EF

표 3. 차트 파싱에 사용된 규칙들

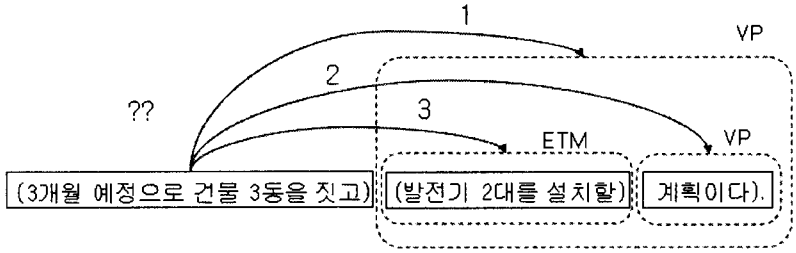

그림 3. 동적 자질을 사용한 의존관계 결정의 예

그림 3은 의존 관계를 결정의 예를 보여주고 있다. 정적 자질만을 사용할 때는 그림 2의 예를 보여주고 있는 표 1의 경우처럼 2가지 경우만 고려하면 되었다. 그러나 동적 자질을 사용할 경우에는 그림 3과 같이 3가지 경우로 확대하여 학습 및 실험에 사용하였다.

예들 들어 설명하면 그림 3에서 정적 자질은 동일하나 동적 자질은 1은 "VP->ET VP", 2번은 "VP" 마지막으로 3번은 "ET" 이렇게 각각 다르게 추출된다.

5. 실험 및 분석

5.1. 말뭉치(Corpus)

본 논문에서 사용한 말뭉치는 STEP2000과제의 결과물인 구문 구조 부착 말뭉치를 변형하여 만들었다. 다음

```
(S
  (ADJP
    (VP
      (NP (NP
        3/nnc/B-NP 개월/nbu/I-NP
          ) 예정/ncpa/I-NP
      ) 으로/jca/I-NP
      (VP
        (NP
          (NP
            건물/ncn/B-NP
          ) 3/nnc/I-NP 동/nbu/I-NP
        ) 을/jco/I-NP 짓/pvg/B-VP
      )
    ) 고/ecc/I-VP
    (ADJP   (ADJP
        (NP
          (VP
            (NP
                발전기/ncn/I-NP
                2/nnc/I-NP 대/nbu/I-NP
            ) 를/jco/I-NP
            설치/ncpa/B-VP 하/xsv/I-VP
            ) 르/etm/I-VP
            계획/ncpa/B-NP
          ) 이/jcs/I-NP
      있/paa/B-VP
      )
      (AUXP   었/ep/I-VP )
    )
  ) 다/ef/I-VP ./sf/O
)
```

그림 4. STEP2000 구문구조 부착 말뭉치

그림 4는 STEP2000구문 부착 말뭉치의 모습이다.

이 구문 구조 부착 말뭉치는 이진 트리 구조로 이루어져 있으며, 이로부터 절들의 경계 및 의존관계 정보를 추출하였다. 이 말뭉치는 12,085개의 문장과 50,658개의 절 111,658개의 구, 321,328개의 단어로 구성되어 있다 절의 개수는 서술어의 개수를 기준으로 한 것으로 평균적으로 한 문장이 4개의 절로 구성되어 있음을 알 수 있다. 본 논문의 실험은 이들 문장에서 하나 이상의 연결 어미를 포함하고 있는 문장 6,934개를 추출하고 이들간의 의존 관계를 실험하였다. 그림 5는 본 논문에서 사용한 말뭉치의 형태이다. 여기서 절의 경계는 CoNLL 2001에서 사용된 표시 형식을 따랐다.

번호	단어	품사	구 단위화	절 경계		의존절 번호
				시작	끝	
1	3	Nnc	B-NP	S	X	
2	개월	Nbu	I-NP	X	X	
3	예정	Ncpa	I-NP	X	X	
4	으로	Jca	I-NP	X	X	
5	건물	Ncn	B-NP	X	X	
6	3	Nnc	I-NP	X	X	
7	동	Nbu	I-NP	X	X	
8	을	Jco	I-NP	X	X	
9	짓	Pvg	B-VP	X	X	
10	고	Ecc	I-VP	X	E	22
11	발전기	Ncn	B-NP	S	X	
12	2	Nnc	I-NP	X	X	
13	대	Nbu	I-NP	X	X	
14	를	Jco	I-NP	X	X	
15	설치	Ncpa	B-VP	X	X	
16	하	Xsv	I-VP	X	X	
17	르	Etm	I-VP	X	E	0
18	계획	Ncpa	B-NP	X	X	
19	이	Jcs	I-VP	X	X	
20	있	Paa	B-VP	X	X	
21	었	Ep	I-VP	X	X	
22	다	Ef	I-VP	X	E	0

그림 5. 학습 및 실험을 위해 변형한 말뭉치

그림 5에서 첫 번째 열은 단어의 순서를 의미하고, 두 번째 열은 단어를, 그 다음은 품사를 표현하고 있다. 여기에 사용된 품사는 KAIST 테그셋[13]을 따른 것으로 52개의 품사로 표현되어 있다. 네 번째는 구 단위 정보를[1] 다섯 번째와 여섯 번째는 절의 시작점과 끝지점에 관한 정보로 시작점인 경우는 'S' 로 끝은 'E' 로 표시하였다. 시작점도 아니고 끝도 아닌 점은 'X' 로 표현하였다. 마지막으로 의존절 번호는 표시된 절이 의존 관계가 있는 서술어의 단어의 위치를 표시하고 있는데 '0' 은 의존관계가 없음을 의미하고 '0' 을 제외한 다른 번호는 의존관계가 있는 절의 위치정보를 표시하고 있다.

5.1. 실험 및 평가

실험에서는 절의 경계가 인식된 말뭉치에서 각 절들의 서술부의 동사와 어미를 추출하여 사용하였기 때문에 각 절을 인식에 관련된 재현율(Recall)과 정확율(Precision)은 측정할 수 없음으로 평가 기준을 정확도(Accuracy)를 기준으로 삼았다.

$$Accuracy = \frac{\text{정확하게 추정한 절의 수}}{\text{전체 절의 수}}$$

Step2000 말뭉치에서 하나이상의 연결어미를 포함한 문장의 수는 6,934개이고 여기서 90%의 문장인 6,240개를 학습에 사용하였고 나머지 10%에 해당되는 694개의 문장을 실험에 사용하였다.

구 분		수
전체 문장의 수		6,934
전체 절의 수		28,872
	전성, 종결어미를 가진 절	18,478
	연결어미를 가진 절	10,394

표 4. 실험에 사용한 말뭉치의 구성

실험에 사용한 말뭉치에 포함된 전체의 절의 수는 28,872개이고 이들 중에서 의존관계를 가질 수 없는 전성 어미 및 종결어미를 포함한 절은 전체의 약 63%를 차지고 하고 있다. 이와 같이 연결어미를 포함 절의 수가 높은 이유는 실험에 사용하기 위해 하나 이상의 연결어미를 포함한 절을 가지고 있는 문장만 추출하였기 때문이다. 표 4에서 실험을 수행한 말뭉치의 구성을 보여주고 있는데, 여기서 연결어미를 가진 절 만이 의존절이 될 수 있고, 지배절은 모든 절이 될 수 있다.

실험에서 사용한 SVM은 SVM ^{Light}[14]를 사용하였다. 실험은 지배절과 의존절의 어미의 정보만 사용한 경우, 정적 자질을 사용한 경우와 동적 자질을 포함한 모든 자질을 포함한 경우로 각각 나누어 실험하였다. 실험 결과는 표 5와 같다.

사용한 자질들	정확도(%)
기준점(Base Line)	57.5
의존절 및 지배절의 어미 어휘 및 Tag	64.4
모든 정적 정보 사용	68.59
+ 동적 자질 정보	73.5

표 5. 종속절의 의존관계 분석 실험 결과

표 5에서 나타난 정확도는 후보들 중에서 가장 큰 마진 값을 가진 하나의 절을 선택했을 때 선택된 절과의 의존 관계가 있는 것을 측정한 값을 의미한다. 기준점(Base Line)은 모든 절의 의존관계가 바로 옆의 경우를 의미한다. 예를 들면 가장 첫 번째 절이 두 번째 절을, 두 번째가 세 번째 절과 의존관계가 있는 것이다. 실험 결과 의존절 및 지배절의 어미의 정보만 사용한 경우는 64.4%이고 모든 자질을 사용한 경우는 73.5%로 나타났다. 이는 각 절의 어미들이 종속절의 의존 관계에 어느 정도의 영향력을 발휘하고 있다는 것을 알 수 있다. 동적 정보까지 모든 정보를 사용한 경우는 73.5%의 결과를 보여주고 있다. 이는 구문 정보를 담고 있는 동적 자질이 많은 정보를 담고 있다는 것을 의미한다.

다른 언어인 유럽 언어권 및 일본어를 대상으로 의존관계 분석의 성능보다 낮게 나타났는데[15] 이는 본 논문에서는 모든 의존관계를 파악하는 것이 아니라 절들의 의존 관계 만을 대상으로 하였기 때문이다. 의존관계 대상을 각각의 어절로 삼고 어절들의 의존관계를 측정하면, 특히 명사구들의 의존관계가 높은 성능이 나올 것으로 예상되고, 이로 인해서 전반적인 성능의 수치는 높게 나온 것이다.

5. 결론 및 향후 연구

본 논문에서는 문장에 소속된 종속절의 의존관계를 인식하는 방법을 제안하였다. 이를 위해 먼저 절을 구성하는 서술어부 즉 동사와 어미의 정보에 따라 의존관계가 성립한다고 가정하였다. 그리고 사용한 자질은 각각의 절의 표면적인 정보인 어휘, POS Tag 및 위치 정보를 사용하였고, CKY 차트 파서를 사용하여 상황에 따른 문법적인 정보를 추출하여 사용하였다.

실험 결과 어휘 정보들만 사용한 경우는 68.59%의 정확도를 보였고 문법적인 정보인 동적 자질을 사용한 경우는 73.5%로 어휘 정보만을 사용한 경우보다 4.91%의 성능 향상됨을 보였다. 향후 연구로 사용된 각종 자질이 성능에 얼마만큼 영향을 주고 있는지 등의 다양한

측정과 보다 높은 성능을 발휘화기 위해서는 본 논문에서 사용한 어휘 정보 및 문법적인 정보들뿐만 아니라 의미적인 정보들이 추가되어야 할 것으로 보인다. 특히 주 동사구에서 주 동사를 인식하는 부분과 인식된 동사의 하위 카테고리(SubCategory) 정보 및 의미역(Semantic Role)등의 정보들이 추가되어야 할 것이다.

참고 문헌

[1] 박성배, 장병탁, "한국어 구 단위화를 위한 규칙 기반 방법과 기억 기반 학습의 결합," 정보과학회 논문지 제 31권 제 3호, pp369-378, 2004.

[2] Xavier Carreras and Luís Màrquez, "Boosting Trees for Clause Splitting," *Proceedings of the CoNLL'2001*, 2001.

[3] Antonio Molina and Ferran Pla, "Clause Detection using HMM," *Proceedings of the CoNLL'2001*. 2001.

[4] Dan Klein and Christopher D.Manning, "Corpus-Based Induction of Syntactic Structure : Models of Dependency and Constituency," *Proceedings of the CoNLL'2001*, 2001.

[5] Kiyotaka Uchimoto, Satoshi Sekine and Hitoshi Isahara, "Japanese Dependency Structure Analysis Based on Maximum Entropy Models," *Proceedings of EACL' 99*, 1999.

[6] Taku Kudo and Yuji Matsumoto, "Japanese Dependency Structure Analysis Based on Support Vector Machines," *Proceedings of the EMLNP*, PP. 18-25, 2000.

[7] Jianfeng Gao and Hisnmi Suzuki, "Unsupervised Learning of Dependency Structure of Language Modeling," *Proceedings of the ACL'03*, 2003.

[8] Takehito Utsuro, Shigeyuki Nishiokauama, Masakazu Fujio and Yuji Matsumoto, "Analyzing Dependencies of Japanese Subordinate Clauses based on Statistics of Scope Embedding Preference," *Proceedings of ACL' 00*, PP.110-117, 2000.

[9] 서광진, *어절 사이의 의존관계를 이용한 한국어 구문 분석기*, 한국 과학기술원 석사학위 논문, 1993.

[10] 김광진, 송영훈, 이정현, "한국어 내포문을 단문으로 분리하는 시스템의 구현," *제 5회 한글 및 한국어 정보처리 학술발표 논문집*, PP.25-34, 1993.

[11] 김미진, *한국어 복합문의 Zero Anaphpra 처리를 위한 분해 및 복원 알고리즘*, 경북대학교 대학원 컴퓨터공학과 박사학위논문, 2003.

[12] 이현주, 김상수, 박성배, 이상조, "SVM 모델을 이용한 절 경계 인식," *제16회 한글 및 한국어 정보처리 학술 발표 논문집*, PP.151-156. 2004.

[13] 최기선, 남영준, 김진규, 한영균, 박석문, 김진수, 이춘택 김덕봉, 김재훈, 최병진, "한국어 정보베이스를 위한 형태, 통사 태그 표준에 관한 연구," 인지과학, 제7권 제4호, PP43-61, 1996.

[14] T. Jochims, "Making large-scale SVM learning practical Technical Report LS8," Universitat Dortmund, 1998.

[15] Takeshi Abekawa and Manabu Okumura, "Japanese Dependency Parsing Using Co-occurrence Information and a Combination of Case Elements," *Proceedings of ACL' 06*, PP.833-840, 2006.