http://dx.doi.org/10.3745/KIPSTB.2011.18B.6.405

명사 어휘의미망을 활용한 문법 검사기의 문맥 오류 결정 규칙 일반화

소 길 자[†] · 이 승 희[††] · 권 혁 철[†††]

요 약

국내에서 가장 일반적으로 사용되고 있는 규칙 기반 오류 검출 방법은 언어 전문가가 한국어 문서에서 자주 발생하는 오류에 대한 검출 규칙을 경험적으로 구축하고 있다. 그러나 이렇게 경험적으로 규칙을 만들면 새로운 패턴의 문장이 나타날 때마다 규칙이 수정되어야 하므로 일관성 있는 오류 검사 및 교정을 기대할 수 없다. 본 논문에서는 이를 해결하려고 최근 개발되고 있는 어휘의미망 중에서 KorLex와 같은 정규화된 언어 자원을 활용하여 단어들의 범주 정보를 추출하고 이를 이용하여 오류 결정 규칙을 일반화한다. 그러나 현재 구축된 KorLex에는 명사의 계층관계 정보는 구축되어 있지만, 문장 요소와의 관계 정보, 즉, 격틀 정보가 부족하다. 본 논문에서는 용언 의미 오류 결정 규칙으로 사용할 선택제약 명사 클래스를 정보이론에 기초한 MDL과 Tree Cut Model을 활용하여 추출하고 이러한 선택제약 명사 클래스를 사용하여 문법 검사기 규칙을 일반화하는 방안을 제안한다. 실험 결과, 혼동하기 쉬운 네 개의 용언에 대해 목적어로 사용된 명사를 선택제약 명사 클래스로 일반화하여 문법 검사기 오류 결정 규칙 수를 평균 64.8%로 줄였고 기존 명사를 사용한 문법 검사기보다 정확도 측면에서 평균 약 6.2%정도 향상된 결과를 얻을 수 있었다.

키워드 : 문법 검사기, 문맥 의존 오류 검사, 선택제약 명사 클래스, 오류 결정 규칙 일반화, MDL(Minimum Description Length), Tree Cut Model

Generalization of error decision rules in a grammar checker using Korean WordNet, KorLex

So Gil-Ja[†] · Lee Seung-Hee[††] · Kwon Hyuk-chul[†††]

ABSTRACT

Korean grammar checkers typically detect context-dependent errors by employing heuristic rules that are manually formulated by a language expert. These rules are appended each time a new error pattern is detected. However, such grammar checkers are not consistent. In order to resolve this shortcoming, we propose new method for generalizing error decision rules to detect the above errors. For this purpose, we use an existing thesaurus KorLex, which is the Korean version of Princeton WordNet. KorLex has hierarchical word senses for nouns, but does not contain any information about the relationships between cases in a sentence. Through the Tree Cut Model and the MDL(minimum description length) model based on information theory, we extract noun classes from KorLex and generalize error decision rules from these noun classes. In order to verify the accuracy of the new method in an experiment, we extracted nouns used as an object of the four predicates usually confused from a large corpus, and subsequently extracted noun classes from these nouns. We found that the number of error decision rules generalized from these noun classes has decreased to about 64.8%. In conclusion, the precision of our grammar checker exceeds that of conventional ones by 6.2%.

Keywords : Grammar Checker, Context Dependent Error Detection, Selectional Constraint Noun Classes, Generalization of an Error Decision Rule, MDL(Minimum Description Length), Tree Cut Model

※ 이 논문은 한국콘텐츠진흥원 2010년 선정 문화기술 공동연구센터 2차년도 사업의 연구결과로 수행되었음.
† 정 회 원 : 영산대학교 게임·콘텐츠학과 전임강사
†† 정 회 원 : (주)나라인포테크 지능시스템 연구소 소장
††† 종신회원 : 부산대학교 정보컴퓨터공학부, 인지과학협동과정 교수
논문접수 : 2011년 3월 3일
수 정 일 : 1차 2011년 5월 30일, 2차 2011년 7월 4일, 3차 2011년 8월 17일
심사완료 : 2011년 9월 5일

1. 서 론

문법 검사기는 문서에 나타난 철자 오류를 검사하고 구문 오류, 의미 오류 등을 검출하는 시스템이다. 문법 검사기에서 문서에 나타난 오류를 검출하는 방법은 단일 어절의 형태소 분석만으로 검출할 수 있는 단일 오류(single-word error)와 오류가 있는 단어를 중심으로 좌우 단어의 언어 정보를 참조해야 오류 검출이 가능한 문맥 의존 오류(context-dependent error)로 구분할 수 있다[1]. 실제 문서에서 발견되는 오류 중 문맥 의존 오류가 차지하는 비중은 30% 이상으로 조사되고 있으나[2,3] 이를 검출하고 교정하는 연구 결과는 아직 미비한 실정이다.

문맥 의존 오류를 다루는 연구가 영어권에서는 n-gram 모델이나 공기정보(collocation) 확률값을 이용한 통계적 방법으로 진행되어 왔고[4,5,6], 국내에서는 규칙 기반 오류 검출, 전체 구문 분석을 이용한 오류 검출 등의 방법이 연구되었다[7,8,9]. 국내에서 가장 일반적으로 사용되고 있는 방법인 규칙 기반 오류 검출 방법은 언어 전문가가 한국어 문서에서 자주 발생하는 오류를 경험적으로 구축하고 있다. 언어 전문가는 오류가 발생한 단어의 오류 여부를 결정할 정보를 오류 결정 규칙으로 기술하는데 주로 단어, 품사, 전자 사전 정보가 사용된다. 그러나 이렇게 경험적으로 규칙을 만들면 새로운 패턴의 문장이 나타날 때마다 규칙이 수정되어야 하므로 일관성 있는 오류 검사 및 교정을 기대할 수 없다. 이를 해결하려면 최근 개발되고 있는 정규화된 언어 자원인 KorLex와 같은 어휘의미망을 활용하여 단어들의 범주 정보를 추출하고 이를 사용해서 규칙을 일반화해야 한다. 특히, 용언에 대한 오류 결정 규칙은 주어나 목적어에 오는 명사 범주 정보인 선택제약 명사 클래스를 사용하면 일반화될 수 있다.

그러나 현재 구축된 어휘의미망 KorLex에는 명사의 계층관계 정보는 있지만, 문장의 다른 요소와의 관계, 즉, 목적어-서술어, 주어-서술어 관계를 나타내는 격틀 정보가 부족하여 문법 검사기에 활용하려면 용언 의미 오류 결정 규칙으로 사용할 선택제약 명사 클래스 정보를 새롭게 추출해야 한다.

문장에서 용언의 주어나 목적어 역할을 할 수 있는 선택제약 명사 클래스를 구하는 문제는 선택제약 조건(Selectional Restriction) 또는 선택 선호도(Selectional Preference)를 구하는 문제로 연구되었다. 그 중 기존 구축된 어휘의미망의 중간 노드를 명사 클래스로 보고 접근한 연구로는 Resnik, Stephen Clak, Abe와 Li 등이 있다[10,11,12].

본 논문은 정보 이론에 기반한 MDL과 Tree Cut Model을 사용하여 KorLex에서 검사 단어의 선택제약 명사 클래스를 추출하고 그 결과를 활용해 문법 검사기의 오류 결정 규칙을 일반화하는 방법을 제안한다.

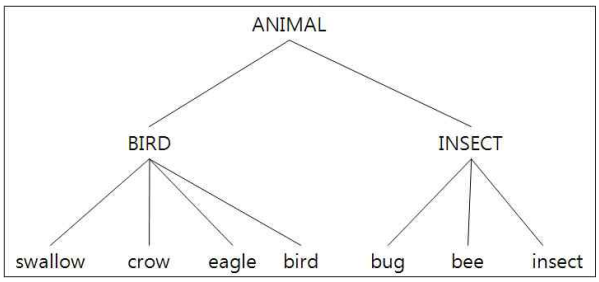
일반화 과정은 다음과 같다. 문장에서 용언의 선택제약 명사들을 추출하고, 추출한 명사들의 범주 정보인 선택제약 명사 클래스에 해당하는 신셋 정보를 어휘의미망 KorLex에서 구한다. 다음, 전 단계에서 구한 KorLex의 신셋 정보를 문법 검사기의 오류 결정 규칙의 제약 조건으로 사용한다.

제안된 오류 결정 규칙의 성능을 규칙 수와 검사 정확도로 평가한다. 선택제약 명사 클래스로 오류 결정 규칙을 일반화하면 규칙의 수가 줄어들게 되는데, 여기서 줄어든 규칙 수는 일반화 수준을 나타낸다. 검사 정확도는 일반화된 규칙이 오류를 얼마나 정확히 검출하는지로 측정된다. 실험에 사용된 문법 검사기 규칙은 혼동하기 쉬운 용언 네 쌍의 검사 규칙이다. 실험 결과, 일반화되기 전 문법 검사기와 비교했을 때 규칙의 수가 평균 64.8%로 감소하였고, 정확도는 평균 약 6.2%정도 향상되는 결과를 보였다.

2. Tree Cut Model과 MDL

2.1 Tree Cut Model

Tree Cut Model은 Abe와 Li(1998)에 의해 제안된 방법이다. 트리 형태의 명사 시소러스에서 동사의 논항에 적합한 서브 트리를 명사 클래스로 선택하는 방법이다[10]. 동사와 클래스 간의 선택 선호도는 $P(n|v,r)$ (n:클래스에 포함된 명사, v:동사, r:slot(문법적 관계의 논항 위치))로 계산한다. 트리 형태 시소러스 내에서 단말 노드는 한 개의 명사를 대표하고 중간 노드는 명사의 클래스를 의미하며 트리 안에서 단말 노드들을 분할할 기준이 되는 노드의 집합을 cut이라고 한다.



(그림 1) 계층적 시소러스

예를 들어 (그림 1)에는 5개의 cut이 존재한다: [ANIMAL], [BIRD, INSECT], [BIRD, bug, bee, insect], [swallow, crow, eagle, bird, INSECT] [swallow, crow, eagle, bird, bug, bee, insect]. Tree Cut Model에서 후보 모델 M은 tree cut과 파라미터 벡터로 표현된다.

$$M = (\Gamma, \Theta)$$
$$\Gamma = [C1, C2, ..., Ck],$$
$$\Theta = [P(C1), P(C2), ..., P(Ck)]$$

이 식에서 C_1, C_2, ..., C_k은 tree cut을 의미한다. 이 때 $\sum_{i=1}^{k} P(C_k) = 1$을 만족해야 한다.

(그림 1)의 시소러스에서 모델 M=([BIRD, INSECT], [P(BIRD), P(INSECT)]은 하나의 tree cut model로 표현된다. Abe와 Li는 트리에서 동사와 선택 선호도가 높은 cut들로 구성된 모델을 찾는 방법으로 MDL을 사용하였다. MDL은 Rissanen[13]에 의해 제안된 방법으로 모델과 관찰된 데이터를 표현하는데 필요한 비트 수를 고려하여 모델을 선택하는 방법이다.

2.2 MDL(Minimum Description Length)

어휘의미망에서 일반화된 명사 클래스가 루트에 가까우면 모델은 단순하지만, 실제 데이터를 표현하기에는 부족하다. 반면 단말노드에 가까운 클래스가 선택되면 모델은 파라미터 수가 많아지므로 복잡해지지만, 실제 데이터를 설명하기에는 더 좋은 모델로 볼 수 있다. 모델의 복잡도(complexity)와 모델의 실데이터와의 적합성(good-fitness) 사이에는 trade-off가 존재한다. 일반화된 명사 클래스를 고려할 때 모델의 복잡도와 적합성 모두를 고려하여 선택할 수 있는 방법으로 MDL을 사용한다. MDL에서는 일반화된 정보를 얻는 과정을 압축능력(ability to compress)으로 간주한다. 즉, 모델과 데이터를 압축하는데 사용되는 비트 수가 가장 적은 모델이 가장 좋은 모델이 된다. 모델 자체를 압축하는데 소요되는 비트 수를 "모델 정보량(model description length)"이라고 하고 모델을 통해 나타나는 데이터를 압축하는데 소요되는 비트 수를 "데이터 정보량(data description length)"이라고 한다. 모델의 총 정보량은 이 두 가지 정보량을 합쳐서 나타낸다.

모델의 총 정보량(L(TOT)) =
모델 정보량(L(MOD)) + 데이터 정보량(L(DATA))

Abe와 Li에 의하면 모델 정보량과 모델에 의한 데이터 정보량은

$$L(MOD) = \frac{k}{2} * \log|S| \quad (1)$$

$$L(DATA) = -\sum_{n \in S} \log P(n|v,r) \quad (2)$$

로 나타낼 수 있다. 이때 k는 모델에 포함된 명사 클래스의 개수이다. S는 실험에 나타난 명사 리스트이다. $|S|$는 S의 크기, 즉 실험에 나타난 명사 출현 회수 합을 의미한다. $P(n|v,r)$은 MLE(maximum likelihood estimation) 방법으로 확률값을 계산한다. 그러나 명사 n은 여러 명사 클래스에 속할 수 있으므로 $P(n|v,r)$은 n이 속한 명사 클래스의 크기로 나누어 정규화한다.

$$P(n|v,r) = \frac{1}{|C|} P(C) \text{ for each } n \in C$$

$$P(C) = \frac{f(C)}{|S|} \quad (3)$$

여기서 C는 명사 n이 속한 명사 클래스를 나타내고 $f(C)$는 명사 클래스 C에 속한 명사들의 출현 빈도수의 합을 나타낸다. MDL은 후보 모델들의 총 정보량을 구한 후 최소의 정보량 값을 가진 모델을 선택한다.

3. 명사 어휘의미망을 활용한 용언 선택제약 명사 클래스 추출

3.1 계층적 명사 어휘의미망 KorLex

2004년부터 개발되기 시작한 부산대학교의 KorLex는 1단계(KorLex 1.0)로 PWN의 명사를 대역한 KorLex Noun 1.0을 공개하였다. 2단계(KorLex 1.5)에서 1.0에 부족한 한국어 어휘의미의 특성을 반영하여 개념 및 의미를 다시 구조화하고 있다. 이때 한국어에 적용될 의미 세분화의 기준은 표준국어대사전에 두었다. KorLex는 개념을 표상하는 최소 단위를 "동일한 어휘의미를 가지는 동의어 집합(synonym set, 신셋(synset))"으로 규정한다[14]. 예로 다의어 "배"는 "복부, 선박, 배수" 등의 의미를 가지는데 이를 KorLex에서는 {복부, 배}, {선박, 배}, {배수, 배} 등으로 표현하여 의미 중의성이 없이 하나의 개념을 나타낸다. KorLex는 명사, 동사, 형용사, 부사에 대해서 구축하였고 그 중 가장 먼저 명사에 대해서 의미망이 구축되었다. 명사 어휘의미망은 신셋 간, 어의 간 의미 관계가 매우 다양하게 표현되어 있다. 명사 어휘의미망의 의미 관계 중 신셋 간의 상위(hypernym)와 하의(hyponym) 계층은 IS-A 방식으로 나타낸다. 명사 어휘의미망은 9개의 최상위 계층을 가지며, 25개의 의미 분류가 있다[14]. 본 논문에서는 KorLex의 신셋 서브 트리를 명사 클래스로 정의하고 용언의 오류 결정 규칙으로 사용할 신셋을 MDL과 Tree Cut Model을 이용해 추출한다.

3.2 용언의 선택제약 명사 추출

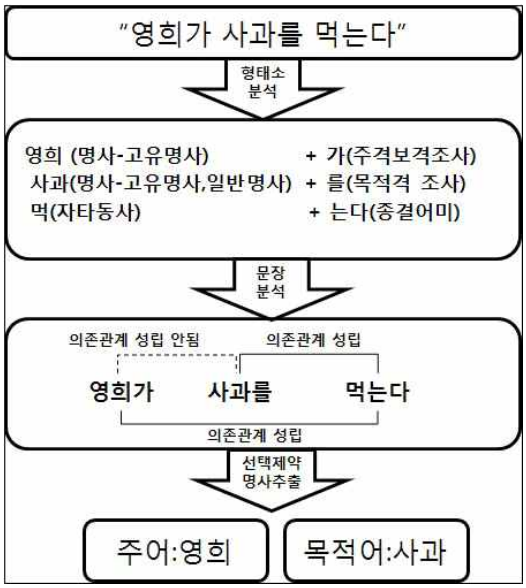
용언의 선택제약 명사 추출은 형태소 분석기와 부분 문장 분석기를 통해 주어, 목적어에 해당하는 명사를 추출한다. 부분 문장 분석기는 문장 성분 간의 의존 관계를 생성하는 의존 문법을 사용하여 문장 분석을 수행한다. 의존 문법에 의한 문장 분석은 문장 내의 성분 중, 어느 것이 지배소이고 어느 것이 의존소인지 형태소 단위부터 추출하고 그들

〈표 1〉 의존문법에 기반한 지배-의존 규칙

지배소	의존소	결과	관계
명사	관형사	수식	명사구
명사	관형사구	수식	명사구
주격보격조사	명사	격부여	격조사구
주격보격조사	명사구	격부여	격조사구
목적격조사	명사	격부여	격조사구
목적격조사	명사구	격부여	격조사구
동사	명사	논항	동사구
동사	격조사구	논항	동사구
동사구	명사	논항	동사구
동사구	명사구	논항	동사구
동사구	격조사구	논항	동사구

간의 관계를 밝히면서 문장 분석을 수행하여 용언의 주어나 목적어를 추출한다. 본 연구에서 사용한 의존 문법의 의존 규칙은 총 138개이고 그 일부는 <표 1>과 같다.

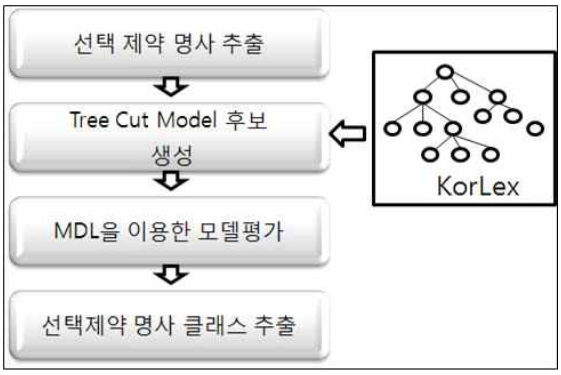
예를 들어 “영희가 사과를 먹는다”의 문장은 (그림 2)와 같이 서술어의 원형이 복원된 뒤 형태소 분석을 한 후 문장 분석기를 통해 선택제약 명사를 추출한다.



(그림 2) 용언의 선택제약 명사 추출

3.3 KorLex를 이용한 선택제약 명사 클래스 추출

용언의 선택제약 명사 클래스를 추출하는 방법은 정보이론에 기반한 MDL과 Tree Cut Model을 활용한다. 선택제약 명사 클래스 추출 시스템은 (그림 3)과 같다. 용언의 선택제약 명사 추출기에서 (명사, 문법관계, 빈도수)=(n, r, f)로 구성되는 트리플 데이터(triple data)를 구한다. Tree Cut Model 후보 생성 단계에서는 명사 n을 포함한 신셋을 KorLex에서 찾고, 이 신셋을 포함하는 KorLex의 서브 트리로 Tree Cut Model 후보를 생성한다. 생성된 후보 Cut Model은 MDL을 이용해 평가되고 적합한 후보 Cut Model이 용언의 선택제약 명사 클래스로 선택된다.



(그림 3) 선택제약 명사 클래스 추출 시스템

후보 Cut Model은 명사 n을 포함한 신셋의 형제 노드들의 집합 또는 이 신셋을 포함하는 KorLex의 상위 노드의 부분집합으로 만들어질 수 있다. 예를 들어 <표 2>에 나타난 “먹다”의 선택제약 명사 리스트를 이용해 KorLex에 맵핑하면 (그림 4)의 트리 구조를 얻을 수 있다.

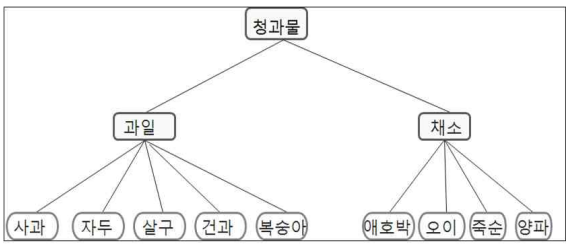
〈표 2〉 “먹다”의 선택제약 명사 리스트

명사	문법관계	빈도수
사과	목적어	4
자두	목적어	2
복숭아	목적어	4
애호박	목적어	2
오이	목적어	4
양파	목적어	4

(그림 4)의 트리 구조에서는 5개의 cut을 구할 수 있다: [청과물], [과일, 채소], [과일, 애호박, 오이, 죽순, 양파],[사과, 자두, 살구, 건과, 복숭아, 채소],[사과, 자두, 살구, 건과, 복숭아, 애호박, 오이, 죽순, 양파].

Cut Model M은 (Cut, Θ)로 구성되는데 Cut과 신셋 확률값 벡터 Θ는 다음과 같다.

$$\text{Cut} = [C_1, C_2, C_3, \ldots, C_k],$$
$$\Theta = [P(C_1), P(C_2), P(C_3), \ldots, P(C_k)]$$



(그림 4) 어휘의미망의 예

이때 $C_1, C_2, C_3, \ldots, C_k$는 KorLex의 신셋이고, $P(C)$는 수식(4)에 의해 구한다.

$$P(C) = \frac{\text{신셋 } C\text{에 속하는 선택제약명사 빈도 합}}{\text{선택 제약명사 빈도 합}} \quad (4)$$

<표 2>의 데이터에서 5개의 cut 모델을 구하면 아래와 같다.

M(1) = {[청과물],1.0}
M(2) = {[과일, 채소], [0.5, 0.5]}
M(3)= {[과일, 애호박, 오이, 죽순,양파],
[0.5, 0.1, 0.2, 0.0, 0.2]}
M(4)= {[사과, 자두, 살구, 건과, 복숭아, 채소],
[0.2, 0.1, 0.0, 0.0, 0.2, 0.5]}

M(5)= {[사과, 자두, 살구, 건과, 복숭아, 애호박, 오이, 죽순, 양파],[0.2, 0.1, 0.0, 0.0, 0.2, 0.1, 0.2, 0.0, 0.2]}

이 5개의 모델에 대해서 모델의 복잡도를 나타내는 모델 정보량(L(MOD))과 모델의 데이터 적합도를 나타내는 데이터 정보량(L(DATA))을 구한다. 예로 M(4)에 대해서 모델 정보량과 데이터 정보량을 구하면 <표 3>과 같다. 이때 단말 노드는 하위 신셋이 없는 명사 클래스로 본다. M(4)는 확률 파라미터 개수가 6이다.

〈표 3〉 모델 정보량 계산 예

C	사과	자두	살구	건과	복숭아	채소
f(C)	4	2	0	0	4	10
|C|	1	1	1	1	1	4
P′(C|v,r)	0.2	0.1	0.0	0.0	0.2	0.5
P′(n|v,r)	0.2	0.1	0.0	0.0	0.2	0.125
모델 정보량	(6-1)/2 × log20 = 10.8048					
데이터 정보량	4*log0.2 + 2*log0.1 + 4*log0.2 + 10*log0.125 = 55.21928					
총정보량	10.8048 + 55.2193 = 66.0241					

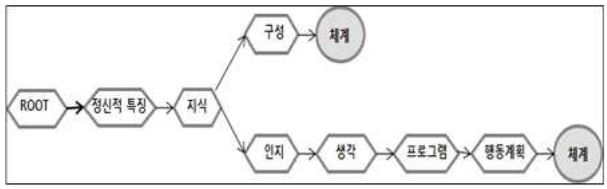
이와 같은 방법으로 5개의 모델에 대해서 모델 정보량과 데이터 정보량을 구한 결과는 <표 4>와 같다.

〈표 4〉 모델들의 정보량 결과값

모델	모델 정보량	데이터정보량	총정보량
M(1)	0	63.3985	63.3985
M(2)	2.1609	63.2193	65.3802
M(3)	8.6439	58.4386	67.0825
M(4)	10.8048	55.2193	66.0241
M(5)	17.2878	50.4386	67.7264

<표 4>를 보면 모델 M(1)이 모델 복잡도가 가장 낮고 M(5)의 모델 복잡도가 가장 높은 것으로 보인다. 그러나 데이터 적합도는 모델 M(1)보다는 모델 M(5)가 좋은 것으로 드러나 모델 복잡도와 데이터 적합도는 trade-off가 있음을 알 수 있다. MDL은 이 두 정보량의 합을 사용하여 가장 정보량이 낮은 모델을 선택한다. <표 4>에서는 모델 M(1)이 선택되어 "먹다"의 선택제약 명사 클래스로 [청과물]을 사용할 수 있다. 부분적인 모델에서도 같은 결과를 보이는데, 하위 신셋 모델 M(hyponym)=[사과, 자두, 살구, 건과, 복숭아]=33.8631, 상위 신셋 모델 M(hypernym)=[과일]=33.2192이므로 [사과, 자두, 살구, 건과, 복숭아]보다 [과일]이 선택된다. 이 사실을 이용하여 선택제약 명사 클래스 추출 과정은 KorLex 전체에 대한 모든 모델을 만들지 않고 두 개의 모델, 즉 하위 신셋들로 구성된 모델 M(hyponym)과 이 신셋들의 직접적인 상위 노드로 구성된 모델 M(hypernym)에 대해서만 정보량을 비교하며 단말 노드에서부터 계층적 구조를 따라 M(hypernym)의 정보량이 M(hyponym)의 정보량보다 작을 동안 반복적으로 수행된다.

KorLex에서 선택제약 명사 클래스를 추출할 때는 두 가지 문제가 해결되어야 한다. 첫 번째는 명사와 맵핑된 신셋이 KorLex의 계층적 구조에서 다른 깊이를 가질 때 발생하는 단말 노드 선택 문제고, 또 하나는 명사와 맵핑된 신셋의 빈도수 계산이다. 선택제약 명사 추출기에서 얻어진 명사는 의미 중의성이 제거되지 않은 상태이므로 KorLex의 여러 신셋과 맵핑될 수 있다. (그림 5)의 예처럼 서로 다른 깊이의 신셋과 맵핑될 때 단말 노드로 어느 신셋을 선택할지 결정해야 한다.



(그림 5) 명사 "체계"의 신셋 깊이

본 논문에서는 첫 번째 문제를 해결하려고 추출된 명사들이 KorLex에 맵핑된 평균 깊이를 계산하였고 단말 노드는 평균 깊이 이하의 노드로만 한정하여 최상위 노드(Root)에 너무 가까운 신셋이 단말 노드로 선택되는 문제를 해결한다. 두 번째 문제는 가장 단순한 의미 중의성 해소 방법으로 명사가 맵핑된 신셋에 동일한 신뢰도를 부여하여 데이터에서 나타난 명사의 빈도수를 명사가 맵핑된 신셋 수로 나누어 신셋의 빈도수를 계산한다. 즉 명사 n의 서로 다른 의미에 의해 갖게 되는 신셋의 빈도수는 아래 식 (5)의 방법으로 계산된다.

$$f'(C) = f(C) / n\text{이 맵핑된 신셋 수} \quad (5)$$

여기서 f(C)는 신셋 C의 빈도수를 의미한다.

(그림 6)은 단말 노드에서부터 KorLex의 계층적 구조를 따라 하위 신셋 모델과 상위 신셋 모델을 비교하며 더는 상

```
function Find_Class(c, node_depth)
    node_depth++
    if(node_depth > leaf_level &&
      is_found_in_corpus(c)) then
        return leaf_node;
    else
        for each hyponym s(i) of c
            ret = Find_Class(s(i), node_depth)
            if ret==LEAF_NODE or
               ret == GENERALIZE_SUCCESS then
                frequency of c += frequency of s(i)
                sub_list = append(s(i))
            else if ret == GENERALIZE_FAIL then
                return ret  //Generalization fail!!
            endif
        loop
        if LTOT(c) <= LTOT(sub_list) then
            return GENERALIZE_SUCCESS
        else
            return GENERALIZE_FAIL
        endif
    endif
end Find_Class
```

(그림 6) 선택제약 명사 클래스 추출 알고리즘

위 신셋 모델이 선택되지 않을 때까지 반복적으로 수행하는 선택제약 명사 클래스 추출 알고리즘이다. 추출된 선택제약 명사 클래스는 문법 검사기의 오류 결정 규칙으로 사용한다.

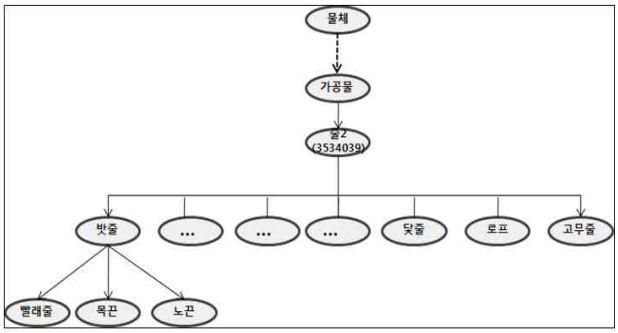
4. 선택제약 명사 클래스를 이용한 오류 결정 규칙 일반화

문법 검사기에 구현된 문맥 오류 검사 규칙은 크게 ①오류가 가능한 단어를 표기하는 표제어, ② 검사의 검사범위에 대한 정보를 제공하는 규칙의 종류, ③오류 이유를 알려주는 도움말, ④오류가 발생하면 대치어로 제시하는 단어를 제공하는 대치어 정보부, ⑤오류가 가능한 어절의 검색에서 사용될 문법적 제약조건을 나타내는 현재 어절 제약 조건부, ⑥검사할 때 오류 여부를 결정할 정보를 포함하는 오류 결정 규칙으로 구성된다. 이렇게 구성된 지식베이스는 한 어절 단위의 철자검사가 수행된 후 문맥 오류 검사기에서 문장의 문맥 오류를 검사하고 교정하는데 사용된다. 검사 단어가 용언일 때 오류 결정 규칙은 검사 단어와는 같이 쓰일 수 없고 대치어로 제시되는 용언의 선택제약 명사로 구성한다. 이 특성을 이용하여 대치어의 선택제약 명사 클래스를 추출하고 이를 오류 결정 규칙으로 사용하여 일반화를 수행한다.

(그림 7)은 “엉기다”를 “엉키다”로 잘못 쓴 오류를 검사하는 규칙이다. "엉기다"는 '한데 뭉쳐 굳어지다', '섞이다'의 의미이고 "엉키다"는 '일이나 물건이 서로 얽히게 되다'의 의미이다. 문장에서 “엉기다”의 주어로 사용된 명사가 KorLex의 신셋 {줄2}의 하위 신셋에 나타나면 “엉키다”의 오용으로 판단하고 “엉기다”를 “엉키다”로 교정한다. 오류

① 검사단어: 엉기
② 검사범위: 일반동사
③ 도움말: 6207
④ 대치어: 엉키
⑤ 검사방향: 앞
////여기서부터 오류 결정 규칙부//////////////////////
⑥ 오류 결정 규칙 : 대상 : 주어
조건 : 주어의 명사 ∈ {줄2;3534039}

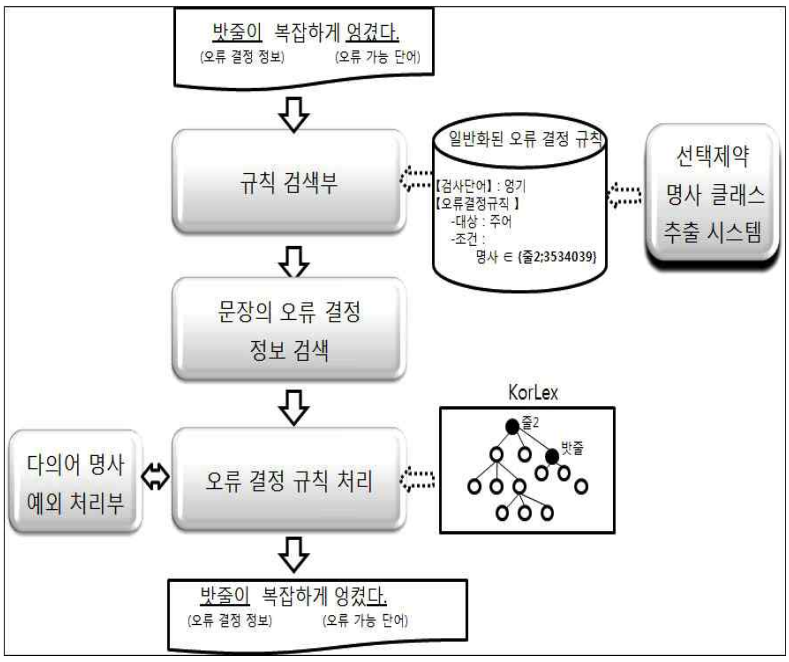
(그림 7) 문맥 오류 검사 규칙 일반화 예



(그림 8) {줄2}의 하위 신셋 구조

결정 규칙에 표기된 ”3534039“는 신셋의 고유 번호로 KorLex의 신셋 검색에 사용된다. (그림 8)은 KorLex에서 {줄2}의 하위 신셋 구조이다.

이렇게 선택제약 명사 클래스로 일반화된 오류 결정 규칙을 갖는 문법 검사기의 문맥 오류 처리 과정은 (그림 9)와 같다. 규칙 검색부에서는 문장에 오류 가능 단어가 있으면 오류를 처리할 규칙을 검색한다. 부분 문장 분석을 통해 오



(그림 9) 문법 검사기의 문맥 의존 오류 처리 시스템 구성도

류 결정 정보가 되는 주어나 목적어를 검출한다. 오류 결정 규칙 처리부에서는 주어나 목적어로 사용된 선택제약 명사 클래스 정보를 KorLex에서 가져오고 가져온 정보가 오류 결정 규칙에 명시된 범주에 포함되는지 비교하여 입력된 문장을 교정할 것인지 결정한다. (그림 9)에서 다의어 명사 예외처리부는 의미 중의성 때문에 명사가 여러 명사 클래스에 포함될 때 호출되는 모듈이다. 선택제약 명사 클래스를 오류 결정 규칙으로 사용하여 검사를 수행하면 명사의 의미 중의성 때문에 맞는 문장을 틀렸다고 알려주는 "잘못된 경고(false alarm)"가 발생할 수 있다. 이런 문제를 최소화하기 위해 명사가 속한 클래스 중 현재 검사 단어의 선택제약 명사 클래스가 있으면 오류 문장으로 판단한 검사 결과를 무시한다. 예를 들어 "말을 가리키다"에서 "말"은 {척추동물}과 {언어수행}에 모두 속하는 명사다. "말"이 {언어수행} 클래스에 속하는 의미로 사용되었다면 "말을 가리키다"는 "말을 가르치다"로 교정되어야 한다. 그러나 {척추동물}의 의미로 사용되었다면 "가리키다"의 주어로 사용될 수 있으므로 오류 검사에서 "오류 문장"이라고 나온 결과를 무시하여, "잘못된 경고"가 발생하지 않도록 한다.

5. 실험 및 평가

문법 검사기의 오류 결정 규칙 일반화 실험 평가는 오류 결정 규칙 수와 문법 검사기의 정확도로 평가한다. 명사를 사용하면 오류 결정 규칙이 열린 목록이 되어 규칙 수가 방대해질 수 있다. 그러나 선택제약 명사 클래스를 이용하면 그 수가 줄어들게 되는데, 이때 줄어든 규칙 수는 논문에서 제안한 규칙 일반화 수준을 측정하는 기준이 될 수 있다.

본 논문에서는 문법 검사기의 오류 결정 규칙으로, 용언의 추출된 선택제약 명사 클래스를 모두 사용하였다. 일반화된 오류 결정 규칙을 가진 문법 검사기의 정확도를 실험하기 위해 조선일보 37만 어절을 가진 기사와 2003년, 2004년 약 2천만 어절을 가진 한겨레 신문에서 문맥 의존 오류 검사를 수행한다.

실험은 조선일보 37만 어절에서 "가르치다/가르키다", 한겨레 신문 약 2천만 어절에서 "늘리다/늘이다", "드러내다/들어내다", "마치다/맞히다" 혼동하기 쉬운 용언 네 쌍을 사용하여 제안한 방법의 검사 정확도를 평가하였다. 여기서 사용한 검사 정확도의 정의는 다음과 같다.

$$\text{검사 정확도} = \frac{\text{오류 검출 문장수}}{\text{오류 문장수}} \times 100$$

5.1 선택제약 명사 클래스 추출

실험에서 사용되는 네 쌍의 용언은 모두 목적어로 사용된 명사를 통해 사용 오류를 판별할 수 있으므로 먼저 선택제약 명사 추출 시스템을 이용해 목적어를 원시 코퍼스에서 추출한다. 선택제약 명사 클래스 추출에 사용한 코퍼스는 천만 어절로 된 세종 현대 문어 말뭉치를 사용한다. 목적어 추출과정에서 '것', '등', '따위', 등의 의존명사는 실험에서 제외한다. 추출된 명사는 KorLex에 맵핑되고 추출된 명사를 포함한 신셋은 단말 노드가 된다. 명사의 의미 중의성이 해소되지 않아 여러 개의 단말 노드로 맵핑되고 이때 신셋은 식 (5)와 같이 명사의 출현 빈도를 명사가 속한 신셋 수로 나누어 빈도를 계산한다.

네 쌍 용언 중 "가르치다/가르키다"를 선택하여 선택제약 명사 클래스 추출 과정을 설명하면 다음과 같다. <표 5>는 동사 "가르치다"의 목적어 위치에 나타난 명사들의 빈도수 및 KorLex에 맵핑되는 신셋을 보여준다. <표 5>의 데이터를 사용하여 선택제약 명사 클래스 추출 알고리즘을 통해 KorLex에서 추출한 '가르치다'의 선택제약 명사 클래스는 (그림 10)과 같다. <표 6>은 실험에서 사용된 하위 신셋(hyponym)으로 구성된 모델과 상위 신셋(hypernym)으로 구성된 모델의 정보량 비교표다. <표 6>을 보면 M(hyponym)의 정보량보다 M(hypernym)의 정보량이 더 작으므로 {복싱 1, 권투 1, 권투경기1}, {레슬링 1}대신 {접촉경기1}을 명사 클래스로 추출한다.

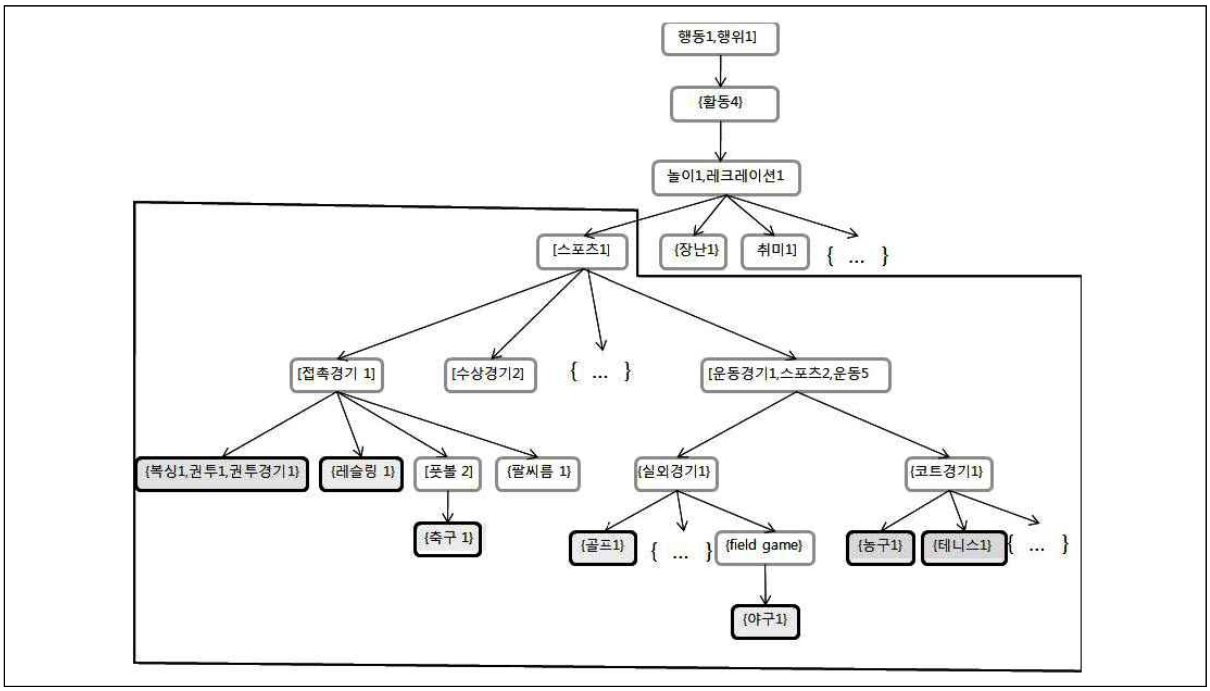
〈표 5〉 "가르치다"의 목적어 빈도수와 신셋 맵핑 예

목적어	빈도	KorLex에 맵핑된 신셋 정보 {신셋;synset 번호}
축구	5	{축구1;453585}
야구	1	{야구1:447288}
농구	2	{농구1:456335} {농기구1,농구2;14443660}
테니스	1	{테니스1,정구1;457626}
골프	2	{골프1;440870}
레슬링	2	{레슬링1;424160}
복싱	2	{복싱1,권투1,권투경기1;422482}

〈표 6〉 "가르치다" 목적어에 대한 두 모델의 정보량 비교

이름	모델	모델 정보량	데이터 정보량	모델총 정보량
M (hypornym)	[복싱1, 권투1, 권투경기1], [레슬링1], [풋볼2] [아이스하키1, 하키2] [씨름1] [팔씨름1]	88.8854	24.1435	113.0289
M(hypernym)	[접촉경기]	99.2304	4.0239	103.2543

<표 7>은 실험에 의해 "가르치다"의 선택제약 명사 클래스로 뽑힌 결과를 보여준다. 문서에 나타난 어휘 중 "영문학, "역사", "미술사"에 의해 {인문학1}이 추출되었고, "생각", "시각", "개인주의"에 의해 {태도1}이 추출되었다. 실험 결과 "가르치다"의 선택제약 명사 881개를 이용해 332개의 클래스를 추출하였다.



(그림 10) "가르치다" 선택제약 명사 클래스 예

〈표 7〉 "가르치다"의 선택제약 명사 클래스 추출 결과 예

선택제약 명사 클래스	문서에 나타난 명사
인문학 1	영문학, 역사, 미술사, 연기
태도1	생각, 시각, 개인주의
스포츠1	복싱, 레슬링, 축구, 야구
방법 1	방법, 기술, 통계, 묘책
마음상태1	준비, 자각, 호기심
사회통제1	의무, 형벌, 지휘, 정직
논리적 사고1	분류, 정책, 가정, 점

5.2 일반화된 오류 결정 규칙을 이용한 문법 검사기 성능 평가

앞 절에서 언급한 "가르치다/가르키다" 용언 쌍에 대해 목적어로 검출된 단어와 사용된 검사 규칙 등을 예로서 검사 정확도를 구하는 과정을 설명하면 다음과 같다. 실제 문서에는 오류 문장이 많지 않으므로 실험의 효율성을 위해 테스트 문서에 있는 "가르치다"를 모두 "가리키다"로 바꾸어 실험이 진행된다. 다른 용언에 대해서도 동일한 방식으로 실험이 진행된다. 실험은 문법 검사기의 부분 문장 분석으로 목적어 검출이 가능한 문장을 대상으로 한다. '가르치다/가르키다'의 경우 32개의 오류 문장이 존재한다. <표 8>은 오류가 있는 문서에서 나타난 "가리키다"의 목적어다.

〈표 8〉 오류 문서에서 "가리키다"의 목적어로 검출된 단어

과정, 연기, 테크닉, 인물화, 한글, 규율, 국어, 존귀함, 영어, 서양화, 합창, 요가, 법, 독일어, 종교, 신기술, 사진, 시계수리학, 컴퓨터, 한글, 교육, 조리법, 실무

<표 9>는 "가르치다/가리키다" 실험에서 나타난 오류 문장과 오류 검출에 사용한 일반화된 오류 결정규칙을 보여준다.

〈표 9〉 오류 문장 및 사용된 검사 규칙

예1	오류 문장	"사회복지·장례지도사 과정을 가리키는 전문대에 입학했다."
	오류 검출에 사용된 규칙	대상 : 목적어 조건 : 명사 ∈ {지식2;6225142}
예2	오류 문장	"인물화를 가리키는 ooo(51) 주얼리디자인과 겸임교수"
	오류 검출에 사용된 규칙	대상 : 목적어 조건 : 명사∈ {영상물1;3782824}
예3	오류 문장	"규율을 가리키고 용기를 키워"
	오류 검출에 사용된 규칙	대상 : 목적어 조건 : 명사 ∈ {방법1 ;5333823}

실험 결과, 명사로 오류 결정 규칙을 구성한 문법 검사기는 27개의 오류를 검출하여 84%의 검사 정확도를 보였고 선택제약 명사 클래스로 오류 결정 규칙을 구성한 문법 검사기는 30개의 오류를 검출하여 93%의 검사 정확도를 보였다.

<표 10>은 혼동하기 쉬운 다양한 용언 쌍에 대하여 제안된 선택제약 명사 클래스로 오류 결정 규칙을 구성한 문법 검사기의 성능을 평가한 결과이다. 실험 결과에서 대부분은 제안한 방법이 기존의 명사로 오류 결정 규칙을 구성한 문법 검사기보다 우수한 성능을 보였고 오류 결정 규칙 수의 감소 비율이 높을수록 성능이 향상되는 것을 확인할 수 있

〈표 10〉 명사와 선택제약 명사 클래스로 각각 오류 결정 규칙을 구성한 문법 검사기 성능 비교

혼동하기 쉬운 용언 쌍	오류 문장 수	규칙 수		규칙 수 감소 비율(%)	검출된 문장 수		검사 정확도(%)	
		명사	명사 클래스		명사	명사 클래스	명사	명사 클래스
늘리다 /늘이다	1108	269	80	70.3%	725	917	65.4%	82.8%
드러내다 /들어내다	954	765	285	62.7%	884	730	92.7%	76.5%
마치다 /맞히다	1938	195	70	64.1%	1267	1552	65.4%	80.1%

었다. 그러나 "드러내다/들어내다" 용언 쌍의 경우는 기존 방법이 제안된 방법보다 성능이 더 좋은데 이는 오류 문장 수에 비해 제공되는 명사 규칙 수가 너무 많아 대부분의 오류를 규칙에서 제공하므로 검사 정확도가 높게 나온다. 이런 특별한 예를 제외하고는 선택제약 명사 클래스를 사용한 방법이 명사를 사용한 방법보다 우수하다.

실험에서 혼동하기 쉬운 용언 네 쌍의 목적어로 사용된 명사를 이용해 선택제약 명사 클래스를 추출하였고 이를 이용하여 문법 검사기 오류 결정 규칙의 수를 평균 64.8%로 줄였다. 일반화된 오류 결정 규칙을 이용한 문법 검사기의 검사 정확도를 측정하기 위해 조선일보와 한겨레 신문을 대상으로 문맥 의존 오류 검사를 수행한 결과 기존 명사를 사용한 문법 검사기보다 네 쌍 용언의 검사 정확도가 평균 약 6.2% 정도 향상된 결과를 얻을 수 있었다.

6. 결 론

한국어 문법 검사기에서 문맥 의존 오류를 처리하는 방법은 언어 전문가가 한국어 문서에서 자주 발생하는 오류에 대해 경험적으로 규칙을 구축하였다. 그러나 이렇게 경험적으로 규칙을 만들면 새로운 패턴의 문장이 나타날 때마다 규칙이 수정되어야 하므로 일관성 있는 오류 검사 및 교정을 기대할 수 없다. 특히, 검사 단어가 용언일 때는 오류 결정 규칙에 용언의 주어나 목적어 명사가 사용되므로 규칙의 수가 방대해질 수 있다.

본 논문에서는 이를 해결하려고 최근 개발되고 있는 어휘의미망 중에서 KorLex와 같은 정규화된 언어 자원을 활용하여 선택제약 명사의 범주 정보인 선택제약 명사 클래스를 추출하고 이를 이용하여 오류 결정 규칙을 일반화하는 방안을 제안하였다. 그러나 현재 구축된 KorLex에는 명사의 계층관계 정보는 구축되어 있지만, 문장 요소와의 관계 정보, 즉, 격틀 정보가 부족하다. 격틀 정보가 없는 KorLex에서 용언 의미 오류 결정 규칙으로 사용할 선택제약 명사 클래스를 추출하기 위해 정보이론에 기초한 MDL과 Tree Cut Model을 활용하였고 추출된 선택제약 명사 클래스를 사용하여 문법 검사기 규칙을 일반화하였다. 선택제약 명사 클래스를 오류 결정 규칙으로 사용하면 의미 중의성 때문에 맞는 문장을 틀린 문장으로 판단하는 "잘못된 경고(false alarm)"가 발생하는데, 명사가 속한 클래스 중 현재 검사 단어의 선택제약 명사 클래스가 있으면 검사 결과를 무시하도록 예외처리부를 두어 이 문제를 최소화하였다. 실험은 혼동하기 쉬운 용언 네 쌍의 목적어로 사용된 명사를 이용해 선택제약 명사 클래스를 추출하였고 이를 이용하여 문법 검사기 오류 결정 규칙의 수를 평균 64.8%로 줄였다. 그리고 문맥 의존 오류 검사를 수행한 결과 기존 명사를 사용한 문법 검사기보다 평균 약 6.2% 정도 향상된 결과를 얻을 수 있었다.

참 고 문 헌

[1] M. Roger, "Spelling checkers, spelling correctors, and the misspellings of poor spellers," *Information Processing and Management*, Vol.23, No.5, pp.495-505, 1987.

[2] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys*, Vol.24, No.4, pp.377-439, Dec., 1992.

[3] A. R. Golding and D. Roth. "A winnow-based approach to context-sensitive spelling correction," *Machine learning*, Vol.34, No.1-3, pp.107-130, 1999.

[4] A. R. Golding, "A Bayesian hybrid method for context-sensitive spelling correction," *Proc. the 3rd workshop on very large corpora*, pp.39-53, 1995.

[5] E. S. Atwell, "How to detect grammatical errors in a text without parsing it," *Proc. EACL '87*, pp.38-45, 1987.

[6] C. Chelba and F. Jelinek, "Recognition performance of a structured language model," *Eurospeech*, 1999.

[7] 김현진, "어절 간 의존관계와 부분 문장 분석을 이용한 한국어 문법 검사기 구현," *부산대학교 전자계산학과 석사학위 논문*, 1997

[8] M. Y. Kang, A. S. Yoon, H. C. Kwon, "Improving partial parsing based on error-pattern analysis for Korean grammar-checker," *ACM Transactions on Asian Language Information Processing*, Vol.2, No.4, pp.301-323, 2003.

[9] 이공주, 황선영 외, "전체 문장 분석에 기반한 한국어 문법 검사기," *정보과학회논문지:소프트웨어 및 응용*, Vol.30, No.10, pp.992-999, 2003.

[10] H. Li and N. Abe, "Generalizing case frames using a thesaurus and the MDL principle," *Computational Linguistics*, Vol.24 No.2, pp.217-244, 1998.
[11] P. Resnik. "Selectional preferences and sense disambiguation," *Proc. ACL SIGLEX Workshop*, pp.52-57, 1997.
[12] S. Clark and D. Weir, "Class-based probability estimation using a semantic hierarchy," *Computational Linguistics*, Vol.28 No.2, pp.187-206, 2002.
[13] J. Rissanen. "Modeling by shortest data description," *Automatic*, Vol.14, No.5, pp.37-38, 1978.
[14] 윤애선, 황순희 외, "한국어 어휘의미망 Korlex 1.5의 구축," *정보과학회논문지:소프트웨어 및 응용*, Vol.36, No.1, pp.92-108, 2009.

소 길 자

e-mail : kjso@ysu.ac.kr
1994년 동의대학교 전자계산학과(학사)
1999년 부산대학교 전자계산학과 (이학석사)
2005년 부산대학교 컴퓨터공학과 공학박사 수료
2001년 9월~현 재 영산대학교 게임·콘텐츠학과 전임강사
관심분야 : 인간언어공학, 게임, 입체영상

이 승 희

e-mail : sheel@pusan..ac.kr
1990년 이화여자대학교 컴퓨터학과(학사)
2004년 부산대학교 컴퓨터학과(공학석사)
2008년 부산대학교 컴퓨터공학과(박사)
2004년~현 재 (주)나라인포테크 근무
2005년~현 재 (주)나라인포테크 지능시스템 연구소 소장
관심분야 : 인간언어공학, 정보검색, 인공지능

권 혁 철

e-mail : hckwon@pusan..ac.kr
1982년 서울대학교 컴퓨터공학과(학사)
1984년 서울대학교 컴퓨터공학과 (공학석사)
1987년 서울대학교 컴퓨터공학과 (공학박사)
1992년~1993년 (미)Stanford 대학교 CSLI 방문 교수
1987년~현 재 부산대학교 정보컴퓨터공학부, 인지과학협동과정 교수
관심분야 : 인간언어공학, 정보검색, 인공지능