전체 문장 분석에 기반한 한국어 문법 검사기

(A Korean Grammar Checker based on the Trees Resulted from a Full Parser)

이 공 주 † 황 선 영 †† 김 지 은 †††
(Kong Joo Lee) (Sun Young Hwang) (Jee Eun Kim)

요 약 문법 검사기는 문장의 문법 오류를 찾고 이에 대한 적절한 대안을 제시하는 것이다. 문법 오류를 찾기 위해서 문법 검사기는 전체 문장을 분석해야 하며 이는 많은 자원이 소요되는 작업이다. 이러한 이유로 대부분의 한국어 문법 검사기는 중의성이 없는 작은 부분에 대해서만 구문 분석을 수행하는 부분 구문 분석기를 이용하고 있다. 본 논문의 구문 분석기는 문법 오류를 검사하기 위해서 전체 구문 분석기를 사용하였다. 이 방식은 여러 단어를 사이에 두고 떨어져 있는 두 단어 간에 문법적 오류가 있을 경우에도 이를 찾아서 고칠 수 있다. 결과적으로 이 방식은 수행 성능을 저하시키는 대신, 문법 오류를 수정하는 정확률의 향상을 기대할 수 있다. 본 논문의 문법 검사기는 문법 오류를 찾고 수정하기 위해서 65개의 규칙을 사용한다. 전체 구문 분석기를 사용하는 한국어 문법 검사기는 약 7백만 어절로 구성된 실험 코퍼스에 대해서 약 96.49%의 교정 정확률을 얻을 수 있었다.

키워드 : 문법 검사기, 전체 문장 분석

Abstract The purpose of a grammar checker is to find a grammatical erroneous expression in a sentence, and to provide appropriate suggestions for them. To find those errors, grammar checker should parse the whole input sentence, which is a highly time-consuming job. For this reason, most Korean grammar checkers adopt a partial parser that can analyze a fragment of a sentence without an ambiguity.

This paper presents a Korean grammar checker using a full parser in order to find grammatical errors. This approach allows the grammar checker to critique the errors between the two words in a long distance relationship within a sentence. As a result, this approach improves the accuracy in correcting errors, but it may come at the expense of decrease in its performance.

The Korean grammar checker described in this paper is implemented with 65 rules for checking and correcting the grammatical errors. The grammar checker shows 96.49% in checking accuracy against the test corpus including 7 million words.

Key words : Korean grammar checker, Full parsing

1. 서 론

문법 검사기는 한 어절 내에서의 맞춤법 오류를 찾아서 교정하던 기존의 맞춤법 검사기와 달리 어절과 어절 간의 상호 관계 속에서 일어나는 문법적 오류를 교정해 주는 프로그램이다. 그렇기 때문에, 여러 어절에 걸쳐서 나타나는 문법적 오류를 정확히 교정하기 위해서는 구문 분석을 수행해야 한다[1]. 구문 분석은 복잡도가 높으며 속도가 느리고 많은 기억공간을 요구한다. 이러한 문제들로 인하여 기존의 연구에서는 분석되는 어절 수를 제한하거나, 문장 분석을 종료하는 조건 등을 둔 부분 구문 분석 방법을 많이 채택하였다. 부분 구문 분석을 이용한 문법 검사기는 효율적이며 적은 노력으로 많은 문법 오류를 검사할 수 있다는 장점이 있다. 그러나 서로 대응되는 어절 사이에 다른 성분들이 나열되어 있거나, 수식어가 포함되거나, 또는 내포문이 첨가된 경우에는 대응되는 어절 간의 거리가 멀리 떨어지게 된다.

(1) "선생님은 우리 어머니가 매우 아름답다고 조심스럽게 말씀하셨습니다."

문장 (1)은 주어 '선생님은'과 그의 서술어 '말씀하셨

† 정 회 원 : 이화여자대학교 컴퓨터학과 교수
kjlee007@ewha.ac.kr
†† 비 회 원 : 연세대학교 국어정보학 협동과정
sunnyhw@hotmail.com
††† 비 회 원 : (주)한국마이크로소프트 연구원
jee_eun_k@hotmail.com
논문접수 : 2002년 12월 11일
심사완료 : 2003년 6월 24일

습니다' 사이에 높임 자질이 서로 맞지 않는 문법 오류를 갖고 있다. 그러나 주어와 해당 서술어 사이가 멀리 떨어져 있고, 그 사이에 내포문이 포함되어 있어서 부분 구문 분석만으로는 문법 오류를 찾아내기가 어렵다.

본 연구에서는 이와 같은 문제를 해결하기 위하여 부분 구문 분석이 아닌, 전체 구문 분석 방법을 사용하여 문법 오류를 찾아내고자 한다. 즉 입력 문장에 대해서 전체 구문 분석을 수행하여 적절한 구문 트리를 얻어내고, 그 구문 트리의 문법 성분 간에 문법 오류가 있을 경우, 이를 찾아내어 교정한다. 이와 같은 방법은 문법 검사기의 수행 속도를 낮추는 대신, 검사 및 교정 결과의 정확도 향상을 기대할 수 있다.

본 연구에서 구현한 문법 검사기는 수동으로 작성된 구문 규칙을 기반으로 한 범용의 구문 분석기를 이용한다. 구문 분석기로부터 얻어진 결과 구문 트리에서 문법 오류를 찾고 이를 교정하는 것 또한 수동으로 작성된 교정 규칙에 의해서 이루어진다. 구문 규칙과 교정 규칙은 모두 규칙 작성자가 한국어 문법을 바탕으로 기술하였다. 현재 구현된 문법 검사기는 59개의 구문 규칙과 65개의 교정 규칙을 사용하고 있다. 특히 교정 규칙은 어문 규정과 함께 중, 고등학교와 대학교 학생들의 작문을 통해 수집된 오류를 바탕으로 해서 구성되었다. 본 연구에서 구현한 문법 검사기는 마이크로소프트 Office XP의 Word에 제공되고 있다.

본 논문의 구성은 다음과 같다. 2절에서는 문법 검사기와 관련된 기존의 연구들을 간단히 살펴보고 3절에서는 문법 검사기의 시스템 구성을, 4절에서는 문법 검사기의 교정 규칙의 구성을 살펴볼 것이다. 5절의 실험 부분에서는 구현된 문법 검사기의 성능을 실제 코퍼스 적용을 통해 보이고 마지막으로 결론을 맺는다.

2. 관련 연구

한 어절 안의 맞춤법 오류를 교정하는 맞춤법 검사기에 대한 연구가 활발히 진행되어 여러 연구와 결과물들을 낸 데 비해서 여러 어절 간의 관계를 바탕으로 해서 문법적인 오류를 교정하는 문법 검사기에 대한 개발과 연구는 활발히 진행되지 않은 편이다. 현재는 부산대에서 발표된 문법 검사기에 대한 연구들[2-4]이 국내의 문법 검사기 개발과 연구를 선도하고 있다고 할 수 있다. [2]와 [3]에서 다루어진 문법 검사기를 간략히 살펴보면, 이 문법 검사기는 우리말의 오류를 어절 단위로 검증하는 철자 오류와 여러 어절을 분석해야 처리할 수 있는 문법 오류로 나누고 문법 오류를 처리하기 위해 부분 문장 분석 방법을 이용하였다. 부분 문장 분석은 검사 단어를 기준으로 한 의존 문법을 이용하였으며 이를 통해 연어 오류 단어를 찾아 교정한다. 그리고 부분 문장 분석시 발생할 수 있는 어휘나 의미적 중의성으로 인한 오류를 막기 위하여 어휘적 중의성 제거 규칙을 사용하고 있다. 이 어휘적 중의성 제거 규칙은 말뭉치 데이타에서 얻은 언어적 지식을 바탕으로 한 경험적 규칙에 기반하고 있다. [4]는 철자/문법 검사기를 웹 기반의 학습 시스템과 통합시킴으로써 학습자 입장에서의 수동적 교정에 머물지 않고, 학습자가 스스로 학습 결과를 평가하고 철자/문법 검사기를 통해 잘못된 내용을 수정, 학습해 나가도록 하는 능동적인 교정 시스템을 구축하고자 한 연구이다.

국내의 문법 검사기 연구에 비해 국외의 문법 검사기에 대한 연구는 활발히 이루어져 여러 시스템이 개발되어 사용되고 있다. 특히 마이크로소프트 Word의 영문 문법 검사기는 MS의 범용 자연언어 처리 시스템인 NLPwin[5]을 기반으로 구현되었다. 마이크로소프트 Word의 영문 문법 검사기는 주로 영어 문장의 수(number)의 일치, 수동태와 구두점 사용법 등을 검사한다.

본 연구에서 구현한 문법 검사기 또한 위에서 언급한 *NLPwin* 시스템을 바탕으로 구현되었다. *NLPwin* 시스템은 규칙 기반의 구문 분석을 채택하는 범용의 자연어 처리 시스템으로서 형태소 분석과 구문 분석, 의미 분석과 구문 생성 등 자연어 처리 단계 전반을 포함하는 시스템이다. 문법 검사기는 이 시스템의 구문 분석 결과를 받아서 구문에 오류가 존재하는지를 검사하여 교정을 제시한다.

3. 문법 검사기의 시스템 구성

문법 검사기는 크게 사전과 형태소 분석/생성기, 구문 분석기, 그리고 교정규칙 처리기로 구성된다. 문법 검사기는 입력문에 대해서 사전정보를 이용하여 형태소 분석을 수행하고(1), 구문 분석을 수행한다(2). 구문 분석 결과의 구문 트리에 대해서 교정규칙을 수행하여, 교정이 필요한 부분을 찾아내어, 이 부분에 대해서는 형태소 생성을 다시 수행한다(3,4). 원래의 입력에서 교정된 부

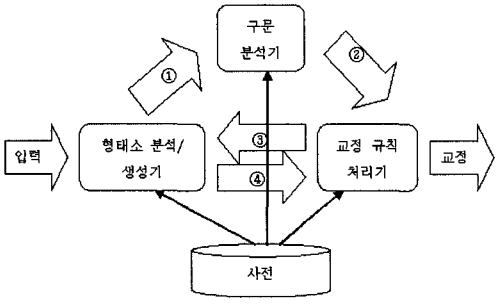

그림 1 문법 검사기의 시스템 구성

분을 교체하면 최종 교정된 문장을 얻을 수 있다(그림 1). 다음 절에서는 각 부분에 대해서 좀더 자세히 살펴보도록 하겠다.

3.1 사전 및 형태소 분석기

사전은 각 단계의 기반이 되는 어휘에 대한 정보를 저장한다. 그러므로 사전이 저장하고 있는 정보는 각각의 단계에서 요구하는 정보들로 구성된다. 즉, 형태소 분석과 생성의 단계에서 필요한 품사, 활용 정보와 구문 분석 단계에서 요구되는 통사 정보, 교정규칙 처리기에서 요구되는, 공기할 수 있는 어휘에 대한 정보 등이 사전에 저장된다. 그림 2는 사전에 저장되는 정보를 간략히 보인 예이다[1].

사업	밥	벌이다
{POS: NOUN	{POS: NOUN	{POS: VERB
Infl: NOUN-REG	Infl: NOUN-REG	Infl: VERB-REG
LI: VN SinoChar	PoliteWord: 진지 }	LI: T1 }
ConfusWord: 벌리다→벌이다}		

그림 2 사전 정보

그림 2를 보면, 교정규칙 처리를 위해서 사전에 추가된 항목으로는 *ConfusWord*와 *PoliteWord*라는 항목이 존재한다. *ConfusWord* 항목은 해당 단어와 함께 쓰이는 동사나 형용사가 자주 틀리는 표현일 때 올바른 표현이 무엇인가에 대한 정보를 담고 있다. 예를 들어 위의 그림 2에서 '사업'에 들어 있는 'ConfusWord: 벌리다→벌이다'라는 정보는 이 명사가 '벌이다'라는 동사와 함께 쓰일 때 '벌리다'로 잘못 쓰이는 경우가 많으며 '벌리다'와 함께 쓰였을 경우에는 '벌이다'로 교정해야 함을 뜻한다. *PoliteWord*는 높임 표현을 뜻하는 것으로 '밥'과 같은 단어가 'PoliteWord: 진지'라는 항목을 지니게 된다. *PoliteWord* 항목을 지닌 단어가 높임 표현을 사용해야 하는 문맥에서 잘못 사용되었을 경우, 교정규칙은 이 정보를 이용해서 *PoliteWord*에 있는 높임 표현으로 단어를 교체할 수 있다.

형태소 분석기에서는 사전 정보를 이용하여 입력에 대한 형태소 분석이 행해진다. 이 단계에서 행해진 분석의 결과는 사전의 정보와 함께 구문 분석 단계에 이용된다. 형태소 분석의 결과가 여러 개일 경우에도 여러 개의 분석 결과가 모두 구문 분석의 단계로 넘어가 구문 분석에 이용되게 된다.

1) LI (Lexical information)는 어휘에 대한 형태, 통사적 정보를 담는 항목이다. '사업'의 LI 항목에 들어가 있는 'VN'은 이 명사가 접미사 '하다'를 취해서 동사가 될 수 있다는 형태론적 정보를 표현하며 'SinoChar'는 이 명사가 한자어임을 뜻한다. 동사 '벌이다'에 들어 있는 정보 중, LI 항목에 들어 있는 'T1'은 이 동사가 타동사임을 뜻하는 것으로 구문 분석시에 이용되는 정보이다.

3.2 구문 분석기

구문 분석 단계에서는 형태소 분석 결과들을 받아서 전체 문장에 대한 구문분석을 행한다. 구문 분석에서는 문장 전체에 대한 구문 분석을 시도하며 문장 전체에 대한 분석이 성공했을 경우에 한해서 교정규칙이 적용된다. 특히 높임법이나 시제 일치와 같은 문장 내 성분들 간의 호응 관계를 이용한 교정 사항들은 전체 문장에 대한 정확한 분석을 전제로 하여 수행되게 된다.

구문 분석은 상향식 차트 파싱 방법을 사용한다. 구문 분석 단계에서는 59개의 한국어 문법을 이용한다. 대부분의 구문 규칙은 기본적으로 이진(binary) 규칙으로 구성되어 있으며 규칙 작성자에 의해 수동으로 기술되었다. 구문 분석의 결과 트리에는 주어, 술어, 목적어 등의 문장 성분과 문장 성분 간의 결합 관계가 표현된다. 그림 3은 구문 분석기가 문장에 대해서 분석 결과를 제시한 예이다. 각 노드 옆에 있는 별표(*)는 그 노드가 바로 위의 상위 노드의 중심어(head)임을 보여준다. 분석 결과 이 문장의 중심어(head)는 서술어 '읽지(VERB2)'로 분석되었으며 주어와 목적어는 각각 NP1과 NP2로 분석되었다.

구문 분석은 하나의 입력문에 대해서 여러 개의 분석 결과를 낼 수 있다. 이는 문장이 가지는 구조적 중의성이나 문장을 이루는 단어들의 사용 형태가 가지는 품사적 중의성 때문이다. 이렇게 하나의 문장에 대해서 여러 개의 구문 분석 결과가 가능할 경우에 최종적으로 하나의 구문 분석 결과를 얻기 위해서 구문 규칙 작성자는 구문 규칙 작성시에 규칙 작성자의 경험과 해당 규칙이 적용될 구의 문법적 특징에 기반한 점수를 각 규칙에 부여한다[2]. 그림 3의 두 트리에 표현된 Score가 이 점

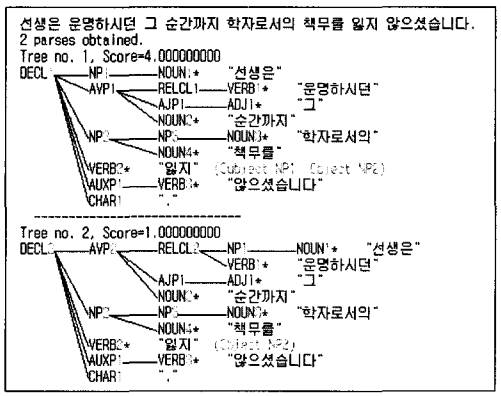

그림 3 구문 분석 결과

2) 예를 들어 특정 부사 부류가 동사와 형용사를 모두 수식할 수 있으나 형용사를 수식하는 경우가 동사를 수식하는 경우보다 월등히 많다면 구문 규칙 작성시에 이 부사 부류에 한해서 형용사를 수식하는 경우에 좀 더 높은 점수를 부여할 수 있다.

수에 해당되는 것으로 높은 점수를 부여받은 것이 최종적인 구문 트리로 얻어진다. 만약 여러 개의 분석 결과가 같은 점수를 가지게 되면 그 중 하나가 임의적으로 선택되어서 최종 구문 트리로 간주된다. 최종 구문 트리는 교정 규칙 처리기로 넘겨져서 교정규칙의 적용을 받게 된다.

3.3 교정 규칙 처리기

교정 규칙 처리기에서는 구문 분석에서 얻어진 전체 문장에 대한 분석 결과를 받아들여서 구문 교정을 행하게 된다. 구문 분석 결과로 만들어진 트리의 각 노드들은 구문 분석의 결과뿐만 아니라 형태소 분석의 결과와 어휘에 대한 사전 정보까지를 담고 있다. 교정 규칙 처리기는 트리의 각 노드들을 차례로 돌면서 각 노드에 표현되어 있는 이러한 정보들을 검사하여 교정 규칙의 조건에 맞을 경우 교정 규칙을 적용하게 된다. 교정 규칙이 처리되는 과정을 보이면 그림 4와 같다.

개개의 교정 규칙은 입력이 교정 대상에 속하는지를 판단하는 조건부(Condition)와 오류에 대한 교정을 제안하는 교정부(Action)로 나뉜다. 교정부에서는 교정 어절(Suggested_word)을 제안하여 형태소 생성기에 넘긴다. 형태소 생성기에서 교정 어절을 생성하여 다시 교정 규칙 처리기로 넘겨주면 교정 규칙 처리기는 교정 대상이 되는 입력 어절을 교정 어절로 교체한다. 그림 5는 교정 규칙의 구성을 간단히 보인 그림이다.

아래의 그림 6은 실제 교정 규칙의 예로서, 주어와 서술어의 높임 자질이 일치하지 않는 오류를 교정하는 교정 규칙이다.

그림 6의 교정 규칙이 그림 3에 나온 예문에 적용되는 과정을 살펴보자. 그림 3의 예문의 경우, 문장 전체

```
tree_list = tree2list(best_result_tree)
                    //depth first 방식을 취하여 각 node 들을 list 로 만듦.
foreach node A in tree_list
{                   // list 에 있는 각 node 들에 대하여
        foreach rule R in grammar_checker_rules
        {           // node A 에 문법 검사기의 교정 규칙(R)들을 차례로 적용
            Suggested_sent = R(A);  // 교정된 문장을 얻어냄.
        }
}
```

그림 4 교정 규칙이 처리되는 과정

```
조건부: if(node A has error_type1){      // node A가 type1의 오류를 가지는지 검사
교정부:         Checked_word = A;        // A를 교정 대상 표현으로 간주
                B = Suggest_to_correct(A);
                                         // A에 대한 교정 표현 B를 만들 것을 제안
                Suggested_word = morph_generation(B); // 교정 표현을 생성
                replace(Checked_word, Suggested_word);
                                         // 생성된 교정 표현으로 대치
    }
```

그림 5 교정 규칙의 구성

를 포함하는 노드인 DECL1의 서술어와 주어의 높임 자질이 일치되지 않아 높임 자질 불일치 교정 규칙의 조건에 맞게 되어 높임 자질 일치 교정이 일어난다. 이를 그림 7에서 표현된 것과 같이 과정별로 살펴보면, 형태소 분석 단계를 거치면, AUXP1에 들어 있는 높임의 선어말어미 '-시-'에 대한 정보가 형태소 분석 결과에 반영되게 되며, NP1의 형태소 분석 결과에 높임의 주격 조사가 없다는 점이 반영된다(1). 구문 분석 단계에서는 구문 분석 결과에 NP1이 AUXP1을 포함한 서술어의 주어임이 반영된다(2). 교정 규칙 처리기의 조건부에서는 형태소 분석과 구문 분석의 이러한 결과들을 이용하여 주어(NP1)와 서술어(VERB2 + AUXP1)의 높임의 자질이 일치하지 않는 높임의 오류가 DECL1에 존재함을 찾아내어 교정 규칙을 적용하게 된다. DECL1에 그림 6의 교정 규칙이 적용되면 교정부에서 높임의 자질

```
if(Subject_of_node_A has not Honorific and
  Predicate_of_node_A has Honorific) {
        B = add Honorific to Subject_of_node_A; // 주어에 Honorific 첨가
        C = remove Honorific from Predicate_of_node_A;
                                  // 서술어에서 Honorific 제거
        Suggested_Subject = morph_generation(B);
        Suggested_Predicate = morph_generation(C);
        replace(Subject_of_node_A, Suggested_Subject);
        replace(Predicate_of_node_A, Suggested_Predicate);
}
```

그림 6 높임 자질 불일치 교정 규칙의 예

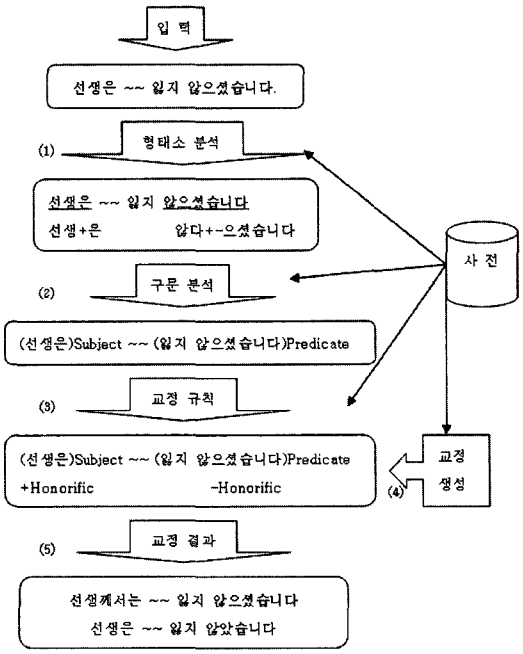

그림 7 교정 규칙의 적용 과정

이 없는 주어, '선생은'은 높임의 자질을 지닌 '선생께서는'으로, 높임의 자질이 있는 술어, '잃지 않으셨습니다'는 높임의 자질이 없는 '잃지 않았습니다'로 교체되도록 제안된다(3). 교정 규칙 처리기에서 제안한 교정이 실제로 이루어지기 위해서는 입력에 부여된 새로운 자질을 이용한 형태소 생성이 필요하다. 형태소 생성기에서는 이러한 생성을 처리하여 생성된 교정 표현을 다시 교정 규칙 처리기로 보낸다(4). 교정 규칙 처리기는 교정 대상이 되는 표현을 생성된 교정 표현으로 대체함으로써 교정 처리를 완료한다(5). 결과적으로 위의 예문이 교정 규칙 처리기를 통과하면 '선생께서는 ~~ 잃지 않으셨습니다', '선생은 ~~ 잃지 않았습니다'와 같은 두 가지의 가능한 교정이 이루어진다.

4. 교정 오류의 유형과 교정 방법

이 절에서는 본 논문의 문법 검사기가 다루고 있는 교정 오류의 유형에 대해서 살펴보도록 한다. 문법 검사기가 행하는 교정은 어문 규정에 제시되어 있는 혼동하기 쉬운 표현들과 중, 고등학교와 대학교 학생들의 작문에서 발견되는 오류들을 바탕으로 하여 구성되었다.

4.1 높임 자질의 불일치

높임 자질이 일치하지 않았을 때 행하는 교정은 (1) 주어와 서술어의 높임의 자질이 일치하지 않을 경우, (2) 서술어와 간접 목적어의 높임의 자질이 일치하지 않을 경우, (3) 서술어와 그에 따른 목적어의 높임의 자질이 일치하지 않을 경우로 나뉠 수 있다. 교정 규칙은 (1), (2)에 대해서는 각 문장 성분 간의 높임의 자질을 비교하여 높임의 자질이 일치하지 않을 경우, 높임 자질이 없는 문장 성분에 높임 자질을 더하거나 높임 자질이 있는 문장 성분에서 높임 자질을 빼는 방식으로 교정을 제안한다. (3)에 대해서는 높임 자질을 비교해서 높임의 자질이 일치하지 않을 경우, 높임의 자질을 가진 단어로 목적어를 교체하도록 제안한다. 이 때에는 목적어로 쓰인 단어의 사전 정보에 *PoliteWord*가 저장되어 있어야 한다. 그림 8은 서술어와 목적어의 높임의 자질이 일치하지 않을 경우에 대한 처리 예제이다.

```
사전
밥: {NOUN; NOUN-REG; PoliteWord 진지}
이름: {NOUN; NOUN-REG; PoliteWord 성함}
.....

교정규칙
if(Object is 밥 and
  Predicate is one of {들다 잡수다 ..... } and
  Predicate has Honorific) {
        Suggested_Object = PoliteWord(밥); // PoliteWord(밥)=진지
        .....
        morph_generation(Suggested_Object);
        replace(Object, Suggested_Object);
}

예
밥을 드시고 계신다 → 진지를 드시고 계신다.
```

그림 8 서술어와 목적어의 높임 자질 불일치에 대한 교정 규칙

표 1에서는 본 절에서 다루고 있는 높임 자질 불일치에 관한 교정 규칙을 정리해 보았다.

4.2 어미에 따른 주어 표현 제약

어미에 따른 주어 표현 제약을 교정하는 규칙은 서술어의 어미가 특정한 서법을 표현할 경우 그에 따른 주어 표현의 오류를 교정하는 규칙이다. 이 교정 규칙은 (1) 서술어의 어미가 청유형일 경우에는 주어에 낮춤의 자질을 가진 '저희'와 같은 표현이 사용될 수 없으며 (2) 서술어의 어미가 약속을 표현할 경우에는 주어가 반드시 1인칭이어야 한다는 문법에 기반한다[6]. 이 교정은 주어와 서술어의 관계에 기반하므로 문장 전체에 대한 정확한 구문 분석이 선행되어야 한다.

4.3 부사의 쓰임

부사의 쓰임에 대한 교정은 부사와 서술어 간의 호응이 제대로 일치하지 않을 경우에 행하는 교정이다. 이에는 (1) 서술어에 표현된 시제가 부사에 표현된 시제와

표 1 높임의 자질의 불일치에 관한 교정 규칙

교정 내용	예
(1) 주어와 서술어의 높임 자질 일치	아버지가 오셨다. → 아버지께서 오셨다/아버지가 왔다
(2) 간접 목적어와 서술어의 높임 자질 일치	선생에게 드렸다. → 선생께 드렸다/선생에게 줬다
(3) 목적어와 서술어의 높임 자질 일치	밥을 드셨다. → 진지를 드셨다 이름을 밝히셨다. → 성함을 밝히셨다 나이를 잡수셨다. → 연세를 잡수셨다

표 2 어미에 따른 주어 표현 제약에 관한 교정 규칙

교정 내용	예
(1) 청유형 어미와 주어	저희가 힘을 합칩시다. → 우리가 힘을 합칩시다.
(2) 약속의 어미와 주어	네가 밥을 지으마. → 내가 밥을 지으마.

표 3 부사의 쓰임에 관한 교정 규칙

교정 내용	예
(1) 부사의 시제와 서술어의 시제 일치	아까 오겠다. → 아까 왔다.
(2) 부정 부사와 서술어의 부정 표현 일치	결코 좋았다. → 결코 좋지 않았다.

표 4 자주 틀리는 표현에 관한 교정 규칙

교정 내용	예
다리다/달이다	한약을 다리고 있다. → 한약을 달이고 있다. 교복을 달이고 있다. → 교복을 다리고 있다.
부치다/붙이다	이 일은 불문에 붙이기로 한다. → 이 일은 불문에 부치기로 한다.
다르다/틀리다	나는 그와 의견이 틀렸다. → 나는 그와 의견이 달랐다.
두텁다/두껍다	이불이 너무 두터운 거 아니냐? → 이불이 너무 두꺼운 거 아니냐?
빚/빗	빗을 다 갚고 나서 얘기하자. → 빚을 다 갚고 얘기하자.

일치하지 않을 경우 서술어의 시제를 교정해주는 규칙과 (2) 부정 부사가 사용되었을 때 서술어에 부정 표현을 제안해 주는 교정이 포함된다.

4.4 자주 틀리는 표현

자주 틀리는 표현은 단어의 모양과 발음이 비슷하여 사람들이 자주 혼동하여 쓰는 단어들에 대한 교정 유형이다. 예를 들어 '띄다/띠다'는 흔히 잘못 쓰는 표현 중에 하나인데 '띄다'는 '간격을 벌리다'라는 의미로 사용하여야 하며 '띠다'는 '어떠한 것을 지니다'라는 의미를 표현할 때 사용해야 한다. '띠다/띄다'는 둘 다 맞춤법 상으로는 옳은 표현들이기 때문에 맞춤법 검사기로는 교정할 수 없으며 목적어나 주어로 무엇을 취했는가를 살펴서 문맥에 따라 교정해 주어야 한다. 즉, '간격', '줄', '칸'과 같은 단어가 '띠다'의 목적어로 올 경우에 한해서 '띠다'를 '띄다'로 교정해야 하며 '미소', '웃음기', '허리띠', '빛깔'과 같은 단어가 '띄다'의 목적어로 올 경우에 한해서 '띄다'를 '띠다'로 교정해야 한다.

자주 틀리는 표현들을 교정하기 위해서는 우선 '간격', '줄', '칸', '미소', '웃음기' 등의 특정 단어들의 사전 정보에 *ConfusWord*가 저장되어 있어야 한다. 또한 구문 분석 결과에 목적어와 술어, 주어와 술어 등 문장 성분 분석이 이루어져 있어야 한다. 교정 규칙은 사전 정보와 구문 분석의 결과를 참조하여 문맥에 따라서 잘못 사용된 표현을 교정한다. 문법 검사기에 사용된 자주 틀리는 표현들에 대한 교정 규칙은 57개로서 여기에는 표 4와 같이 동사, 형용사, 명사들로 구성된 규칙들이 존재한다. 아래 목록은 자주 틀리는 표현 오류에 해당하는 규칙들의 예이다.

4.5 띄어쓰기

문법 검사기는 품사적 중의성이 있는 어절의 띄어쓰기에 오류가 있을 경우 이를 교정한다. 띄어 쓴 어절의 붙여 쓰기 교정은 앞뒤의 어절을 참조하는 것만으로도 충분히 해결할 수 있으므로 굳이 구문 분석을 이용하지 않고 맞춤법 검사기에서도 처리할 수 있다. 그러나, 구문 분석 결과를 이용한 문법 검사기의 붙여 쓰기는 품사적 중의성이 있는 단어에 대해선 구문 전체에 대한 분석 결과를 참조해서 중의성을 해소하고 이를 이용해 붙여 쓰기 교정을 함으로써 좀 더 정확한, 그리고 폭 넓은 교정을 행할 수 있다는 장점을 지닐 수 있다. 예를 들어 '사실'과 같은 형태는 '밝혀진 *사실* 만큼이나 중요하다'와 같은 문장에서는 명사로 쓰였으므로 뒤에 오는 '만큼'을 조사로 파악하여 붙여 써야 하는 반면, '여기에 *사실* 만큼 부자입니다'와 같은 문장에서는 동사의 활용형이므로 뒤에 오는 '만큼'을 의존 명사로 파악하여 띄어 써야 한다. 앞에 오는 어절에 대한 품사 중의성 해소는 이렇듯 구문 분석의 결과를 참조하여 이루어지며 이를 통해 정확한 붙여 쓰기 교정이 가능해진다.

5. 구현 및 실험

이제까지 살펴본 문법 검사기는 마이크로소프트의 Word에 포함되어서 사용자들이 Word에 입력하는 문장을 입력문으로 받아서 교정을 행하게 된다. 문서 작성시의 교정 프로그램이라고 할 수 있는 한국어 문법 검사기는 맞춤법 검사기와 함께 2001년 출시된 마이크로소프트 Office XP의 Word에 제공되고 있다. 또한 기능을 좀 더 강화한 문법 검사기가 2003년에 출시될 다음 버전의 Office에 제공될 예정이다. 문법 검사기가 실제로 마이크로소프트 Word의 입력문을 처리해서 내 놓는 결과는 그림 9와 같다.

본 연구에서는 구현된 문법 검사기의 성능을 실험하기 위해서 국립국어연구원에서 21세기 세종 계획의 결과물로 2001년도에 배포한 약 700만 어절의 말뭉치를 실험 데이타로 이용하였다. 실험 데이타의 구성은 표 5와 같다.

문법 검사기가 행한 교정이 옳은지를 의미하는 교정 정확률을 측정하기 위해서 실험 데이타의 교정된 문장

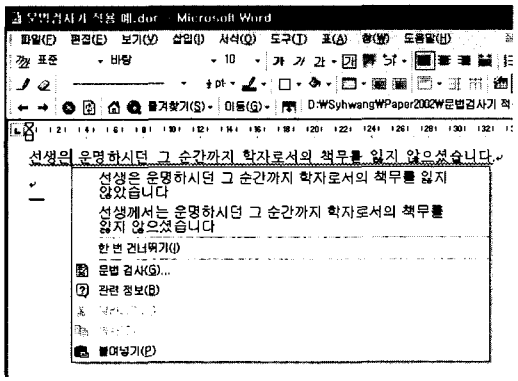

그림 9 문법 검사기 적용의 예

표 5 실험 대상 말뭉치의 구성

실험 데이타	문장 수	어절 수	문장 당 어절 수
	589,874 문장	6,901,526 어절	11.7 어절

표 6 교정 정확률

교정된 문장	옳은 교정	틀린 교정	교정 정확률
2,109 문장	2,035 문장	74 문장	96.49%

중에서 옳은 교정의 비율을 산출하였다. 표 6은 실험 데이타에서 행해진 전체 교정에 대한 교정 정확률이다. 교정된 문장의 평균 어절 수는 14.92 어절이며 가장 짧은 문장은 2 어절로 이루어졌으며 가장 긴 문장은 118 어절로 이루어졌다.

실험 결과가 보여 주듯, 문법 검사기는 교정 정확률에 있어서 높은 정확도를 보여 주고 있다. 이렇듯 높은 정확률은 몇 가지 사실을 바탕으로 하는 것으로 보인다.

우선 이는 문법 검사기가 Word에 포함된 상업적 제품으로 개발되었다는 사실에 기인한다. 마이크로소프트의 문법 검사기는 상품의 성격을 지니기 때문에 가능한 한 사용자의 불편을 최소화하면서 정확한 교정을 제공해 주는 것을 목적으로 한다. 이를 위해서 문법 검사기는 문법적으로 틀린 것이 확실한 경우에 한해서 정확한 교정을 제시하도록 디자인되었다. 두 번째로 이는 문법 검사기가 문장 전체에 대한 분석에 기반하여 교정을 행한다는 점에 기인한다. 전체 문장 분석에 기반을 둔 문법 교정은 문장 전체에 대한 분석이 성공할 경우에 한해서 교정을 행하기 때문에 높은 교정 정확률을 보인다는 장점을 지니고 있다. 이 외에 실험 말뭉치가 몇 번의 교정 작업을 거친 말뭉치라는 점이 높은 정확률의 한 요인으로 지적될 수 있다. 국립국어연구원에서 배포한 말뭉치는 수작업을 통한 몇 번의 교정 과정을 거친 말뭉치로서 오류가 거의 없는 양질의 말뭉치로 간주되고 있다. 따라서 상대적으로 전체 구문 분석이 쉽게 이루어져 높은 교정 정확률을 보이게 되는 것이다.

표 7은 교정된 문장들을 교정 유형 별로 나누어 산출한 교정 정확률이다.

교정 유형 별 실험 결과에서 나타나듯이 틀린 교정은 주로 구문 분석이 어려운 교정 유형에서 나타났다. 교정 정확률이 낮은 '부사의 쓰임'은 우리말의 '부사'가 문장에서 자유로운 위치 이동을 보인다는 점 때문에 정확한 구문 분석이 어렵다는 문제점을 지닌다. '어미에 따른 주어 표현 제약'에서도 틀린 교정은 주로 우리말의 주어가 쉽게 생략된다는 점 때문에 문장 성분 간의 결합관계가 잘못 분석되어서 일어 났다. 구문 분석이 상대적으로 어려운 두 유형을 빼 놓고 다른 유형의 교정 정확률은 전체 문장의 구문 분석에 기반하여 높은 교정 정확률을 보였다. 그러므로 교정 정확률을 높이기 위해서는 부사의 유형 분류와 생략된 요소의 처리를 통해서 구문 분석의 정확도를 높이는 작업이 이루어져야 할 것이다.

6. 결론 및 남은 과제

본 연구에서는 이제까지 전체 문장에 대한 구문 분석을 바탕으로 한 문법 검사기의 시스템 구성과 교정 규칙 등을 소개하고 그 성능을 평가하였다. 전체 문장에 대한 구문 분석을 바탕으로 한 문법 검사기는 정확한 교정을 행할 수 있다는 장점을 지녔다. 그러나 한 편으로는 전체 문장에 대한 구문 분석의 결과가 오분석일 경우, 그에 따라 교정 정확률이 낮아 질 수 있다는 한계 역시 지녔다. 이러한 한계점을 해결하기 위해서는 각 범주별 어휘 분류와 이를 구문 규칙에 반영하는 작업 등을 통해서 전체 문장에 대한 구문 분석의 정확도를 향상시키는 노력이 선행되어야 할 것이다. 더불어서 교정 대상이 되는 어휘들에 대한 광범위한 수집과 이를 사전

표 7 교정 유형 별 교정 정확률

교정 유형(교정 규칙의 수)	교정된 문장	옳은 교정	틀린 교정	교정 정확률
높임 자질의 불일치(3)	1127	1088	39	96.53%
어미에 따른 주어 표현 제약(2)	12	9	3	75.00%
부사의 쓰임(2)	22	16	6	62.72%
자주 틀리는 표현(57)	549	529	20	96.35%
띄어쓰기(1)	399	393	6	98.49%

에 충실히 반영하는 작업이 이루어져야 할 것이다.

참 고 문 헌

[1] 채영숙, 연어 규칙에 기반한 한국어 문서 교정 시스템의 구현, 부산대학교 박사학위 논문, 1998.
[2] 소길자, 권혁철, 어휘적 중의성 제거 규칙과 부분 문장 분석을 이용한 한국어 문법 검사기, 정보과학회 논문지, 제23권 제3호, p. 305-315, 2001.
[3] 한국전자통신연구원, 구문 분석을 이용한 한국어 문법/문체 검사기의 구현, 한국전자통신연구원 연구 보고서, p. 21-47. 1999.
[4] 남현숙, 김상훈, 김지원, 권현주, 정유진, 권혁철, 한국어 철자/문법 검사기와 웹 기반 언어 학습 시스템의 통합 환경 구축, 한국인지과학회 춘계 학술대회, pp. 37-40, 2000.
[5] MS Internal Documents: *NLPwin* Overview
[6] 고영근, 남기심, 표준국어문법론, 탑출판사, 1985.

이 공 주
1992년 서강대학교 전자계산학과(학사) 1994년 한국과학기술원 전산학과(공학석사). 1998년 한국과학기술원 전산학과(공학박사). 1998년~2003년 (주)한국마이크로소프트 연구원. 2003년~현재 이화여자대학교 컴퓨터학과 대우전임강사. 관심분야는 자연언어처리, 자연어 인터페이스, 기계번역, 정보검색

황 선 영
1996년 연세대학교 국문학과(학사). 1998년 연세대학교 국문학과(문학석사). 1999년~2002년 (주)한국마이크로소프트 연구원. 1999년~현재 연세대학교 국어정보학 협동과정. 관심분야는 자연언어처리, 한국어 코퍼스 구축

김 지 은
1985년 외국어대학교 영어과(학사). 1987년 조지타운 Univ. 언어학과(MS). 1993년 조지타운 Univ. 언어학과(Phd). 1995년~2003년 (주)한국마이크로소프트 연구원. 관심분야는 한국어정보처리, 기계번역