(제 10회 한글 및 한국어 정보처리 학술대회)

Beam Search 알고리즘을 이용한 효율적인 한국어 의존 구조 분석[1)]

°김학수
서강대학교 전자계산학과
서울시 마포구 신수동 1
우: 121-742
hskim@nlpzodiac.sogang.ac.kr

서정연
서강대학교 전자계산학과
서울시 마포구 신수동 1
우: 121-742
seojy@ccs.sogang.ac.kr

Efficient Analysis of Korean Dependency Structures Using Beam Search Algorithms

Harksoo Kim
Department of Computer Science
Sogang University

Jungyun Seo
Department of Computer Science
Sogang University

요 약

구문분석(syntactic analysis)은 형태소 분석된 결과를 입력으로 받아 구문단위간의 관계를 결정해 주는 자연어 처리의 한 과정이다. 그러나 구문분석된 결과는 많은 중의성(ambiguity)을 갖게 되며, 이러한 중의성은 이후의 자연어 처리 수행과정에서 많은 복잡성(complexity)를 유발하게 된다. 지금까지 이러한 문제를 해결하기 위한 여러 가지 연구들이 있었으며, 그 중 하나가 대량의 데이터로부터 추출된 통계치를 이용한 방법이다. 그러나, 생성된 모든 구문 트리(parse tree)에 통계치를 부여하고, 그것들을 순위화하는 것은 굉장히 시간 소모적인 일(time-consuming job)이다. 그러므로, 생성 가능한 트리의 수를 효과적으로 줄이는 방법이 필요하다. 본 논문에서는 이러한 문제를 해결하기 위해 개선된 beam search 알고리즘을 제안하고, 기존의 방법과 비교한다. 본 논문에서 제안된 beam search 알고리즘을 사용한 구문분석기는 beam search를 사용하지 않은 구문분석기가 생성하는 트리 수의 1/3정도만으로도 같은 구문 구조 정확률을 보였다.

1. 서 론

언어를 모형화하는 것은 언어에 존재하는 다양한 성질들을 발견하고 언어를 설명하려는 노력이다. 모형화의 대상인 자연언어는 오랜 기간 동안 다양하게 의사소통의 편의에 따라 변화하고 발전하여 매우 복잡한 양상을 띠며, 그 복잡한 양상 때문에 자연언어를 분석할 때에는 두 가지의 어려움이 발생한다. 기하급수적으로 폭증하는 애매성(ambiguity)의 문제와, 프로그래밍 언어와 달리 그것의 경계가 분명하지 않다는 개방성(open ended)의 문제가 그것이다[4]. 이러한 자연언어의 애매성과 개방성은 자연어처리의 여러 단계에서 나타나며, 구문분석 단계에서도 물론 나타난다. 이러한 문제를 해결하기 위해서 많은 통계적 모델들이 제안되었다. 그러나, 생성된 모든 구문 트리에 확률 모델을 적용하고, 이를 순위화하는 것은 매우 시간 소모적인 일이다. 그러므로, 효율적으로 트리의 수를 줄이는 방법이 연구되었으며, 그 중 하나가 beam search algorithm이다.

본 논문에서는 통계 모델을 구문분석에 적용하는데 있어서 문제점으로 지목되어온 파싱타임(parsing time)의 문제를 해결하면서도, 구문구조 정확률을 유지하는 효율적인 beam search 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 beam search 알고리즘의 효율성을 비교하기 위해 사용한 한국어 통계적 구문 분석기[2)]를 소개한다. 그리고, 3장에서는 기존의 beam search 알고리즘을 개선하기 위한 방법들을 설명한다. 4장에서는 제안된 방법의 효율성을 기존의 방법과 비교, 실험한다. 마지막으로 5장에서 결론을 내리고, 향후과제를 제시한다.

2. 어휘정보를 이용한 한국어 통계적 구문분석기

1) 본 논문은 과학기술부 소프트과학 특정 기초 연구, "대화 이해 및 생성을 위한 대화 인지 모형 연구"의 일부분임.

2) "어휘 의존 정보에 기반한 한국어 통계적 구문 분석기"로 오른쪽 우선 차트 파싱 방법을 사용한다[1,2].

2.1 확률 모델

어휘정보를 이용한 통계적 구문분석기는 가능한 모든 구문 트리의 신뢰성을 [식1]과 같이 추정하며, 가능한 모든 구문 트리 중에 최대 확률값을 갖는 구문 트리를 결정한다.

$$T_{best} = \arg\max_{T} P(T|S) \quad (1)$$

이때, 문장을 어절과 그들간의 의존 관계로 구성되었다는 가정에 의해서 [식2]와 같이 구문 트리의 확률값을 추정할 수 있으며, 우변의 두 확률값을 각각 인접어절 확률과 의존 확률이라 한다.

$$P(T|S) = P(B, D|S) = P(B|S) \times P(D|S, B) \quad (2)$$

따라서, 학습 자료로부터 추출된 인접어절 확률과 의존 확률을 이용하여 각 구문 트리의 확률값을 추정함으로서 가능한 구문 트리들의 순위화가 가능하다.

2.2 의존 확률

[식2]에서 문장을 이루는 두 어절간의 지배/의존 관계를 나타내는 $P(D|S, B)$를 의존 확률이라 한다. 이때, j번째 어절과 h_j번째 어절간에 R_j이란 의존 관계가 존재함을 [식3]과 같이 정의하면, j번째 어절의 의존 확률을 [식4]와 같이 정의할 수 있다.

$$AF(j) = (h_j, R_j) \quad (3)$$

$$P(AF(j) = (h_j, R_j)|S, B) \quad (4)$$

따라서, m개의 어절로 구성된 문장 S의 의존 확률 $P(D|S, B)$는 [식5]와 같이 정의할 수 있다.

$$P(D|S, B) = \prod_{j=1}^{m} P(AF(j)|S, B) \quad (5)$$

$$\approx \frac{\sum_{k=-1}^{1} F(R_j|\langle w_j, t_j\rangle, \langle w_{hj}, t_{hj}\rangle, \triangle_{j, hj+k})}{\sum_{k=1..m, k\neq j, p\in P} F(p|\langle w_j, t_j\rangle, \langle w_k, t_k\rangle)} \quad (6)$$

$$F(R_j|\langle w_j, t_j\rangle, \langle w_{hj}, t_{hj}\rangle, \triangle_{j, hj+k}) \approx \frac{C(R_{j,\langle w,\rangle}, t_j\rangle, \langle w_{hj}, t_{hj}\rangle, \triangle_{j, hj+k})}{C(\langle w_j, t_j\rangle, \langle w_{hj}, t_{hj}\rangle, \triangle_{j, hj+k})} \quad (7)$$

[식5]의 확률값은 학습 자료로부터 얻어낸 두 어절 각각의 단어/태그(tag) 쌍을 이용하여 [식6]과 같이 추정하게 되며, 문자는 [식7]과 같이 추정한다. 이때, 두 어절간의 거리 정보 [식8]을 이용하여 두 어절간의 의존 관계에 대한 가중치를 부여받게 된다.

$$\triangle_{j, hj} = h_j - j \quad (8)$$

[식6]에서 나타나는 희소 데이터 문제(sparse data problem)를 해결하기 위해서는 [부록]에 첨가한 삭제 보간(deleted interpolation)을 사용한다.

2.3 인접어절 확률

인접어절 확률이란 문장내의 인접하는 두 어절간의 관계를 나타내는 확률로서 [식9]와 같이 추정한다.

$$P(B|S) = \prod_{i=2..n} P(G_i|w_{i-1}, t_{i-1}, w_i, t_i, c_i) \quad (9)$$

[식9]의 의미는 두 어절사이에 존재할 수 있는 관계를 [표1]과 같이 정의하고, 학습자료로부터 추출된 인접한 단어와 품사 태그 쌍이 특정 관계로 나타날 수 있는 확률값을 구문분석에 반영하는 것이다.

< 표 1 : 어절간의 관계 >

S(tart)	C(ontinue)	E(nd)	B(etween)
괄호 시작	괄호 내부	괄호 끝	괄호 사이

3. 개선된 beam search 알고리즘

3.1 Collins의 beam search 알고리즘

[10]에서 정의한 beam search 알고리즘은, 현재 어절까지 구문 분석된 부분 트리가 가지는 확률값들 중에 가장 큰 값을 P_h라고 하고 beam의 크기를 β라고 했을 때, $\frac{P_h}{\beta}$보다 작은 확률값을 가지는 부분 트리는 잘라버리는 것이다. 그러나, 이렇게 지역적인 확률의 최대값과 고정된 beam의 크기를 이용한 방법은 몇 가지 문제점을 가지고 있다. 첫째, 부분 트리들의 존속 여부를 단지 현재 어절까지의 지역적(local) 확률값만을 사용함으로써 문장 전체로 봤을 때 높은 확률을 가지는 구문 트리가 미리 잘려 나가는 탐색 오류(search error)를 많이 포함한다.

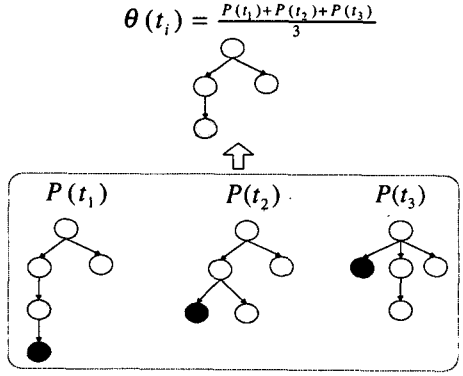
둘째, 고정된 beam의 크기를 사용함으로써 생성된 트리의 수에 효율적으로 대처하지 못한다. 즉, 생성된 트리의 수가 적을 때는 조금만 잘라내고, 많을 때는 많이 잘라내는 방법이 보다 효율적이지만 [10]에서의 방법은 그렇지 못하다. 셋째, 임계값(threshold value) 계산에 단순히 부분 트리의 최대값만을 반영함으로써 확률 분포가 가지는 평균이나 표준 편차와 같은 여러 가지 특성을 반영하지 못한다.

3.2 전역적 확률(global probability)의 계산

3.1에서 언급했듯이 기존의 beam search 알고리즘은 단지 현재 어절까지의 지역적인 확률값만을 이용하여 부분 트리의 존속 여부를 결정하기 때문에 전체적인 구문 트리가 가지는 특성을 효과적으로 반영하지 못한다. 본 논문에서는 이러한 문제점을 보완하기 위해 [식10]과 같이 부분 트리 확률 θ를 정의하여 사용한다.[3)]

$$\theta(t_i)=\frac{\sum_{j=1}^{n} P(t_j)}{n} \qquad (10)$$

[식10]에서 t_i는 현재 어절이 생성한 부분 트리이고, t_j는 t_i로 인해 생성될 수 있는 부분 트리들이다. 즉, [식10]의 의미는 현재 어절의 부분 트리를 포함하는 다음 어절의 부분 트리들이 가지는 확률값의 평균을 현재 어절이 생성한 부분 트리의 확률로 이용하는 것이다. [그림1]은 [식10]을 도식화하여 나타낸 것이다.



< 그림 1 : global beam search >

3) [식9]에서 계산되는 확률은 단지 beam search알고리즘에서 임계값과의 비교를 위한 것이다.

3.3 표준 편차를 반영한 임계값의 계산

부분 트리의 표준 편차 또한 임계값을 계산하는데 중요한 정보로 사용될 수 있다. 그러나, 기존의 방법은 임계값 계산에 부분 트리의 최대값만을 반영하여 확률 분포가 가지는 특성을 살리지 못하였다. 본 논문에서는 [식11]과 같이 임계값 ε을 정의하여 사용한다. P_{min}은 부분 트리의 최소값이고, σ는 부분 트리들의 표준편차이다.

$$\varepsilon = P_{min} + \sigma \qquad (11)$$

[식11]과 같이 임계값을 계산한 이유는 조밀한 분포를 가지는 가진 확률값들은 보다 작은 임계값을 적용하고, 그렇지 않은 것은 큰 임계값을 적용하기 위한 것이다.

[식10]과 [식11]에 의해서 현재 부분 트리의 $\theta(t_j)$가 ε보다 작으면 잘라내고, 그렇지 않으면 존속시킨다. 그러나, 이렇게 현재 생성된 부분 트리의 수를 전혀 고려하지 않고 임계값을 적용하면 구문 분석 초기에 옳은 구문 구조가 잘려나가는 경우가 많이 발생한다. 예를 들어, 현재 어절에서 단지 2개의 부분 트리가 생성되었다고 가정하면 둘 중에 하나는 임계값 계산에 의해 잘려 나갈 가능성이 커진다. 이러한 경우는 10개의 부분 트리에서 2~3개의 구문 구조가 잘려나가는 것보다 더 큰 탐색 오류를 발생시킬 수 있다. 본 논문에서는 이러한 오류를 고려하여 임계값을 적용하지 않고 그대로 유지할 최소의 부분 트리 개수를 정의하여 사용한다. 그대로 유지할 부분 트리의 개수 n은 [식12]와 같다.

$$n = 4.01 \times (i-1) \qquad (12)$$

[식12]에서 i는 현재까지 구문 분석(오른쪽 우선 챠트 파싱)이 끝난 어절의 수이고, 4.01은 [1]의 확률 추출 문서에서 얻은 어절당 평균 트리의 수이다.

4. 실험 및 평가

4.1 확률 추출 문서의 분포 및 성격

신문사설, 소설, 교과서를 중심으로 한 494문장을 이용하였으며, 실험 대상 문서는 제외하였다. 문법 규칙도 같은 문장에서 추출하였으며 총 1352개이다.

4.2 실험 대상 문서의 분포 및 성격

시스템을 평가하기 위해 대상으로 삼은 문서는 [표2]와 같으며, 문장의 평균 어절 길이는 7.8이다.

< 표 2 : 실험 대상 문서의 구성 >

문서 종류	문서 이름	문장수
신문사설	동아일보 사설	20
소설	농민	20
소설	겨울	20
초등학교 교과서	자연	20
초등학교 교과서	사회	20

4.3 구문 분석 정확도 평가

실험 데이터 100문장에 대해, beam search를 사용하지 않고 구문 분석을 한 결과, 3119개의 구문 트리를 생성했으며, [표3]과 같은 정확도를 보였다. [표3]에서 보는 바와 같이 올바른 구문 트리가 상위 1위로 나타난 경우가 45%, 상위 5위 안에 나타난 경우는 70%, 상위 10위 안에 나타낸 경우는 82%를 보였다.

< 표 3 : 구문분석 정확도 >

순위	개수	순위	개수
1	45	6	5
2	11	7	5
3	10	8	1
4	4	9	1
5	0	10	0

4.4 Beam search 알고리즘의 성능 평가

기존의 beam search 알고리즘과의 비교를 위해 두 가지 실험을 하였다. 먼저, 3.2에서 제안한 전역적 확률 계산법의 효율성을 평가하기 위해서 임계값의 계산 방법은 기존의 방법(최대 부분 트리 확률/빔의 크기)으로 유지하면서 3.1의 부분 트리 확률 계산 방법과 비교하였다. 그리고, 임계값 계산 방법의 효율성을 평가하기 위해서 3.3에서 제안한 방법을 3.1과 3.2의 부분 트리 확률값 계산 방법에 각각 적용해 보았다.

[표4]는 beam의 크기 β를 10단위로 변화시키면서 두 알고리즘을 비교한 것이다.

< 표 4 : 성능 평가1 >

Beam Size	Collins의 방법			전역적 확률		
	P	T	F	P	T	F
1	53.7	99	4	51.2	102	0
10	89.0	786	1	89.0	782	0
20	89.0	1130	1	89.0	1129	0
30	90.2	1375	1	90.2	1373	0
40	91.5	1522	1	91.5	1531	0
50	92.7	1595	1	91.5	1608	0
60	95.1	1760	0	93.9	1755	0
70	95.1	1868	0	95.1	1865	0
80	95.1	1945	0	95.1	1938	0
90	95.1	1992	0	95.1	1987	0
100	95.1	2070	0	95.1	2068	0
960	100.0	2769	0	100.0	2777	0

P: (beam search를 사용한 상위10위의 개수)/(beam search를 사용하지 않은 상위 10위의 개수)*100
T: 생성된 구문 트리의 개수
F: 구문 분석을 실패한 개수

[표4]에서 보듯이 전역적 확률 계산법은 beam의 크기에 관계없이 구문 분석이 실패하는 경우가 없었다. 이것은 전역적 확률 계산법이 기존의 방법보다 전체 구문 구조의 특성을 잘 반영한다는 것을 보여준다. 그러나, 전역적 확률 계산법은 현재 어절의 부분 트리 확률을 계산하기 위해 현재 어절의 부분 트리를 포함하는 다음 어절의 부분 트리의 값들을 계산해야 하기 때문에 기존의 방법보다 약 2배의 시간이 소요되었다.

[표5]는 3.2에서 제안한 방법을 Collins의 부분 트리 확률 계산 방법과 3.2에서 제안한 방법에 각각 적용해 본 결과이다.

< 표 5 : 성능 평가2 >

Collins의 방법			전역적 확률		
P	T	F	P	T	F
98.8	967	0	100	984	0

[표5]에서 보듯이 두 가지 방법 모두 1/3정도의 구문 분석 트리를 생성하면서도 100%에 가까운 재현율을 얻었다. 특히, 두 가지 모두 beam

search를 사용하지 않은 것과 상위 10위를 비교했을 때 [표6]과 같이 향상된 결과를 얻었다.

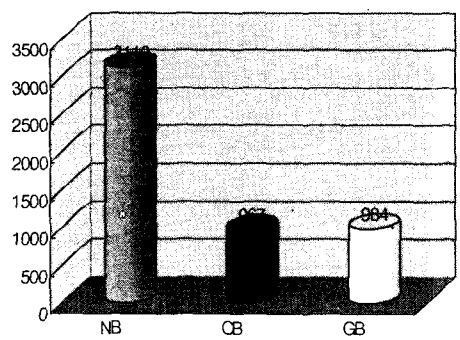
< 표 6 : 구문분석 정확도 비교 >

순위	NB	CB		GB	
	개수	개수	순위변화	개수	순위변화
1	45	45		45	
2	11	11		11	
3	10	11	4 → 3	11	4 → 3
4	4	3		3	
5	0	0		1	6 → 5
6	5	5		4	
7	5	5		5	
8	1	0	8 → X	1	
9	1	1		1	
10	0	0		0	

NB: beam search를 사용하지 않은 경우
CB: Collins의 방법을 사용한 경우
GB: 전역적 확률 계산법을 사용한 경우

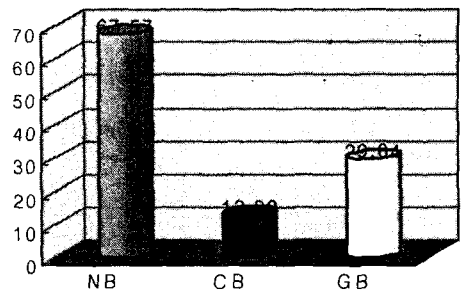
[표6]과 같이 beam search를 사용한 것에서 상위 랭커(ranker)가 더 많이 나온 이유는, 문장 전체의 확률값은 같지만 구문 분석 도중에 부분 트리에서 보다 낮은 확률값을 가지는 옳지 못한 트리들이 잘려나가기 때문이다.

[그림2]는 beam search를 사용하지 않은 것과 사용한 것에서 생성된 트리의 수를 비교하여 도식화한 것이다.



< 그림2 : 생성된 트리의 수 비교 >

[그림3]은 beam search를 사용하지 않은 것과 사용한 것의 구문 분석 시간을 비교하여 도식화한 것이다.



< 그림 3 : 구문 분석 시간 비교 >

5. 결론 및 향후과제

본 논문에서는 통계 모델을 적용하여 구문 분석을 할 때 나타나는 긴 구문 분석 시간 문제를 해결하고, 정확률을 유지하기 위해 효율적으로 beam search 알고리즘을 개선하는 방법을 제안하였다. 전역적 확률 계산법은 구문 분석의 안정성과 정확률을 유지하는데 적당했으며, 표준 편차를 이용하여 임계값을 계산하는 방법은 구문 분석 속도의 향상에 많은 기여를 했다. 특히, 표준 편차를 임계값 계산에 적용한 방법은 beam search 알고리즘을 적용하지 않은 방법이 생성한 트리수의 1/3 정도만으로도 비슷한 정확률을 보였다. 정확률, 안정성 면에서 전역적 확률 계산법과 표준 편차를 반영한 임계값 계산법을 혼합하는 것이 가장 좋은 결과를 얻었다. 물론, 구문 분석 시간 면에서는 전역적 확률 계산법을 사용하지 않은 것이 2배 정도 빨랐으나 안정성과 정확률을 생각한다면 큰 부담이 되지는 않을 것으로 생각된다.

본 논문에서 실험한 데이터의 수는 비교적 적은 100문장이고 평균 어절의 길이도 7.8로 알고리즘의 효율성을 단정적으로 말하기에는 아직 충분하지 못한 것으로 생각된다. 그러므로, 더 많은 데이터에서의 평가가 이루어져야 할 것으로 보이며, 어절의 길이에 따른 성능의 저하 등도 좋은 실험 대상이 될 것으로 생각된다. 또한, 한국어의 특징으로 인해 구문 트리의 수를 줄일 수 있는 많은 휴리스틱과 어절 정보가 조사나 어미에 포함되어 있을 것으로 보인다. 이것에 대한 연구는 구문 분석의 속도와 정확률 면에서 많은 성능 향상을 가져올 것이다. 마지막으로, 본 논문에서는 다루지 않았지만 확률 모델 자체의 정확도를 증가시키기 위한 연구도 필요할 것으로 보인다. 특히, 거리 정보의 범위를 더 넓힌다던가 아니면 용언의 격분석이나 문형정보의 이용하는 방법도 생각해 볼 수 있을 것이다.

참고 문헌

[1] 김학수, 서정연, "어휘 의존 정보에 기반한 한국어 통계적 구문분석기," *'97년도 한국정보과학회 인공지능 연구회 춘계 발표 논문집*, pp.74-80, 1997.

[2] 김지훈, 김학수, 서정연, "통계적 처리 방법을 이용한 한국어 의존구조 분석기," *'97년도 인지과학회 춘계 학술 발표 논문집*, pp.200-209, 1997.

[3] 김창현, 한국어 구문 분석을 위한 오른쪽 우선 챠트 파서, 한국과학기술원, 전산학과, 석사학위 논문, 1992.

[4] 김형근, 확률적 의존 문법과 한국어 구문 분석, 한국과학기술원, 전산학과, 석사학위 논문, 1994.

[5] 안동언, 기계번역을 위한 한국어 해석에서 형태소로부터 구문요소의 형성에 관한 연구, 한국과학기술원, 전산학과, 석사학위논문, 1987.

[6] 우승균, 구문관계를 이용한 한국어 구문분석, 한국과학기술원, 전산학과, 석사학위논문, 1992.

[7] 서광준, 어절 사이의 의존관계를 이용한 한국어 구문분석기, 한국과학기술원, 전산학과, 석사학위논문, 1992.

[8] 조영환, 구문요소화, 한국과학기술원, 전산학과, 컴퓨터시스템 실험식 내부메모, 1991.

[9] Joshua Goodman, "Global Thresholding and Multiple-Pass Parsing," *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pp.11-25, 1997.

[10] Michael John Collins, "A New Statistical Parser Based on Bigram Lexical Dependencies," *Proceedings of ACL*, 1996.

부 록

Sparse Data Problem:

$C(<\bar{w}_j,\bar{t}_j>,<\bar{w}_{hj},\bar{t}_{hj}>,\Delta_{j,hj})$ 의 해결 방안?

4개의 estimates를 구하고 그것을 (Jelinek 90)에 있는 deleted interpolation 방법으로 근사화하기 위해 3개로 줄이고 interpolation을 수행한다.

①

$$E_1=\frac{\eta_1}{\delta_1}\ E_2=\frac{\eta_2}{\delta_2}\ E_3=\frac{\eta_3}{\delta_3}\ E_4=\frac{\eta_4}{\delta_4}$$

where

$$\delta_1=C(<\bar{w}_j,\bar{t}_j>,<\bar{w}_{hj},\bar{t}_{hj}>,\Delta_{j,hj})$$
$$\delta_2=C(<\bar{w}_j,\bar{t}_j>,<\bar{t}_{hj}>,\Delta_{j,hj})$$
$$\delta_3=C(<\bar{t}_j>,<\bar{w}_{hj},\bar{t}_{hj}>,\Delta_{j,hj})$$
$$\delta_4=C(<\bar{t}_j>,<\bar{t}_{hj}>,\Delta_{j,hj})$$
$$\eta_1=C(R_j,<\bar{w}_j,\bar{t}_j>,<\bar{w}_{hj},\bar{t}_{hj}>,\Delta_{j,hj})$$
$$\eta_2=C(R_j,<\bar{w}_j,\bar{t}_j>,<\bar{w}_{hj},\bar{t}_{hj}>,\Delta_{j,hj})$$
$$\eta_3=C(R_j,<\bar{w}_j,\bar{t}_j>,<\bar{w}_{hj},\bar{t}_{hj}>,\Delta_{j,hj})$$
$$\eta_4=C(R_j,<\bar{w}_j,\bar{t}_j>,<\bar{w}_{hj},\bar{t}_{hj}>,\Delta_{j,hj})$$

②

$$E_{23}=\frac{\eta_2+\eta_3}{\delta_2+\delta_3}$$

③ deleted interpolation

If E_1 *exists, i.e.* $\delta_1>0$

$$F(R_j|<\bar{w}_j,\bar{t}_j>,<\bar{w}_{hj},\bar{t}_{hj}>,\Delta_{j,hj})=\lambda_1\times E_1+(1-\lambda_1)\times E_{23}$$

Else If E_{23} *exists, i.e.* $\delta_2+\delta_3>0$

$$F(R_j|<\bar{w}_j,\bar{t}_j>,<\bar{w}_{hj},\bar{t}_{hj}>,\Delta_{j,hj})=\lambda_2\times E_{23}+(1-\lambda_2)\times E_4$$

Else $F(R_j|<\bar{w}_j,\bar{t}_j>,<\bar{w}_{hj},\bar{t}_{hj}>,\Delta_{j,hj})=E_4$

여기에서

$$\lambda_1=\frac{\delta_1}{\delta_1+1}$$

$$\lambda_2=\frac{\delta_2+\delta_3}{\delta_2+\delta_3+1}$$