규칙 기반 학습에 의한 한국어의 기반 명사구 인식

(Base Noun Phrase Recognition in Korean using Rule-based Learning)

양 재 형 †

(Jaehyung Yang)

요 약 한국어의 기반 명사구, 즉 비재귀적인 단순 명사구를 인식하는 비통계적인 규칙 기반 학습 기법을 제안한다. 학습 말뭉치에 기반 명사구에 대한 초기 예측이 표시되어 있고 목표 말뭉치에는 올바른 기반 명사구가 태그(tag)의 형식으로 표시되어 있다면, 규칙 기반 학습은 먼저 인접한 주위 형태소들의 다양한 문법적 정보를 나타내는 규칙 템플릿을 이용하여 기반 명사구 태그를 수정하는 규칙 후보들을 생성해 내고, 이 후보들 가운데 학습 말뭉치를 목표 말뭉치에 가장 가깝게 변환하는 일련의 규칙들을 차례로 얻어낸다. 국어정보베이스의 15만 단어 규모의 트리 태그 부착 말뭉치를 이용한 실험 결과 386개의 변환 규칙을 얻었으며, 이를 이용하여 90% 이상의 높은 기반 명사구 인식 정확도를 얻을 수 있었다.

Abstract A rule-based non-statistical learning technique for recognizing base NPs (noun phrases) in Korean sentences is proposed. Base NP is defined as a simple, non-recursive noun phrase. The sentences in training corpus are annotated with possibly incorrect initial base NP marks. Target corpus consists of the same sentences annotated with correct base NP marks. Rule-based learning is a non-statistical procedure which obtains a sequence of ordered rules that transform training corpus to target corpus as close as possible. In experiment, the proposed algorithm acquired 386 transformation rules from the tree-tagged Korean corpus of 150,000 words and showed accuracy of over 90%.

1. 서 론

각 문장에 대해 하나의 완전한 분석 구조를 얻는 전통적인 전체 파싱(full parsing) 방법에 근거한 자연언어 파서는, 아주 제약된 영역을 제외하고는 아직도 만족할 만한 성능과 강건성(robustness)을 보이지 못하고 있다. 이러한 어려움을 타개하기 위해 최근에는 얕은 파싱(shallow parsing) 혹은 부분 파싱(partial parsing)이라는 새로운 접근법이 제시되고 있다[1,2,3]. 얕은 파싱은 하나의 문장을 서로 겹치지 않는 일련의 청크(chunk)들로 쪼개는 것을 목표로 한다. 여기서 청크[1]는 몇 개의 연속된 단어들이 모인 의미 있는 연속체를 말하며, 얕은 파싱에서는 문장을 이러한 청크 단위로 쪼개는 것, 혹은 이에 더하여 이 청크들 간의 간단한 문법적 관계를 파악하는 정도를 파싱의 목표로 삼고 있다. 청크는 대략 완전한 파스(full parse)의 한 구절 혹은 구절의 일부에 해당하지만, 반드시 일치하는 것은 아니다. 얕은 파싱의 결과는 응용 목적에 따라 다양한 형태를 가질 수 있는 것이지만, 가능한 한 가지 예를 보이면 (1)과 같다[2].

(1) [SBJ [NP South Korea]] [V registered] [OBJ [NP a trade deficit]] of [NP $101 billion] in [NP October], [V reflecting] [NP the country]'s [OBJ [NP economic sluggishness]].

이 예에서는 문장을 대략 명사형 청크와 동사형 청크로 나누고, 주어, 목적어 등의 문법 관계를 파악하는 것을 얕은 파싱의 역할로 보고 있다. 이러한 청크의 파악이 문장 전체의 완전한 분석 구조를 생성하지는 못하지만, 전체 파싱의 전단계의 역할을 할 수 있을 뿐만 아니라 그 자체로도 정보 추출(information extraction)[3] 이나 기타 대규모의 자연언어 처리 응용에 유용하게 사

† 종신회원 : 상남대학교 지식정보공학부 교수
jhyang@kns.kangnam.ac.kr
논문접수 : 1999년 9월 8일
심사완료 : 2000년 9월 20일

용될 수 있다. 또 대규모의 제약되지 않은 텍스트로부터 사전 정보를 자동 습득하려는 사전 습득 시스템의 중요한 구성 요소가 되기도 한다[4, 5].

지금까지 많이 연구되어 온 청크는 기반 명사구(base noun phrase)인데, 기반 명사구는 비재귀적인(non-recursive) 명사구를 말한다. 즉, 그 구성 요소로 다른 명사구를 포함하고 있지 않은 단순한 명사구를 기반 명사구라 한다. 기반 명사구의 인식은 흔히 얕은 파싱의 첫 단계에 해당한다. 한국어 예문 (2) - (4)에 기반 명사구들을 괄호로 표시하였다.[1]

(2) [1930 년대]+에는 [형식주의자들]+의 [공격]+이 [극도]+에 달하였으며, [자유주의적 마르크스주의 비평가들]+에 의해 [전혀 다른 방면]+으로부터 [공격]+을 받았다.

(3) [지정 좌석]+도 없이 [아무 데]+나 [빈 자리]+를 찾아 앉는 [바람]+에 [조사단]+은 뿔뿔이 흩어질 수밖에 없었다.

(4) 그래서 [칭기즈칸]+은 [뛰어난 기마 기술]+로 [인류 역사상 최대]+의 [제국]+을 이루었다.

본 연구는 한국어의 기반 명사구 인식을 위해, 규칙의 자동 학습에 근거한 접근 방안을 제안한다. 이 기법은 말뭉치 기반의 학습 방법이지만, 학습된 정보가 수많은 확률 정보로 표현되는 통계적 기법과는 달리, 규칙 기반의 접근법을 택하여 학습 과정 및 결과를 이해하기 용이하다는 장점을 갖는다. 국어정보베이스의 구문 태그 부착 말뭉치에 대해 실험한 결과를 보인다.

2. 관련 연구

청크의 인식에 의한 얕은 파싱의 개념은 [1]에서 처음 제안되었다. 여기서 청크를 특정한 통사적 기능을 갖는 단어들의 연속체로 보았는데, 하나의 내용어(content word)와 이를 둘러싼 몇 개의 기능어(function words)들로 구성되는 패턴으로 보았다.

[3, 7] 등에서는 여러 단계의 유한 상태 오토마타(finite state automata)가 직렬로 연결되어 부분 파싱을 수행한다. 즉, 파싱은 여러 단계로 이루어지는데 앞 단계에서 명칭이나 간단한 명사구가 인식되고 뒷 단계로 가면서 보다 복잡한 명사 그룹, 동사 그룹 등이 인식되는 방식이다.

[8]의 시스템은 간단한 확률 모델에 근거하여 품사 태그 부착뿐만 아니라 기반 명사구를 인식하는 프로그램인데, Brown 말뭉치를 사용하여 '고무적인'(encouraging) 결과를 얻을 수 있었다고 한다. 또 [9]의 NPtool 시스템은 수동으로 구축된 문법 규칙을 이용하여 명사구를 추출하였는데, 95% 이상의 놀랄만큼 높은 정확도가 주장되었다. 그러나 [10]에서 지적한 대로 위 연구의 부록에 첨부된 예를 검토해 보면 NPtool에서의 명사구 청크의 정의가 다소 느슨하거나 부정확한 것으로 보인다.

[11]은 메모리 기반 학습(memory-based learning paradigm) 방법에 의한 얕은 파싱을 제안하고 있다. 이 기법에서는 학습용 말뭉치로부터 필요한 언어 정보를 미리 추출해 두는 것이 아니라, 학습용 예제들을 형태의 변환 없이 그대로 트라이(trie)의 형태로 저장하여 두는데 이를 메모리라고 부른다. 어떤 입력이 특정 패턴에 맞는지를 판단할 때 그 입력이 메모리에 저장된 학습용 말뭉치에 나타났는지 조사하여 긍정적인 증거와 부정적인 증거를 바탕으로 판단하게 된다. Penn treebank를 사용한 NP 인식 실험에서 91.6%의 정확도 및 재현율을 보고하고 있다.

[12, 13, 2]에서도 메모리 기반 기법과 유사한 방식의 학습 방법에 의해 기반 명사구를 인식하는 방안이 제시되고 있다. 먼저 treebank의 NP 예제들로부터 품사 태그열로 이루어지는 NP 문법 규칙들을 뽑아 낸 후, 이를 학습용 말뭉치의 긍정적/부정적 증거들에 의해 가지치기(pruning)해 감으로써 일련의 순서화된 문법 규칙을 얻는 방식이다. 90.7%의 정확도 및 91.1%의 재현율을 얻었다고 한다.

[14]에서는 품사 태그가 부착된 단어들에 부분적 구문 구조를 할당하기 위해 최대 엔트로피(maximum-entropy) 모형에 기반한 통계적 기법을 적용하고 있는데, 독일어 텍스트에 대해 NP 및 PP 청크 분할 실험에서 87.6%의 정확도 및 88.9%의 재현율이 나타났다고 한다.

SPARKLE 프로젝트[4, 5][2]는 유럽과 같은 다국어 정보 환경에서 상용화가 가능한 강건하고 이식성 있는 언어 도구를 개발하려는 목표를 가지고 있는데, 일차적으로 영어, 불어, 독일어, 이탈리어어에 대한 얕은 파서의 개발이 독립적으로 진행되고 있다. 각 파서는 서로 다른 접근법을 취하고 있는데, 예를 들면 독일어에 대해서는 HMM 기반의 청크 파싱이, 불어에 대해서는 유한 자동 기법이, 영어에 대해서는 구절 규칙에 의한 LR 파

1) 예문은 본 연구의 실험에 사용한 국어정보베이스[6]의 구문 태그 부착 말뭉치에 포함된 문장들이다.

2) SPARKLE 프로젝트 홈페이지(http://www.ilc.pi.cnr.it/sparkle.html) 참조.

서와 확률적 모호성 해소를 통합한 방식 등이 적용되고 있다고 한다. 아직 정확한 성능은 보고되지 않고 있다.

[15]는 규칙 기반의 비통계적 학습 기법의 일종인 변형 기반 학습(transformation-based learning) 방법을 제시하고 이를 품사 태그부착 문제에 적용한 예를 보였다. 이 기법은 말뭉치로부터 언어 지식을 자동 습득한다는 점에서 최근에 각광을 받고 있는 경험적 접근법들의 일종으로 볼 수 있지만, 통계적 기법들과는 달리 규칙 기반의 접근을 취하고 있다. 따라서 학습에 의해 얻어지는 결과는 통계값들이 아니라 일련의 규칙들이 된다. 태그부착 문제에 이 기법을 적용하면, 먼저 간단한 태그 할당기가 학습 말뭉치 내의 각 단어에 임의의 태그를 할당한 후, 태그가 잘못 할당된 부분에서 이 태그를 올바르게 교정하는 규칙들을 만들어 내는데, 전체 말뭉치에 대해 태그 할당 정확도를 많이 개선시키는 순서에 따라 일련의 규칙들을 얻게 된다.

[10]는 Penn treebank를 이용하여 영어의 청크 분할 문제에 [15]의 기법을 적용하였다. 오류를 교정하는 규칙을 체계적으로 생성하기 위해 규칙 템플릿(rule template)의 개념을 도입하였으며 100개의 규칙 템플릿을 사용한 실험 결과를 제시하고 있다. 영어의 명사 청크 분할 문제에서 91.8%의 정확도 및 92.3%의 재현율을 보였고, 명사 및 동사 청크 분할 문제에서 87.7%의 정확도 및 88.5%의 재현율을 나타내었다.

한국어 파싱에 말뭉치 기반의 접근법을 채택한 기존의 연구 결과들[16, 17]은 실험 대상이 수백 문장에 불과하여 정확한 평가가 어렵다. 예를 들어 [16]은 확률 모델에 근거한 한국어의 의존 구조 파싱에서 85.7%의 정확도를 보고하고 있는데, 실험 문장이 소규모인 점을 고려하며 영어에 대한 연구 결과[18, 19]를 크게 벗어나지 않고 있다고 판단된다.

3. 한국어의 기반 명사구 인식

본 논문에서는 [15]에서 제안되고 [10]에서 영어에 대해 적용된 규칙 기반 학습 기법을 한국어의 기반 명사구 인식 문제에 적용하였다.

3.1 문제의 표현

기반 명사구의 인식은 형태소 태그 부착 후에 이루어지며, 형태소 태그 정보를 이용한다. 본 연구에서는 (5)의 형태소 태그들을 사용하였는데, 이들은 [20]의 태그셋(tagset)에 기반을 두고 있다.

(5) AD : 부사
AJ : 형용사
CP : 서술격 조사 '이'
DT : 관형사
EM : 어말 어미
IJ : 감탄사
JO : 조사
ND : '명사+적' eg) 일반적, 효과적
NN : 보통명사
NP : 대명사
NQ : 고유명사
NU : 수사
NX : 의존명사
SE : 마침표
SY : 기호
VV : 동사
VX : 보조용언

본 연구에서는 파생된 형태만 고려하였으므로 "고려+하+는"은 기반 명사구 인식 단계 이전에 "고려하/VV+는/EM"으로 바꾸었다. 파생적 용법이 아닌 접사들과 선어말 어미는 본 연구와 관련한 문법적 영향이 미미할 것으로 판단하여 무시하였다. 따라서 (5)에는 선어말어미(EP), 형용사 파생접미사(SJ), 명사 파생접미사(SN), 등사 파생접미사(SV), 접두사(PF), 접미사(SF) 등의 형태소 태그들이 나타나지 않는다. 그리고 ND는 "일반/NN+적/SF"이나 "효과/NN+적/SF" 같은 단어에 할당된다. 이런 단어들이 경우에 따라 명사나 관형사로 쓰일 수 있어서 독립된 태그 ND를 할당하였다.

청크 태그(chunk tag)라는 상위의 태그들을 도입함으로써, 기반 명사구 인식의 문제를 일종의 태그 부착 문제로 볼 수 있다. [10] 등의 전례를 따라 {I, O, B}의 세 가지로 구성된 청크 태그를 도입하는데, 본 연구에서는 이러한 청크 태그가 각 형태소에 할당된다. 어떤 형태소가 기반 명사구의 일부이면 I(In) 태그가 주어지고, 기반 명사구에 포함되지 않으면 O(Out) 태그가 주어진다. 또, B(Begin) 태그는 하나의 기반 명사구에 연이어 나타나는 기반 명사구의 첫 형태소에 할당하여 앞의 기반 명사구와 구분한다. 예를 들어 앞의 예문 (4)에 품사 태그를 할당한 결과가 (6)이고, 여기에 다시 청크 태그를 할당한 결과가 (7)인데, (7)에서 (4)의 기반 명사구를 손쉽게 추출할 수 있다.

(6) 그래서/AD 칭기스칸/NQ+은/JO 뛰어나/AJ+ㄴ/EM 기마/NN 기술/NN+로/JO 인류/NN 역사상/NN 최대/NN+의/JO 제국/NN+을/JO 이루/VV+었/PF+다/EM+./SE

(7) 그래서/AD/O 칭기스칸/NQ/I+은/JO/O 뛰어나/AJ/I+ㄴ/EM/I 기마/NN/I 기술/NN/I+로/JO/O 인류

/NN/I 역사상/NN/I 최대/NN/I+의/JO/O 제국/NN/I+을/JO/O 이루/VV/O+었/EM/O+다/SE/O

이제 기반 명사구 인식은, 각 형태소에 대해 어떤 청크 태그를 줄 것인지를 결정하는 일종의 청크 태그 부착 과정이 되며, 이 결정을 위해 전·후의 문맥 정보를 고려하여야 할 것이다.

3.2 규칙 기반 학습

규칙 기반 학습은 그림 1과 같이 진행된다.

학습 말뭉치(training corpus)는 규칙을 학습할 대상이 되는 형태소 태그 부착 말뭉치이며, 이 학습 말뭉치의 각 형태소에 올바른 청크 태그를 할당한 것이 목표 말뭉치(target corpus)이다. 규칙 기반 학습은 학습 말뭉치에 할당된 현재의 청크 태그들을 목표 말뭉치의 태그들과 비교하여, 학습 말뭉치를 가능한 한 목표 말뭉치에 가깝게 바꾸어주는 규칙들을 학습하는 것이다. 따라서 먼저 어떤 단순한 방법으로 학습 말뭉치에 청크 태그를 초기할당해야 하는데, 이를 담당하는 부분이 초기 시스템(baseline system)이다.

초기 시스템은 다양한 방법으로 구현할 수 있겠으나, 본 연구에서는 형태소 태그별로 목표 말뭉치 내에서 가장 자주 할당받았던 청크 태그를 구하여, 해당 형태소 태그를 갖는 형태소에는 모두 같은 청크 태그를 할당하는 방법을 택하였다. 예를 들어 목표 말뭉치를 검사하여 본 결과 태그 AJ를 받은 형태소들의 청크 태그가 O인 경우가 가장 많았다면, 초기 시스템은 모든 AJ 형태소에 대해서 일단 O를 할당한다는 뜻이다.

학습 시스템은 현재의 말뭉치(current corpus)를 목표 말뭉치와 비교하여 조금이라도 현재 말뭉치를 개선할 가능성이 있는 모든 가능한 규칙 후보들을 생성한 다음, 이 가운데서 현재 말뭉치를 목표 말뭉치에 가장 가깝게 바꾸어주는 하나의 규칙을 선택한다. 선택된 규칙을 현재의 말뭉치에 적용하여 청크 태그들을 변경하고 나서, 다시 학습 사이클이 반복된다. 이와 같은 방식으로 일련의 규칙들이 학습된다. 학습 과정이 끝나고 새로운 말뭉치에 대해 청크 태그 부착을 행할 때에도 같은 순서로, 초기 시스템으로 청크 태그를 할당한 후 학습 과정에서 얻어진 일련의 규칙들을 순서대로 적용한다.

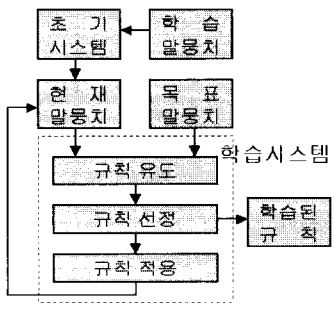

그림 1 규칙 기반 학습 과정

여기서 규칙은 IF-THEN의 형식을 갖는 대치 규칙으로서, "어떤 형태소와 그 주변 문맥이 특정 조건을 만족하면, 그 형태소의 청크 태그를 새로운 태그로 바꾸어라"는 형식이다. 가령 (7)의 예문에 초기 시스템을 적용한 결과가 (8)와 같다고 하자.

(8) 그래서/AD/O 칭기스칸/NQ/I+은/JO/O 뛰어나/AJ/O+ㄴ/EM/O 기마/NN/I 기술/NN/I+로/JO/O 인류/NN/I 역사상/NN/I 최대/NN/I+의/JO/O 제국/NN/I+을/JO/O 이루/VV/O+었/EM/O+다/SE/O

(8)을 목표 말뭉치인 (7)과 비교해 보면 밑줄 그은 부분에서 청크 태그가 잘못 되어 있음을 알 수 있다. 현재의 말뭉치에서 잘못되어 있는 청크 태그를 올바른 것으로 바꾸어 줌으로써 목표 말뭉치에 가까워질 가능성이 있기 때문에, 청크 태그가 현재 잘못되어 있는 부분마다에서 새로운 규칙 후보를 만들어낸다. 예를 들어 "뛰어나/AJ/O"라는 단어에 대해 (9)와 같이, "직전 형태소의 형태소 태그가 JO이고, 직전 형태소의 청크 태그가 O이며, 현재 형태소의 청크 태그가 O이면, 이 청크 태그를 I로 바꾸라"는 내용의 규칙을 생성할 수 있을 것이다.

(9) P_{-1} = JO, T_{-1} = O T_0 = O → T_0 = I

여기서 (9)는 물론 "뛰어나/AJ/O"의 청크 태그를 올바르게 고칠 수 있는 가능한 규칙의 한 가지 예에 불과하며, 말뭉치의 이 위치에서만 수많은 규칙을 생성해낼 수 있다. 어떤 규칙 후보들을 만들어낼 것인지는 다음 절에서 설명하는 규칙 템플릿에 의해 지정된다.

한편 규칙 (9)가 이 특정한 위치에서는 올바로 작용하지만 말뭉치의 다른 위치에서는 맞는 태그를 틀리게 바꿀 수도 있을 것이다. 따라서 현재 말뭉치와 목표 말뭉치의 비교에 의해 수많은 규칙 후보들을 생성한 다음, 각 후보 규칙들이 말뭉치 전체에 대해 미치는 '(고친 회수) - (틀린 것을 맞게 고친 회수) - (맞는 것을 틀리게 고친 회수)'를 비교하여 가장 좋은 규칙을 선택한다. 이러한 과정을 반복하여 얻어지는 일련의 순서 있는(ordered) 규칙이 학습의 결과가 된다.

3.3 규칙 템플릿

현재 말뭉치의 n번째 형태소인 M_n에서 청크 태그가 틀려져 있으면 그 위치에서 이 태그를 올바르게 고치는 여러 규칙 후보를 생성하게 되는데, 이때 어떤 규칙을

표 1 템플릿을 구성하는 패턴들

P-패턴	PW-패턴	T-패턴
P_0	P_0 W_0	T_0
P_{-1}	P_{-1} W_{-1}	T_{-1} T_0
P_1	P_1 W_1	T_0 T_1
P_{-1} P_0	P_{-1} P_0 W_1	T_{-2} T_{-1}
P_0 P_1	P_{-1} P_0 W_0	T_1 T_2
P_{-1} P_1	P_0 P_1 W_0	
P_{-2} P_{-1}	P_0 P_1 W_1	
P_1 P_2	P_1 P_1 W_{-1}	
P_{-3} P_{-2} P_{-1}	P_1 P_1 W_1	
P_{-2} P_{-1} P_0	P_2 P_{-1} W_{-2}	
P_{-1} P_0 P_1	P_{-2} P_{-1} W_{-1}	
P_0 P_1 P_2	P_1 P_2 W_1	
P_1 P_2 P_3	P_1 P_2 W_2	
P_{-3} P_{-2} P_{-1} P_0	P_{-2} P_{-1} P_0 W_{-2}	
P_{-2} P_{-1} P_0 P_1	P_2 P_{-1} P_0 W_{-1}	
P_{-1} P_0 P_1 P_2	P_2 P_1 P_0 W_0	
P_0 P_1 P_2 P_3	P_0 P_1 P_2 W_0	
P_{-4} P_{-2} P_{-1} P_0	P_0 P_1 P_2 W_1	
P_0 P_1 P_2 P_4	P_0 P_1 P_2 W_2	
P_{-5} P_{-3} P_{-1} P_0		
P_0 P_1 P_3 P_5		

만들어 내느냐는 것은 주변 문맥의 어떤 특성이 기반 명사구 인식 문제에 영향을 미칠 수 있을 것으로 보는지에 달려 있다. 대체로 보아 M_n 및 전후 형태소들(..M_{n-2}, M_{n-1}, M_n, M_{n+1}, M_{n+2}..)이 어떤 어휘 형태(string)를 갖는지, 그리고 어떤 형태소 태그 및 청크 태그를 갖는지 등이 영향을 미칠 수 있을 것이다. 이를 위해 미리 규칙 템플릿(template)의 집합을 정의해 두어, 규칙 인스턴스들을 생성할 때에 이 템플릿을 따르도록 한다. 본 연구에서는 200개의 규칙 템플릿을 사용하였는데, 표 1에 보인 기본 패턴들로부터 조직적으로 구성하였다.

여기서 P는 형태소 태그를, W는 형태소의 스트링 형태를, 그리고 T는 청크 태그를 각각 나타낸다. 첨자는 청크 태그를 할당하고자 하는 현재의 형태소를 중심으로 하는 좌우의 거리를 나타낸다. 가령, P_{-2}는 좌측으로 두 번째 형태소의 형태소 태그를 말한다. 형태소 태그만을 참조하는 P-패턴이 21개, 형태소 태그 및 스트링 형태를 참조하는 PW-패턴이 19개, 그리고 청크 태그를 참조하는 T-패턴이 5개이므로, (P-패턴)×(T-패턴), 그리고 (PW-패턴)×(T-패턴)으로 결합하면 총 200개의 템플릿이 만들어는데, 이 템플릿들이 후보 규칙의 좌측편(lefthand side)을 구성한다. 가령 "P_{-1}" 패턴과 "T_{-1} T_0" 패턴을 결합하여 "P_{-1} T_{-1} T_0" 템플릿을 얻는데, 이 템플릿이 실제의 문맥에 매칭되어 적절한 값이 할당되면 (9)와 같은 규칙 인스턴스가 만들어질 수 있다.

한 형태소의 청크 태그를 결정하기 위해 보다 큰 문맥을 고려에 넣을수록, 또 가능한 모든 결합 패턴을 고려할수록 좋지만, 윈도우가 커지고 결합 가능성이 증가할수록 규칙의 수가 급속히 늘어나므로 학습에 걸리는 시간이 너무 길어진다. 본 연구에서는 표 1에서 보는 바와 같이 C 태그와 W 형태는 좌우 두 번째 형태소[-2..+2]까지만 보는 것으로 하였고, P 태그는 [-5..+5]의 범위를 고려하되 [-3..+3]까지는 남김없이 보고 그 이상 떨어진 태그는 제한적으로만 보는 것으로 하였다.

특히 주변 문맥의 형태소 스트링을 보는 W-패턴의 경우는 이른바 어휘화된(lexicalized) 규칙을 만들어내는데, 이를 일반적으로 허용하면 규칙의 수가 너무 늘어날 뿐만 아니라, 자료 희소성(data sparseness)의 문제로 인해 그 유용성이 떨어질 가능성이 많다. 따라서 첫째, 모든 규칙은 최대 하나의 W 요소만 가질 수 있도록 제한하였고, 둘째, W 요소가 있는 템플릿은 W가 조사, 어미 등의 문법 형태소에 매칭되는 경우에만 규칙 인스턴스를 구성하도록 하였다. 즉, W 요소가 포함된 템플릿으로부터 후보 규칙을 구성할 때에는 W에 일치되는 형태소의 태그가 JO나 EM, NX인 경우에 한하여 후보 규칙이 만들어지도록 하였다. 이와 같이 함으로써 규칙의 수와 학습에 걸리는 시간을 상당히 줄일 수 있다. 다른 연구들에서 지적된 바와 같이 어휘화된 규칙은 자료 희소성의 문제를 가지므로, 명사나 용언과 같은 어휘 형태소에 대해 어휘화된 규칙의 생성을 억제하는 것이 전체적인 성능에 큰 영향을 주지는 않는 것으로 판단된다. 충분히 큰 학습 말뭉치를 사용한다면 보다 일반적인 어휘화 규칙의 도입을 고려해 볼 수 있을 것이다. 가장 효과적인 템플릿의 결정은, 계산 자원이 허용하는 범위 내에서 이 문제의 해결에 유용한 정보를 최대한 포함하여야 하므로, 추후의 연구가 필요한 부분이다.

3.4 학습 알고리즘

그림 2에 학습 알고리즘을 보였다. 청크 태그의 초기 할당(1행)은, 목표 말뭉치에서 각 형태소 태그가 어떤 청크 태그를 가장 자주 할당받았는지에 근거하여 이루어졌다. 예를 들면 형용사인 AJ는 학습 말뭉치 내에서 f(O) = 5849, f(I) = 3029, f(B) = 76 등의 빈도로 청크 태그를 할당받은 것으로 밝혀졌는데, 이런 경우 초기 시스템은 모든 AJ에 대해 일단 O 태그를 할당하게 된다.

초기 할당이 이루어진 다음 2-12행의 루프가 반복되는데, 말뭉치를 스캔하면서 청크 태그가 잘못되어 있는 부분에서 앞에서 설명한 바와 같이 템플릿에 따라 새로운 규칙 후보들을 생성한다. 청크 태그가 잘못되어 있는 각 위치에서 최대 템플릿의 개수만큼의 새로운 규칙 후

보들이 생성되는데(5-11행), 생성된 규칙 후보가 이미 *candidate_rules*에 존재하는 것이면 그 규칙의 *positive_score*만 증가시키고(7-8행), 새로운 것이면 *candidate_rules*에 추가한다(9-10행). *positive_score*는 틀린 것을 맞게 고친 회수를 저장하는 변수이다. 규칙 후보의 생성을 마치면, *candidate_rules*를 *positive_score*에 따라 내림차순으로 정렬한다(13행). 이제 *positive_score*가 큰 규칙 후보부터 차례로 말뭉치에 대한 부정적 기여도인 *negative_score*를 구하여 순기여도(*positive_score* - *negative_score*)가 가장 큰 규칙을 뽑는다(15-23행). 만일 다음에 고려할 규칙 후보의 *positive_score*가 현재까지의 최대 순기여도(net_maxscore)보다 크지 않다면, 그 규칙 후보 혹은 그 이후의 규칙 후보들은 더 나은 순기여도를 가질 가능성이 없으므로, 더 이상 진행할 필요 없이 그 때의 최대 순기여 규칙 후보를 택하면 된다(16-17행).

선택된 규칙 후보를 *learned_rules*에 다음 규칙으로 추가하고(26행), 이 규칙을 말뭉치 전체에 적용하여 청크 태그를 변경한다(27행). 그런 다음 다시 규칙 후보 생성 단계부터 반복한다. 만일 더 이상 0보다 큰 순기여도를 갖는 규칙을 찾지 못하면 학습은 종료된다(24-25행).

```
1) apply baseline system to Corpus;
2) loop {
3)   for corpus_position ← 1 to corpus_size {
4)     if current_tag(corpus_position) ≠ correct_tag(corpus_position)
5)       for t ← 1 to template_size {
6)         make a candidate rule r from t th template;
7)         if r found among candidate_rules
8)           increment r's positive_score;
9)         else
10)          add r (with positive_score ← 1) to candidate_rules;
11)      }
12)  }
13)  sort candidate_rules in descending order of the positive score;
14)  net_maxscore ← 0;
15)  for each r in candidate_rules {
16)    if r's positive_score < net_maxscore
17)      break;
18)    compute r's negative_score;
19)    if (r's positive_score - r's negative_score > net_maxscore) {
20)      net_maxscore ← r's positive_score - r's negative score;
21)      cur_maxrule ← r;
22)    }
23)  }
24)  if net_maxscore is 0    // no more rule to apply
25)    halt;
26)  append cur_maxrule to learned_rules;
27)  apply cur_maxrule to Corpus;
28) endloop;
```

그림 2 규칙 기반 학습 알고리즘

3.5 비교

[15]는 규칙 기반의 비통계적 학습 기법을 제안하였고, 이 기법이 자연언어 처리에 최근 도입되고 있는 말뭉치 기반의 경험적 접근법의 일종으로 통계적 기법에 대한 대안이 될 수 있음을 주장하였고, [10]은 이를 체계화하여 영어의 청크 분할 문제에 적용하였다. 본 연구는 이 기법을 한국어의 기반 명사구 인식 문제에 적용하기 위해 필요한 문법적 틀과 알고리즘을 개발하였으며, 이를 위하여 한국어의 특성과 학습 효율성을 고려하였다.

첫째, 여러 개의 형태소가 결합되어 단어를 구성하는 한국어의 특성에 따라, 단어가 아니라 형태소가 태그 부착의 기본 단위로 설정되었다. 하나의 한국어 단어는 어휘 형태소와 문법 형태소를 포함하여 둘 혹은 그 이상의 형태소를 가지는데, 이중 일부만 기반 명사구에 포함되어야 하는 경우가 흔하므로, 단어를 하나의 단위로 볼 수 없다. 뿐만 아니라 어휘 형태소와 문법 형태소의 문법적 성격이 판이하므로 이들을 분리하여 취급함이 옳다.

둘째, 형태소가 기본적 조작 단위가 되면 한 단어가 여러 개의 형태소로 쪼개지므로, 규칙의 수가 급속히 늘어나 학습 효율성이 떨어진다. 따라서 학습 과정 이전에 "명사+접미사→동사/형용사"를 이루는 파생을 미리 처리하여 이들을 하나의 용언으로 취급하였다. 또한 기타 접사와 선어말 어미 등 기반 명사구의 인식 문제와 관련하여 문법적 영향이 미미할 것으로 보이는 문법형태소들을 무시하였다. 그리고 규칙 수를 늘리는 주된 요소인, W 패턴으로부터 구성되는 어휘화된 규칙의 생성을 상당히 제한하였다. 이를 통하여 성능 저하를 최소화하면서 효율적인 규칙 학습이 가능하였다.

본 연구의 접근법에 대한 대안으로, 학습에 의하지 않고 명사구 유형으로부터 규칙을 직접 만드는 방안을 생각해 볼 수 있으나, 그렇게 얻어지는 유형들은 인접 문맥을 고려하지 않는 지역적 패턴이 되므로 실제 문장에 적용했을 때 많은 모호성이 발생되는데, 이러한 모호성의 해소가 결국 파싱 작업에 근접하게 된다. 이에 비해 본 연구에서 학습되는 규칙들은 기본적으로 규칙 템플릿이 지정한 범위까지의 주위 문맥을 고려하는 패턴이 된다. 또한 모호성 없이 빠른 속도로 비교적 정확하게 기반 명사구를 인식해 낼 수 있다.

한편 기존의 파서로 문장 전체에 대한 파싱을 행하고 그 결과에서 기반 명사구를 추출하는 기법보다 본 연구의 제안이 우월한 정확도를 보일 것인지의 의문을 제기할 수 있다. 그러나 이러한 단순 비교는 적절치 못하다.

본 연구는 현재 기술 수준의 전체 파싱이 정확도나 속도 면에서 만족할 만한 성과를 얻지 못하고 있으며, 일부의 응용에 있어서는 부분구조의 효율적인 추출만으로도 쓸모가 있다는 전제에서 출발한다. 즉, 기존의 전체 파서(full parser)는 대상 문장의 길이가 길어질수록 급격히 정확도나 속도가 저하되어 파싱이 실패하는 비율이 높아지는 반면에, 부분 파싱 기법은 문장의 길이에 관계없이 빠른 속도로 비교적 정확하게 부분 구조를 생성할 수 있다. 따라서 파싱 결과에서 기반 명사구를 추출했을 때의 정확도와, 파싱을 거치지 않는 기반 명사구 추출의 정확도를 수치상으로 단순 비교하는 것은 적절치 못하다. 다만 확률적 파싱에 대한 국내외의 연구들[16, 17, 18, 19]이 대략 85% 내외의 정확도를 보이고 있는데 반해, 본 연구를 포함한 기반 명사구 추출 기법들은 다음 장에서 살펴보는 바와 같이 90% 이상의 정확도를 보이고 있음을 지적할 수 있다.

4. 실 험

4.1 실험 결과

한국과학기술원에서 만들어 배포한 국어정보베이스[6]에 포함된 트리 태그 부착 말뭉치를 이용하여 실험을 행하였다. 약 15만 단어 규모(322,057 형태소)의 이 말뭉치는 원래 54개의 세분된 태그셋(tagset)에 따라 형태소 태그 부착이 이루어져 있는데, 이와 같이 세분된 태그셋은 자료 희소성의 문제를 겪을 수 있고, 이 태그들 가운데 일부는 불필요하였으므로, 3.1절에서 설명한 (5)의 태그셋으로 변환하여 사용하였다. 또한 트리에 포함된 중첩된 명사구들로부터 비재귀적 기반 명사구들을 추출하고 적절한 청크 태그를 할당하여 본 연구의 알고리즘이 가정하는 형태로 변환하였다. 원래의 말뭉치 자체에도 상당수의 트리 태그 부착 오류가 포함되어 있는 것으로 보이며, 위의 자동 추출 과정 자체도 일부의 조직적인 오류를 가질 가능성이 있으므로, 이렇게 얻어진 말뭉치에 어느 정도의 오류가 포함되어 있을 것이다.[3)]

이와 같이 본 연구의 형태대로 고친 말뭉치의 크기는 295,682 형태소였으며, 이로부터 임의로 연속된 영역을 택하여 검증 말뭉치(75,002 형태소)로, 나머지를 학습 말뭉치(220,680 형태소)로 사용하였다. 학습 말뭉치를 이용하여 앞에서 보인 알고리즘에 따라 규칙의 학습이 이루어졌는데, 학습 시간을 제한하기 위해 규칙의 순기여도가 3 미만으로 떨어질 때에 학습을 중단하였다. 이와 같이 하여 386 개의 규칙이 얻어졌는데, 현재 규모의 말뭉치에서는 그 이후의 규칙들이 전체 성능에 미치는 영향은 미미한 것으로 여겨진다.

얻어진 초기 시스템 및 규칙들을 이용하여 검증 말뭉치에 대해 행해진 본 연구의 기반 명사구 인식 실험 결과는 표 2와 같았다. 초기 시스템에 의한 초기 결과(baseline result)와 학습된 규칙에 의한 최종 결과로 나누어 보였는데, 기반 명사구를 단위로 한 정확도(precision)와 재현율(recall)로 나타내었다. 즉 기반 명사구의 시작과 끝 지점이 올바로 파악되었을 때에만 그 명사구에 대한 인식이 성공한 것으로 보았다. 재현율은 검증 말뭉치에 있는 전체 기반 명사구들에 대해 바르게 인식된 기반 명사구들의 비율이며, 정확도는 인식된 전체 기반 명사구들 가운데 바르게 인식된 것들의 비율을 말한다. 검증 말뭉치는 3,195 문장으로 이루어져 있는데, 약 4만 단어 혹은 75,002 형태소 규모였다. 검증 말뭉치에 있는 기반 명사구의 수는 17,107개로서 문장 당 약 5.4개의 기반 명사구가 있는 셈이다. 초기시스템의 정확도는 표에 보인 바대로 영어에 대한 [10]의 결과와 크게 다르지 않았는데, 이는 두 언어에서의 기반 명사구 인식 문제의 난이도가 엇비슷함을 의미하는 것으로 보인다. 표의 마지막 열에 보인 것은 태그 단위의 정확도로서, 올바른 청크 태그들의 전체에 대한 비율을 말한다.

표 2의 아래쪽에는 2장에서 언급한 관련 연구인 [10], [11], [12]에서 영어에 대해 이루어진 유사한 실험의 결과를 보였다. [12]와 [11]은 품사 태그 정보만을 이용하였고, [10]은 품사 태그 및 어휘 정보를 모두 이용한 결과이다. 본 연구는 어휘 정보를 이용하였지만 앞서 설명한 대로 조사, 어미, 의존 명사 등의 문법 형태소들에만 한정하였다. 본 연구의 결과는 영어에 대해 현재까지 가장 우수한 결과인 [10]의 실험 결과와 비교하여 정확도는 비슷하고 재현율에서 2% 가량 낮게 나타났는데, 대상 언어와 실험 조건 등이 다르므로 실험 결과의 직접적인 비교는 어렵지만, 본 연구의 결과가 경쟁력이

표 2 기반 명사구 인식 실험 결과

	정확도	재현율	태그단위 정확도
본 연구 (초기/최종)	81.1/**91.8**%	82.2/**90.7**%	91.4/**96.9**%
[10] (초기/최종)	81.9/91.8%	78.2/92.3%	94.5/97.4%
[12]	90.7%	91.1%	
[11]	91.6%	91.6%	

3) 트리로부터 기반 명사구를 자동 추출하는 과정의 오류들은 학습 말뭉치와 검증 말뭉치에 모두 일관되게 포함될 것이므로 실험 결과에 직접적인 영향을 미치지는 않는다.

있음을 알 수 있다.

4.2 결과의 분석

규칙 기반 학습에 의해 얻어진 규칙들 가운데 첫 10개를 보이면 표 3과 같다. 규칙 기반 학습 기법이 갖는 중요한 장점의 하나가, 학습된 결과를 쉽게 해석할 수 있다는 점인데, 이런 점이 모델의 문제점을 파악하고 개선하는 데 기여할 수 있다. 가령 표 3의 첫 번째 규칙은 "다르/AJ/O+ㄴ/EM/O"로 태그부착된 부분을 "다르/AJ/O+ㄴ/EM/I"로 바꾸라는 뜻이다. 즉, 초기 시스템에 의해 EM은 항상 O로 태그부착되어 있는데, 이와 같은 형용사+관형형 어미의 구성은 그 뒤의 체언과 합하여 기반 명사구를 형성할 가능성이 많으므로 청크 태그를 I로 바꾸어주는 것이다. 일단 어미의 청크 태그가 I로 바뀌면, 앞의 형용사 "다르/AJ"도 두 번째 규칙에 의해 I를 할당받게 되어 "다르/AJ/I+ㄴ/EM/I"로 바뀌게 된다. 그러므로 초기의 태그 할당이 "다른 [사람]+을"이었다면, 이 두 규칙을 거친 후에는 "[다른 사람]+을"로 변경되는 것이다.

다른 규칙들도 살펴보면 그 뜻을 헤아리기 그리 어렵지 않다. 3번째 규칙은 "-ㄹ 수 있/없(다)" 구절과 관계되는데, 실험에 사용한 말뭉치에 따르면 이런 구성은 보조용언구(AUXP)로 간주되었으므로 기반 명사구로 파악되지 않는다. 따라서 규칙에 나타난 대로 '수'의 태그를 O로 변경한다. 4번째 규칙은 명사 뒤에 딸린 콤마와 같은 구두점을 앞 명사에 딸린 요소로 보아 기반 명사구의 일부로 간주한다. 5번째와 10번째 규칙은 예문 (10)에서의 "-ㄹ 것이(다)" 구절과 관련된 것인데, 이것도 역시 학습 말뭉치에서 보조용언구로 간주되어 기반 명사구에서 제외되었다.

표 3 상위 10개의 기반 명사구 인식 규칙

순번	적 용 조 건		새태그 (T_0)
1	P_{-1}=AJ P_0=EM W_0=ㄴ	T_0=O	I
2	P_0=AJ P_1=EM P_2=NN	T_1=I T_2=I	I
3	P_{-1}=EM P_0=NX W_{-1}=ㄹ	T_1=O T_2=O	O
4	P_0=SY	T_{-1}=I T_0=O	I
5	P_{-1}=EM P_0=NX P_1=CP P_2=EM	T_0=I T_1=O	O
6	P_1=EM W_{-1}=기	T_{-1}=O T_0=O	I
7	P_0=EM	T_{-2}=O T_{-1}=I	I
8	P_0=EM W_0=ㅁ	T_0=O	I
9	P_1=EM W_1=ㅁ	T_{-1}=O T_0=O	I
10	P_1=NX P_2=CP P_3=EM	T_0=I T_1=O	O

(10) [문명 세계 손님들]+에게는 [이것]+도 [쉬운 일]+이 아닐 것이다.

나머지 규칙들은 어미에 의한 명사화를 다루고 있는데, 6번째와 7번째 규칙은 어미 ' 기', 8번째 및 9번째 규칙은 전성어미 '-ㅁ'에 의한 명사화를 다루고 있다.

4.3 오류 유형

검증 말뭉치에 대한 실험 결과의 약 10%를 직접 검토한 결과, 다음과 같은 주된 오류 유형들을 발견할 수 있었는데, 비교적 서로 엇비슷한 빈도로 나타났다. 각 패턴에서 [..]는 본 알고리즘에 의해 예측된 기반 명사구를 나타내고 {..}는 원래의 학습 말뭉치에 있는 기반 명사구를 뜻한다.

i) [{NP}..{NP}] (21%)

이 오류 유형은 예문 (11)과 같이 두 개나 그 이상의 기반 명사구로 분리되어야 할 것이 합쳐져 하나로 예측된 오류이다.

(11) [{그 결과} {인류 문명 사상 최대}]+의 [구조물]+인 [만리 장성]+이 [중국]+과 [{몽골} {사이}]+에 만들어 졌다.

ii) {[NP]..[NP]} (17%)

이 유형은 i)과 반대로 학습 말뭉치에서는 합쳐서 하나의 기반 명사구로 간주한 것을 본 알고리즘이 분리하여 예측한 경우이다. 이 유형에는 (12)와 같은 동격 어구가 주로 나타난다.

(12) [우리 나라]+에서도 [파평 윤씨들]+은 {[물고기], 특히 [잉어]}+를 먹지 않는다.

iii) {..[NP]} (25%)

비교적 자주 나타난 유형으로, (13)과 같이 기반 명사구에 포함되어야 할 수식어를 너무 적게 잡은 오류 유형이다. 이 유형은 비록 본래의 기반 명사구 전체를 정확히 포착하지는 못하였으나, 대체로 그 머리에 해당하는 명사를 찾은 것이고 후속 단계에서의 추가적인 결합도 기대해 볼 수 있으므로, 그 피해가 크지 않은 안전한 오류들이라고 볼 수 있다.

(13) [대화]+가 [매 사냥]+으로 접어들자 [두툼한씨]+는 {비스듬히 뉘+었던 [상체]}+를 벌떡 일으켜 세우며 [억양]+도 부쩍 높였다.

iv) [..{NP}] (19%)

이 유형은 예측된 기반 명사구가 원래의 기반 명사구보다 더 많은, 불필요한 요소를 포함하는 예들인데, (14)와 같은 예는 구문 구조를 파괴하는 것이므로 후속 단계에 피해를 준다. "용언+연결어미 명사"가 결합된 이러한 예는 일반적인 문법적 구성과 어긋나므로 후속 단계의 처리를 통해 교정할 수 있는 여지도 있을 것으로 보

인다.

(14) [늙어서 {사냥}]-을 못 하는 [매]+는 [산]+으로 올라가 날려 보낸다.

(15) [나]-는 [날카로운 {매}]+의 [눈매]+처럼 내쏘는 듯한 [그]+의 [눈동자]+에 [사진기]+의 [초점]+을 맞추었다.

(15)와 같은 예도 이 유형에 속하는데, 수식의 모호성 때문에 발생한 오류로 볼 수 있다. 즉, '날카로운'이 '눈매'를 꾸며야 하므로 [날카로운 매]로 기반 명사구를 삼을 수 없다. 이 경우도 보다 심층적인 문법적 처리가 요구되는 부분이다.

vi) 접속과 관련한 오류 (8%)

명사 접속에서 접속항(conjunctor)의 예측이 틀리거나, 명사 접속을 꾸미는 수식어의 수식 범위와 관련한 오류이다. 그러나 (16)과 같은 접속항 오류는 사실상 그 판단이 애매하다. 본 연구가 사용한 학습 말뭉치에서는 (16)의 접속 부분을 (17)과 같이 분석하여 "[A]와 [B, 그리고 C]" 형태로 기반 명사구를 가정하고 있지만, 견해에 따라서는 본 알고리즘이 예측한 방식과 같이 각 접속항을 모두 독립된 기반 명사구로 보는 입장도 옳다고 볼 수 있을 듯하다.[4] 여기서는 학습 말뭉치와 다르므로 일단 오류로 보았다.

(16) [이 사냥]+에서는 [매]-의 [주인]+인 [수알치]+와 {[털이꾼], 그리고 [배꾼]}+이 합세한다.

(17) (NP (NP (VP (NP (NP 매/ncn)+의/jcm
주인/ncn)+이/jp)-ㄴ/etm
수알치/ncn)+와/jcj
털이꾼/ncn+,/sp 그리고/maj 배꾼/ncn)

(18) 그렇지 않고서는 [그 {모양}]+이나 [기능]+이 그렇게 꼭 닮아 있을 수가 없겠다.

한편 (18)은 명사 접속에 수식어가 있을 때 그 접속 범위와 관련한 문제인데, 여기서는 '그'가 '모양이나 기능' 전체를 수식하는 것으로 보아야 하므로 [그 모양]을 기반 명사구로 볼 수 없다. 이는 원래 접속의 수식 범위와 관련한 까다로운 문제로서 후속 단계에서의 심도 있는 대책이 필요한 부분이다.

오류 유형을 정리하면, 접속과 관련한 다섯 번째 유형은 이 단계에서 처리 가능한 범위를 벗어나는 것으로 판단되며, 첫째와 넷째 유형에 포함된 상당수의 오류들은 구문 구조를 깨뜨리는 것으로서 주된 오류에 속한다. 반면에 둘째와 셋째 유형은 그 피해가 심하지 않거나 후속 단계에서 추가의 결합을 기대할 수 있는 오류들이다. 따라서 기반 명사구 단위로 약 10% 가까운 오류가 발생한 것으로 나타나지만, 이들이 전부 심각한 영향을 미치는 오류인 것은 아님을 알 수 있다.

한편 본 연구가 예측한 기반 명사구가 비록 검증 말뭉치의 기반 명사구와 일치하지 않더라도 오류라고 볼 수는 없는 경우가 다수 발생하였다. 이는 사용된 검증 말뭉치 자체의 구문 태그 부착 오류이거나 혹은 구문 트리로부터 기반 명사구를 자동 추출하는 데 있어서의 오류에 의한 것들이다. 이런 경우가 오류로 간주된 것들의 10% 이상을 차지하였는데, (19)가 그 한가지 예이다.

(19) [이 동등한 권리]+가 [불평등한 노동]+에 대해서는 [불평등한 {권리}]+인 것이다.

또한 학습 말뭉치로 사용한 국어정보베이스는 "-ㄹ 수 있/없(다)", "-기 때문이(다)" 등을 보조용언구로 파악하여 '수'나 '때문'을 따로 명사구로 포착하지 않는 원칙을 따르고 있는 것으로 보이나, 실제의 말뭉치에는 이 원칙에 어긋나게 태그부착된 예가 다수 발견되었고, 이러한 것들이 실험 결과에서는 오류로 간주되었다. 따라서 이런 점들을 고려하면 전체 정확도는 현재의 결과보다 약간 높을 것으로 기대할 수 있다.[5]

5. 결 론

규칙 기반 학습에 근거하여 한국어의 기반 명사구를 인식하는 기법을 제안하였다. 규칙 기반 학습은 통계적 학습법과 달리 규칙 기반의 방법에 의해 일련의 규칙을 말뭉치로부터 학습하는데, 학습 과정 및 학습 결과의 이해가 용이하다. 얻어진 규칙은 큰 공간을 차지하지 않으며, 이를 이용한 기반 명사구 추출은 전통적인 구절 규칙에 근거한 파싱 기법에 비해 효율적이고 강건하게(robust) 실행될 수 있다. 본 연구에서는 이 기법이 한국어의 기반 명사구 인식 문제에 효과적으로 적용될 수 있음을 보였다. 통합국어베이스의 구문 태그 부착 말뭉치를 이용하여 실험하였는데, 90% 이상의 높은 정확도를 얻을 수 있었다.

실험 결과 나타난 오류들 가운데 상위의 문법 정보를 필요로 하는 수식어의 수식 범위나 접속과 관련한 문제들은 후속 단계의 연구를 통해 대처되어야 할 것이다. 또한 성능 개선을 위해서 일부 오류를 교정하고 추가의 구절 결합을 담당하는 후처리 단계를 두는 방안을 검토

4) (17)에 나타난 태그들은 국어정보베이스[5]가 채택한 형태소 태그들이다.

5) 단순 계산에 의한다면 태그 단위의 오류율이 3.1%이므로, 이 오류들 가운데 10% 정도를 맞는 것으로 보아 전체적으로 약 0.3% 가량의 정확도 상승을 예측할 수 있다.

해 볼 수 있을 것이다. 보다 큰 학습 말뭉치를 사용하기 위해서는 규칙 템플릿의 확장과 학습 과정의 효율화를 검토해야 할 것이다.

본 연구에서 개발된 기반 명사구 인식 기법은 그 자체로도 다양한 응용에 사용될 수 있지만, 특히 얕은 파싱의 더 진전된 단계를 위한 입력으로 사용될 수 있을 것이며, 제안된 기법을 보다 확장한다면, 한국어 문장으로부터 명사, 동사 및 기타의 청크를 인식해 내고 주어, 목적어 등 청크들 간의 중요한 문법적 관계를 파악하는 얕은 파서를 구현할 수 있을 것이다.

참 고 문 헌

[1] S. Abney, "Parsing by Chunks," in R. Berwick, S. Abney, C. Tenny, *eds.*, *Principle-Based Parsing*, Kluwer, pp.257-78, 1991.

[2] C. Cardie, S. Mardis, D. Pierce, "Combining Error-Driven Pruning and Classification for Partial Parsing," *Proc ICML-99 (International Conference on Machine Learning)*, 1999.

[3] J. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, M. Tyson, "FASTUS: A Cascaded Finite State Transducer for Extracting Information from Natural-Language Text," in Roche, Schabes, *eds.*, *Finite-State Language Processing*, MIT, pp.383-406, 1997.

[4] Briscoe, E.J., J. Carroll, "Automatic Extraction of Subcategorization from Corpora," *Proc ANLP (ACL Conference on Applied Natural Language Processing)*, 1997.

[5] Carroll, J., G. Minnen, T. Briscoe, "Corpus Annotation for Parser Evaluation," *Proc. EACL'99 Workshop on Linguistically Interpreted Corpora*, 1999.

[6] 한국과학기술원, 국어정보베이스, v. 1.0 (CD 배포판), 1997.

[7] S. Abney, "Partial Parsing via Finite-State Cascades," *Proc Robust Parsing Workshop ESSLLI'96*, pp.8-15, 1996.

[8] K. Church, "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," *Proc 2nd Conference on Applied Natural Language Processing*, pp.136-43, 1988.

[9] A. Voutilainen, "*NPtool*, a Detector of English Noun Phrases," *Proc Workshop on Very Large Corpora*, pp.48-57, 1993.

[10] L. Ramshaw, M. Marcus, "Text Chunking using Transformation Based Learning," *Proc 3rd Workshop on Very Large Corpora*, pp.82-94, 1995.

[11] S. Argamon-Engelson, I. Dagan, Y. Krymolowski, "A Memory Based Approach to Learning Shallow Natural Language Patterns," *Proc ACL/Coling*, pp.67-73, 1998.

[12] C. Cardie, D. Pierce, "Error-driven Pruning of Treebank Grammars for Base Noun Phrase identification," *Proc ACL/Coling*, pp.218-24, 1998.

[13] C. Cardie, D. Pierce, "The Role of Lexicalization and Pruning for Base Noun Phrase Grammars," *Proc AAAI 99*, 1999.

[14] W. Skut, T. Brants, "A Maximum-Entropy Partial Parser for Unrestricted Text," *Proc 6th Workshop on Very Large Corpora*, 1998.

[15] E. Brill, "Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Parts-of-Speech Tagging," Computational Linguistics, Vol.21, No.4, pp.543-65, 1995.

[16] K. Seo, K. Nam, K. Choi, "A Probabilistic Model for the Korean Dependency Parsing Using Ascending Dependencies," *Proc NLPRS'97*, pp.145-54, 1997.

[17] 윤준태, 김선호, 송만석, "전역적 연관 표를 이용한 한국어 구문분석", 정보과학회논문지(B), 제24권 11호, pp.1297-1306, 1997.

[18] D. Magerman, "Statistical Decision-Tree Models for Parsing," *Proc ACL*, pp.276-283, 1995.

[19] Michael Collins, "Three Generative, Lexicalised Models for Statistical Parsing, *Proc ACL*, pp.16-23, 1997.

[20] 서영훈 외, 한국어 구문 Tagged Corpus 구축 및 구문 분석 데이타 사전 개발, 연구보고서, 한국전자통신연구원, 1998.

양 재 형

1988년 서울대학교 컴퓨터공학과 학사. 1990년 서울대학교 컴퓨터공학과 석사. 1995년 서울대학교 컴퓨터공학과 박사. 1995년 ~ 현재 강남대학교 지식정보공학부 조교수. 관심분야는 한국어정보처리, 자연언어처리, 인공지능 등임.