□ 기술해설 □

# 자연어 처리를 위한 품사 태깅 시스템의 고찰

고려대학교 임해창* · 임희석** · 이상주** · 김진동**

● 목 차 ●



## 1. 서 론

인간이 사용하는 언어를 컴퓨터를 이용하여 처리하고자 하는 자연어 처리에 대한 연구는 크게 통계 기반 접근방법(Statistical Approach)과 규칙 기반 접근방법(Rule-Based Approach),[1] 그리고 통계 기반 접근방법과 규칙 기반 접근방법을 통합한 통합 접근방법(Hybrid Approach)으로 구분할 수 있다[15, 58].

통계 기반 접근방법은 실세계 자연어 용례들과 부속 정보를 포함하는 대량의 원시(Raw) 또는 태깅된(Tagged) 코퍼스(Corpus)를 분석하고 자연어에 대한 통계 정보를 추출하여 얻은 확률(Probability) 또는 불확실성(Uncertainty)을 이용하여 자연어처리의 중의성 문제를 확률적으로 해결하는 방법이다. 통계 기반 접근방법은 확률 또는 불확실성을 나타내는 통계 정보를 사용하므로 거의 모든 언어 현상에 적용할 수 있지만, 실세계 언어 현상을 충분히 대표할 수 있는 양과 질의 코퍼스가 존재하지 않아 데이터 부족 문제(Data Sparseness Problem)로 인해 정확도가 저하된다.

한편, 규칙 기반 접근방법은 자연어에 적용되는 공통적인 원리(Common Principle)나 결정적인 규칙(Deterministic Rule)을 찾아내고, 이를 이용하여 자연어처리의 중의성 문제를 결정적으로 해결하는 방법으로 지식 기반 방법(Knowledge-Based Approach)이라고도 한

*종신회원
**학생회원

1) 일반적으로 통계 기반 접근방법을 경험주의(Empiricism), 데이터 집약적(Data-Intensive; Data-Driven) 방법 또는 코퍼스 기반 방법(Corpus-Based Approach)이라고 하고, 규칙 기반 접근방법을 합리주의(Rationalism)이라고 분류해 왔으나, 최근의 자연어처리 연구 추세가 주로 코퍼스 중심으로 이루어지고 있고 대부분 데이터 집약적인 방법을 사용하고 있어, 경험주의와 합리주의 개념으로 통계 기반 접근방법과 규칙 기반 접근방법을 구분하는 것은 불합리하다. 본 논문에서는 통계 기반 접근방법과 규칙 기반 접근방법을 중의성 해결을 위해 사용되는 정보가 확률적인지 또는 결정적인 규칙인지를 기준으로 구분한다.

다. 규칙 기반 접근방법은 일관성 있는 결정적 규칙을 얻기가 어렵고 많은 규칙들을 잘 제어하기가 쉽지 않아 일반적으로 견고하지 못하지만, 규칙이 적용될 수 있는 현상에 대해서 높은 정확도를 보인다.

통합 접근방법은 대량의 데이터에서 추출한 확률 및 통계 정보와 언어 지식을 모두 사용함으로써 통계 기반 접근방법과 규칙 기반 접근방법의 장단점을 상호 보완하고자 하는 방법이다.

품사 태깅(Part-of-Speech Tagging)이란 문장내 각 단어에 해당하는 품사를 할당하는 작업으로, 원시 코퍼스에 부속 정보로서 품사 정보를 추가하여 태깅된 코퍼스를 구축하기 위한 것이다. 많은 단어가 중의성을 포함하므로, 품사 태깅에는 중의성 해결(Disambiguation) 과정이 반드시 포함된다. 품사 태깅은 어휘적 중의성으로 인한 구문 분석 단계에서의 과다한 부담을 줄이기 위해 구문 분석의 전처리 과정으로 사용되며, 정보 검색 시스템에서 높은 재현률 및 정확도를 갖는 색인어와 검색어 추출을 위해 사용될 수 있다. 또한 기계 번역, 언어 정보 획득 도구, 철자 검사, 사전 구축 등 자연어 처리의 제반 분야에서 필수적인 과정으로 인식되고 있다.

품사 태깅 시스템은 통계 기반 접근방법을 이용한 시스템[1, 2, 7, 8, 11-16, 18, 20-22, 25-29, 31, 35-37, 40-45, 48-54, 57], 규칙 기반 접근방법을 이용한 시스템[3-6, 19, 20, 39, 59] 그리고 통합 접근방법을 이용한 시스템[23, 38, 46, 47]으로 구분할 수 있다. 통계 기반 접근방법을 사용하는 시스템은 확률 모델을 사용하는 시스템과 신경망 및 퍼지망을 사용하는 시스템 등이 있으며[2, 31, 41, 43, 50], 규칙 기반 접근방법은 태깅하고자 하는 단어의 좌우 문맥을 참조하여 긍정적(Positive) 또는 부정적(Negative)으로 중의성을 해결하는 규칙을 사용하는 중의성 해결 규칙 기반(Disambiguation Rule-Based) 시스템[10, 18-20, 34, 39]과 초기 태깅 오류를 최소화하기 위해 오류를 올바른 태그로 변경시키는 규칙을 사용하는 변형 규칙 기반(Transformation Rule-Based) 시스템 등이 있다[3-6, 59].

또한 최근에는 통계 기반 접근법을 사용한 시스템과 규칙 기반 접근방법을 사용한 시스템을 통합함으로써 광범위한 데이터 처리가 가능하고, 높은 정확도를 갖는 태깅 시스템 개발을 위한 연구가 활발히 진행되고 있다.

본 논문은 자연어 처리를 위한 기존의 품사 태깅 시스템을 태깅에 사용된 방법 및 특징, 품사 집합, 실험 환경 및 정확도, 그리고 각 시스템의 문제점을 중심으로 살펴보고, 이들을 비교한다. 또한 새로운 품사 태깅 시스템 구축 시 반드시 고려하여야 할 사항과 태깅 시스템의 올바른 평가를 위하여 고려되어야 할 사항에 대해서도 살펴보고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 올바른 품사 태깅 시스템 구축 시 고려되어야 할 제반 사항에 대해서 설명하고, 3장에서는 통계 기반 태깅 시스템에 대하여 설명하며, 4장에서는 전통적인 규칙 기반 태깅 시스템과 변형 규칙 기반 태깅 시스템에 대하여 살펴보기로 한다. 또한 5장에서는 통계 기반 접근방법과 규칙 기반 접근방법을 통합한 품사 태깅 시스템에 대해서 설명한다. 6장은 품사 태깅 시스템의 평가에 사용되는 기준에 대하여 설명하고, 마지막으로 7장에서는 품사 태깅 시스템의 향후 연구 방향에 대하여 논의한다.

## 2. 품사 태깅의 고려 사항

품사 태깅 시스템을 구축하고자 할 때 고려해야 할 사항은 다음과 같다.

### 품사 집합(Part-of-Speech Tag Set) 결정 문제

품사 집합 결정은 태깅 시스템 구축 시 가장 먼저 대두되는 사항으로 매우 신중히 다루어져야 한다. 세분화된 품사 집합을 사용할 경우 태깅하고자 하는 대상 언어에서 중의성이 매우 심각하게 발생하지만, 태깅 결과에서 매우 자세한 언어 정보를 추출할 수 있다. 반면 적은 개수의 세분화되지 않은 품사 집합을 사용할 경우 세분화된 품사 집합을 사용하는 경우에 비해 비교적 정확한 품사 태깅이 가능하나 태깅된 결과로부터 추출할 수 있는 정보가 제한

되므로 품사 집합은 품사 태깅 시스템이 사용될 응용 분야와 목적에 따라 알맞게 결정되어야 한다.

### 미등록어(Unknown Word) 처리 문제

미등록어 처리는 태깅 시스템의 견고성을 결정하는 데에 중요한 영향을 미친다. 미등록어란 태깅 시스템 또는 형태소 분석기에서 사용하는 사전에 등록되어 있지 않은 단어라고 할 수 있다. 대용량의 코퍼스에 품사를 태깅하기 위한 경우라면 대상 코퍼스로부터 미등록어 추출기에 의해서 미등록어 후보들을 추출하고, 이를 사람이 편집하여 사전에 등록하여 사용하는 것이 매우 효율적이다. 그러나 태깅 시스템이 실시간 처리를 요하는 다른 시스템과 연계되어 사용되는 경우, 태깅하고자 하는 언어의 특성을 고려한 미등록어 처리 기능이 형태소 분석 시스템 또는 품사 태깅 시스템에 반드시 필요하다. 기존의 품사 태깅 시스템에서 사용하는 미등록어 처리 방법에 대해서는 각 시스템을 설명할 때 다루기로 한다.

### 자료 부족(Data Sparseness) 문제

자료 부족 문제란 미등록어 문제와는 달리 특정 형태소, 단어 또는 어절에 대한 품사 정보가 전자 사전에는 있으나 이에 관련된 어휘 정보 또는 문맥 정보과 같은 통계 정보를 학습 코퍼스로부터 추출할 수 없거나, 아주 적은 빈도로 존재하기 때문에 추출된 통계 정보의 신뢰도가 아주 낮은 경우를 말하며, 태깅 시스템의 견고성을 좌우하는 요인이다. 예를 들어, will은 '할 수 있다'의 의미를 갖는 '조동사' 품사와 '유언'의 의미인 '명사' 품사를 갖는데, will이 '유언'의 의미로 사용된 예가 학습 코퍼스에 한 번도 나타나지 않을 경우 P(will|명사)에 대한 정보를 얻을 수 없다.

일반적으로 자료 부족 문제는 학습 코퍼스가 다양한 언어 현상을 모두 반영하지 못하기 때문에 발생된다. 따라서 견고성을 갖기위한 품사 태깅 시스템은 자료 부족 문제를 극복할 수 있는 해결 방법(Smoothing)을 가지고 있어야 하며, 학습에 대량의 균형있는 코퍼스(Balanced Corpus)를 사용하는 것이 바람직하다.

### 학습 방법 결정

품사 태깅 시스템의 학습 방법은 자율 학습(Unsupervised Learning)과 지도 학습(Supervised Learning)으로 나눌 수 있다. 지도 학습은 태깅된 코퍼스를 이용하여 통계 정보나 규칙 정보를 추출하여 태깅 시스템을 학습시키는 방법이고, 자율 학습은 원시 코퍼스로부터 태깅에 필요한 유용한 정보를 추출하여 태깅 시스템을 학습시키는 방법이다. 일반적으로 지도 학습에 의한 품사 태깅 시스템이 비교적 높은 정확도를 보이나, 품사 태깅 시스템 구축 시에 지도 학습에 필요한 태깅된 코퍼스를 구축해야 하는 어려움이 따른다. 따라서 학습 방법 결정은 품사 태깅 시스템 구축 시 고려되어야 할 매우 중요한 사항이다.

### 응용분야에 적합한 설계

이상적인 품사 태깅 시스템이라면 새로운 품사 집합이나 새로운 코퍼스에 대한 적응성(Adaptability)이 뛰어나야 한다. 하지만 현실적으로 새로운 환경에 높은 적응성을 갖는 품사 태깅 시스템을 구축하는 일은 매우 어려운 일이다. 따라서 품사 태깅 시스템은 사용될 응용 분야에 따라 알맞게 설계되어야 한다. 응용 분야에 맞도록 설계된 품사 태깅 시스템은 그 분야에 맞는 영역 지식(Domain Knowledge)을 활용하여 사전을 구축하고 태깅을 위한 정보를 추출함으로써 높은 정확도를 갖을 수 있다.

### 한국어 품사 태깅의 단위

한국어는 교착어로서 굴절어인 영어와는 다른 특성을 갖는다. 교착어란 의미를 나타내는 실질 형태소에 어법적 관계를 나타내는 형식 형태소가 붙음으로써 문법 기능을 하는 언어를 말한다. 굴절어란 실질 형태소와 형식 형태소의 구분이 뚜렷하지 않고, 어형의 변화로 어법 관계를 나타내는 언어를 말한다. 이러한 언어적 특성의 차이 때문에 영어와 한국어에 대한 태깅 양상이 다르게 나타난다. 즉, 영어에서의 품사 태깅은 띄어쓰기의 단위가 되는 단어에 적절한 품사를 할당하는 것이지만(e.g. see →

see/동사), 한국어에서의 품사 태깅은 띄어쓰기의 단위가 되는 어절이 어떤 품사의 형태소로 구성되었는지를 결정하는 것이다(e.g. 본다 → 보/동사+ㄴ다/종결어미). 이러한 차이점 때문에 영어권에서 오랫동안 연구되어온 품사 태깅 모델을 한국어의 품사 태깅에 이용하는 데 많은 어려움이 따르게 된다.

한국어 품사 태깅 모델은 품사 태깅의 단위에 따라 어절 단위 품사 태깅 모델[44, 47, 48, 51, 52]과 형태소 단위 품사 태깅 모델[44, 49, 53, 54, 57]로 나누어진다. 어절 단위(Eojeol-Unit) 품사 태깅 모델은 어절의 문법적 기능(Grammatical Function) 즉, 어떤 품사들의 결합으로 이루어져 있는지만을 결정하는 모델이고, 형태소 단위(Morpheme-Unit) 품사 태깅 모델은 어절의 형태(Form)와 문법적 기능 즉, 어떤 형태소와 품사들의 결합으로 이루어져 있는지를 결정하는 모델이다. 따라서, 어절 단위 모델은 어절의 형태소 분석 결과에 동일한 품사 결합을 나타내는 동품사 중의성[48]을 표현할 수 없는 반면, 어절 단위 문맥을 고려하므로 문맥 확장이 쉬운 장점을 갖고 있고, 반대로 형태소 단위 모델은 형태소 단위 문맥을 고려하므로 문맥 확장이 비교적 힘들지만, 형태소와 품사를 모두 결정하므로 동품사 중의성을 표현할 수 있다.

## 3. 통계 기반 품사 태깅 시스템

통계에 기반한 품사 태깅 방법은 크게 어휘 확률만을 이용하는 방법, HMM(Hidden Markov Model)의 자율 학습을 이용하는 방법, N-gram의 문맥 확률과 어휘 확률을 이용하는 방법으로 분류할 수 있다. 이 밖에도 통계 기반 품사 태깅 방법으로 신경망을 이용하는 방법[2, 31, 35, 41, 50]과 퍼지망을 이용하는 방법[43]이 제안된 바 있으나, 기존의 통계 기반 방법과는 이론적인 배경이 다르기 때문에 본 논문에서는 기존의 통계 기반 방법을 중심으로 설명하고자 한다.

품사 태깅 모델을 확률적인 수식으로 정의하기 위해서, 품사 태깅의 대상 언어가 W개의 유한한 단어(Word) 집합($w^1, w^2, w^3, \cdots, w^W$)으로 이루어져 있고, 우리는 C개의 품사(Category) 집합($c^1, c^2, c^3, \cdots, c^C$)을 사용하여 태깅을 하려 한다고 가정하자.

이 때, 품사 태깅의 문제는 "길이가 N인 단어열(문장) $w_{1,N}=w_1\ w_2\ \ldots\ w_N$이 주어졌을 때, 가장 확률이 높은 품사열 $c_{1,N}=c_1\ c_2\ \ldots\ c_N$을 구하는 것"으로 [식 1]과 같이 정의할 수 있다. [식 1]에서 $w_i$는 문장에서 $i$번째에 나타나는 단어를 나타내며, $c_i$는 $i$번째 단어에 할당되는 품사를 의미한다.

$$T(w_{1,N}) \stackrel{\text{def}}{=} \underset{c_{1,N}}{argmax}\ P(c_{1,N}|w_{1,N}) \qquad [식\ 1]$$

[식 1]은 문장 단위의 통계 정보를 필요로 하는 매개변수(Parameter) $P(c_{1,N}|w_{1,N})$를 가진다. 그러나 자연어에서 문장은 매우 다양한 형태로 발생하기 때문에 문장 단위의 통계 정보를 획득하는 것은 거의 불가능한 일이다.

$$T(w_{1,N}) = \underset{c_{1,N}}{argmax}\ \frac{P(c_{1,N}, w_{1,N})}{P(w_{1,N})} \qquad [식\ 2]$$

$$= \underset{c_{1,N}}{argmax}\ P(c_{1,N}, w_{1,N}) \qquad [식\ 3]$$

이제부터 [식 1]에 대한 적절한 변형을 통하여 이를 통계 정보 획득이 가능한 형태로 만들어 보도록 하자. 먼저 조건부 확률의 정의에 의해 [식 1]을 [식 2, 3]과 같이 변환한다. [식 2]에서 분모의 $P(w_{1,N})$은 모든 $c_{1,N}$에 대해 상수이므로 [식 3]에서 생략되었다. [식 3]을 개별 단어에 대한 확률의 곱으로 나타내기 위해 Chain Rule을 적용할 수 있다. 이 때 $P(w_i)$을 먼저 분리시키느냐 $P(c_i)$을 먼저 분리시키느냐에 따라 두 가지의 변환이 가능하다.

$$\begin{aligned} & P(c_{1,N}, w_{1,N}) \\ &= P(w_1)\ P(c_1|w_1) \\ &\quad P(w_2|c_1, w_1)\ P(c_2|c_1, w_{1,2}) \\ &\quad \cdots \\ &\quad P(w_N|c_{1,N-1}, w_{1,N-1})\ P(c_N|c_{1,N-1}, w_{1,N}) \qquad [식\ 4] \\ &= P(w_1)\ P(c_1|w_1) \end{aligned}$$

$$\prod_{i=2}^{N} P(w_i|c_{1,i-1}, w_{1,i-1})\ P(c_i|c_{1,i-1}, w_{1,i}) \qquad [식\ 5]$$

$$= \prod_{i=1}^{N} P(w_i|c_{1,i-1}, w_{1,i-1})\ P(c_i|c_{1,i-1}, w_{1,i}) \qquad [식\ 6]$$

[식 4, 5, 6]은 $P(w_1)$을 먼저 분리시켰을 경우이고, [식 7, 8, 9]는 $P(c_1)$을 먼저 분리시켰을 경우이다. [식 5]를 [식 6]으로, [식 8]을 [식 9]로 단순화시키기 위해서, $w_{1,0}$을 "단어열(문장)의 시작, $w_0$"으로, $c_{1,0}$을 "단어열(문장)에 대응되는 품사열의 시작, $c_0$"으로 정의한다.

$$P(c_{1,N}, w_{1,N})$$
$$= P(c_1)\ P(w_1|c_1)$$
$$P(c_2|c_1, w_1)\ P(w_2|c_{1,2}, w_1)$$
$$\cdots$$
$$P(c_N|c_{1,N-1}, w_{1,N-1})\ P(w_N|c_{1,N}, w_{1,N-1}) \quad [식\ 7]$$

$$= P(c_1)\ P(w_1|c_1)$$
$$\prod_{i=2}^{N} P(c_i|c_{1,i-1}, w_{1,i-1})\ P(w_i|c_{1,i}, w_{1,i-1}) \quad [식\ 8]$$

$$= \prod_{i=1}^{N} P(c_i|c_{1,i-1}, w_{1,i-1})\ P(w_i|c_{1,i}, w_{1,i-1}) \quad [식\ 9]$$

결과적으로 [식 6]과 [식 9]가 유도되었으며 여기까지는 아직 어떠한 가정도 사용되지 않았다. 그러나 [식 6]과 [식 9]는 여전히 통계 획득이 불가능하며, 이를 통계 획득이 가능한 형태로 만들기 위해서 적절한 가정(Assumption)의 도입이 필요하다. 이 때 어떠한 가정을 도입하느냐에 따라 여러 가지 통계 모델이 유도될 수 있다.

### 3.1 어휘 확률 기반 품사 태깅 시스템

[식 6]에 다음과 같은 마르코프 가정(Markov Assumption)을 도입함으로써 통계에 기반한 가장 간단한 품사 태깅 모델을 유도할 수 있다.

$$P(w_i|w_{1,i-1}, c_{1,i-1}) \cong P(w_i|w_{i-1}) \quad [가정\ 1]$$
$$P(c_i|c_{1,i-1}, w_{1,i}) \cong P(c_i|w_i) \quad [가정\ 2]$$

[가정 1]은 "현재 단어의 발생은 바로 이전의 단어에만 의존한다"라는 것을 가정한 것이며, [가정 2]는 "현재 품사의 발생은 현재의 단어에만 의존한다"라는 것을 가정한 것이다.

$$T(w_{1,N}) = \underset{c_{1,N}}{argmax} \prod_{i=1}^{N} P(w_i|w_{i-1})P(c_i|w_i) \quad [식\ 10]$$

[가정 1]과 [가정 2]를 [식 6]에 대입하고, 그 결과를 [식 3]에 대입하면 다음과 같은 [식 10]이 얻어진다. [식 10]에서 $P(w_i|w_{i-1})$은 모든 $c_{1,N}$에 대해 상수이므로 생략하면 통계에 기반한 첫 번째 품사 태깅 모델인 [모델 1]을 얻을 수 있다.

$$T(w_{1,N}) = \underset{c_{1,N}}{argmax} \prod_{i=1}^{N} P(c_i|w_i) \quad [모델\ 1]$$

[모델 1]은 단어 $w_i$와 함께 가장 많이 쓰이는 품사를 $w_i$의 품사 $c_i$로 결정한다. 이 모델은 단어에 대한 품사 발생 정보만을 고려할 뿐 문맥 정보는 전혀 고려하지 않는 가장 단순한 품사 태깅 모델이라고 할 수 있다.

Charniak 등[11]은 [모델 1]을 적절한 평탄화(Smoothing) 기법과 결합하여 통계 정보 획득에 사용되지 않은 문서에 대한 태깅 실험을 수행한 결과 91.51%의 정확도를 보고하였다.

### 3.2 HMM 기반 품사 태깅 시스템

[식 9]에 다음과 같은 마르코프 가정(Markov Assumption)을 도입함으로써 품사 문맥을 고려하는 품사 태깅 모델을 유도할 수 있다.

$$P(c_i|w_{1,i-1}, c_{1,i-1}) \cong P(c_i|c_{i-1}) \quad [가정\ 3]$$
$$P(w_i|w_{1,i-1}, c_{1,i}) \cong P(w_i|c_i) \quad [가정\ 4]$$

[가정 3]은 "현재 품사의 발생은 바로 이전의 품사에만 의존한다"라는 것을 가정한 것이며, [가정 4]는 "현재 단어의 발생은 현재의 품사에만 의존한다"라는 것을 가정한 것이다.

[가정 3]과 [가정 4]를 [식 9]에 대입하고, 그 결과를 [식 3]에 대입하면 통계에 기반한 두 번째 품사 태깅 모델인 [모델 2]를 얻을 수 있다.

[모델 2]는 품사 문맥 정보와 품사에 대한 어휘 발생 정보를 사용하여 품사 태깅을 수행하는 모델이다.

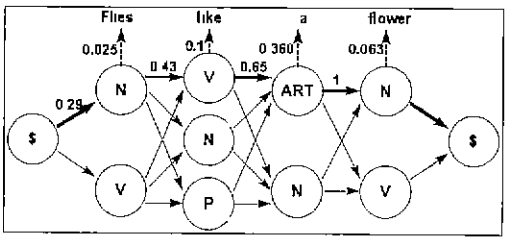



그림 1 영어 문장의 발생을 HMM으로 모델링한 예

$$T(w_{1,N}) = \underset{c_{1,N}}{argmax} \prod_{i=1}^{N} P(c_i|c_{i-1})P(w_i|c_i)$$

[모델 2]

이 모델은 지난 몇 년 동안 실제로 개발된 대부분의 태깅 시스템들의 기본 모델이 되어왔다[11].

Charniak 등[11]은 [모델 2]를 적절한 평탄화(Smoothing) 기법과 결합하여 통계 정보 획득에 사용되지 않은 문서에 대한 태깅 실험을 수행한 결과 95.97%의 정확도를 보고하였다.

또한 [모델 2]는 태깅되지 않은 코퍼스로부터 자율 학습(Unsupervised Learning)을 통해 매개 변수 값을 획득할 수 있다는 특징을 가진다. 이러한 경우에 [모델 2]를 은닉 마르코프 모델(Hidden Markov Model)[32, 33]이라고 부르며, $P(c_i|c_{i-1})$을 상태 전이 확률(State Transition Probability), $P(w_i|c_i)$를 관측 심볼 확률(Observation Symbol Probability)이라고 한다.

Kupiec[21, 22]은 [모델 2]와 같은 HMM을 자율 학습시킬 때, 단어의 발생 형태가 너무 다양하기 때문에 관측 심볼 확률이 안정적으로 학습되기 어려운 문제를 극복하기 위해 의사 부류(Equivalence Class)를 정의하였다. 의사 부류란 중의성이 나타나는 유형을 하나의 부류로 묶은 것이다. 예를 들면, "type"과 "store"는 둘다 Noun-or-Verb 부류에 속하게 된다. Kupiec이 의사 부류를 활용하여 실제로 품사 태깅에 사용한 모델은 [모델 2-1]과 같다. [모델 2-1]에서 $Eqv_i$는 문장에서 $i$번째에 나타나는 단어의 의사 부류를 나타낸다.

$$T(w_{1,N}) = \underset{c_{1,N}}{argmax} \prod_{i=1}^{N} P(c_i|c_{i-1})P(Eqv_i|c_i)$$

[모델 2-1]

Kupiec의 시스템은 42개의 품사 집합을 사용하여 자율 학습을 하였고 Brown 코퍼스에 대해 품사 태깅 실험을 수행한 결과 95.7%의 정확도를 얻었다.

한편, [모델 2]의 문맥 정보를 3-gram으로 확장하면 [모델 3]을 얻을 수 있다.

$$T(w_{1,N}) = \underset{c_{1,N}}{argmax} \prod_{i=1}^{N} P(c_i|c_{i-1}, c_{i-2})P(w_i|c_i)$$

[모델 3]

[모델 3]도, [모델 2]와 마찬가지로, 태깅되지 않은 코퍼스로부터 자율 학습을 통해 매개변수 값을 획득할 수 있다. 이러한 경우에 [모델 3]을 2차 은닉 마르코프 모델(Second Order Hidden Markov Model)이라고 부른다.

Merialdo[27]는 자율 학습(Relative Frequency Training)과 지도 학습(Maximum Likelihood Training)에 대한 비교 실험 결과를 발표하는 논문에서 [모델 3]을 사용하였다. 이 실험에서는 76개의 품사 집합을 사용하였으며 Treebank에 대한 품사 태깅 실험 결과 자율 학습의 경우 86.6%, 지도 학습의 경우 97.0%의 정확도를 나타내었다.

### 3.3 N-gram 기반 품사 태깅 시스템

[모델 2]에서 품사에 대한 어휘 발생 정보인 $P(w_i|c_i)$대신에 어휘에 대한 품사 발생 정보인 $P(c_i|w_i)$를 사용하면 [모델 4]를 얻을 수 있다.

$$T(w_{1,N}) = \underset{c_{1,N}}{argmax} \prod_{i=1}^{N} P(c_i|c_{i-1})P(c_i|w_i)$$

[모델 4]

[모델 4]에서 어휘 확률로서 $P(w_i|c_i)$ 대신에 $P(c_i|w_i)$가 사용된 것은 직관적으로는 타당한 듯 하나 이론적인 근거는 미약하다[11].

Marshall[26]은 LOB 코퍼스의 태깅에 사용된 CLAWS에서 [모델 4]에 기초한 품사 태깅 모델을 사용하였으나, 어휘에 대한 품사 발생 정보로는 통계값이 아니라 1, 1/2, 1/8의 값을 가지는 3단계의 Rarity Marker를 사용하였다. Marshall은 이러한 모델을 바탕으로 130개의 품사 집합을 사용하여 94%의 정확도를 얻었으며, 여기에 숙어 처리를 위한 전처리 시스템을 포함시켜서 97%까지 정확도를 향상시켰다.

DeRose[16]는 CLAWS를 개선시킨 VOLSUNGA에서 [모델 4]와 동일한 모델을 사용하였으며, 동적 프로그래밍(Dynamic Programming) 기법을 사용하여 태깅 수행 시간을 선형 시간(Linear Time)으로 크게 단축시켰다. DeRose는 [모델 4]를 바탕으로 97개의 품사 집합을 사용하여 96%의 정확도를 얻었으며, 여기에 숙어 처리를 포함시켜서 99%까지 정확율을 향상시켰다.

[모델 4]의 문맥 정보를 3-gram으로 확장하면 [모델 5]를 얻을 수 있다.

$$T(w_{1,N}) = \underset{c_{1,N}}{argmax} \prod_{i=1}^{N} P(c_i|c_{i-1}, c_{i-2})P(c_i|w_i)$$

[모델 5]

Church[14]는 [모델 5]와 유사한 모델을 사용하여 Brown 코퍼스에 대한 품사 태깅 실험을 수행한 결과 95~99%의 정확도를 보고하였다.

### 3.4 어절 단위 한국어 품사 태깅 모델

한국어에 대한 품사 태깅 모델을 확률적인 수식으로 정의하기 위해서, 한국어가 M개의 유한한 형태소(Morpheme) 집합($m^1$, $m^2$, $m^3$, …,$m^M$)으로 이루어져 있고, 우리는 $C$개의 품사(Category) 집합($c^1$, $c^2$, $c^3$,…,$c^C$)을 가지고 태깅한다고 가정하자.

한국어에서 띄어쓰기의 단위가 되는 어절(Eojeol) $e$는 형태소 결합 $\overline{m}$과 이에 대응되는 품사 결합 $\overline{c}$의 쌍으로 분석될 수 있다. 하나의 어절 $e$는 하나 이상의 형태소 결합 $\overline{m}$에 대응되고, 하나의 형태소 결합 $\overline{m}$은 하나 이상의 품사 결합 $\overline{c}$에 대응될 수 있다. 문장에서 $i$번째 어절을 $e_i$라 하면, 형태소 결합 $\overline{m}_i$는 어절 $e_i$를 구성하는 형태소의 열(Sequence of Morphemes) $m_{i2}$ $m_{i2}$ … $m_{ik}$를 의미하며, 품사 결합 $\overline{c}_i$는 형태소 결합 $\overline{m}_i$의 각 형태소에 대응되는 품사로 구성된 품사의 열(Sequence of Categories) $c_{i1}$ $c_{i2}$…$c_{iL}$를 의미한다. 여기서 K(K≥1)는 i번째 어절에 대응되는 형태소 결합 $\overline{m}_i$의 형태소 수를, $m_{ij}$는 $i$번째 어절에 대응되는 형태소 결합 $\overline{m}_i$의 j번째 형태소를, $c_{ij}$는 i번째 어절에 대응되는 형태소 결합 $\overline{m}_i$의 j번째 형태소에 대응되는 품사를 의미한다.

따라서, 어절 단위 품사 태깅의 문제는 "길이가 N인 어절열(문장) $e_{1,N}=e_1$ $e_2$ … $e_N$이 주어졌을 때, 가장 확률이 높은 품사 결합열 $\overline{c}_{1,N}$ $=\overline{c}_1$ $\overline{c}_2$ …$\overline{c}_N$을 구하는 것"으로 [식 11]과 같이 정의될 수 있다.

$$T(e_{1,N}) \overset{\text{def}}{=} \underset{\overline{c}_{1,N}}{argmax}\ P(\overline{c}_{1,N}|e_{1,N}) \qquad [식\ 11]$$

[식 11]은 문장 단위의 통계 정보를 필요로 하는 매개변수 $P(\overline{c}_{1,N}|e_{1,N})$을 가진다. 그러나 문장 단위의 통계 정보를 획득하는 것은 거의 불가능한 일이다.

이제부터 [식 11]에 대한 적절한 변형을 통하여 이를 통계 정보 획득이 가능한 형태로 만들어 보도록 하자. 먼저 조건부 확률의 정의에 의해 [식 11]을 다음과 같이 변환한다.

$$T(e_{1,N}) = \underset{\overline{c}_{1,N}}{argmax} \frac{P(\overline{c}_{1,N}, e_{1,N})}{P(e_{1,N})} \qquad [식\ 12]$$

$$= \underset{\overline{c}_{1,N}}{argmax}\ P(\overline{c}_{1,N}, e_{1,N}) \qquad [식\ 13]$$

[식 12]의 분모 $P(e_{1,N})$은 모든 $\overline{c}_{1,N}$에 대해 상수이므로 [식 13]에서 생략되었다. [식 13]에 Chain Rule을 적용하고 1차 마르코프 가정을 도입하면 [모델 6]이 얻어진다.

$$T(e_{1,N}) \cong \underset{\overline{c}_{1,N}}{argmax} \prod_{i=0}^{N} P(\overline{c}_i|\overline{c}_{i-1})P(e_i|\overline{c}_i)$$

[모델 6]

[모델 6]은 영어에서의 품사 태깅 모델인 [모델 2]와 개념적으로 유사한 모델로서 자율 학습이 가능한 HMM 모델이다. 그림 2는 [모델 6]을 이용하여 한국어 문장의 발생을 어절 단위 HMM으로 모델링한 예이다. 이 모델에서 사용하는 문맥 정보인 $P(\overline{c}_i|\overline{c}_{i-1})$ 과 어휘 발생 정보인 $P(e_i|\overline{c}_i)$는 모두 어절 단위의 통계 정보를 필요로 한다. 그러나 어절의 형태가 매우 다양하게 발생하는 한국어의 특성상, 어절 단위의 통계 정보 획득에는 자료 부족 문제가 심각하게 나타난다. 이러한 문제를 완화시키기 위해 어절 단위 품사 태깅 모델에서는 매우 단순화된 품사 집합을 사용하는 것이 일반적이다. 그러나 품사 집합을 단순화시키더라도 어휘 발생 정보 획득시에 나타나는 자료 부족 문제는 여전히 심각한 문제로 남는다.

이운재[52]는 [모델 6]과 같은 HMM에 기반한 모델을 자율 학습시킨 품사 태깅 시스템

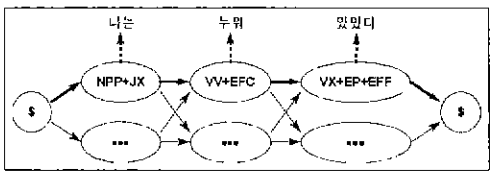



그림 2 한국어 문장의 발생을 어절 단위 HMM으로 모델링한 예

을 개발하였다. 이 시스템에서는 품사 3-gram에 기반한 상태 전이 확률을 사용하였고, 형태소 단위의 어휘 발생 확률을 결합하여 관측 심볼 확률을 계산하였다. 이운재의 시스템에서 실제로 사용된 모델은 [모델 6-1]과 같으며, 17개의 품사 집합을 사용하여 품사 태깅 실험을 수행한 결과 90%의 정확도를 얻었음이 보고되었다.

$$T(e_{1,N}) \cong \underset{\bar{c}_{1,N}}{argmax} \prod_{i=0}^{N} P(\bar{c}_i|\bar{c}_{i-1}, \bar{c}_{i-2}) \sqrt[N_i]{\prod_{j=0}^{N_i} P(m_{ij}|c_{ij})}$$

[모델 6-1]

이상주[48]는 한국어 품사 태깅 과정을 어절 단위의 중의성 최소화 단계와 형태소 단위의 품사 태깅 단계로 분리시켰다. 이 때 중의성 최소화 단계에서 [모델 6]에 기반한 은닉 마르코프 모델을 자율 학습시켰는데, 어절의 발생 형태가 너무 다양하게 나타나기 때문에 관측 심볼 확률이 안정적으로 학습되기 어려운 문제를 극복하기 위해 Kupiec이 사용한 것과 같은 의사 부류를 사용하였다. 실제로 사용된 품사 태깅 모델은 [모델 6-2]이며, 18개의 품사 집합을 사용하여 품사 태깅 실험을 수행한 결과 94%의 정확도를 얻었음이 보고하였다. 여기에서 $Equ_i$는 $i$번째 어절에 대응되는 한국어 의사 부류를 의미한다.

$$T(e_{1,N}) \cong \underset{\bar{c}_{1,N}}{argmax} \prod_{i=0}^{N} P(\bar{c}_i|\bar{c}_{i-1}) P(Equ_i|\bar{c}_i)$$

[모델 6-2]

어절 단위 품사 태깅 모델은 어절 단위로 문맥과 어휘의 발생을 관측하는 방법으로 태깅의 결과 어절 단위의 품사 정보를 얻게 된다. 이 방법은 한국어에서 중의성 해결의 중요한 단서가 되는 어절 단위의 문맥을 관측할 수 있다는 장점이 있다. 하지만 어절의 형태가 매우 다양하게 발생하는 한국어의 특성상, 통계 정보를 획득할 때 자료 부족 문제가 심각하게 나타나게 되고, 따라서 매우 간소화된 품사 집합을 사용할 수 밖에 없다. 그리고 태깅의 결과가 어절 단위로 주어지기 때문에, 형태소 단위의 품사 정보를 얻기 위해서는 태깅된 결과에 대해 다시 추가적인 분석을 수행해야 하는 부담이 따른다.

### 3.5 형태소 단위 한국어 품사 태깅 모델

형태소 단위 품사 태깅의 문제는 "길이가 N인 어절열(문장) $e_{1,N}$이 주어졌을 때, 가장 확률이 높은 형태소열 $m_{1,X}=m_1\ m_2 \ldots m_X$과 품사열 $c_{1,X}=c_1c_2 \ldots c_X$을 구하는 것"으로 [모델 7]과 같이 정의될 수 있다. 여기에서, 형태소열 $m_{1,X}$는 형태소 결합열 $\overline{m}_{1,N}$에 대응되는 것으로 문장 $e_{1,N}$을 구성하는 X개의 형태소로 이루어진 형태소의 열을 의미하며, 품사열 $c_{1,X}$는 품사 결합열 $\bar{c}_{1,N}$에 대응되는 것으로 형태소 열 $m_{1,X}$의 각 형태소에 대응하는 품사의 열을 의미한다. X는 문장 $e_{1,N}$을 구성하는 형태소의 개수를 나타내며, 어절을 구성하는 어절들의 형태소 분석 결과에 따라 다양하게 나타난다.

[모델 7]에서 사용하는 문맥 정보인 $P(c_i|c_{i-1})$과 어휘 발생 정보인 $P(m_i|c_i)$는 모두 형태소 단위의 통계 정보를 필요로 하므로, 통계 정보 획득시의 자료 부족 문제가 상대적으로 덜 심각하게 발생한다.

$$T(e_{1,N}) \cong \underset{m_{1,X},\, c_{1,X}}{argmax} \prod_{k=0}^{X} P(c_k|c_{k-1}) P(m_k|c_k)$$

[모델 7]

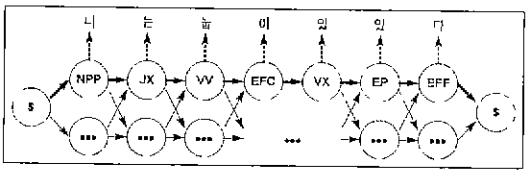



그림 3 한국어 문장의 발생을 형태소 단위 HMM으로 모델링한 예

[모델 7]은 자율 학습이 가능한 HMM모델이다. 그림 3은 [모델 7]을 이용하여 한국어 문장의 발생을 형태소 단위 HMM으로 모델링한 예이다. 그러나, 한국어에서 어절은 서로 다른 개수의 형태소로 분석될 수 있기 때문에 한 문장에 대해 다중 관측열(Multiple Observation)이 발생하게 되므로, 계산 상에 과부하(Overhead)가 걸리게 된다.

임철수[57]는 공유 단어열(Shared Word Sequence)과 가상 단어(Virtual Word) 개념

을 도입하여 다중 관측열에 대한 효율적인 태깅 알고리즘을 제안하였다. 공유 단어열은 문장에 대응되는 형태소 열에서 공통적인 부분 형태소 열을 의미하는 것으로, Viterbi 알고리즘에서 중복된 계산을 줄이기 위해 사용된다. 가상 단어는 Viterbi 알고리즘에서 바로 전 상태의 태그를 결정하기 전에 미리 형태소를 선택하기 위해 형태소 열에 첨가되는 것으로, Viterbi 알고리즘에서 Maximization 연산을 줄이기 위해 사용된다. 임철수의 태깅 시스템은 79개의 품사 집합을 사용하여 실험 코퍼스에 대한 태깅 실험을 수행한 결과 88%의 정확도를 보였다.

$$T(e_{1,N}) \cong \underset{m_{1,N},\ c_{1,N}}{argmax} \prod_{N_i=1}^{N} \prod_{j=1}^{K} P(c_{ij}|c_{i(j-1)})P(m_{ij}|c_{ij})$$

[모델 6-3]

이상호[49]는 Viterbi 알고리즘을 어절 단위로 전개시켜 최적의 형태소 결합열과 품사 결합열을 구하는 [모델 6-3]과 같은 모델을 사용하였다. 그리고, 한국어의 형태적 특성을 고려한 미등록어 처리 기법을 결합시켜서 실험 코퍼스에 대한 태깅 실험을 수행한 결과 93.6%의 정확도를 보고하였다. [모델 6-3]에서 $P(c_{i1}|c_{i0})$는 $P(c_{i1}|c_{(i-1)M-1})$로 정의된다.

[모델 7]과 [모델 6-3]은 형태소 단위의 문맥 정보와 어휘 발생 정보를 효과적으로 모델링하여 자료 부족 문제가 줄어드는 반면, 한국어에서 중요한 어절 단위의 문맥 정보를 고려하기 어렵고 어절 사이의 공백 정보를 효과적으로 이용하지 못하는 단점을 갖고 있다.

김진동[44]은 어절 단위 품사 태깅 시스템과 형태소 단위 품사 태깅 시스템의 단점을 보완하고 장점을 취하기 위하여, [모델 6]을 기본으로 하되 어절 단위의 문맥 정보와 어휘 발생 정보를 모두 형태소 단위 요소들로 모델링한 [모델 6-4](이중 은닉 마르코프 모델; Twoply HMM)과 같은 모델을 사용하였다. 여기에서 $h_i$는 $i$번째 어절의 실질 형태소에 해당하는 품사(Head Category)를, $t_i$는 $i$번째 어절의 형식 형태소에 해당하는 품사(Tail Category)를 의미한다.

$$T(e_{1,N})$$

$$\cong \underset{m_{1,N},\ c_{1,N}}{argmax} \prod_{i=0}^{N} \left[ \sqrt[2]{P(h_i|t_{i-1})P(t_i|h_{i-1})} \prod_{j=1}^{K+1} P(c_{ij}|c_{i(j-1)})P(m_{ij}|c_{ij}) \right]$$

[모델 6-4]

이 모델은 어절 단위의 문맥을 관측하면서도 형태소 단위의 통계 정보만을 필요로 하기 때문에 자료 부족 문제가 그리 심각하게 나타나지 않고 따라서 품사 집합을 축소시킬 필요가 없다. 또한 품사 태깅의 결과가 형태소 단위로 주어진다는 장점을 지닌다. 59개의 품사 집합을 사용하여 실험 코퍼스에 대한 태깅 실험을 수행한 결과 94.3%의 정확도를 얻었다.

## 4. 규칙 기반 품사 태깅 시스템

규칙 기반 품사 태깅 시스템은 태깅하고자 하는 대상 언어에 대한 언어 지식을 결정적인(Deterministic) 규칙 형태로 표현하며, 그 규칙을 품사 태깅에 사용한다. 일반적으로 규칙 기반 접근방법(Rule-Based Approach)은 지식 기반 접근방법(Knowledge-Based Approach) 또는 제약 기반 접근방법(Constraint-Based Approach)라고도 한다.

규칙 기반 품사 태깅 시스템에서 사용하는 규칙을 예를 들어 설명하면, 규칙 '[PREP + TNS]→TNS [VERB+N]'[2]는 PREP나 TNS 품사를 갖는 단어가 VERB나 N의 품사를 갖는 단어 앞에 사용될 경우 TNS 품사로 태깅하라는 것을 의미한다. 규칙 기반 품사 태깅 시스템은 제한된 범위 내에서 비교적 적은 개수의 규칙으로 높은 정확도를 갖는 태깅이 가능하며, 태깅 결과에 대한 설명이 가능하고, 시스템 성능 향상에 드는 노력을 적게 요구한다는 장점이 있다. 그러나 태깅 규칙 구축에 많은 수작업을 필요로하며, 새로운 환경에 대한 적응력이 낮고 견고성이 떨어진다는 단점이 지적되어 왔다. 실제로 전통적인 규칙 기반 품사 태깅 시스템에서 사용하는 규칙은 언어학자나 전문적 언어 지식을 가지고 있는 전문가에 의해서 수동으로 구축되었다. 그러나 최근에는 사용가능해진 대량의 원시 코퍼스 및 태깅된

2) PREP : 전치사, TNS : To 부정사 용법의 To, VERB : 동사, N : 명사

코퍼스의 구축과 컴퓨터 성능 향상에 힘입어 태깅 규칙을 자동으로 학습하고자 하는 연구가 활발히 진행되고 있다.

본 장에서는 먼저 수동으로 구축된 규칙을 사용한 Klein과 Simmons의 시스템[20], Green과 Rubin의 TAGGIT[18], Chanod와 Tapanainen 시스템[10], 그리고 Voutilainen 시스템[39]에 대하여 살펴본다. 그리고 학습을 통하여 자동으로 추출된 규칙을 사용하는 Hindle 시스템[19], 변형 규칙을 사용하는 Brill의 시스템[3-6], 그리고 변형 규칙을 한국어의 특성에 맞게 적용한 시스템[59]에 대하여 살펴보고자 한다.

### 4.1 Klein과 Simmons의 시스템

Klein과 Simmons 시스템은 수동으로 구축된 규칙을 이용한 최초의 규칙 기반 품사 태깅 시스템이다[17]. 품사 태깅 과정은 다음과 같다.

- 400여개의 기능어 사전을 참조하여 기능어의 품사를 할당
- 규칙에 적용되지 않는 예외 단어를 처리, 1500여개의 예외 사전을 사용
- 접미어와 특수 기호 등을 처리
- 태깅 규칙을 적용

이 시스템은 30개의 품사를 사용하며, 백과사전의 표본 텍스트에 대해서 약 90%의 정확도를 보였으며, 규칙 기반 태깅 시스템의 시효라는 점에 의의가 있는 시스템이다.

### 4.2 Green과 Rubin의 시스템(TAGGIT)

TAGGIT[18]은 Brown 코퍼스 태깅을 위하여 만들어진 규칙 기반 태깅 시스템이다. TAGGIT은 단어에 가능한 모든 품사를 할당하는 초기 태깅(Initial Tagging) 단계와 각 단어가 문장에서 사용된 품사를 결정하는 중의성 해결(Disambiguation) 단계로 구분된다.

초기 태깅 단계에서는 먼저 3,000개 정도의 예외 단어를 가지고 있는 예외 사전을 참조한다. 다음 축약형, 특수 기호나 대문자로 시작하는 단어 등 여러 가지 특별한 경우에 대하여 처리한다. 예외 처리와 특별한 경우의 처리를 수행한 후, 약 450개정도 크기의 접미사 사전을 이용하여 단어의 마지막 부분(Ending)에 대한 처리를 수행한다. 앞의 여러 가지 처리 후에도 단어에 대해 품사를 하나도 할당할 수 없을 경우, 그 단어에 대한 품사로 명사(NN), 동사(VB), 형용사(JJ) 등 세 개의 품사를 할당한다.

중의성 해결 단계는 초기 태깅에 의해서 할당받은 품사 중에서 각 단어의 문장 내에서 사용된 품사를 문맥 규칙을 이용하여 결정한다. TAGGIT은 3,300개의 문맥 규칙(Context Frame Rule)을 사용하였고, 문맥 규칙은 특정 조건이 만족될 때 어떤 품사를 제거하거나 선택하는 형태이다. 예를 들면, 규칙 "W X ? Y Z→A"는 현재 단어의 앞에서 두 번째, 첫 번째 단어의 품사가 각각 W, X이고 뒤에서 첫 번째, 두 번째 품사가 각각 Y, Z일 경우 중의성을 갖는 단어, ?의 품사를 A로 결정한다를 의미한다. 또한 규칙 "W X ? Y Z→not A"는 현재 단어의 앞에서 두 번째, 첫 번째 단어의 품사가 W, X이고 뒤에서 첫 번째, 두 번째 품사가 각각 Y, Z일 경우 중의성을 갖는 단어, ?의 품사는 A가 될 수 없음(단어 ?의 품사에서 A를 제거)을 나타낸다. TAGGIT은 문맥 규칙에서 중의성 해결을 위해 사용하는 문맥[3]을 좌우로 최대 2개 단어로 한정하였다. 즉 문맥 규칙의 형태는 'X ? Y → A'이거나 'W X ? Y Z→A'의 형태를 갖는다. 이는 중의성 해결을 위해 대상 어절의 근접 정보(Local Information)만을 사용한다는 것을 의미한다.

TAGGIT에서 사용한 품사는 86개였고, CLAWS[26]나 VOLSUNGA 시스템[16]은 이 품사 집합을 기본으로 각각의 품사 집합을 만들었다. Brown 코퍼스의 약 백만 단어에서 실험한 결과 TAGGIT은 약 77%의 정확도를 보였다. 이는 Klein과 Simmons 시스템에 비하여 비교적 낮은 결과이지만 대량의 실제 코퍼스에 적용한 첫 번째 시도였다는 점에서 TAGGIT은 큰 의의를 남겼다. 또한 실험 결과, 좌우 문맥을 하나만 고려하는 규칙은 전체 규칙의 25%에 지나지 않지만, 이들 규칙을 이용하여 중의성의 약 80%를 해결할 수 있음을

3) 중의성 해결을 위해 참조하는 단어의 수

보였다. 이렇게 근접 정보가 중의성 해결에 큰 도움이 된다는 사실은 연어 확률(Collocational Probability)을 태깅에 사용한 CLAWS 시스템 개발에 큰 동기를 부여하기도 하였다.

### 4.3 Hindle의 시스템

Hindle 시스템[19]은 Fidditch라는 결정적 구문 분석기(Deterministic Parser)에서 사용된 시스템으로 태깅을 위한 언어 지식을 대용량 코퍼스로부터 자동 추출이 가능함을 보였다. Fidditch는 구문 구조의 용례 추출을 위해 개발된 결정적 구문 분석기로 이 시스템은 100,000 단어의 사전, 형태소 분석기, 4,000개 크기의 예외 사전, 300 단어 크기의 복합 단어 사전, 350개의 구문 구조 규칙, 그리고 350개의 어휘적 중의성 해결 규칙을 사용한다.

Hindle 태깅 시스템은 구문 분석기와 연계되어 동작하므로 태깅에 필요한 문법적 정보를 제공받을 수 있다는 잇점이 있다. 따라서 품사 집합도 문법 구조를 이용하여 구분이 가능한 품사를 일반화함으로써 46개의 품사 집합만을 사용하였다. 이는 Brown 코퍼스 태깅을 위해 개발된 87개의 품사 집합을 사용한 TAGGIT에 비해 매우 적은 개수이다. 이와 같이 적은 품사 집합의 사용이 가능한 이유는 Hindle 태깅 시스템이 구문 분석기와 연계되어 사용되기 때문이다. 예를 들면, TAGGIT에서는 I와 me를 주격 대명사, 목적격 대명사로 구분하지만 Hindle 시스템에서는 이를 PRO라는 품사로 일반화하였는데, 이는 구문 구조를 참조하면 PRO가 주격인지 목적격인지의 정보를 알 수 있기 때문이다.

중의성 해결 규칙 학습은 구문 구조 분석을 위한 규칙, 문맥과는 독립적으로 중의성을 해결하는 기본 규칙(Default Rule), 그리고 품사 및 구문 주석이 달린 코퍼스를 이용하여 수행된다. 기본 규칙은 버퍼의 두 번째 또는 세 번째 단어와는 상관없이 중의성을 갖는 현재 단어에 품사를 할당하는 규칙이며 그 예는 아래와 같다.

[ADJ+N+V]=N [*] [*]

위의 기본 규칙은 현재 단어가 형용사(ADJ), 명사(N) 그리고 동사(V)의 품사를 갖는 중의성이 있을 때 이 단어의 품사를 N으로 할당하라는 의미를 나타낸다. 학습 초기에 136개의 기본 규칙을 사용되며 학습을 통하여 세부적인 새로운 규칙들이 생성된다.

중의성 해결 규칙 학습 과정은 다음과 같다. 입력 문장의 구문 구조 분석 시 버퍼의 첫 번째 또는 두 번째에 위치한 단어가 중의성을 갖을 경우, 현재의 규칙들을 이용하여 그 단어의 중의성을 해결하고자 한다. 현재의 문맥에 맞는 규칙을 적용하여 중의성이 해결되면 구분 분석을 계속 진행한다. 만약 실패할 경우, 현재까지의 구문 분석된 정보(버퍼와 스택의 정보)와 현재 사용되고 있는 중의성 해결 규칙을 이용하여 새로운 규칙을 생성한다. 새롭게 생성된 규칙은 현재 사용 중인 규칙보다 더 세부적인 규칙이므로 이전의 규칙보다 우선 순위를 갖게된다. 규칙 학습에 의해서 추출된 규칙의 예는 다음과 같다.

[PREP+TNS]=TNS [N+V]

이 규칙은 PREP나 TNS 품사를 갖는 단어가 N이나 V의 품사를 갖는 단어 앞에 사용될 경우 TNS 품사로 태깅하라는 것을 의미한다.

새롭게 생성된 규칙이나 기본 규칙은 항상 올바르게 적용되는 규칙일 수 없다. 따라서 규칙 학습 과정은 적절하지 못한 규칙을 제거하는 과정을 포함한다. 규칙을 적용했을 때, 실패 비율이 주어진 임계값(Threshold, 10-20%)을 초과할 경우 그 규칙은 제거된다.

Brown 코퍼스의 500개의 표본을 Fidditch의 46개의 품사에 맞게 수정하여 이중 90%를 사용하여 규칙 학습에 사용하였고, 10%는 실험 코퍼스로 사용하였다. 학습 코퍼스를 한 번 구문 분석하는 동안 새로운 규칙이 학습되며, 불필요한 규칙은 제거된다. 이렇게 학습된 규칙을 이용하여 학습 코퍼스를 다시 구문 분석을 수행하면 좀더 정련된 규칙을 추출할 수 있게 된다. 총 5번에 걸친 구문 분석을 통해 35,000개의 규칙이 추출되었다. 추출된 규칙을 학습 코퍼스에 적용한 결과 전체적인 정확도는 98%였고, 중의성 있는 단어에 대한 정확도는

95%였다. 실험 코퍼스에서는 전체적으로 약 97%의 정확도를 보였고, 중의성 있는 단어에 대한 정확도는 약 90% 정도의 정확도를 보였다.

### 4.4 Chanod와 Tapanainen의 시스템

Chanod와 Tapanainen 시스템[9, 10]은 프랑스어 태깅을 위한 규칙 기반 시스템으로 확률 기반 방법과 통합되어 사용되기도 하였다. 통합 시스템에 관한 설명은 5장에서 살펴보기로 하고 여기서는 Chanod와 Tapanainen 시스템을 중심으로 살펴보고자 한다. Chanod와 Tapanainen 시스템의 개발 동기는 프랑스어에 대한 중의성에 대한 조사에서부터 시작된다. 중의성에 대한 조사를 약 100만 어절에 대해서 수행한 결과, 16개의 단어가 갖는 중의성이 전체 중의성의 50%이상을 차지하였고, 전체 중의성의 약 2/3가 97개의 중의적 단어에 의한 것이었다. 따라서 먼저 출현 빈도가 높은 중의적 단어들에 대한 주 규칙(Principle Rule)을 수동으로 작성하였다. 그 다음 이들 규칙에 의해서 해결될 수 없는 단어들을 태깅하기 위하여 휴리스틱 규칙을 작성하였고, 주 규칙과 휴리스틱 규칙에 의해서 해결될 수 없는 중의성에 대해서는 비 문맥 규칙(Non-Contextual Rule)을 만들었다. 규칙 구축을 위해 신문 기사 문장 50개가 사용되었으며, 약 1개월 가량의 기간이 소요되었다.

규칙의 형태는 '패턴-처리 방법' 형태이고, 각 규칙은 어느 경우에 어떤 품사가 부적절한가를 의미하는 제약 조건(Constraint)을 나타낸다. 규칙은 여러 개의 FST(Finite State Tranducer)에 의해서 표현되었고, 각각의 Transducer가 적용될 때 중의성을 갖는 단어의 중의성을 감소 또는 제거한다. 모든 Transducer[4]가 적용된 후에는 각 단어가 하나의 품사만을 갖게함으로써 품사 태깅을 수행한다.

이 시스템은 37개의 품사를 사용하며, 주 규칙 39개, 25개의 휴리스틱 규칙 그리고 비 문맥 규칙 11개, 총 75개의 태깅 규칙을 경제에 관련된 문서에서 추출한 5752개의 단어[5]에 적용한 결과, 98.7%의 정확도를 보였다. 또한 철자 오류와 이미 사전에 존재하는 특정 단어와 같은 형태의 고유 명사를 포함하고 있는 신문에서 추출한 12만 단어에 대해서 실험한 결과 97.5%의 정확도를 보였다.

Chanod와 Tapanainen은 이러한 실험 결과를 통해 품사 태깅 문제에 있어서 통계 기반의 방법이 정확도 측면에서 규칙 기반 방법보다 우수하다는 기존의 주장을 반박하였다. 또한 통계 기반 시스템과 비슷한 정확도를 갖기 위해서는 규칙 기반 시스템에서 사용될 규칙 구축 작업도 예상보다 그리 어렵지 않다는 것을 주장하였다.

### 4.5 Voutilainen의 시스템(ENGCG)

Voutilainen 시스템[39]은 ENGCG(English Constraint Grammar) 중의성 해결 시스템과 Finite-State Intersection Grammar [38]로 알려진 구문 분석 시스템을 이용한 규칙 기반 태깅 시스템이다. 여기서 사용된 구문 분석기는 ENGCG에 의해서 해결되지 못한 중의성을 구문 분석의 부수 효과(Side Effect)에 의해 처리하게 된다. 이 시스템은 다음과 같은 다섯 가지 요소로 구성되어 있다.

- 토큰 분리기
- ENGCG 형태소 분석기
- ENGCG 중의성 해결기
- 구문 태그 할당기
- Finite State 구문 분석기

토큰 분리기는 단어, 문장 부호 등을 구별하는 역할을 하며, 형태소 분석기는 단어에 가능한 품사를 할당하는 작업을 수행한다. 사전에 등록되어 있지 않은 단어를 분석하기 위해서는 휴리스틱 규칙을 사용하며, 규칙 적용에 실패한 단어에 대해서는 명사로 할당한다. 그리고 80,000개의 어휘로 구성된 사전을 사용하며, 139가지의 품사 집합을 사용한다. ENGCG 중의성 해결기는 규칙을 이용하여 각 단어의 중의성을 제거하는 기능을 하며, 구문 태그 할당기는 구문 분석에 사용될 각 단어의 구문 태그를 결정하는 역할을 수행한다. Finite State 구문 분석기는 ENGCG에서 제거되지 못한 중의

4) 75개의 규칙을 11개의 Finite State Tranducer로 나타내었다.

5) 이중 약 54%의 단어가 중의성을 갖는 단어였다.

성을 해결한다.

ENGCG 태거에서의 중의성 해결은 다음과 같은 세 가지에 의해서 이루어 진다.

- ENGCG 중의성 해결기
- 휴리스틱 규칙
- Finite-State Intersection Grammar

ENGCG 중의성 해결기는 '패턴-처리 방법' 형태의 규칙을 사용한다. 각 규칙은 어떤 품사가 어느 문맥에 부적절한 품사인가를 나타낸다. 따라서 이들 규칙은 문맥이 일치하는 단어에서 부적절한 품사를 제거함으로써 중의성을 감소시킨다. 현재 1,185개의 규칙을 사용하며 이중 844개(71%) 규칙이 한 단어 이상의 거리에 위치한 단어를 참조할 수 있는 규칙이다. ENGCG 중의성 해결기는 중의성 해결이 어려운 단어는 처리하지 않고 남겨둠으로써 오류의 가능성을 줄인다. 예를 들면, 1,185개 규칙을 사용하여 중의성을 해결하게 되면 전체 중 약 3-7% 단어가 중의성을 그대로 가지고 있게되며, 한 어절당 평균 분석 개수는 1,04-1.08이었다. 하지만 전체 어절의 99.68% 단어가 정확한 품사를 가지고 있게 된다.[6] 여기에 나머지 중의성을 제거하기 위하여 200개의 휴리스틱 규칙이 적용된다. 휴리스틱 규칙의 적용은 전체 정확도를 99.41%로 감소시키지만, 나머지 중의성의 약 50%가량을 해결한다. ENGCG 중의성 해결기와 휴리스틱 규칙을 통과한 후에도 남아있는 중의성은 Finite-State Intersection Grammar를 사용하는 구문 분석 과정을 통해서 제거된다.

ENGCG 태거는 38,202 단어 크기의 실험 코퍼스를 이용하여 실험하였고, 실험 결과는 그림 4와 같다. 그림 4의 두 번째 열은 실험 단어 중 중의성을 갖는 단어의 비율을 나타낸 것이고, 세 번째 열은 실험 단어에 할당된 전체 품사 개수를 의미한다.

최근 컴퓨터 성능 향상과 사용 가능한 대량의 코퍼스 구축에 힘입어 통계 기반 접근방법이 품사 태깅을 위해 많이 사용되며, 좋은 결과를 보여 왔다[14]. 그러나, 그림 4의 실험 결과를 보면, ENGCG 태거의 정확도가 통계 기반 접근방법을 사용하는 어떤 시스템보다도 좋은 성능을 나타내고 있다는 것을 알 수 있다. 이와 같은 실험 결과를 바탕으로 Voutilainen은 품사 태깅 문제도 자연어 처리의 다른 문제와 같이 규칙 기반 접근방법으로 충분히 해결할 수 있다는 것을 주장하였다.

### 4.6 Brill의 시스템

Brill은 변형에 기반한 오류에 의한 학습(Transformation-Based Error-Driven Learning)을 제안하였고, 이를 이용한 품사 태깅 시스템을 개발하였다[3-6]. Brill 시스템의 태깅 과정을 간단히 설명하면 먼저, 초기 태거를 이용하여 각 단어의 품사를 태깅하고, 다음에 변형 규칙을 이용하여 초기 태거에 의한 오류를 수정하여 전체적인 정확도를 증가시키는 것이다. Brill 시스템을 통계 기반의 방법과 비교하여 설명하면, 초기 태거가 어절 발생 확률 정보로 태깅을 수행하고, 변형 규칙이 품사 간의 문맥 전이 확률을 고려하여 초기 태거의 오류를 수정하는 것이라 할 수 있다. Brill 시스템은 기본적으로 다음과 같은 요소로 구성된다.

- 초기 태거
- Scoring 함수
- 규칙 틀
- 변형 규칙 학습기
- 변형 규칙을 이용한 품사 태깅

초기 태거는 각 단어가 가질 수 있는 품사 중 가장 확률이 높은 또는 가장 자주 사용되는 품사로 그 단어를 태깅하는 작업을 수행한다.[7] 규칙 틀과 Scoring 함수, 그리고 변형 규칙 학습기는 태깅된 학습 코퍼스로부터 초기 태거의 태깅 오류를 수정할 수 있는 규칙 추출을 위해 사용된다. 변형 규칙 학습 과정은 아래와 같다.

1) 초기 태거를 이용하여 학습 코퍼스를 태깅한다.
2) 학습 코퍼스의 태깅 결과와 태깅된 코퍼스의 분석 결과를 비교하여 혼동 행렬[8]

6) 한 단어가 여러 가지 품사를 가질 수 있으므로 해당 단어에 적합한 품사를 가지고 있는 경우에 정확한 것으로 계산한 수치

7) Brill은 단어의 여러 가지 품사 중 가장 자주 사용되는 (확률이 높은) 품사에 대한 정보를 가지고 있는 사전을 사용하였음

8) 초기 태거에 의한 오류 유형을 빈도수에 따라 내림차순으로 정렬한 결과

| | 중의성 단어 | 전체 품사 수 | 어절당 품사 수 | 오류 수 | 오류율 |
|---|---|---|---|---|---|
| D0(형태소 분석기) | 39.0% | 67,737 | 1.77 | 31 | 0.08% |
| D1(D0+ENGCG) | 6.2% | 40,450 | 1.06 | 124 | 0.32% |
| D2(D1+휴리스틱 규칙) | 3.2% | 38,946 | 1.02 | 226 | 0.59% |
| D3(D2+구문 분석기) | 0.6% | 38,342 | 1.00 | 281 | 0.74% |

그림 4 ENGCG 태거의 실험 결과

작성

3) Scoring 함수와 규칙 틀을 이용하여 초기 태거에 의한 오류를 가장 많이 수정할 수 있는 규칙 추출
4) 추출된 태깅 규칙을 학습 코퍼스에 적용
5) 추출된 규칙 저장
6) 찾아진 규칙의 오류 수정 빈도가 임계값(Threshold)보다 작을 때까지 단계 2-5를 반복

Scoring 함수는 후보 규칙을 학습 코퍼스에 적용하였을 때의 오류 감소 개수와 새롭게 발생한 오류 개수와의 차이를 계산하는 함수이다. 학습 과정의 2-3 단계에서 모든 규칙 틀에 의해서 생성된 후보 규칙들 중 가장 Score값이 높은 후보 규칙 하나만이 변형 규칙으로 추출된다. 규칙 틀은 변형 규칙 생성을 위한 탐색 공간으로 사용되며, 올바른 변형 규칙 추출을 위하여 규칙 틀의 형태는 매우 중요하다. Brill은 현재 어절에서 최대 3어절 거리의 품사를 고려할 수 있는 규칙틀을 사용하였다. 또한 어휘 간의 관계를 고려할 수 있는 규칙 틀을 사용함으로써 태깅 시 어휘 간의 유용한 정보를 사용할 수 있도록 하였다.[9]

학습에 의해서 추출된 규칙의 예는 다음과 같다.

Change the tag

From **preposition** to **adverb** if the two positions to the right is **as**

이 규칙은 현재 어절 오른쪽 2번째 어절의 품사가 as일 때, preposition으로 태깅되어 있는 현재 어절의 품사를 adverb로 수정하라는 것을 의미하는 것이다.

60만 어절 크기의 코퍼스를 이용한 학습 결과, 품사 문맥을 고려할 수 있는 규칙 틀만을 사용한 경우, 378개의 규칙이 추출되었고, 어휘 문맥까지 고려할 수 있는 규칙 틀을 사용하였을 때 447개의 규칙이 추출되었다. 어절의 품사 문맥만을 고려한 규칙 378개를 15만 어절 크기의 실험 코퍼스에 적용한 결과, 97.0%의 정확도를 보였다. 같은 실험 코퍼스에 품사 및 어휘 문맥을 고려할 수 있는 규칙 447개를 적용하였을 때, 97.2%의 높은 정확도를 보였다.[10]

Brill은 변형 규칙을 이용하여 K-best 태깅 방법도 제안하였다[4]. K-best 태깅은 변형 규칙을 적용할 때, 변형 규칙의 올바른 품사와 현재 품사를 모두 그 어절의 품사로 간주하는 것이다. 즉, 'change tag x to y if context is matched' 형태의 변형 규칙을 'add tag x if context is matched'와 같이 바꾸어 적용하는 것이다.

최근에 Brill은 태깅된 코퍼스가 아닌 원시 코퍼스로부터 변형 규칙을 자동 학습할 수 있는 방법을 제안하였고[5], Ramshaw는 변형 규칙 학습 속도를 현저하게 개선한 새로운 학습 알고리즘을 제안하였다[34].

### 4.7 변형 규칙 기반 한국어 품사 태깅 시스템

한국어의 한 어절은 보통 하나 이상의 형태소로 구성되며, 한 어절내의 태깅 오류가 하나 이상의 형태소 품사 태깅 오류에 의한 것들이 종종 발생한다. 따라서 '문맥 $\alpha$에서 품사 x를 품사 y로 변형한다'와 같이 영어에 적용된 규칙 형태를 한국어에 그대로 사용할 수 없다.

9) 마르코프 모델에 기반한 대부분 태깅 시스템은 어휘 간의 관계를 모델링하는 것이 매우 힘들

10) 이 결과는 미등록어가 존재하지 않는다는 closed vocabulary 가정에 의해 수행된 것임.

또한 변형 규칙을 이용한 한국어 품사 태깅 시스템은 첨가어적인 한국어 특성을 고려할 수 있는 규칙 틀을 사용하여야 한다.

임희석[59]은 첨가어적인 한국어 특성을 고려할 수 있도록 변형 규칙의 형태를 수정하고, 한국어에 적합한 변형 규칙 추출을 위한 규칙 틀을 사용한 시스템을 개발하였다. 태깅을 위한 규칙으로는 '문맥 $\alpha$에서 어절 태그[11] x를 어절 태그 y로 변형한다'와 같은 어절 변형 규칙 형태를 사용하였다. 이러한 변형 규칙 형태는 어절을 구성하는 형태소 품사 열인 어절 태그를 수정함으로써 여러 가지 형태소 품사를 수정할 수 있다. 예를 들면, '내가 먹은 감'에서 '먹은'이 '명사(먹)+조사(은)'로 잘못태깅되었을 때, "문맥 $\alpha$에서 어절 태그, '명사+조사'를 어절 태그, '동사+어미'로 변형한다"와 같은 변형 규칙을 사용하면, '동사(먹)+어미(은)'으로 오류를 수정할 수 있는 것이다.

규칙 틀은 크게 어절 태그 문맥, 형태소 품사 문맥, 그리고 어휘 문맥을 고려할 수 있는 형태로 나뉘어 진다. 형태소 품사 문맥은 이웃 어절의 실질 형태소나 형식 형태소가 특정 어절에 미치는 문법적 영향을 의미하며, 어절 태그 문맥은 주위 어절의 어절 태그가 현재어절에 미치는 문법적 영향을 고려하기 위한 것이다. 세 가지 규칙 틀을 이용한 학습 결과 추출된 변형 규칙의 예는 그림 5와 같다.[12]

57개의 품사 집합을 사용하여 10만 어절 크기의 학습 코퍼스를 이용하여 학습한 결과, 101개의 규칙이 추출되었다. 101개의 변형 규칙 중 53개(52%)가 형태소 태그 문맥을 고려한 규칙틀에 의해서 만들어 진 것이었으며, 30개(18%)는 어절 태그 문맥을 이용한 규칙 틀에 의해서 추출된 것이었다. 어휘 문맥을 고려한 규칙 틀로부터는 18개(18%)의 규칙이 추출되었다. 101개의 변형 규칙을 약 2만 어절 크기의 실험 코퍼스에 적용한 결과 약 94.8%의 정확도를 보였다.

## 5. 통합 접근 품사 태깅 시스템

11) 어절태그란 어절을 구성하는 형태소의 품사열을 의미한다.

12) 품사 집합은 [59] 논문을 참조

통합 접근 방법은 통계 기반 접근방법과 규칙 기반 접근방법의 장단점을 상호 보완하여 좀더 견고하고, 정확성 높은 태깅 시스템을 개발하기 위한 목적으로 시도되고 있는 방법이다. 통합 접근방법도 통합에 사용되는 방법에 따라 다양하게 분류될 수 있다[23, 38, 46]. 이 장에서는 Tapanainen과 Voutilainen 시스템[38] 그리고 TAKTAG 시스템[23, 46]의 통합 방법 및 실험 결과를 중심으로 설명하고자 한다.

### 5.1 Tapanainen과 Voutilainen의 시스템

이 시스템은 규칙 기반 태깅 시스템인 ENGCG[39]과 마르코프 모델을 이용한 통계 기반 시스템인 Xerox 태거(XT)[15]를 통합한 시스템이다[38]. 일반적으로 규칙 기반 태깅 시스템은 모든 중의성을 해결할 수 있는 것은 아니지만, 규칙이 사용되는 경우에는 매우 높은 정확도로 중의성을 해결한다. 반면에 통계 기반 태깅 시스템은 모든 중의성을 해결할 수는 있지만, 그 정확도가 규칙 기반 태깅 시스템과 비교하여 낮다.

이 시스템의 통합 방법은 ENGCG 태거와 XT가 독립적으로 태깅을 수행하고, 그 결과를 비교한다. 만약 태깅 결과가 서로 다를 경우에는 규칙 기반 태거인 ENGCG의 결과를 선호하도록 하고, ENGCG가 처리하지 못한 단어에 대한 품사는 XT에 의한 결과를 따르도록 하는 것이다. 만약 ENGCG가 2개 이상의 품사로 단어를 태깅한 경우에는 XT의 결과와 비교하여 하나의 품사를 선택(Unambiguious Mode)하거나 2개 이상의 품사를 선택(Carefule Mode)한다. Tapanainen과 Voutilainen 시스템의 품사 태깅 과정을 예를 들어 설명하면, 그림 6과 같다.

ENGCG는 A-F까지 6개의 단어를 입력으로 받아, D는 처리하지 못하고, B는 3개의 품사를, 그리고 나머지 단어들은 하나의 품사만을 할당하였다. ENGCG에 의해서 태깅되지 못한 단어, B, D에 대해서는 XT의 결과를 이용한다. XT는 ENGCG와는 달리 Brown 코퍼스 품사 집합을 사용한다. 따라서 XT의 결과를 바로 사용할 수는 없으며, XT의 결과를

| 번 호 | 변 경 전 | 변 경 후 | 조 건(문맥) |
|---|---|---|---|
| 1 | VV+EFD | VX+EFD | [-1 Tail] EFC |
| 2 | VV+EFD | VX+EFD | [-1 어절 태그] NNCV+XSV+EFC |
| 3 | NNB+JCV+SS. | NNB+JCP+EFF+SS. | [-1 Tail] JCO |
| 4 | NNCG+JN | NNCG+JCA | [1 어휘] 같은 |

그림 5 한국어 품사 태깅을 위한 변형 규칙의 예

| 입력 문장 : | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| ENGCG의 태깅 결과 : | a | b1, b2, b3 | c | 처리 못함 | e | f |
| XT의 태깅 결과 : | k | j | l | m | n | o |
| XT의 태깅 결과를 ENGCG 품사로 사상시킨 결과 : | a | b1, b2 | c, x | d | e | f |
| 통합결과(방식1) : | a | b1, b2 | c | d | e | f |
| 통합결과(방식2) : | a | b1 | c | d | e | f |

그림 6 Tapanainen과 Voutilainen 시스템의 태깅 과정

| | 중의성 비율 | 단어당 품사수 | 정확도(%) |
|---|---|---|---|
| D1(ENGCG) | 6.4% | 1.08 | 99.65% |
| D2(D1+휴리스틱 규칙) | 3.6% | 1.04 | 99.37% |
| D3(D2+XT+방식1) | 2.2% | 1.02 | 99.18% |
| D4(D2+XT+방식2) | 0.0% | 1.00 | 98.54% |

그림 7 Tapanainen과 Voutilainen 시스템의 평가

ENGCG 품사 집합에 맞도록 사상시켜야 한다. 그 결과가 그림 6에서 'XT의 결과를 ENGCG 품사로 사상시킨 결과'에 나타나 있다. '통합 결과'는 두 시스템의 태깅 결과를 규칙 기반 시스템인 ENGCG의 결과를 선호하여 통합한 것이다. 통합 결과에서 방식 1(Careful Mode)은 한 단어에 2개 이상의 품사를 허용하도록 한 것이고, 방식 2(Unambiguious Mode)는 한 단어에 하나의 품사만을 허용하여 통합한 결과이다.

Tapanainen과 Voutilainen 시스템은 26,711단어로 구성된 실험 코퍼스에서 평가되었으며, 그 결과는 그림 7과 같다.

### 5.2. TAKTAG 시스템

TAKTAG(Two phase learning Architecture for Korean part-of-speech TAGger) 시스템[23, 46]은 자율 학습에 의한 HMM에 기반한 품사 태깅 시스템의 오류를 수정하기 위하여 변형 규칙을 적용한 시스템이다. 이 시스템은 19개의 품사 집합을 사용하며, HMM을 학습하기 위하여 약 5만개의 형태소를 사용하였다. 약 1만개의 형태소에 대하여 실험한 결과, HMM의 정확도는 약 78.4%였다.

변형 규칙은 그림 8과 같은 형태소 변형 규칙을 사용하였다.

그림 9는 학습 결과 추출된 형태소 변형 규칙의 예이다. 그림 9의 규칙은 접속조사(jJ)로 태깅되어 있는 형태소 '와'를 다음 어절의 첫 번째 형태소가 '같이'일 때, jC 품사로 변형한다는 의미이다.

```
[현재 형태소][현재 형태소 태그];[문맥]→[변경될 형태소][변경될 형태소 태그]
[문맥]={[규칙 틀][해당 형태소, 해당 형태소 태그]}+
```

그림 8 형태소 변형 규칙

```
[와][jJ];[N1FMO] : [같이]→[와] : [jC]
```

그림 9 형태소 변형 규칙의 예

100여 개의 규칙 틀과 1만여 개의 형태소를 이용한 학습 결과, 450여개의 변형 규칙이 추출되었다. 변형 규칙을 HMM의 태깅된 결과에 적용했을 때, 약 13% 정도의 오류를 수정할 수 있었고, TAKTAG 시스템의 전체적인 정확도는 약 91.5%였다.

## 6. 품사 태깅 시스템의 평가 기준

본 장에서는 품사 태깅 시스템의 객관적인 평가에 사용될 수 있는 기준과 그 기준들에 따른 평가 방법을 설명하고자 한다.

품사 집합의 크기는 태깅하고자 하는 언어의 중의성 정도에 많은 영향을 미치므로, 태깅 시스템 평가 시 그 시스템이 몇 가지의 품사 집합을 사용하는가를 참조하여야 한다. 많은 개수의 품사 집합을 사용하는 시스템은 대상 언어에 발생하는 중의성 정도가 매우 높으므로 품사 태깅이 매우 어려우며 정확도가 낮을 수 밖에 없다. 반면, 적은 개수의 품사 집합을 사용하는 시스템의 경우, 대상 언어에 발생하는 중의성이 낮으므로 품사 태깅이 쉬우며, 정확도가 높게 나타난다. 따라서 품사 태깅 시스템을 평가할 때, 정확도만을 참조하여서는 안되며 평가할 시스템의 품사 집합 크기도 참조하여야 한다.

품사 태깅 시스템의 정확도를 계산하기 위해서 여러 가지 방법이 사용될 수 있다. 먼저 정확도 계산을 위해 사용되는 단위에 따라 형태소 단위, 어절 단위, 문장 단위의 계산 방법이 있다. 형태소 단위의 계산 방법은 전체 형태소 중에 태깅 시스템이 얼마나 많은 형태소를 올바르게 태깅하였는가를 계산하는 것이다. 어절 단위의 계산 방법은 태깅의 결과, 어절내의 모든 형태소가 올바르게 태깅되었을 경우에만 정확한 태깅이 이루어 졌다고 계산하는 방법이다. 따라서 이 방법은 일반적으로 형태소 단위의 평가 방법보다 낮은 수치의 정확도를 보이며, 한국어와 같이 첨가어적인 언어의 품사 태깅 시스템에서 자주 사용된다. 문장 단위의 계산 방법은 문장내의 모든 어절이 올바르게 태깅되었을 경우에만 정확한 태깅으로 계산하는 방법이다. 이 방법은 형태소 단위, 어절 단위의 정확도에 비해서 매우 낮은 수치의 정확도를 보인다. 예를 들어 10개의 어절로 이루어진 10개의 문장이 있다고 가정하자. 이 때 5개의 문장에서 각각 한 어절이 잘못 태깅되었다면, 문장 단위의 정확도 계산 방법으로는 50%(5/10*100)의 정확도를 보이지만, 어절 단위 계산 방법으로는 95%(95/100*100)의 정확도를 보인다. 정확도 계산을 위해 사용되는 범위에 따라 중의성있는 형태소, 어절에 대해서만 계산하는 방식과 입력으로 사용된 모든 형태소, 어절을 계산에 포함하는 방법이 있다. 일반적으로 두 가지 방법 중 어느 방법이 더 우수하다고 할 수 없으나, 첫 번째 경우의 계산 방법은 두 번째 계산 방법보다 낮은 수치를 나타낸다.

일반적으로 품사 태깅 시스템은 입력으로 미등록어가 사용될 수 있다는 가정(Open Vocabulary Assumption) 또는 미등록어가 사용되지 않는다는 가정(Closed Vocabulary Assumption)을 사용한다. 미등록어 처리는 품사 태깅 시스템이 해결해야 할 문제로서 현재까지 만족스러운 해결 방법이 없는 실정이다. 따라서, 미등록어를 허용하는 시스템은 미등록어를 허용하지 않는 시스템보다 비교적 낮은 정확도를 보이게 된다. 하지만, 미등록어를 고려하는 시스템은 그 자체로 매우 의미있는 것이므로 정확도는 높지만 미등록어를 고려하지 않는 시스템보다 뒤떨어진다고 평가되어서는 안된다.

학습 방법이 품사 태깅 시스템의 또 다른 평가 기준으로 사용될 수 있다. 일반적으로 지도 학습으로 개발된 품사 태깅 시스템이 높은 정확도를 보이지만, 지도 학습을 위해 태깅된 학습 코퍼스를 구하거나 작성해야하는 어려움이 따르게 된다. 따라서 정확도는 비교적 낮지만, 자율 학습을 이용하는 품사 태깅 시스템이 더 우수하다고 평가 될 수도 있다.

실험 코퍼스의 양과 질도 품사 태깅 시스템

의 평가에 사용되는 매우 중요한 요소이다. 실험 코퍼스의 크기가 클수록 태깅의 정확도가 낮게 나타날 수 있지만 태깅 시스템의 견고성을 고려한 평가가 이루어 질 수 있다.

## 7. 결 론

품사 태깅은 문장에 사용된 각 단어에 알맞는 품사를 할당하는 작업으로 구문 분석기, 기계 번역 시스템, 정보 검색 시스템 등 자연어 처리 시스템에서 매우 중요한 역할을 한다. 따라서 품사 태깅 시스템은 사용될 응용 분야에 맞도록 구축되는 것이 바람직하다. 또한 품사 태깅 시스템의 구축 시에는 품사 집합의 크기, 미등록어 처리 문제, 학습 방법, 그리고 자료 부족 문제를 어떻게 다룰 것인가를 신중히 결정하여야 한다.

본 논문은 자연어 처리를 위한 품사 태깅 시스템을 통계 기반 접근방법을 사용하는 시스템, 규칙 기반 접근방법을 사용하는 시스템, 그리고 통합 접근방법을 중심으로 살펴보았다. 통계 기반 접근방법을 사용하는 품사 태깅 시스템은 견고하며, 태깅을 위한 정보를 자동으로 추출할 수 있다는 장점을 갖는다. 하지만, 태깅에 유용하게 사용될 수 있는 어휘간의 관계를 고려하기가 매우 힘들며, 태깅 결과를 인간이 이해, 분석하기가 어렵다는 단점이 있다. 규칙 기반 접근방법을 이용한 품사 태깅 시스템은 일반적으로 태깅에 사용될 규칙 추출에 많은 노력을 필요로 하고, 견고하지 못하다는 단점을 가지고 있다. 하지만, 규칙이 적용될 수 있는 현상에 대해서는 높은 신뢰도로 문제를 해결할 수 있다는 장점이 있다. 통합 접근방법은, 통계 기반 접근방법과 규칙 기반 접근방법의 장점을 취하고 단점을 보완하기 위한 것으로, 규칙이 적용될 수 있는 현상들은 높은 신뢰도로 해결하고 그렇지 않은 현상들은 통계 기반 접근방법으로 해결한다. 따라서, 높은 신뢰도를 보이면서도 견고한 장점이 있다.

현재 품사 태깅의 정확도 면에서 영어의 경우 97%-99%의 높은 정확도를, 한국어의 경우 약 94%-95%의 비교적 높은 정확도를 보이고 있지만, 태깅된 코퍼스를 구축하는 데에 태깅 시스템을 사용하기에는 현실적으로 한계가 있다. 예를 들어, 99%의 정확도를 가진 태깅 시스템을 사용한다고 하더라도, 100만 어절 코퍼스에 대해서 1만 어절이 오류이며, 이 1만 어절을 후처리하고자 해도 무엇이 오류인지 모르기 때문에 100만 어절을 모두 확인해야 하는 부담이 따르게 된다. 따라서, 태깅 시스템을 실용화하기 위해서는 태깅 오류에 대한 효율적인 후처리(Postprocessing) 방법에 대한 연구가 수행되어야 할 것이다.

품사 태깅 시스템의 궁극적인 목표는 견고하고, 이식성이 뛰어나며 정확한 품사 태깅 시스템을 구축하는 것이다. 이와 같은 목표를 달성하기 위해서는 미등록어 처리 문제, 자료 부족 문제 등 해결해야 할 많은 문제점을 안고 있으며, 이에 대한 향후 연구가 계속되어야 할 것이다. 최근에는 통계 기반 접근방법과 규칙 기반 접근방법을 통합하여 각 접근방법이 가지고 있는 한계를 극복하고자 하는 연구가 활발히 진행되고 있다. 또한 응용 분야에 적합한 태깅 시스템의 구축으로 미등록어 문제나 자료 부족 현상 등 대용량 코퍼스에서 발생하는 여러 문제점을 해결하고자 하는 연구도 진행되고 있다.

## 참고문헌

[1] Andrew David Beale, "Lexicon and Grammar in Probabilistic Tagging of Written English," *Proc. of the 26th Annual Meeting of the ACL*, pp.211-216, 1988

[2] Julian Benello, Andrew W. Mackie, James Anderson, "Syntatic Category Disambiguation with Neural Networks," *Computer Speech and Language*, Vol.3, pp. 203-217, 1989.P

[3] Eric Brill, "A simple Rule-Based Part-of-Speech Tagger," *Proc. of the third Conf. on Applied NLP*, Trento, Italy, pp.153-155, 1992.

[4] Eric Brill, "Some Advances in Transformation-Based Part-of-Speech Tagging," *Proc. of the twelvth National Conference on*

*Artificial Intelligence(AAAI)*, Seatlle, Wa, pp.722-727, 1994.

[5] Eric Brill, "Unsupervised Learning of Disambiguation Rules for Part-of-Speech Tagging," *Proc. of the 3rd Workshop on Very Large Corpora*, pp. 1-13, 1995.

[6] Eric Brill, "Transformation-Based Error-Driven Learning and Natural Language Processing : A Case Study in Part-of-Speech Tagging," *Computational Linguistics*, Vol.21, No.4, pp.543-564, 1995.

[7] Christoper S. Bulter, *Computers and Written Texts*, (Ed.), Blackwell : Oxford UK. & Cambridge USA(Pub.), pp.115-140, 1992.

[8] Chao-Huang Chang and Cheng-Der Chen, "HMM-Based Part-of-Speech Tagging for Chinese Corpora," *Proc. of the First Workshop on Very Large Corpora*, June, pp.40-47, 1993.

[9] Jean-Pierre Chanod, Pasi Tapanainen, "Statistical and Constraint-Based Taggers for French," *Technical report(MLTT-016)*, Rank Xerox research centre, Grenoble, 1994.

[10] Jean-Pierre Chanod, Pasi Tapanainen, "Tagging French - Comparing a Statistical and a Constraint-Based Method," *Proc. of the 7th conference of the European chapter of the ACL*, Dublin, pp.149-156, 1995.

[11] Eugene Charniak, Curtis Hendrickson, Neil Jacobson, Mike Perkowitz, "Equations for Part-of-Speech Tagging," *Proc. of the 11th National Conference on Artificial Intelligence(AAAI)*, pp.784-789, 1993.

[12] Eugene Charniak, *Statistical Language Learning*, (Ed.), The MIT Press : Cambridge Massachusetts & London England, 1993.

[13] Kenneth W. Church, "Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," *Proc. of the 2nd Conference on Applied NLP*, pp.136-143, 1988.

[14] Kenneth W. Church, Robert L. Mercer, "Introduction to the Special Issue on Computational Linguistics Using Large Corpora." *Computational Linguistics*, Vol.19, No. 1, pp. 1-24, 1993.

[15] Doug Cutting, Julian Kupiec, Jan Pedersen, Penelope Sibun, "A Practical Part-of-Speech Tagger," *Proc. of 3rd Conf. on Applied NLP*, pp.133-140, 1992.

[16] Steven J. DeRose, "Grammatical Category Disambiguation by Statistical Optimization," *Computational Linguistics*, Vol.14, No.1, pp.31-39, 1988.

[17] Kjell Elenius, "Comparing a Connectionist and a Rule Based Model for Assigning Parts-of Speech," *Proc. of Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP-90)*, pp.597-600, 1990.

[18] Roger Garside, Geoffrey Leech, Geoffrey Sampson (Ed.), *The Computational Analysis of English : a Corpus-Based Approach*, Longman Grousp UK Limited, 1987.

[19] Donald Hindle, "Acquiring Disambiguation Rules from Text," *Proc. of 27th Annual Meeting of the ACL*, pp.118-125, 1989.

[20] Sheldon Klein, Robert F. Simmons, "A Computational Approach to Grammatical Coding of English Words," *Journal of the ACM*, Vol.10, NO.3, pp.334-347, 1963.

[21] Julian Kupiec, "Augmenting a Hidden Markov Model for Phrase-Dependent Word Tagging," *Proc. of Darpa Speech and Natural Language Workshop*, pp.92-98, 1989.

[22] Julian Kupiec, "Robust Part-of-Speech Tagging Using a Hidden Markov Model," *Computer Speech and Language*, pp.225-242, 1992.

[23] Geunbae Lee, Jong-Hyeok Lee, Sanghyun Shin, "TAKTAG : Two-phase Learning Method for Hybrid Statistical/Rule-Based Part-of-speech Disambiguation," *Proc. of the 1995 Int. Conf. on Computer Processing of Oriental Languages(ICCPOL)*, pp.158-

163, 1995.

[24] Carl G. de Marcke, "Parsing The LOB Corpus," *Proc. of the 28th Annual Meeting of the ACL*, pp.243-251, 1990.

[25] Mitch Marcus, "Statistical Natural Language Processing : Current Trends and Future Directions," *Proc. of ATR Int. Workshop on Speech Translation*, 1993.

[26] Ian Marshall, "Choice of Grammatical Word-Class without Global Syntactic Analysis : Tagging Words in the LOB Corpus," *Computers in Humanities*, Vol.17, pp. 139-150, 1983.

[27] Bernard Merialdo, "Tagging Text with a Probabilistic Model," *Proc. of ICASSP*, pp. 809-812, 1991.

[28] Bernard Merialdo, "Tagging English Text with a Probabilistic Model," *Computational Linguistics*, Vol.20, No.2, pp.155-171, 1994.

[29] Marie Meteer, Richard Schwartz, Ralph Weiscedel, "Studies in Part of Speech Labelling," *Proc. of the DARPA Speech and Natural Language Workshop*, pp.331-336, 1991

[30] Robert Milne, *Lexical Ambiguity Resolution : Perspectives from Psycholinguistics*, Neuropsychology, and Artificial Intelligence, Steven L. Small, Garrison W. Cottrell, Michael K. Tanenhaus(Ed.), Morgan Kaufmann : San Mateo, Califonia (Pub.), pp.45-71, 1988.

[31] Mami Nakamura, Katsuteru Maruyama, Takeshi Kawabata, Kiyohiro Shikano, "Neural Network Approach of Word Category Prediction for English Texts," *Proc. of Int. Conf. on Computational Linguistics*, pp.213-218, 1990.

[32] L. R. Rabiner, B. H. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP MAGAZINE*, Jan., pp.4-16, 1986.

[33] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. of the IEEE*, Vol. 77, No.2, pp.257-286, 1989.

[34] Lance A. Ramshaw, Mitchell P. Marcus, "Exploring the Statistical Derivation of Transformational Rule Sequences for Part-of-speech Tagging," *The Balancing Act : Proc. of the ACL Balancing Act Workshop on Combining Symbolic and Statistical Approaches to Language*. 1994.

[35] Hinrich Sch tze, "Part-of-Speech Induction from Scratch," *Proc. of the 31st Annual Meeting of the ACL*, pp.251-258, 1993.

[36] Hinrich Sch tze, Yoram Singer, "Part-of-Speech Tagging Using A Variable Memory Markov Model," *Proc. of the 32nd Annual Meeting of the ACL*, pp.181-187, 1994.

[37] Hinrich Sch tze, "Distributional part-of-speech tagging," *Proc. of the 7th Conf. of the European Chapter of the ACL*, Dublin, pp.141-148, 1995.

[38] Pasi Tapanainen, Atro Voutilainen, "Tagging accurately - Don't guess if you know," *Proc. of the 7th conference of the European chapter of the Association for Computational Linguistics*, pp.149-156, 1994.

[39] Atro Voutilainen, "A Syntax-Based Part-of-Speech Analyser," *Proc. of the 7th Conf. of the European Chapter of the ACL*, pp.157-164, Dublin, 1995.

[40] Ralph Weischedel, Marie Meteer, Richard Schwartz, Lance Ramshaw, Jeff Palmucci, "Coping with Ambiguity and Unknown Words through Pababilistic Models," *Computational Linguistics*, Vol.19 No.2. pp.359-382, 1993.

[41] 김재훈, 김진형, 서정연, "단어 품사 예측을 위한 신경회로망에서의 불균등 학습," *춘계 인공지능연구회 학술발표 논문집*, pp.14-20, 1993.

[42] 김재훈, 서정연, "품사 태깅 : 검토 및 다루기 어려운 문제들", *한국과학기술원 전산학과 기술 보고서*, 1995.

[43] 김재훈, 조정미, 김창현, 서정연, 김길창, "퍼지망을 이용한 한국어 품사 태깅," *제 5 회 한글 및 한국어 정보처리 학국어 정보처*

*리 학술대회 발표 논문집*, pp.539-603, 1993.

[44] 김진동, 임희석, 임해창, "어절 단위의 문맥을 고려한 형태소 단위의 한국어 품사 태깅 모델," *인지과학회 춘계학술발표 논문집*, pp. 97-106, 1996.

[45] 박혜준, 윤준태, 송만석, "말뭉치 품사꼬리달기 시스템 구현," *정보과학회 봄 학술발표논문집*, pp.829-832, 1994.

[46] 신상현, 이근배, 이종혁, "TAKTAG : 통계와 규칙에 기반한 2단계 학습을 통한 품사 중의성 해결," *제 7회 한글 및 한국어 정보처리 학술대회 발표 논문집*, pp.169-174, 1995.

[47] 신상현, 이근배, 홍남희, 이종혁, "확률과 규칙을 사용한 품사 태깅." *제 6회 한글 및 한국어 정보 처리 학술대회 발표 논문집*, pp. 318-321, 1994.

[48] 이상주, 임희석, 임해창, "은닉 마르코프 모델을 이용한 두단계 한국어 품사 태깅," *제 6회 한글 및 한국어 정보처리 학술대회 발표 논문집*, pp.305-312, 1994.

[49] 이상호, "미등록어를 고려한 한국어 품사태깅 시스템 구현", *한국과학기술원 전산과학과 석사학위논문*, 1995.

[50] 이선정, "신경망을 이용한 한국어 단어범주 예측 및 애매성 해소", *서울대학교 계산통계학과 박사학위논문*, 1994.

[51] 이운재, 최기선, 김길창, "한국어 문서 태깅 시스템," *한국정보과학회 봄 학술발표 논문집*, Vol.20, No.1, pp.805-808, 1993.

[52] 이운재, "한국어 문서 태깅 시스템의 설계 및 구현", *한국과학기술원 전산학과 석사학위논문*, 1993.

[53] 이하규, "Tail-Head 공기 정보를 이용한 한국어 어휘 중의성 해소," *미게재 논문*

[54] 이하규, 김영택, "통계정보에 기반을 둔 한국어 어휘중의성해소," *한국통신학회지*, Vol. 19, No.2, pp.265-275, 1994.

[55] 임권묵, "형태 중의성 해결을 위한 말마디 사전 설계에 관한 연구," *대신대학 논문집*, Vol. 12, pp.341-357, 1992.

[56] 임권묵, "한국어 형태소 분석에서의 오분석 제거와 중의성 해결", *연세대학교 박사학위 논문*, 1996.

[57] 임철수, "HMM을 이용한 한국어 품사 태깅 시스템 구현", *한국과학기술원 전산학과 석사학위 논문*, 1994.

[58] 임해창, 임희석, 윤보현, "자연어처리 연구 동향 : 통계 기반의 자연어 처리," *한국정보과학회지*, Vol.12, No.9, pp.20-30, 1994.

[59] 임희석, 김진동, 임해창, "한국어 특성을 고려한 변형 규칙 기반 품사 태깅," *춘계 인공지능연구회 학술발표 논문집*, pp.3-10, 1996.


**임 해 창**

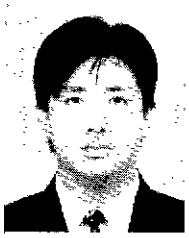
1979 고려대학교 독어독문학과 학사
1983 Missori 주립대학 전산학 석사
1990 Texas 주립대학 전산학 박사
1991～93 고려대학교 전산과학과 조교수
1994～현재 고려대학교 전산과학과 부교수
관심분야 : 자연어처리, 정보 검색, 인공지능

**임 희 석**

1992 고려대학교 전산과학과 학사
1994 고려대학교 전산과학과 석사
1996 고려대학교 전산과학과 박사과정 수료
관심분야 : 자연어처리, 정보 검색, 인공지능, 데이타베이스

**이 상 주**

1992 고려대학교 전산과학과 학사
1995 고려대학교 전산과학과 석사
1995～현재 고려대학교 전산과학과 박사과정
관심분야 : 한국이정보처리, 정보 검색, 기계 번역



김 진 동

1994 고려대학교 전산과학과 학사
1994~현재 고려대학교 전산과학과 석사
관심분야 : 자연어처리, 정보 검색, HCI


## ● 제23회 정기총회 및 추계학술발표회 ●

- 행사일정 : 1996년 10월 25(금)~26일(토)
- 행사장소 : 한국외국어대학교(용인)
- 발표논문 접수마감 : 1996년 8월 24일(토)
- 문의 및 접수처 : 한국정보과학회 사무국
  Tel. 02-588-9246, Fax. 02-521-1352
  서울시 서초구 방배3동 984-1(머리재빌딩)㉾137-063

## ● 제22차 전문대학 전산관련학과 교수 세미나 ●

- 일　　시 : 1996년 7월 9(화)~11일(목)
- 장　　소 : 낙산비치호텔(Tel. 0396-672-4000~15)
- 주　　최 : 전문대학 전산교육연구회
- 문 의 처 : 인하공업전문대학 전자계산과 박진양 교수
  T. 032-870-2325