형태소의 모호성 축소를 위한 포섭조건의 자동 추론

김재훈†○ 장병규† 김길창† 서정연‡

† 대전 유성구 구성동 한국과학기술원, 전산학과
‡ 서울 마포구 신수동 서강대학교, 전자계산학과

Inducing Subsumption Conditions for Morphological Ambiguity Reduction

Jae-Hoon Kim†○ Byung-Gyu Jang† Gil Chang Kim† Jungyun Seo‡

† Department of Computer Science, KAIST, Taejon
‡ Department of Computer Science, Sogang University, Seoul

요 약

한국어는 교착어이기 때문에 형태소 해석은 자연언어 처리에서 매우 중요한 역할을 담당한다. 한국어 형태소 해석에서 주로 사용되는 정보는 두 형태소의 결합 가능 유무를 나타내는 접속정보이다. 이 접속정보는 단순한 품사정보에 의해서 표현되기 때문에 형태소 해석기의 과잉 해석의 원인이 된다. 이를 줄이기 위해 언어 지식의 일종인 포섭관계가 제안되었다[5]. 그러나, 포섭관계를 검사하기 위한 포섭조건들은 수작업에 의해서 작성되었다. 수작업으로 작성된 이들 조건들은 일관성 유지하기 어려울 뿐 아니라 완전한 조건들을 구하기도 어렵다. 따라서, 본 논문에서는 이와 같은 문제를 해소하기 위해서 품사 태깅된 말뭉치를 이용해서 포섭조건의 자동 추출방법을 제안한다.

1. 서론

한국어는 교착어(agglutinative language)이다. 여기서 말하는 교착(agglutination)은 단순한 어근(root)과 접미어(suffix)가 첨가되는 과정을 의미한다. 그러나, 더 일반적으로는 서로 다른 의미를 가진 두 개 이상의 구성요소(constituent element)들이 결합하여 복합어(compound word)나 파생어(derivative word)가 생성되는 과정을 말한다.

본 논문에서는 자연언어 처리 입장에서 이들을 다루고자 한다. 자연언어 처리에서는 교착과정을 통해서 생성된 단어 즉, 어절을 처리 대상의 기본 단위로 삼기에는 그 수가 많게 된다. 다행히도, 교착과정을 통해서 생성된 단어(어절)는 어떠한 규칙에 의해서 원래의 단어를 찾을 수 있다. 이와 같은 과정을 형태소 해석이라고 한다. 따라서, 형태소 해석은 언어학에서 의미하는 교착과정의 역과정으로 설명될 수 있으며, 교착어에서 형태소 해석은 자연언어 처리에서 매우 중요한 역할을 담당한다.

한국어의 형태소 해석 방법은 여러 가지가 제안되었다[1]. 이들 방법의 대부분에서 이용되는 규칙은 형태소 배열규칙(morphotactics)이다. 형태소 배열규칙은 형태소들의 배열에 관한 제약(ordering restriction)이다[14]. 한국어 형태소 해석기의 대부분은 형태소 배열규칙으로서 접속정보(connectivity information)를 사용한다. 접속정보는 두 형태소의 결합 유무를 표현한 것으로 형태소 품사(morpheme category)에 대한 이진관계(binary relation)이다. 접속정보는 한국어의 모든 형태소 배열규칙을 표현할 수는 없지만, 간단하면서도 거의 대부분의 형태소 배열규칙을 표현할 수 있기 때문에 많은

형태소 해석기에서 이용되는 정보(knowledge)이다. 이와 같은 접속정보는 필요 이상의 많은 해석을 생성하며, 이를 형태소의 **과잉해석**(over-analysis)이라고 한다. 예를 들어 명사와 명사가 결합할 수 있다는 접속정보가 있다고 가정하면, 한국어 단어 '소나무'에 대한 형태소 해석결과는 (1)과 같다[1].

(1) ㄱ. "소나무/명사"
ㄴ. "소/명사 + 나무/명사" ㄷ. "소/명사 + 나/명사 + 무/명사"

이와 같은 두 해석결과는 형태론적으로는 물론이고, 구문적으로도 전혀 모순이 없는 결과이다. 그러나, 의미론적인 입장에서 해석결과 (1ㄱ)과 (1ㄴ)을 살펴보면, (1ㄴ)은 여러 가지의 의미로 해석될 수 있으며, 그 중에 하나는 (1ㄱ)의 의미해석을 포함하게 될 것이다[2]. 이와 같은 의미에서 (1ㄱ)은 (1ㄴ)에 비해 훨씬 더 구체적인 해석이라고 할 수 있다. 본 논문에서는 이와 같이 어느 하나의 해석이 다른 해석의 구체적인 해석이 될 수 있는 조건을 찾아서, 더 구체적인 해석을 올바른 형태소 해석의 결과로 삼고자 한다. 이와 같은 두 형태소 해석들 사이의 관계를 포섭관계(subsumption relation)라고 하며, 이 포섭관계는 단어의 형성과정을 모형화한 언어지식의 일종이다.

포섭관계는 포섭조건에 의해서 검사되어지며, 이 포섭조건은 일일이 손으로 작성되었다[5]. 그러나, 손으로 작성된 조건들은 일관성 유지에 많은 문제를 가지게 된다. 본 논문에서는 이와 같은 문제를 해결하기 위해서 품사태깅된 말뭉치로부터 포섭조건을 자동으로 추론하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 포섭관계에 대해서 기술하고, 3장에서는 포섭조건의 자동 추론 방법에 대해서 논한다. 포섭관계를 이용한 형태소 축소 방법에 대해서는 4장에서 기술하고, 5장에서는 예비 실험 및 평가에 대해서 기술하고, 6장에서 결론을 맺고자 한다.

2. 형태소 해석의 포섭관계

한국어 단어의 많은 경우는 단어의 형성 과정을 통해서 만들어진 복합어들이다[6, 8]. 복합어의 일부는 형태소 해석의 처리대상에 포함되고, 또 다른 일부는 하나의 단어(단일어, simple word)로 간주된다. 복합어를 단일어로 간주할 경우에도 형태소 해석에서는 복합어에 포함된 각각의 어근들로 해석되는 경우가 발생한다. 본 논문에서는 이와 같이 두 형태소의 해석결과들 사이의 복합어 관계를 찾아서, 어근들의 결합으로 이루어진 해석을 최종적인 형태소 해석결과로부터 제거하고자 한다. 복합어에 대한 형태소 해석결과를 **구체적인 형태소 해석결과**라고 하고, 어근들의 결합에 의한 해석결과를 **일반적인 형태소 해석결과**라고 한다.

2.1 포섭관계의 정의 및 성질

정의 1 어절 W에 대한 형태소 해석결과를 $A = \{A_1, A_2, \ldots A_n\}$라고 하자. 이 때, 해석결과 A_i가 해석결과 A_j보다 더 일반적인*(more general)* 해석이라면, A_i가 A_j를 **포섭한다***(subsume)*라고 하며, $A_i \sqsubseteq A_j$로 표기한다. 여기서, A_i를 **포섭하는 해석***(subsuming analysis)*이라고 하고 A_j를 **포섭되는 해석***(subsumed analysis)*이라고 한다. □

예를 들면, 어절 '**날아오다**'에 대한 형태소 해석결과는 (2)와 같다[3]. (2ㄱ)은 합성법에 의해서 생성된 복합어이다[6]. 정의에 따라서, (2ㄴ)이 (2ㄱ)을 포섭한다고 말할 수 있다. 즉, (2ㄴ)이 (2ㄱ)보다 더 일반적인 형태소 해석결과이다.

(2) ㄱ. "날아오/pv + 다/ef"
ㄴ. "날/pv + 아/ecx + 오/px + 다/ef"

정의 2 어절 W에 대한 형태소 해석결과를 $A = \{A_1, A_2, \ldots, A_n\}$라고 하자. 형태소 해석결과 A에 관한 **포섭관계***(subsumption relation)* $S = (A, \sqsubseteq)$는 식 *(1)*과 같이 정의된다.

$$S = \{(A_i, A_j) | A_i \sqsubseteq A_j\} \qquad \square(1)$$

성질 1 포섭관계는 부분순서 관계*(partial ordered relation)*이다.

예를 들면, 어절 '소나무'에 대한 형태소 해석결과 (1)로부터 포섭관계 S_1(표 1)을 얻을 수 있다[4]. 일반적으로 부분순서 관계는 Hasse 다이그램으로 표현될 수 있으며, 포섭관계 S_1의 Hasse 다이그램은 그림 1과 같다. Hasse 다이그램에서는 이행관계(transitive relation)는 나타나지 않는다.

정의 3 어절 W에 대한 형태소 해석결과 A에 대한 포섭관계 $(A, \sqsubseteq)$가 부분순서 관계라고 하면, 어떤 해석 A_j

[1] 참고로 "소나무/명사"는 형태소 '소나무'의 품사가 명사임을 의미한다. 또, "소/명사 + 나무/명사"에서 +는 형태소의 부분해석이 연이어서 나타남을 표현하는 연산자(concatenator)이다. 본 논문에서 형태소의 해석결과는 선형구조(linear structure)를 갖는 것으로 가정한다.

[2] 단어의 형성 과정에서 '소나무'는 '숲'과 '나무'가 결합하여 하나의 단어로 굳어진 복합어이다.

[3] pv는 동사, ecx는 보조적 연결어미, px는 보조용언, ef는 어말어미를 나타내는 품사태그들이다. 이 품사태그에 대한 상세한 설명은 [3]를 참조하기 바란다.

[4] nc는 보통명사를 나타낸다.

표 1: 어절 '소나무'에 대한 형태소 해석결과 (1)로부터 포섭관계 S_1

포섭하는 해석결과	포섭되는 해석결과
(소/nc + 나무/nc,	소나무/nc)
(소/nc + 나/nc + 무/nc,	소나무/nc)
(소/nc + 나/nc + 무/nc,	소/nc + 나무/nc)

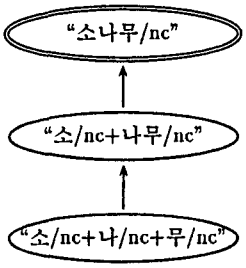

그림 1: 어절 '소나무'에 대한 포섭관계 S_1의 Hasse 다이그램

도 포섭하지 않는 해석 A_i를 **최대해석**(*maximal analysis*)이라고 한다. □

그림 1에서 최대해석은 "소나무/nc"이고, 복선으로 표현되었다.

성질 2 어절 W에 대한 형태소 해석결과 A에 대한 포섭관계 $(A, \sqsubseteq)$가 부분순서 관계일 때, W에 대한 형태소 해석 결과는 최대해석들만 있으면 충분하다. □

최대해석을 포섭하는 모든 해석들은 최대해석보다 더 일반적인 해석(more general analysis)이기 때문에 최종적으로는 최대해석만 있으면 된다.

2.2 포섭 조건

어떤 두 형태소 해석 결과가 포섭관계에 포함되는 지를 검사하기 위해서는 포섭관계를 이룰 수 있는 **포섭조건**(subsumption condition)이 필요하다. 포섭조건은 다음과 같은 것들에 의해서 표현된다.

1. 어절 W의 형태소 해석 결과, $\{A_1, A_2, \ldots, A_n\}$
2. 문장 상에서 어절 W의 위치, $loc(W)$
3. 예외적인 해석, $exception(A_i)$

표 2에서는 이들 정보를 이용한 포섭관계에 대한 예를 보여 주고 있다. 표 2에서 T_i와 T_j는 각각 포섭하는 해석 A_i와 포섭되는 해석 A_j의 품사열이다. 대부분의 포섭조건은 품사열에 의해서 구성된다. 그러나, 다섯 번째 조건의 경우에는 주어진 어절의 위치가 추가되어 있다. 구체적인 예를 보면, 형태소 해석 "따라서/ajs"는 문장의 시작위치에 있을 경우에만 형태소 해석 "따르/pa+아서/ecs"에 비해 더 구체적인 해석이 될 수 있다[5].

시작상태 :	{ 0 }
종결상태 :	{ 2 }
시작조건 :	{ loc = {} & execption = {} }
생성규칙 :	0 → ?/nc 1
	0 → ?/nq 1
	1 → ?/nc 2
	1 → ?/nq 2
	2 → ?/nc 2
	2 → ?/nq 2

그림 2: 고유명사(nq)의 포섭조건을 표현한 오토마타

포섭조건은 오토마타에 의해서 표현될 수 있다. 즉, 정규문법(regular grammar)에 의해서 표현될 수 있다. 포섭조건을 표현하는 정규표현의 단말기호(terminal symbol)는 *morpheme/pos*와 같은 형태로 표현된다. 여기서, *morpheme*은 형태소이거나 '?'(어떤 형태소와도 정합이 가능함을 의미함)이고, *pos*는 품사 중의 하나이다. 시작상태(starting state)에는 $loc(W)$, $exception(A_i)$이 부가되며, 이를 시작조건이라고 한다. 포섭조건의 검사는 선형 시간 내에서 이루어질 수 있다. 예를 들면, 고유명사(nq)에 대한 오토마타는 그림 2와 같으며, 시작조건으로 $loc(W)$, $exception(A_i)$에 대한 제약조건은 없다.

3. 태깅된 말뭉치를 이용한 포섭조건의 자동추론

2.2절에서 설명했듯이 포섭조건은 변형된 정규문법에 의해서 표현 가능하다. 따라서, 알려지지 않은 정규문법에 의해서 생성된 예(examples)를 통한 정규문법의 추론이 가능하다[11, 12]. 포섭조건의 개략적인 추론과정은 그림 3과 같으며, 각 모듈들을 아래의 절에서 차례로 설명한다.

3.1 품사태깅된 말뭉치로부터 포섭패턴의 추출

주어진 어절에 대한 형태소 해석결과 $A = \{A_1, \ldots, A_n\}$와 품사태깅된 말뭉치로부터 가지고 온 올바른 형태소 해석결과 B를 비교해서, 하나의 예로서의 포섭패턴을 추출한다(그림 4). 이 알고리즘은 두 해석결과의 상대적인 차($B \ominus A_i$)를 이용한다. 예를 들면, (3)과 같은 두 결과의 상대적인 차($(3\urcorner) \ominus (3\llcorner)$)는 (4)와 같다.

[5] ajs는 문장접속부사, pa는 형용사, ecs는 종속적 연결어미를 나타낸다.

표 2: 포섭관계의 예

포섭하는 해석(A_i)	포섭되는 해석(A_j)	포섭조건
(매일/nc, 우유/nc)	(매일우유/nq)	$T_i = (nc, nc)$ & $T_j = (nq)$
(서울/nq, 우유/nc)	(서울우유/nq)	$T_i = (nq, nc)$ & $T_j = (nq)$
(호텔/nc, 신라/nq)	(호텔신라/nq)	$T_i = (nc, nq)$ & $T_j = (nq)$
(소/nc, 나무/nc)	(소나무/nc)	$T_i = (nc, nc)$ & $T_j = (nc)$
(따르/pv, 아서/ecs)	(따라서/ajs)	$T_i = (pv, ecs)$ & $T_j = (ajs)$ & $loc(W) = 0$
(날/pv, 아/ecx, 오/px)	(날아오/pv)	$T_i = (pv, ecx, px)$ & $T_j = (pv)$
(즐겁/pa, 어/ecx, 지/px)	(즐거워지/pv)	$T_i = (pa, ecx, px)$ & $T_j = (pv)$
(동시/nc, 에/jca)	(동시에/a)	$T_i = (nc, jca)$ & $T_j = (a)$
(이/npd, 대로/jca)	(이대로/ad)	$T_i = (npd, jca)$ & $T_j = (ad)$

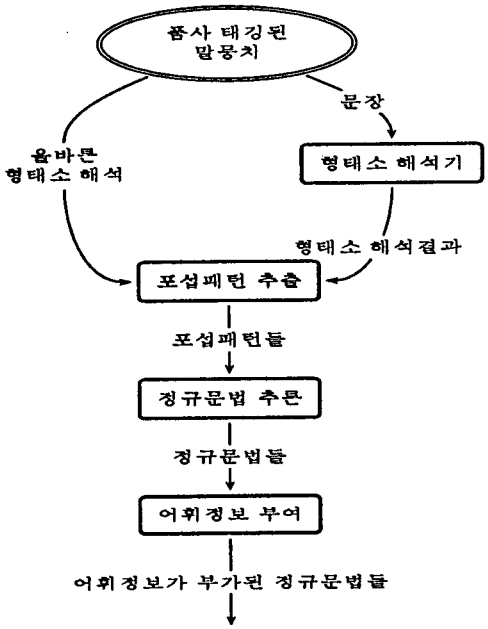

그림 3: 포섭조건의 개략적인 추론과정

(3) ㄱ. '중국/nq + 의/jcm"
ㄴ. '중/nc + 국/nc + 의/jcm"

(4) ㄱ. '중국/nq"
ㄴ. '중/nc + 국/nc"

상대적인 차는 해석결과가 선형구조를 가지고 있기 때문에 전후방향으로 각각 가장 길게 일치하는 부분을 제외함으로서 구할 수 있다. 왜냐하면, 포섭조건은 바로 차이를 보이는 이 부분에 의해서 결정되기 때문이다. 그 차에 단지 하나의 올바른 부분 해석결과만 있을 경우(그림 4의 알고리즘에서 *distance*가 0)에 하나의 복합어가 올바른 형태소 해석결과에 나타났음을 의미한다. 이와 같은 방법으로 말뭉치 내에 포함된 모든 어절에 대해서 포섭패턴을 추출한다.

그림 4의 알고리즘에서 head_admissible(X, Y)와 tail_admissible(X, Y)는 각각 포섭패턴의 시작과 끝에서 X가 포섭되기 위해 Y가 충분한 지를 검사하는 함수이고, pos(X)는 X의 품사를 구하는 함수이다. 그리고, suffix는 접미어 무리를 총칭하는 상수이다.

B가 (1ㄱ)이고, A_i가 (1ㄴ)이라고 할 때, 이 알고리즘에 의해서 추출된 포섭패턴은 "소나무/nc ← 소/nc + 나무/nc"이다. 이 때, 포섭조건의 작성에 필요한 자질들 중 하나인 $loc(W)$도 함께 추출된다.

3.2 정규문법 추론

앞절에서 추출된 포섭패턴으로부터 형태소를 나타내는 부분은 모두 변수에 의해서 일반화시킨다. 따라서, 포섭패턴 "소나무/nc ← 소/nc + 나무/nc"은 "?/nc ← ?/nc + ?/nc"로 일발화시켜서(generalize) k-tail을 기반한 정규문법 추론 알고리즘[12, 13]을 이용해서 품사태그를 단말기호로하는 정규문법을 구한다. 본 논문에서는 하나의 포섭되는 해석결과마다 하나의 정규문법을 구했으며, k를 2로하여 정규문법을 구했다.

3.3 추론된 정규문법에 어휘정보 부가

3.2절에서 포섭패턴에 나타난 형태소는 일반화되어 모두 변수로 처리되었다. 즉, 품사들의 정보만 이용해서 정규문법을 얻었다. 본 절에서는 형태소를 정규문법에 포함시키는 과정으로 정규문법의 특수화 과정이다. 정규문법의 특수화에서는 엔트로피(entropy) 개념을 도입한다.

먼저 정규문법에 속하는 하나의 규칙 p에 대응하는 모든 형태소들을 빈도순으로 정렬한다. 정렬된 형태소들$(M_i, i = 1, \ldots, P)$의 빈도수를 $n(M_1), \ldots n(M_P)(n(M_i) \geq n(M_{i+1}))$이라고 할 때, 규칙 p에서 이들 형태소가 발생할 확률 $p(M_i)$는 $\frac{n(M_i)}{N}$으로 구한다. 여기서 N은 규칙 p에 속하는 모든 형태소 수$(N = \sum_{i=1}^{P} n(M_i))$이다. 이들을 이용해서 규칙 p에서 형태소 M에 대한 엔트로피 $I_p(M)$가 θ(본 논문에서

Input . $A_i = A_{i,1}, A_{i,2} \ldots A_{i,M}$ and $B = B_1, B_2, \ldots B_L$
Output . NULL or $B_k \ldots B_l \leftarrow A_{i,j} \ldots A_{i,m}$
  where $1 \leq k \leq l \leq L$ and $1 \leq j \leq m \leq M$
Method .
  $x = y = 1$;
  while ($A_{i,x} == B_y$) do % forward matching
    if ($x \geq M$ or $y \geq L$) break ;
    $x = x + 1$; $y = y + 1$;
  enddo
  $forwardX = x$; $forwardY = y$;
  $x = M$; $y = L$;
  while ($A_{i,x} == B_y$) do % backward matching
    if ($x \leq forwardX$ or $y \leq forwardY$) break ;
    $x = x - 1$; $y = y - 1$;
  enddo
  $backwardX = x$; $backwardY = y$;
  $distance = backwardY - forwardY$;
  if ($distance == 0$) then
    $B_k = B_l = B_{forwardY}$;
    $A_{i,j} \ldots A_{i,m} = A_{i,forwardX} \ldots A_{i,backwardX}$;
    if (not head_admissible($B_k, A_{i,j}$)) return NULL ;
    if (not tail_admissible($B_l, A_{i,m}$)) return NULL ;
    return $B_k \ldots B_l \leftarrow A_{i,j} \ldots A_{i,m}$;
  endif
  if ($distance == 1$ and pos($B_{forwardY}$) == suffix) then
    $B_k = B_{forwardY}$; $B_l = B_{backwardY}$;
    $A_{i,j} \ldots A_{im} = A_{i,forwardX} \ldots A_{i,backwardX}$;
    if (not head_admissible($B_k, A_{i,j}$)) return NULL ;
    if (not tail_admissible($B_l, A_{i,m}$)) return NULL ;
    return $B_k \ldots B_l \leftarrow A_{i,j} \ldots A_{i,m}$;
  endif
  return NULL ;

그림 4: 품사태깅된 말뭉치로부터 포섭패턴의 추출 알고리즘

θ는 0.2임) 이하일 경우에는 어휘정보를 규칙 p에 포함시킨다.

$$I_p(M) = -\sum_{M_i \in M} p(M_i) \log_2 p(M_i) \qquad (2)$$

어휘정보가 포함되어야 할 경우에 모든 형태소를 정규문법에 포함시키는 것이 아니라, 정규문법에 포함되어야 하는 형태소의 무리와 그렇지 않은 무리로 나눈다. 본 논문에서 이들 두 무리의 경계를 찾기 위해서 엔트로피의 변화량 $\Delta I_p(i, M)$(식 (3))[10]을 이용하고, 이 변화량을 최대로 하는 i를 찾아서 이를 중심으로 두 무리로 나눈다.

$$\Delta I_p(i, M) = I_p(M) - p_L I_p(M_L) - p_R I_p(M_R) \qquad (3)$$

3.4 시작상태의 조건 부여

시작상태에 $loc(W)$와 $exception(A_i)$의 정보가 부여되어야 한다. $loc(W)$의 포함 유무는 3.3절에서 설명한 엔트로피의 개념을 이용해서 구할 수 있다. 그러나, $exception(A_i)$의 경우에는 이전에 모든 방법에 의해서

Input . 형태소 해석결과 $A[1..n]$
Output . 축소된 형태소 해석결과 $R[1..m]$
Method .

```
for i = 1 to n do reduced[i] = false ; enddo

for j = 1 to n do
    for i = 1 to n do
        if (j == i) continue ;
        if (reduced[i] == true ) continue ;
        if (A[i] ⊑ A[j]) then
            reduced[i] = true ;
        endif
    enddo
enddo

j = 0;
for i = 1 to n do
    if (reduced[i] == true ) continue ;
    R[j] = A[i]; j = j + 1;
enddo
```

그림 5: 형태소 모호성을 축소하기 위한 알고리즘

처리 안되는 해석이 발생되었을 경우에 예외규칙을 포함시킨다. 즉, 어휘정보가 부여된 정규문법에 의해서 올바른 해석이 제거될 때, $exception(A_i)$의 정보에 의해서 이를 구제할 수 있도록 하였다.

4. 포섭관계를 이용한 형태소의 모호성 축소

그림 5에서는 포섭관계를 이용한 형태소의 모호성 축소 알고리즘을 가상코드로 표현하고 있다. 이 알고리즘은 형태소 해석결과에 대한 포섭관계를 찾고, 그 포섭관계로부터 최대해석을 찾기 위한 알고리즘이다. 그림 5의 알고리즘에서 포섭관계의 검사는 2.2절에서 설명한 포섭조건을 검사함으로써 이루어진다. 포섭조건을 효율적으로 검사하기 위해서, 먼저 두 해석의 상대적인 차를 구한다(3.1절을 참조할 것). 얻어진 상대적인 차에 대해서 포섭조건을 검사한 후, 그 조건을 만족할 경우, 포섭관계가 성립하는 것으로 간주한다. 예를 들면, 어절 '중국의'를 형태소 해석하면, (3)과 같다. 이들의 상대적인 차는 (4)와 같으며, (4)에 대해서 포섭조건을 검사하면, (4ㄴ)이 (4ㄱ)을 포섭하므로 (4ㄴ)이 최종적인 해석에서 제거될 수 있다.

5. 예비실험

5.1 실험환경

실험을 위해서 사용된 품사는 총 53개이다[3]. 이들 품사를 이용해서 형태소 해석기의 배열규칙 정보인 접속정보를 구성하였다. 접속정보는 품사 태깅된 한국어 말뭉치[4]로부터 자동적으로 만들어졌다. 한국어 형태소 해석기[2]는 변형된 챠트 파싱에 의해서 구현되었다[6].

학습과 시험을 위해서 각각 36,163개와 13,860개의 어절로 구성된 학습 및 시험 말뭉치를 사용하였다.

5.2 자동추론된 포섭조건

학습 말뭉치에 의해서 추론된 정규문법의 수는 총 27개이고, 그 중에는 포섭되는 해석의 품사가 하나일 경우가 25개로 대부분을 차지하고 있다. 이들이 가지는 총 정규규칙의 수는 773개로 하나의 정규문법 당 평균 28.63개의 정규규칙을 가지게 되었다. 이들 정규규칙들 중에는 어휘정보를 포함하고 있는 경우가 301개로 전체의 38.94%를 차지하고 있다. 이 결과는 3.3절에서 설명한 θ를 조절함으로서 그 비율을 조절될 수 있다. 극단적으로 θ를 0일 경우에는 어휘정보를 전혀 사용하지 않는 경우이다.

5.3 성능 평가

시험 말뭉치의 총 형태소 수는 105,735개이다. 이 결과를 4.장에서 설명한 형태소의 모호성 축소 알고리즘을 실행했을 때, 형태소 해석의 총 수는 69,773개였다. 따라서 전체 해석수의 34.01%를 축소할 수 있었다.

6. 결론

본 논문은 형태소 과잉해석으로 발생되는 형태소 해석의 모호성을 축소하는 방법에 대해서 기술하였다. 과잉해석은 간단한 형태소 배열규칙과 불규칙 현상 처리에 의해서 발생되는데, 이를 줄이기 위해서 단어의 형성과정에서 이용되는 언어적 지식을 사용하였다. 이를 효과적으로 처리하기 위해서 포섭관계를 정의하였으며, 이 포섭관계를 이용해서 형태소의 모호성을 축소할 수 있었다. 또한 포섭관계를 검사하는 포섭조건을 품사태깅된 말뭉치로부터 자동으로 추론하였다. 자동 추론된 포섭조건을 이용해서 약 34.01%의 오류를 축소할 수 있었다. 그러나, 이 결과는 여러가지의 매개변수를 잘 조절함으로써 더 좋은 결과를 얻을 수 있을 것이다. 또한, 이 결과는 형태소 해석의 모호성을 제거하는 품사 태깅이나, 그 밖의 구문해석 등 자연언어 처리 전반에 응용될 수 있을 것이다. 특히, 언어지식의 부족에 의해서 발생되는 품사태깅의 오류를 줄이는데 큰 도움을 줄 것으로 기대된다.

[6] 이 한국어 형태소 해석기에 대한 여러 가지 자료는 "http://hanul.kaist.ac.kr/~bgjang/MoA/"에서 구할 수 있다.

참고문헌

[1] 김재훈, **한국어 형태소 처리에 관한 고찰**, 한국과학기술원, 전산학과, 컴퓨터 시스템 연구실 내부 메모, 1992.

[2] 김재훈, 서정연, **실용적인 한국어 형태소 해석**, 한국과학기술원, 전산학과, 컴퓨터 시스템 연구실 내부 메모, 1994.

[3] 김재훈, 서정연, **자연언어 처리를 위한 한국어 품사 태그**, 한국과학기술원, 인공지능연구센터, CAIR-TR-94-55, 1994.

[4] 김재훈, **품사 태깅된 한국어 말뭉치(kHQ)의 작성요령**, 한국과학기술원, 전산학과, 컴퓨터 시스템 연구실 내부 메모, 1995(작성중).

[5] 김재훈, 장병규, 김길창, 서정연, "한국어 형태소 해석의 모호성 축소," **제1회 지능기술 공동학술대회**, 서울, 과학기술진흥센터, pp 116-121, 1995.

[6] 이주행, 현대국어문법론, 대한교과서주식회사, 1993.

[7] 임희석, 윤보현, 임해창, "배제 정보를 이용한 효율적인 한국어 형태소 분석기," **한국정보과학회 논문지**, 제22권, 제6호, pp. 987-964, 1995.

[8] 정도환, **국어 복합어의 의미 연구**, 서광 학술 자료사, 1993.

[9] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A Maximum Likehood Approach to Continuous Speech Recognition," *IEEE Tr. on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 2, pp. 179-190, 1983.

[10] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth Inc., 1984.

[11] R. C. Gonzalez and M. G. Thomason *Syntactic Pattern Recognition: An Introduction*, Addison-Wesley Publishing Company, Inc., 1978.

[12] K. S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice-Hall, Inc., 1982.

[13] L. Miclet, "Regular Inference with Tail-Clustering Method," *IEEE Tr. on Systems, Man, Cybernetics*, vol. SMC-10, no. 11, pp. 737-743,1980.

[14] R. Sproat, *Morphology and Computation*, The MIT Press, 1992.