KTS : 미등록어를 고려한 한국어 품사 태깅 시스템

이상호[O*], 서정연[**], 오영환[*]
[*]한국과학기술원 전산학과, [**]서강대학교 전자계산학과

KTS : A Korean Part-of-Speech Tagging System with Handling Unknown Words

Sangho Lee[O*], Jungyun Seo[**], Yung-Hwan Oh[*]
[*]Dept. of Computer Science, KAIST, [**]Dept. of Computer Science, Sogang University

요 약

자연언어 처리 시스템의 전단부인 형태소 분석 모듈은 해결해야 할 두 가지 문제를 갖고 있다. 하나는 형태소 분석기가 여러 개의 분석 결과를 출력하여 생기는 품사 중의성이고, 다른 하나는 주어진 문장에 미등록어가 사용되어 형태소 분석이 실패되었을 때이다.

본 논문에서는 이 문제들을 해결하는 한국어 품사 태깅 시스템 *KTS*를 소개한다. *KTS*는 주어진 어절에 대해 모든 가능한 분석을 하는 형태소 분석기, 미등록어를 예측하는 미등록어 추정 모듈, 음절 정보와 단서(clue) 형태소를 이용하여 미등록어 후보의 수를 줄이는 미등록어 후보 여과기, 그리고 미등록어의 출현을 모델안에 포함한 품사 태깅 모듈로 구성되어있다.

*KTS*의 품사 태깅 모듈에는 두가지 태깅 방법인 경로 기반 태깅과 상태 기반 태깅의 유일 출력과 다중 출력 기능이 모두 구현되어 있으며, 실험에 의하면, 미등록어가 포함되지 않은 어절에 대해서 89.12%, 미등록어가 포함된 어절에 대해서 68.63%의 정확률을 각각 나타내었다.

1 서론

한국어에서 하나의 어절은 여러개의 형태소 분석 결과로 해석되는 경우가 있으며, 이 중 문장에서 요구하는 분석 결과 선택을 품사 태깅이라 한다. 품사 태깅 문제를 해결하는 품사 태거는 음성 합성[7], 문자 인식[12], 파서의 전단부[8] 등 많은 응용 분야를 갖고 있어서 최근에 활발한 연구가 진행 중이다.

품사 태깅은 방법론에 의해 전체 문장 수준에서 최적화를 시키는 경로 기반 태깅(path-based tagging)과 매 어절 수준에서 최적화를 시키는 상태 기반 태깅(state-based tagging)으로 나뉘며, 품사 태거는 출력 형식에 의해 하나의 결과만 출력하는 유일 출력(single output)과 최적 k개를 출력하는 다중 출력(multiple output)으로 나뉜다[6, 9]. 하지만 방법론의 선택과 무관하게, 사전에 등록되어 있지 않은 단어, 즉 미등록어의 출현에 대해서도 고려를 해야 정확한 품사 태거를 구현할 수 있다. 영어의 경우는 형태소 분석기의 도움을 받아 가능한 접사(suffix)를 구한 후, 품사에서 단어가 출현할 확률 $P(w_i|t_i)$ 대신 품사에서 접사가 출현할 확률 $P(s_i|t_i)$를 이용하는 것이 일반적이다[6, 5, 13].

하지만 한국어의 경우는 그 문제의 양상이 다르다. 첫째, 영어에서 단어는 두 스페이스(space) 사이에 존재하여 미등록어를 분명히 알 수 있는 반면, 한국어의 경우는 미등록어가 어절 안에 존재하여 단지 그 어절의 형태소 분석이 실패했다는 정보만 얻을 수 있을 뿐 어느 부분이 미등록어인지도 알지 못한다. 둘째, 영어에서는 미등록어의 접사 정보를 이용하여 미등록어의 품사를 추정하는 반면, 한국어에서는 미등록어 뒤에 오는 조사나 어미 정보를 이용하여 품사를 추정하게 된다. 바로 이러한 점들이 미등록어를 처리하는데 더욱 어려움을 주고 있다.

본 논문에서는 위의 문제들을 해결하기 위해 구현된 한국어 품사 태깅 시스템 *KTS*를 소개한다[1]. *KTS*는 통계에 기반한 품사 태깅 방법을 이용하며, 그림 1에 보여지듯이 크게 네 개의 모듈, 형태소 분석기, 미등록어 추정 모듈, 미등록어 후보 여과기, 태깅 모듈로 구성되어있다. 입력 문장에 대해 형태소 분석기는 어절의 오른쪽에서부터 읽어들여 모든 가능한 형태소 분석 결과를 만든다. 본 형태소 분석기는 [2]에서 제안된 분석기를 이용하므로, 모든 가능한 분석 결과를 형태소 격자 구조로 표현하고 격자 내의 가능한 경로가 하나의 분석 결과를 의미한다. 만약 입력 어절에 미등록어가 있을 경우에는 미등

[1]본 논문에서 구현한 *KTS*는 cair-archive.kaist.ac.kr::/pub/AI/nlp/parsing/kortaggers/kts/kts.tar.gz에 공개되어 있습니다.

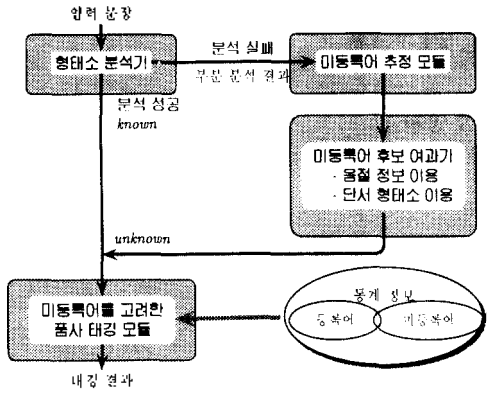

그림 1: *KTS* 구성도

록어 추정 모듈에서 분석기가 남긴 미 완성의 격자를 바탕으로 모든 가능한 미등록어를 추정한다. 이 때 미등록어는 항상 어절의 앞 부분에 있다는 가정을 이용한다. 이렇게 복구된 격자는 미등록어 후보의 수가 상당히 높으므로 미등록어 후보 여과기에서는 한국어의 음절 정보와 단서(clue) 형태소를 이용하여 후보 개수를 줄인다. *KTS*의 최종 모듈인 태깅 모듈은 기존의 태깅 방법과 달리 미등록어의 출현을 예외적인 현상으로 간주하지 않고, 미등록어의 출현 사전도 모델안에 포함시켰다.

본 논문의 구성은 다음과 같다. 2장에서 *KTS*를 이루는 각 모듈들을 설명하고 3장에서 등록어 및 미등록어에 대한 태깅 정확률을 알아본다. 아울러 경로 기반 태깅과 상태 기반 태깅의 성능 비교도 함께 이루어지며 4장에서 결론을 맺는다.

2 시스템 구성

2.1 형태소 분석기

*KTS*의 형태소 분석기는 주어진 어절의 오른쪽에서 왼쪽으로 형태소를 찾으며, 형태소 분석 결과를 하나의 형태소 격자로 표현한다[2]. 예를 들어 문장의 i번째 어절, w_i가 '빔'일 경우, "좌석이 빔(emptiness)"의 'B/형용사 + D/명사형 전성어미'와 "레이저 빔(beam)"의 'B/보통명사'로 분석이 되고 이는 그림 2와 같이 두 개의 경로를 갖는 형태소 격자로 표현된다. 형태소 분석기가 오른쪽에서 왼쪽으로 형태소를 찾는 이유는 한국어의 어절 구조는 크게 "체언 + 조사", "용언 + 어미"로 이루어지고, 대부분의 경우 미등록어는 체언 혹은 용언이므로 바로 형태소 분석기가 분석을 실패하더라도 그 어절에서 가능한 모든 조사 혹은 어미는 찾아 줄 수 있기 때문이다[2].

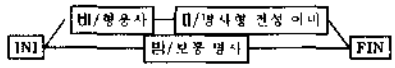

그림 2: '빔'의 형태소 격자

본 절에서 설명하는 형태소 분석기는 주어진 어절에 대해 만들어진 격자가 최소 하나 이상의 경로를 갖으면 그 어절에 미등록어가 없다고 가정하지만, 사실 이러한 가정은 엄격하게 말해 옳지 못한 가정이다. 예를 들면, *KTS*의 사전에 'B/형용사'가 등록되어 있지 않고 'B/체언'은 등록되어 있다면, 그림 2에서 아래의 경로만이 존재하게 될 것이다. 이런 경우에 *KTS*는 미등록어가 존재한다고 판단하지 않으므로, 위 결과만을 가지고 태깅을 수행하게 되며, 만약 주어진 문장에서 'B/형용사 + D/명사형 전성어미'로 분석되는 것이 옳은 경우에 항상 그릇된 태깅 결과를 선택하게 된다.

본 논문에서는 주어진 어절을 분석할 때, 하나의 경로도 만들지 못한 어절을 '미등록 어절'이라고 부르며, 위에서 기술한 것처럼 잘못된 결정에 의한 태깅 오류를 '미등록 어절 정의 오류'라고 한다. 실험에 의하면 이러한 오류가 약 6% 정도 태깅 정확률을 떨어뜨려 시스템의 성능을 저하시키는 가장 큰 원인이 되나 형태소 수준의 정보만을 이용하여 해결하기에는 힘든 문제이고 더 많은 연구가 필요하다.

2.2 미등록어 추정 모듈

예를 들어 다음의 문장을 분석한다고 하자.

"홀로그래피는 사진과 레이저 빔을 이용하여 만드는 삼차원 영상이다."

위의 문장에서 '홀로그래피'가 사전에 등록되어 있지 않다고 가정하면, '홀로그래피는'에 대한 형태소 격자는 그림 3에서 처럼 단지 '는/{전성 어미, 보조사}'와 'L/전성 어미'의 정보만 얻을 수 있고 하나의 경로도 존재하지 않는다. 이런 경우에 *KTS*는 미등록어 추정 모듈을 수행하여 모든 가능한 미등록어를 추정하게 된다. 미등록어 추정 모듈은 형태소 분석기가 남긴 부분 격자를 가지고 품사 접속표를 이용하여 추정을 하게 되는데, 예를 들면, '는/전성 어미' 앞에 올 수 있는 품사는 형용사와 동사이므로 '홀로그래피'는 이 두 가지 품사 중 하나일 가능성이 있는 것이다[3].

이와 같은 방법으로 미등록어를 추정하게 되면 그림 4에서 보듯이 등록 어절의 형태소 격자와 전혀 다름이 없고 단지 경로가 많다는 특징이 있을 뿐이다. 실험에 의하면, 등록 어절의 경우는 경로의 갯수가 평균 3.32개인데 반해 미등록 어절의 경우는 평균 20.01개 정도이다.

[2] 본 논문에서 이용하는 품사 집합은 [1]에서 정의한 52개의 품사 집합에서 '부사형 전성 어미'와 '부사 과생 접미사'를 포함시켜 총 51개의 품사를 갖고 있다.

[3] 본 논문에서는 미등록어가 될 수 있는 품사단위로 다음과 같이 총 8개를 선택하였다. 동작성 보통 명사, 상태성 보통 명사, 보통 명사, 고유 명사, 동사, 형용사, 접두사, 부사

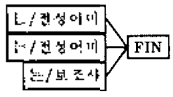

그림 3: '홀로그래피는'의 분석 실패된 격자

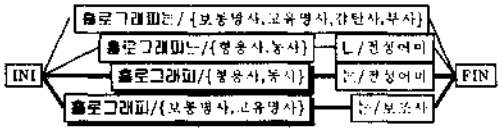

그림 4: '홀로그래피는'의 복구된 격자

2.3 미등록어 후보 여과기

미등록 어절의 복구된 격자는 상당히 많은 경로를 갖고 있다. 이는 곧 높은 중의성을 뜻하고 결과적으로 시스템의 성능을 저하시킨다. 본 논문에서는 시스템의 성능을 높이기 위해 두 가지 휴리스틱을 이용하여 미등록어 후보의 갯수를 줄이다. 먼저 그림 4의 경우 음절 '는'은 절대 체언, 감탄사, 부사 등의 마지막 음절로 사용될 것이라고 생각되지 않으므로 '홀로그래피는/{보통명사,고유명사,감탄사,부사}' 노드는 제거한다. 또한 'ㄴ'는 형용사와 동사의 마지막 음절로 절대 사용되지 않으므로 '홀로그래피는/{형용사, 동사}' 노드를 제거하여 최종적으로 4개의 경로만을 갖는 격자를 얻게 된다.

이렇게 한국어의 음절 정보를 이용하여 후보를 여과하는 방법 외에 단서(clue) 형태소를 이용하여 후보를 여과하는 방법이 있다. 이는 형태소 분석기가 남긴 격자내에 그 어절의 구성을 추정하기에 충분한 단서 형태소가 발견되면 그 형태소를 지나지 않는 경로는 모두 제거하는 방법이다. 예를 들어 "우회시켜"라는 어절을 분석할 때, '우회'가 미등록어일 경우 미등록어 추정 모듈에서는 형태소 격자를 그림 5와 같이 복구시킨다. 이 때, '시키/동사 파생 접미사'는 단서 형태소이므로 나머지 '우회시키', '우회시켜' 노드들은 모두 제거하여 최종적으로 미등록어 후보로는 '우회' 노드 하나만을 갖는 형태소 격자가 된다.

2.4 품사 태깅 모듈

한국어의 어절은 그림 2에서 보듯이 여러개의 길이가 다른 형태소 분석 결과를 얻게 된다. 이런 특징은 영어에서의 품사 태깅과는 다른 정의가 필요하다. 즉, 영어에서는 주어진 단어에 대해 최적 품사를 선택하는 것으로 품사 태깅을 정의하나, 한국어의 경우는 가능한 형태소 분석 결과 중 최적의 분석 결과를 선택하는 것으로 생각하는 것이 옳다.

본 논문에서는 i번째 어절을 w_i로 표현하고 그 어절의 형태소 분석 결과를 구성하는 형태소 열은 m_i로, 형태소 열에 해당하는 품사 열은 t_i로 각각 표현한다. 예를

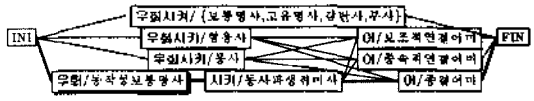

그림 5: '우회시켜'의 복구된 격자

들면, '밤'의 형태소 분석 결과 중 하나가 "비/형용사 + ㅁ /명사형 전성어미"일 경우에 w_i는 '밤', m_i는 '비,ㅁ ', t_i는 '형용사,명사형 전성어미'이다. 또한 m_i와 t_i는 각각 $m_{i1}m_{i2}\ldots m_{iN_i}$, $t_{i1}t_{i2}\ldots t_{iN_i}$로 나누어 질 수 있어서 위의 경우에 m_{i1}은 '비', t_{i1}은 '형용사'이다.

한국어의 품사 태깅은 다음과 같이 주어진 문장에 대해 최적의 형태소 분석 결과를 선택하는 $\phi(w_{1..n})$로 표현할 수 있다.

$$\phi(w_{1..n}) \stackrel{\text{def}}{=} \arg\max_{m_{1..n},t_{1..n}} P(m_{1..n},t_{1..n}|w_{1..n}) \quad (1)$$

식 (1)은 다음과 같이 전개된다.

$$\begin{aligned}
\phi(w_{1..n}) &\stackrel{\text{def}}{=} \arg\max_{m_{1..n},t_{1..n}} P(m_{1..n},t_{1..n}|w_{1..n}) && (2)\\
&= \arg\max_{m_{1..n},t_{1..n}} \frac{P(w_{1..n}|m_{1..n},t_{1..n})}{P(w_{1..n})} P(m_{1..n},t_{1..n}) && (3)\\
&= \arg\max_{m_{1..n},t_{1..n}} P(m_{1..n},t_{1..n}) && (4)\\
&= \arg\max_{m_{1..n},t_{1..n}} P(m_{1..n}|t_{1..n})P(t_{1..n}) && (5)\\
&\cong \arg\max_{m_{1..n},t_{1..n}} \prod_{i=1}^{n} P(m_i|t_i)P(t_i|t_{i-1}) && (6)
\end{aligned}$$

식 (3)에서 $P(w_{1..n}|m_{1..n},t_{1..n})$은 항상 1.0으로 간주되며 $P(w_{1..n})$은 전체 함수에 영향을 끼치지 않으므로 제거한다. 식 (6)은 식 (5)에서 일차 마르코프(Markov) 가정을 이용한 것이고 식 (6)을 근사시키기 위해 먼저 $P(t_i|t_{i-1})$을 다음과 같이 간략화시킨다.

$$\begin{aligned}
P(t_i|t_{i-1}) &\stackrel{\text{def}}{=} P(t_{i1}..t_{iN_i}|t_{i-1,1}..t_{i-1,N_{i-1}}) && (7)\\
&\cong P(t_{i1}..t_{iN_i}|t_{i-1,N_{i-1}}) && (8)\\
&= P(t_{i1}|t_{i-1,N_{i-1}})P(t_{i2}|t_{i1}t_{i-1,N_{i-1}}) \\
&\qquad \ldots P(t_{iN_i}|t_{iN_i-1}..t_{i1}t_{i-1,N_{i-1}}) && (9)\\
&\cong P(t_{i1}|t_{i-1,N_{i-1}}) \prod_{j=2}^{N_i} P(t_{ij}|t_{ij-1}) && (10)
\end{aligned}$$

식 (8)이 의미하는 바는 이전 어절의 품사 열에서 현재 어절의 품사 열로 전이하는 확률은 이전 어절의 마지막 품사에서 현재 어절의 품사 열로 전이하는 확률로 근사시킬 수 있다는 뜻이다. 또한 식 (9)에서 어절 내의 품사는 바로 이전 품사에만 영향을 받는다는 가정을 이용하여 식 (10)을 얻었다.

식 (6)의 확률 $P(m_i|t_i)$를 계산하기에 앞서, 미등록어를 처리하기 위해 k_i라는 새로운 변수를 도입하자. 즉, k_i가 0일 경우에는 w_i가 미등록 어절이고 1일 경우에는 w_i가 등록 어절임을 표현한다고 하자. 또한 k_i도 다른 m_i, t_i와 같이 $k_{i1}k_{i2}\ldots k_{iN_i}$로 세분될 수 있고, 미등록어는 항상 어절의 앞 부분에 존재하므로 $k_{i2}\ldots k_{iN_i}$는 항상 1이며

의미가 있는 변수는 k_{i1} 뿐이다. 이와 같은 과정에 의해 $P(m_i|t_i)$는 다음과 같이 표현될 수 있다.

$$P(m_i|t_i) = P(m_{i1}, k_{i1}|t_i) \quad (11)$$
$$\cong \hat{P}(k_{i1}|t_i)P(m_i|t_i, k_{i1}) \quad (12)$$
$$\cong \hat{P}(k_{i1}|t_i)[k_{i1}P(m_i|t_i, k_{i1}=1) + (1-k_{i1})P(m_i|t_i, k_{i1}=0)] \quad (13)$$

이제 $\hat{P}(k_{i1}|t_i)$와 $P(m_i|t_i, k_{i1})$을 다음과 같이 간략화시킨다.

$$\hat{P}(k_{i1}|t_i) \stackrel{def}{=} \hat{P}(k_{i1}|t_{i1}..t_{iN_i}) \quad (14)$$
$$\cong \hat{P}(k_{i1}|t_{i1}) \quad (15)$$

$$P(m_i|t_i, k_{i1}) \stackrel{def}{=} P(m_{i1}..m_{iN_i}|t_{i1}..t_{iN_i}k_{i1}) \quad (16)$$
$$= \frac{P(k_{i1}m_{i1}t_{i1}..m_{iN_i}t_{iN_i})}{P(t_{i1}..t_{iN_i}k_{i1})} \quad (17)$$
$$= P(t_{iN_i})P(m_{iN_i}|t_{iN_i})P(t_{iN_i-1}|t_{iN_i}m_{iN_i}) \cdot P(m_{iN_i-1}|t_{iN_i-1}t_{iN_i}m_{iN_i}) \cdots P(k_{i1}t_{i1}|t_{i2}..t_{iN_i}m_{i2}..m_{iN_i}) \cdot P(m_{i1}|k_{i1}t_{i1}..t_{iN_i}m_{i2}..m_{iN_i}) / P(t_{i1}..t_{iN_i}k_{i1}) \quad (18)$$
$$\cong P(m_{i1}|k_{i1}t_{i1}..t_{iN_i}m_{i2}..m_{iN_i}) \cdot \prod_{j=2}^{N_i} P(m_{ij}|t_{ij}) \quad (19)$$

식 (19)의 $P(m_{i1}|k_{i1}t_{i1}..t_{iN_i}m_{i2}..m_{iN_i})$는 k_{i1}의 값에 의해 다음과 같이 서로 다르게 근사시킨다.

$$P(m_{i1}|k_{i1}=1, t_{i1}..t_{iN_i}m_{i2}..m_{iN_i}) \cong P(m_{i1}|t_{i1}) \quad (20)$$
$$P(m_{i1}|k_{i1}=0, t_{i1}..t_{iN_i}m_{i2}..m_{iN_i}) \cong P(t_{i1}|t_{i2}, m_{i2}) \quad (21)$$

식 (21)은 m_{i1}대신 t_{i1}을 이용한 것인데 미등록어 m_{i1}은 형태소를 이루는 음절의 특성이 품사를 결정하는데 도움을 준다고 간주되지 않고 오히려 주위의 품사에 영향을 받는다고 가정한 것에 기인한다.

이 식들을 식 (19)에 대입하고 식 (19)를 식 (13)에 적용하여 최종적으로 다음과 같은 품사 태깅 함수 $\phi(w_{1..n})$을 얻게 된다.

$$\phi(w_{1..n}) = \arg\max_{m_1..n, t_1..n} \prod_{i=1}^{n} [P(m_i|t_i) \prod_{j=2}^{N_i} P(t_{ij}|t_{ij-1}) \cdot P(t_{i1}|t_{i-1N_{i-1}})] \quad (22)$$

$$P(m_i|t_i) \cong \hat{P}(k_{i1}|t_{i1})[k_{i1} \prod_{j=1}^{N_i} P(m_{ij}|t_{ij}) + (1-k_{i1})P(t_{i1}|t_{i2}m_{i2}) \prod_{j=2}^{N_i} P(m_{ij}|t_{ij})] \quad (23)$$

KTS는 식 (22)를 기본 식으로 하고 다중 출력을 할 수 있는데, 두 개의 모델 변수를 갖는 경로 기반 태깅과 한 개의 모델 변수를 갖는 상태 기반 태깅을 선택할 수 있도록 구현되어 있다. 먼저, 경로 기반 태깅은 다음 조건을 만족하는 형태소 분석 열 $m_{1..n}^{(j)}, t_{1..n}^{(j)}$를 구한다.

- $P(m_{1..n}^{(j)}, t_{1..n}^{(j)}|w_{1..n}) \geq \sigma P(m_{1..n}^{(b)}, t_{1..n}^{(b)}|w_{1..n})$, $m_{1..n}^{(b)}, t_{1..n}^{(b)}$는 최적의 형태소 분석 열이고 σ는 0과 1사이의 임의의 값이 될 수 있는 모델 변수이다. (σ가 1일 경우에는 하나의 분석 결과만 구하는 유일 출력 태깅가 된다.)
- 최대 k개의 형태소 분석 열을 구할 수 있다.

상태 기반 태깅은 전체 문장 수준에서 최적화를 하지 않고 매 어절에 대해 다음 조건을 만족하는 형태소 분석 결과 $m_i^{(j)}, t_i^{(j)}$를 구한다.

- $P(m_i^{(j)}, t_i^{(j)}|w_{1..n}) \geq \sigma P(m_i^{(b)}, t_i^{(b)}|w_{1..n})$, $m_i^{(b)}, t_i^{(b)}$는 w_i의 최적 형태소 분석 결과이고 σ는 0과 1사이의 임의의 값이 될 수 있는 모델 변수이다.

KTS는 두가지 태깅 방법의 다중 출력을 구현하기 위해 경로 기반 태깅은 tree-trellis 알고리즘을, 상태 기반 태깅은 Forward Backward 알고리즘을 각각 이용한다[4, 10, 11]. 복히 연속 음성 인식에서 이용되는 tree-trellis 알고리즘은 많은 backpointer를 저장해야하는 한편, 본 논문의 품사 태깅 문제는 이산적인 심볼 열을 처리하는 것이므로 이를 간략화시켜 이용하고 있다[3].

3 실험 결과

KTS의 태깅 정확률을 평가하기 위해서 총 54,232 어절(약 116,031 형태소)의 말뭉치를 다음과 같이 세 개의 말뭉치로 나누었다.

- 학습 말뭉치-I : 45,851 어절
- 학습 말뭉치-II : 3,652 어절
- 실험 말뭉치 : 4,729 어절

학습 말뭉치-I은 $P(t_{ij}|t_{ij-1})$, $P(t_{i1}|t_{i-1N_{i-1}})$, $P(m_{ij}|t_{ij})$ 그리고 $P(t_{i1}|t_{i2}m_{i2})$를 구하기 위해서 사용되었으며, 학습 말뭉치-II는 $P(k_{i1}|t_{i1})$을 얻기 위해 이용되었다. 이 확률은 학습 말뭉치-I에 존재하지 않고 학습 말뭉치-II에 나타난 단어를 미등록어라고 가정하여 얻어진다. 실험에 와 하면 말뭉치와 크기가 작은 관계로 보통 명사가 미등록어로 될 확률이 높은데 아는 더 많은 말뭉치, 충분히 큰 사전으로 학습하면 실제 상황을 더욱 잘 모델링할 수 있을 것이라 믿는다.

경로 기반 태깅과 상태 기반 태깅의 성능을 비교하기 앞서, 경로 기반 태깅은 문장 수준에서 최적화시키지만 상태 기반 태깅은 어절 수준에서 최적화를 시키므로, 본 논문에서는 경로 기반 태깅의 결과를 병합하여 어절 수준으로 평가하고 비교하였다. 표 1, 2에 경로 기반 태깅과 상태 기반 태깅의 성능이 비교되어 있는데 σ가 0.2일 때 경로 기반 태깅은 등록 어절에 대해 평균 1.1080개, 미등록 어절에 대해 평균 1.1794개의 후보 결과들을 선택하여 각각 90.99%, 72.18%의 정확률을 얻어 상태 기반 태깅보다 약간 좋은 결과를 내었다. 물론 경로 기반 태깅은 전체 후보의 갯수를 제어하는 모델 변수 k의 도입이 도움이 되었을 것이라고 생각되나, 근본적으로 경로 기반

표 1: 경로기반 태깅 다중출력의 정확률

경로 기반 $k=10$	$\sigma=1.0$ 정확률	$\sigma=1.0$ 어절당 후보갯수	$\sigma=0.2$ 정확률	$\sigma=0.2$ 어절당 후보갯수	$\sigma=0.1$ 정확률	$\sigma=0.1$ 어절당 후보갯수
등록 어절	89.12	1.0000	90.99	1.1080	91.28	1.1636
미등록 어절	68.63	1.0000	72.18	1.1794	76.13	1.3708

표 2: 상태기반 태깅 다중출력의 정확률

상태 기반	$\sigma=1.0$ 정확률	$\sigma=1.0$ 어절당 후보갯수	$\sigma=0.2$ 정확률	$\sigma=0.2$ 어절당 후보갯수	$\sigma=0.1$ 정확률	$\sigma=0.1$ 어절당 후보갯수
등록 어절	89.15	1.0000	90.99	1.1162	91.37	1.1885
미등록 어절	68.63	1.0000	72.18	1.1834	77.71	1.4615

태깅은 상태 기반 태깅보다 더 큰 탐색 공간을 갖기때문에 더욱 정밀한 탐색을 할 수 있다는 장점을 주고 있다. σ가 1.0일 경우, 즉 유일 출력의 경우에는 두 태깅 방법 모두 등록 어절에 대해 89.15%정도의 비교적 낮은 정확률을 나타내었는데 이것은 2.1장에서 설명한 것처럼 '미등록 어절 정의 오류'에 기인한다. 이 오류를 무시할 경우 등록 어절에 대해 96.58%로 정확률이 향상되는 것을 관찰할 수 있었다.

4 결론

본 논문에서는 미등록어를 고려한 한국어 품사 태깅 시스템 *KTS*를 소개하였다. *KTS*는 주어진 문장의 각 어절에 대해 형태소 격자를 만드는 형태소 분석기, 미등록어가 어절 내에 있다고 간주될 때 모든 가능한 미등록어를 추정하는 미등록어 추정 모듈, 음절 정보와 단서 형태소를 이용하여 후보의 갯수를 줄이는 미등록어 후보 여과기, 미등록어의 출현 사건을 모델 안에 포함시킨 품사 태깅 모듈로 구성되어 있다. 품사 태깅 모듈은 또한 두 가지 태깅 방법, 경로 기반 태깅과 상태 기반 태깅의 다중 출력이 모두 구현되어 있어 본 시스템의 이용자가 태깅 방법을 선택할 수 있도록 했고, 두 방법의 성능은 경로 기반 태깅이 약간 우수한 것으로 관찰되었다.

*KTS*의 태깅 정확률은 등록 어절과 미등록 어절에 대해 각각 89.15%, 68.63% 정도의 비교적 낮은 정확률을 보였으나, 이는 '미등록 어절 정의 오류'에 기인한다. 이 오류를 무시할 경우 등록 어절에 대해 96.58%의 정확률을 얻어 더 많은 표제어의 사전을 이용할 경우, 본 시스템은 정보 검색 시스템의 문서 분석 모듈, 문서-음성 변환 시스템에서의 언어 처리 모듈, 자연어 처리 시스템의 파서 전단부등 많은 응용 분야에 쉽게 적용될 수 있다고 생각한다.

5 감사의 글

본 논문의 시스템을 구현하는 동안 많은 도움을 주신 김 재훈, 조 정미, 김 형근씨에게 감사드립니다.

참고 문헌

[1] 김 재훈, 서 정연, 자연언어 처리를 위한 한국어 품사 태그, Technical Report, CAIR-TR-94-55, 한국과학기술원, 인공지능연구센터, 1994.

[2] 이 상호, 김 재훈, 조 정미, 서 정연, "부분 분석 결과를 공유하는 한국어 형태소 분석", 제 11회 음성통신 및 신호처리 워크샵 논문집, pp. 75-79, 1994.

[3] 이 상호, 미등록어를 고려한 한국어 품사 태깅 시스템 구현, 한국과학기술원 석사학위논문, 1995.

[4] L. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Process," *inequalities*, Vol. 3, pp. 1-8, 1973.

[5] E. Charniak, C. Hendrickson, N. Jacobson, and M. Perkowitz, "Equations for Part-of-Speech Tagging," *Proc. of National Conf. on Artificial Intelligence (AAAI-93)*, pp. 784-789, 1993.

[6] E. Charniak, G. Carroll, J. Adcock, A. Cassandra, Y. Gotoh, J. Katz, M. Littman and J. McCann, "Taggers for parsers," *Technical Report, CS-94-06*, Dept. of Computer Science Brown University, 1994.

[7] K. W. Church, "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," *Proceedings of Applied Natural Language Processing*, Austin, Texas, 1988.

[8] Carl G. de Marcken, "Parsing the LOB Corpus," *Proceedings of the 1990 Conference of the Association for Computational Linguistics*, pp 243-259, 1990

[9] G. F. Foster, *Statistical Lexical Disambiguation*, Master's thesis, McGill Univ. School of Computer Science, Montreal, Canada, 1991.

[10] Frank K. Soong and Eng-Fong Huang, "A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition," *IEEE International Conference on Acoustic Speech and Signal Processing*, pp. 546-549, 1991.

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

[12] Rohini K. Srihari and Charlotte M. Baltus, "Combining Statistical and Syntactic Methods in Recognizing Handwritten Sentences," *Fall Symposium Series, Probabilistic Approaches to NL, AAAI*, pp. 121-127, 1992.

[13] R. Weischedel, R. Scewartz, J. Ralmucci, M. Meteer and L. Rawshaw, "Coping with Ambiguity and Unknown Words through Probabilistic Models," *Computational Linguistics*, Vol. 19, No. 2, pp. 359-382, 1993.