

# 음성합성을 위한 품사태깅시스템의 속도 개선

강정세, 박준

한국전자통신연구원, 네트워크연구소, 음성정보연구센터

## A fast POS tagging method for speech synthesis

Jeong-se Kim, Jun Park

Speech Technology Research Center, Network Laboratory, ETRI

E-mail : jungskim @etri.re.kr

### 요약

본 논문에서는 음성합성을 위한 의사형태소 품사 태깅 시스템의 속도를 개선하는 방법으로 정확률을 다소 희생하더라도 속도개선이 될 수 있는 방법을 제안하고자 한다. 형태소 해석 시에는 종성으로 올 수 있는 자모를 제외한 나머지에 대해서는 음절단위로 구성하는 변형된 Tabular 파싱법으로 해석하는데, 여기에다 일반적으로 적용 가능한 몇 가지의 규칙을 추가함으로써 해석 가능한 노드들을 줄였다. 태깅 시에는 한국어의 특성상 어절 하나씩을 품사 태깅하였을 경우에도 상당히 정확하다는 점을 이용하여 어절 내부에서는 full search 를 하고 그 다음 어절은 이전 어절의 제일 높은 값을 가지는 품사열 정보를 활용하는 방법을 제안한다. 제안한 시스템은 32 개 품사 태그셋에 2 만 형태소 사전을 이용해 실험한 결과, 기존의 시스템보다 약 60%이상의 속도 개선을 보였으며, 정확률은 약 1% 정도 떨어졌다.

### 1. 서론

품사 태깅 시스템은 자연언어 처리 뿐 아니라, 음성인식, 음성합성, 언어현상발견, 정보검색 등 여러 분야에 사용되고 있다. 물론 각각의 적용분

야에 따른 정의가 조금씩 다르겠지만, 품사 태깅 시스템의 정확률을 높이며 동시에 검색 속도를 줄이기 위해서 많은 연구를 해왔다. 그러나, 속도는 합성기에서 원하는 만큼의 수준에 미치지 못하고 있다. 따라서 본 논문에서는 정확률이 좀 떨어지더라도 합성기의 속도를 개선하기 위한 방안을 제안한다.

본 논문의 구성은 다음과 같다. 2 장에서는 기존의 품사 태깅 시스템에 대해서 기술하고 3 장에서는 음성 합성용 품사 태깅 시스템에 대해서 제안하며 4 장에서는 실험 및 평가를 하고 5 장에서 결론을 맺는다.

### 2. 기존 품사 태깅 시스템

#### 2.1. 형태소 해석 방법

한국어 형태소 해석 방법으로는 Head-tail 구분법과 Tabular 파싱법, 최장/최단/양방향 일치법, 음절단위 분석법등이 있다. 본 논문에서 사용하고 있는 해석 방법은 Tabular 파싱법을 이용하므로 아래에는 Tabular 파싱법에 대해서 설명한다.

Tabular 파싱법은 bottom-up 방식에 의한 형태소 분석으로 먼저 단어를 이루고 있는 모든 자모 단위의 문자열에 대해서 사전을 검색하여

모든 가능한 어휘 형태소와 문법 형태소를 추출하고, 이렇게 추출된 형태소들은 삼각 테이블에 저장되고 여기에 형태소들의 접속 관계를 검사하기 위해서 접속 정보표를 사용하여 연결 가능한 모든 형태소들의 리스트를 찾는 방법이다.

마등록어를 추정할 시에는 bottom-up 방식으로 해석 가능한 부분 외의 나머지 부분에 대해서 추정한다.

이 방식은 필요 없는 형태소들을 추출하여 삼각 테이블에 가지고 있어서 메모리의 낭비가 있다. 또한 bottom-up 방식을 이용하기 때문에 형태소 태깅으로 해석 자료를 넘기기 위해서 다시 자료구조를 바꾸어야 하는 문제점이 있다.

## 2.2. 형태소 태깅 시스템.

형태소 태깅 방법은 은닉 마르코프 모델을 사용하여 식은 다음과 같다.

$$P = \operatorname{argmax}_P \sum_i \log \Pr(p_i | p_{i-2} p_{i-1}) + \sum_i \log \Pr(W_i | P_i)$$

여기서 P 는 최적의 품사열,  $W_i$  는  $i$  번째의 단어,  $P_i$  는  $W_i$  의 태그를 나타낸다. HMM 접근법의 태깅 과정에서 Viterbi 알고리즘을 적용하여 단어열의 길이에 대해 선형적인 시간 내에 구할 수 있다. 그리고 문맥확률과 어휘확률은 말뭉치로부터 추출된 통계자료를 이용하여 다음과 같이 계산할 수 있다. 여기서 Freq 는 말뭉치에서의 출현빈도를 나타낸다.[1]

$$\Pr(P_i | P_{i-2} P_{i-1}) \approx \frac{\text{freq}(P_{i-2} P_{i-1} P_i)}{\text{freq}(P_{i-2} P_{i-1})}$$

$$\Pr(W_i | P_i) \approx \frac{\text{freq}(W_i, P_i)}{\text{freq}(P_i)}$$

위의 공식을 사용하고 Viterbi 알고리즘을 써서 최적의 값을 구할 수 있다. 그러나 형태소 해석의

결과를 Viterbi 알고리즘에 적용하기 위해서는 자료구조의 변환이 있어야 한다. 실질적으로 계산 자체는 그렇게 많은 시간이 요구되지는 않으나 이 자료구조의 변환 시간이 태깅 시간의 거의 다를 차지한다고 해도 과언이 아니다.

## 2.3. 어절 품사 태깅 시스템

태거의 문제는 속도와 정확율로 나뉜다. 정확율은 태거의 사전을 보완함으로써 어느 정도까지는 해결이 된다고 본다. 그러나, 속도는 합성기에서 원하는 만큼의 수준에 미치지 못하고 있다. 따라서 어절별로 태깅을 하여 그 속도를 높여 보고자 한다. 현재의 접속 정보표와 품사 trigram 은 문장으로 구성되었을 때의 접속 정보표와 품사 trigram 이다. 따라서 어절 태깅 시에는 어절 태깅용 접속 정보표와 품사 trigram 을 사용해야 할 것이다. 그 결과는 기존 것을 사용할 때보다 약 50% 정도 오류가 줄어들었다.

이것은 아래의 실험 결과를 통해서 알 수 있다. 테스트 셋은 2,008line, 23,114 어절을 open test 하였다.

사전의 형태소 수 : 110,952 개  
문장 태깅 시의 오류율 : 5.86%  
어절 태깅 시의 오류율 : 11.5%

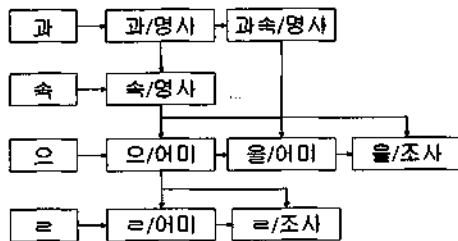
오류를 분석해 본 결과 거의 대부분이 조사, 어미에서 틀리고, 단음절의 단어일 경우는 빈도가 가장 많이 나타난 형태로 태깅이 되었다. 이것을 어느 정도 보정하기 위해서는 어절 태깅을 하되 어절과 어절 사이에는 다른 trigram 을 적용하는 방법이 필요하다.

## 3. 음성 합성용 품사 태깅 시스템

### 3.1. 형태소 해석

기본적으로 Tabular 파싱법을 따르나, 2.1 에서 제시한 문제점들을 해결하기 위해 분리가 가능한 음소(ㄴ, ㄹ, ㅁ, ㅂ, ㅅ)들을 제외한 나머지에 대해서는 음절 단위의 변형된 Tabular 파싱법으로 해석을 한다.

변형된 Tabular 파싱법은 삼각 테이블을 작성 하는 것이 아니라 이와 유사한 자료구조를 가지는 linked-list 를 작성하여 이것을 대신한다. 이것은 어절 내의 최소의 분리 가능한 형태소로 나누어 하나의 리스트를 만들고, 그 list 의 옆으로 형태소들을 조합해서 리스트를 하는 것이다. 여기서 사전을 미리 검색하여 불필요하게 생성되는 형태소들을 미연에 방지할 수 있다. 이렇게 생성된 리스트들의 접속 관계를 따져서 어절의 끝까지 연결되는 것이 있으면 나머지는 삭제하면 된다. 아래 [그림 1]은 그 예이다.



[그림 1] “과속들”에 대한 예

이렇게 함으로써 bottom-up 방식이 아니라 top-down 방식으로 접속 정보표를 이용하여 형태소 해석을 할 수 있다. 따라서 bottom-up 으로 형태소 해석을 해서 다시 자료구조를 바꾸어야 할 이유가 없을 뿐 아니라 삼각 테이블을 위한 메모리의 낭비를 줄일 수 있게 된다.

여기다가 아래의 한국어의 특성을 살리는 규칙들을 적용한다. 규칙들이 많으면 많을수록 좀 더 나은 해석결과가 나올 것이나 이것 또한 시스템의 속도를 생각해야 한다.

(1) “ 형태소 해석 시에 어절의 처음 형태소이면

조사, 접미사, 어미가 올 수 없다” 라는 어절 구조를 반영

(2) 형태소 해석 시에 “ 체언+어미” 로 끝나는 것에 대해서 “ 체언+조사” 가 있으면 조사만 선택

(3) 형태소 해석 시에 “ 용언+조사” 로 끝나는 것에 대해서 “ 용언+어미” 가 있으면 어미만 선택.

(4) 형태소 해석 시에 받침이 있는 체언은 뒤에 “ 은, 이, 을”이 와야 정상. “는, 가, 를”이 오면 다르게 해석

(5) 형태소 해석 시에 받침이 없는 체언은 뒤에 “ 는, 가, 를”이 와야 정상. “은, 이, 을”이 오면 다르게 해석.

### 3.2 미등록어 해석

미등록어 추정도 바뀌어진다. 이것은 해석 시에 어절의 해석이 실패할 경우에 일어나는 것으로 미등록어의 98.6%가 명사 미등록어임을 감안해서 다음과 같이 바꾸었다.[2]

우선 2 음절까지 어절의 첫 부분에 올 수 있는 품사들을 리스트하고 2 음절 이후의 해석 가능한 품사들을 찾는다. 찾는 방법은 형태소 해석과 규칙을 제외하고는 동일하다. 끝까지 해석되는 것이 있으면 여기서 마치고 없으면 3 음절로 하여 다시 계산한다.

### 3.3. 품사 태깅

통계 기반 품사 태깅은 대체로 문장 전체에 대한 Viterbi search 나 Viterbi beam search 를 수행하기 때문에 정확률은 높으나 다채널 음성 합성용으로 활용하기에는 Viterbi search 를 위한 자료구조의 변경에 의해 많은 시간이 소요되고 또한 끝까지 계산을 해야 정확한 결과를 얻을 수

있다. 따라서 Viterbi search 나 Viterbi beam search 를 사용하지 않고 어느 정도의 정확도를 낼 수 있는 것이 2.3 의 어절 품사 태깅이고, 정확도는 조금 떨어지더라도 어절별로 계산하는 것으로 바꾼다. 그러다 보니, 형태소 해석의 결과인 자료구조를 그대로 사용할 수 있어서 첫 어절 내에서는 full search 로 계산했다. 2.3 의 어절 품사 태깅 시스템의 주 오류가 어절 간에 일어나는 오류가 많았기 때문에 어절 간에는 이전 어절의 제일 좋은 값을 이용해서 현재 어절을 full search 했다.

그 결과 Viterbi search '를 위한 자료구조 변경 부분이 삭제되고, 첫 어절이 끝나면 바로 합성을 준비할 수 있게 되어서 실제로 느끼는 속도는 훨씬 빠르다. 즉, 첫 어절의 형태소 태깅 시간만큼의 속도가 늦어진다.

#### 4. 실험 및 평가

실험은 2 만 형태소 사전, 32 품사 셋을 이용하였으며, 아래의 결과는 테스트 셋 2,008 문장, 23,118 어절을 대상으로 open test 를 한 결과이다. 라인당 평균 속도는 테스트 문장을 전체 수행한 시간에 문장수를 나누었다.

[표 1] 성능 비교표

성능비교	오류율	라인당 평균 속도 향상율
원래 PC 태깅	19.81%	0%
음절단위변경	19.60%	0%
해석 변경	19.30%	20%
태깅 변경	20.53%	60%
태깅변경+규칙	20.33%	60%
해석변경+규칙	19.28	30

초기 시스템은 1.7G 2CPU 256M PC 에서 라인당 평균 0.005 초의 속도였으나 형태소 태깅까지 수정한 결과 0.002 초로 약 60%의 속도 개선을 보였으며, 오류율은 19.81%에서 20.33%로 약 1% 정도 정확률이 떨어졌다.

음절단위로 바꾸면서 정확률이 올라가고 해석 시스템을 바꾸면서 정확률이 올라감을 볼 수 있다. 이것은 규칙들과 미등록어 처리 부분에서 효과를 보았기 때문이라고 판단한다.

#### 5. 결론

본 논문에서는 음성 합성용 품사 태깅 시스템을 제안하였다. 제안된 모델은 1% 정도의 정확률이 떨어지더라도 60% 정도의 속도 개선을 보였음을 알 수 있다. 그리고, 첫 어절이 끝나면 바로 합성을 준비할 수 있게 되어서 실제로 느끼는 속도는 이보다 훨씬 빠름을 알 수 있다.

추후에는 본 연구에서 제안된 모델을 음성 합성기에 적용할 계획이며, 제안된 시스템에서의 정확도 향상을 위해서 연구하고자 한다.

#### Acknowledgement

본 연구는 정보통신부 출연 “대화체 연속음성인식 기술 개발” 과제의 일환으로 수행되었습니다.

#### 참고문헌

- [1] 김영택 외 공저, “자연언어처리”, 생능출판사
- [2] 양장모, 김민정, 권혁철, “언어 정보를 이용한 한국어 미등록어 추정”, 한국정보과학회 동 학술발표논문집, pp. 957-960, Vol.23, No.1, 1996