영한 기계번역에서 구문 분석 정확성 향상을 위한 구문 범주 예측

김 성 동†

요 약

실용적인 영한 기계번역 시스템은 긴 문장을 빠르고 정확하게 번역할 수 있어야 한다. 보다 빠른 번역을 위해 문장 분할을 이용한 부분 파싱 방법이 제안되어 속도 향상에 기여하였다. 본 논문에서는 보다 정확한 분석을 위해 결정 트리를 이용한 구문 범주 예측 방법을 제안한다. 문장 분할을 적용한 영어 분석에서 각각의 분할된 부분은 개별적으로 분석되며 각 분석 결과들이 결합되어 문장의 구조가 생성된다. 여기서 각 분할의 구문 범주를 미리 예측하여 부분 파싱 후에 보다 정확한 분석 결과를 선정하고 예측된 구문 범주에 근거하여 올바르게 다른 문장의 분할결과와 결합함으로써 문장 분석의 정확도를 향상시키는 것이 본 논문에서 제안한 방법의 목적이다. 본 논문에서는 Wall Street Journal의 파싱된 말뭉치에서 구문 범주 예측에 필요한 특성을 추출하고 결정 트리를 이용하여 구문 범주 예측을 위한 결정 트리를 생성하였다. 실험에서는 사람이 구축한 규칙을 이용한 방법, trigram 확률을 이용한 방법, 신경망을 이용한 방법 등에 의한 구문 범주 예측 성능을 측정, 비교하였으며 제안된 구문 범주 예측이 번역의 품질 향상에 기여한 정도를 제시하였다.

키워드 : 영한 기계번역, 구문 분석, 파싱, 문장 분할, 구문 범주 예측

Syntactic Category Prediction for Improving Parsing Accuracy in English-Korean Machine Translation

Sung-Dong Kim†

ABSTRACT

The practical English-Korean machine translation system should be able to translate long sentences quickly and accurately. The intra-sentence segmentation method has been proposed and contributed to speeding up the syntactic analysis. This paper proposes the syntactic category prediction method using decision trees for getting accurate parsing results. In parsing with segmentation, the segment is separately parsed and combined to generate the sentence structure. The syntactic category prediction would facilitate to select more accurate analysis structures after the partial parsing. Thus, we could improve the parsing accuracy by the prediction. We construct features for predicting syntactic categories from the parsed corpus of Wall Street Journal and generate decision trees. In the experiments, we show the performance comparisons with the predictions by human-built rules, trigram probability and neural networks. Also, we present how much the category prediction would contribute to improving the translation quality.

Key Words : English-Korean Machine Translation, Syntactic Analysis, Parsing, Intra-Sentence Segmentation, Syntactic Category Prediction

1. 서 론

최근의 영한 기계번역(English-Korean machine translation)은 비교적 짧은 문장의 경우에는 구문 분석 과정에서 큰 문제가 없으며 어느 정도 제대로 된 번역을 생성하고 있으나, 쉼표로 분리되는 요소를 가지는 긴 문장의 경우 각 요소간의 관계가 제대로 분석되지 못하여 매우 부자연스러운 번역이 생성되어 그 이해도가 상당이 낮다는 문제를 보이고 있다. 짧은 문장의 번역에서의 문제는 세련된 의미의 전달이 불충분하다는 점을 들 수 있는데 이는 단어가 가지는 의미적인 부분을 파악하여야 해결할 수 있는 의미적인 문제이며(semantic problem), 긴 문장의 번역에서 발생하는 문제는 각 요소간의 관계를 보다 정확하게 분석하면 어느 정도 해소될 수 있는 구조적인 문제(syntactic problem)이다. 기계번역의 성능 향상을 위해서는 의미적인 문제도 해결하

※ 이 논문은 한국과학재단의 2004년 해외 Post-doc. 연수지원에 의하여 연구되었음. (KRF-2004-214-D00158)
† 정 회 원 : 한성대학교 컴퓨터공학과 조교수
논문접수 : 2006년 2월 23일, 심사완료 : 2006년 4월 26일

여야 하지만 이를 위해서는 많은 노력과 비용이 필요하다. 따라서 본 논문에서는 의미 문제에 비해 적은 비용과 노력으로 어느 정도 해결 방안을 얻을 수 있는 구조적인 문제에 대한 해결에 초점을 맞추었다.

긴 문장의 빠른 분석을 위해 문장 분할(intra-sentence segmentation) 이 제안되었다[1-3]. 이를 이용한 구문 분석에서는 문장 분할 방법을 이용하여 문장을 분할하고 각 분할(segment)을 독립적으로 분석하여 각 분석 결과를 결합하여 문장의 분석 구조를 생성하였다. 즉, 긴 문장을 여러 개의 보다 짧은 분할로 나누고 각 분할을 부분적으로 분석(partial parsing)하는 과정을 통해 입력 문장의 구조를 생성한다. 부분 분석을 하지 않는 경우에는 구문 분석 완료 이후에 많은 후보 구조 중에서 하나의 구조를 선택하는 구조 선정(structure selection) 단계가 한번 존재하지만, 이 경우에는 각 분할의 분석이 끝난 후에 구조를 선택하여야 하므로 여러 번의 구조 선정 단계가 존재한다. 한번의 선정 보다는 여러 번의 선정 과정에서의 오류의 가능성이 높은데, 여기서 한번의 선정 오류는 결국 잘못된 문장 분석 결과를 생성하게 된다. 따라서 문장 분할을 이용한 부분 분석은 분석의 효율성(속도) 향상에는 기여하였지만 정확성의 향상에는 기여했다고 볼 수 없다.

문장 분할을 이용한 구문 분석에서 보다 정확한 구문 구조의 생성을 위해서는 각 분할의 분석 이후에 올바른 분할의 분석 결과를 선정할 수 있어야 하며, 선정된 분석 결과를 그 구문 범주에 따라 적절한 방식으로 다른 분할 결과와 결합할 수 있어야 한다. 각 분할의 구문 분석 이전에 분할의 구문 범주를 예측한다면 올바른 분할의 분석 결과를 선정하고 이를 올바른 방법으로 결합함으로써 정확한 문장 구조를 생성할 수 있을 것이다. 본 논문에서는 구문 분석 이전에 분할의 구문 범주(syntactic category)의 예측을 위한 규칙 또는 함수를 데이터를 이용한 통계적(statistical), 기계학습적(machine learning) 방법에 의해 획득하는 방법을 제안한다. 논문에서는 펜실바니아 대학(University of Pennsylvania)에서 구축한 구문 분석된 말뭉치(Penn Treebank)를 이용하여 구문 범주 예측(syntactic category prediction)을 위한 데이터를 생성하였으며 이를 이용하여 결정 트리(decision tree), 신경망(neural networks), 품사 태그 trigram의 확률(part-of-speeches tag trigram probability)을 이용한 예측 규칙(prediction rules) 또는 함수(prediction functions)를 생성하였다. 구문 범주 예측 성능 평가 결과 결정 트리 학습 방법을 통해 생성된 예측 규칙이 가장 높은 예측 정확성을 보였다. 따라서 이를 이용하여 구문 범주 예측이 구문 분석의 정확성 개선을 통해 번역의 품질 향상에 얼마나 기여하였는지를 분석하였다.

2장에서는 구문 범주 예측이 필요한 문장 분할을 이용한 문장 분석 방법에 대해서 설명한다. 3장에서는 구문 범주 예측 규칙과 함수의 생성 방법을 설명하고 4장에서는 여러 가지 예측 함수와 규칙에 의한 예측의 정확성을 비교 분석한 후 구문 범주 예측이 번역의 품질 향상에 기여한 정도를 제시한다. 5장에서는 본 논문을 결론 지으며 앞으로의 과제를 제시한다.

2. 문장 분할을 이용한 문장 분석과 구문 범주 예측

Abney에 의한 부분 분석 방법[4-6]은 명사구(noun phrases)나 전치사구(prepositional phrases) 분석 등에 활용되었으며 이후의 보다 빠른 문장의 구문 분석을 위한 기초 연구가 되었다. 실용적인 영한 기계번역 시스템을 위해 [7]에서는 문장 패턴(sentence patterns)을 이용한 구문 분석 방법을 제안하였다. 이는 정해진 긴 문장의 패턴에 맞는 문장의 경우에 대해서 만족할만한 결과를 얻을 수 있었으나 패턴의 적용률(coverage)에 한계가 있기 때문에 이후에 활발하게 이용되지 못하였다. 이러한 단점을 보완하기 위해 [2, 3]에서는 통계적인 방법에 의해 문장을 분할하고 부분 파싱을 통해 문장을 분석하는 방법이 제안되었으며 이 방법에서는 문장 분할의 적용률이 문장 패턴에 비해 크게 향상되었기 때문에 긴 문장의 효율적인 분석 방법으로 실용적이라 할 수 있다.

일반적으로 긴 문장은 쉼표로 분리되는 여러 요소의 결합으로 이루어진다. 따라서 긴 문장은 1차적으로 쉼표를 기준으로 분할된다. 그리고 일정한 길이 이상의 분할(segment)은 통계적인 방법으로 다시 분할되어 파서가 구문 분석을 용이하게 할 수 있는 크기로 나누어진다. 쉼표에 의해 분할되는 분할들은 문장에서 특정한 역할을 하는데 그 역할은 분할을 분석한 후 판단할 수 있다. 이러한 분할의 역할을 정확하게 파악하기 위해서는 분할의 분석 결과 구조를 올바르게 선택해야 한다. 분할의 분석 결과로 많은 구조들이 생성되는데 같은 구문 범주에 대해서도 많은 결과 구조가 만들어진다. 따라서 올바른 결과 구조를 선택하는 것은 매우 어려운 문제이다. 이 선택을 용이하게 하기 위해서 본 논문에서는 각 분할의 분석 이전에 분할의 구문 범주를 미리 예측하고 예측된 결과에 의해 분할의 분석 결과를 결정함으로써 선택의 정확성을 높일 뿐만 아니라 분할의 역할을 보다 정확하게 판단하고자 하였다.

(그림 1)은 문장과 쉼표로 분리되는 분할들, 분할들의 문장에서의 역할과 가능한 번역을 보여준다. (그림 1)에서 [번역 1]은 구문 범주 예측을 하지 않았을 경우인데 "분할 1"에 대해서는 명사구의 분석 결과가 선택되었고, "분할 2"에 대해서는 하나의 문장(SENT)으로 분석된 결과가[1], 그리고 "분할 3"에 대해서는 동사구(verb phrase)의 결과가 선택되었다. [번역 1]을 생성하는데 사용된 분할의 분석 결과 중 "분할 2"의 분석 결과가 잘못되었는데 이는 구문 분석기 내부의 구조 선택, 즉 모호성 해소(syntactic disambiguation)와 구문 분석 결과 선택(parse tree selection) 문제를 개선

1) 파서의 영어 문법에서는 RLCL ← SENT라는 규칙에 의해 관계절(relative clause) 결과가 생성되는데, 현재 구조 선택을 위한 점수(score) 체계에서 SENT와 RLCL간의 점수 차이가 없기 때문에 SENT 결과를 가지는 구조가 선택될 수 있다.

A small TCL interpreter, which can be linked into the code, interprets the strings.

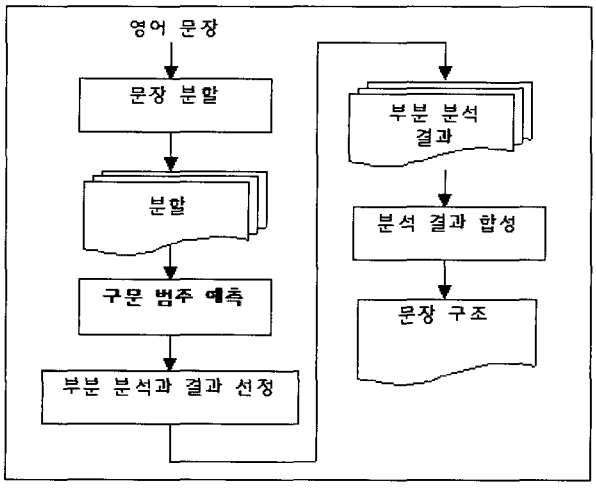
분할 1: A small TCL interpreter → 명사구
분할 2: which can be linked into the code → 관계절
분할 3: interprets the strings → 동사구

[번역 1] 작은 TCL 인터프리터, 그 코드로 연결될 수 있는 것, 그 문자열을 해석한다.
[번역 2]그 코드로 연결될 수 있는 작은 TCL 인터프리터는 그 문자열을 해석한다.

(그림 1) 문장의 분할과 분할의 역할, 가능한 번역들

하여 해결할 수도 있다. 물론 구문 분석기의 결과 구조 선택 과정을 간단하게 수정하여 [번역 1]의 오류를 없앨 수 있다. 그러나 이들 문제들은 자연언어 구문 분석기가 가지는 고유한 문제로서 쉽게 개선할 수 있는 문제는 아니며 이를 위해서는 의미적인 연구가 추가되어야 하는 매우 광범위한 연구가 필요한 문제라고 할 수 있다[2]. 본 논문에서 제안한 구문 범주 예측에 의하여 "분할 2"를 관계절(relative clause)로 예측하고 관계절로 분석된 결과를 취한다면 [번역 2]의 올바른 번역을 얻게 된다.

이와 같이, 구문 범주 예측은 구문 분석기를 수정하지 않고 구문 분석기가 해야 할 각 분할의 올바른 결과 선정과 그 역할을 판단하는 기능을 도와주는 역할을 한다. 이를 통해 문장 분할을 이용한 문장 분석의 정확성을 향상시켜서 보다 자연스러운 번역을 생성하는 것을 목적으로 한다.

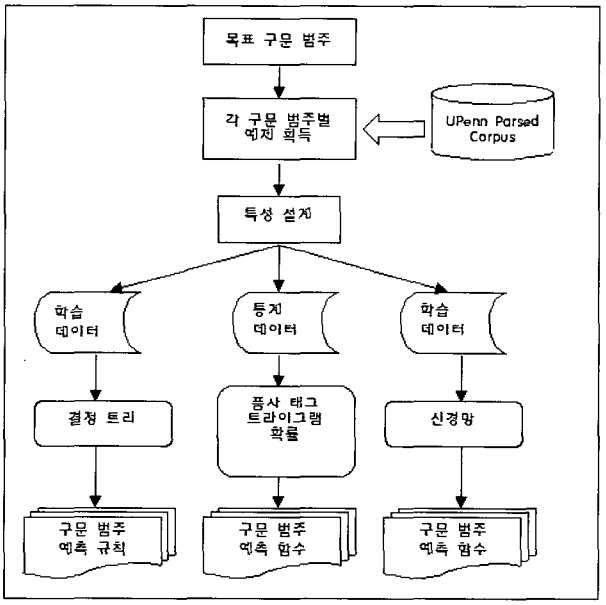


(그림 2) 문장 분할에 의한 문장 분석 과정

(그림 2)는 구문 범주 예측 과정이 추가된 문장 분할에 의한 문장 분석 과정을 보여준다. 영어 문장이 입력되면 어휘 분석(lexical analysis)을 통해 어휘 분석 결과가 문장 분할과정에 전달된다. 각 분할의 분석 이전에 분할의 구문 범주를 예측하고 예측된 구문 범주에 의해 부분 분석의 결과를 선정한다. 분석 결과의 합성 과정에서는 각 분할의 분석 결과에 따라 분할들을 결합하여 입력 문장의 최종 분석 구조를 생성한다. 이후 이 결과물이 한국어 생성(Korean generation) 단계로 전달되어 한국어 번역이 생성된다 [8].

3. 구문 범주 예측 방법

본 절에서는 Penn Treebank를 이용하여 구문 범주 예측 함수와 규칙을 생성하는 과정을 설명한다. 데이터를 이용한 통계적인 방법으로 품사 태그의 트라이그램 확률을 이용하는 방법과 결정 트리, 신경망 등의 기계학습적인 방법을 설명한다. (그림 3)은 예측 규칙 또는 예측 함수의 생성과정을 보여준다.



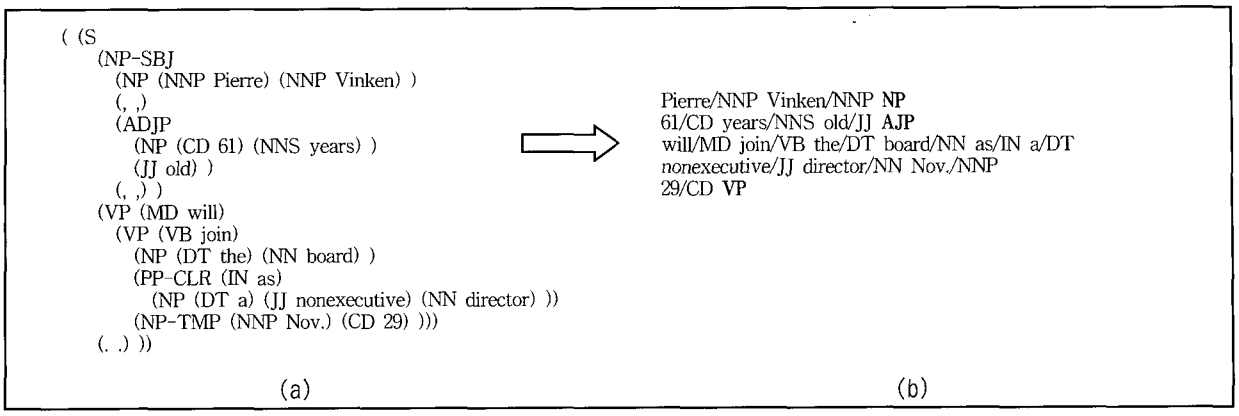
(그림 3) 구문 범주 예측 규칙과 함수의 생성 과정

3.1 목표 구문 범주

본 논문에서는 7가지 구문 범주를 예측의 대상으로 하였다: NP(명사구), VP(동사구), PP(전치사구), ADJP(형용사구), ADVP(부사구), SUBCL(종속절), RLCL(관계절). 영한 번역 시스템으로 문장 번역 테스트를 하는 과정에서 쉼표로 분리되는 분할들이 주로 위에서 열거한 구문 범주를 가졌다는 경험을 바탕으로 목표 구문 범주를 결정하였다.

여기에는 문장을 나타내는 SENT에 해당하는 구문 범주는 대상에서 제외하였는데 이는 SENT로 판단할 만한 분할에서의 특징을 찾기 힘들어 예측이 어렵기 때문이며 SENT를 예측한다 하더라도 다른 분할과의 관계를 파악하는 과정에 큰 유리한 점이 없기 때문이다. 그러나 다른 구문 범주에 대해서는 예측 결과를 이용하여 분할의 분석 결과 선정이나 다른 분할과의 관계를 분명하게 판단할 수 있는 근거 정보를 얻을 수 있으므로 예측의 구문 분석에서의 유용성을 기대할 수 있다.

2) SETN 구조 보다는 RLCL 구조에 높은 점수를 줄 수 있도록 문법의 점수 체계를 조정하여 쉽게 오류를 수정할 수 있지만 이러한 방법은 다른 문장의 경우에 올바른 구조 선택을 방해할 수 있으므로 근본적인 해결책이 될 수 없다.



(그림 4) Penn Treebank로부터 구문 범주별 예제 추출

3.2 각 구문 범주별 예제 획득

이 단계에서는 Penn Treebank로부터 대상이 되는 구문 범주에 해당하는 구나 절을 추출한다. Penn Treebank에는 구문 분석의 결과가 태그되어 있으며 태그된 정보들을 함께 추출하여 각 단어를 [단어/품사 태그]의 형태로 표현하고, 하나의 구나 절 뒤에 이들 구나 절의 구문 범주를 함께 기록한다. (그림 4)는 Penn Treebank로부터 예제의 추출 과정을 보여준다.

(그림 4)의 (a)는 "Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29." 문장의 구문 분석된 결과이고 (b)는 분석 결과에서 쉼표로 분리되는 3개의 분할들을 나타낸다. [단어/품사 태그]열과 분할의 구문 범주(굵은 글씨로 표현)로 하나의 분할을 표현한다.

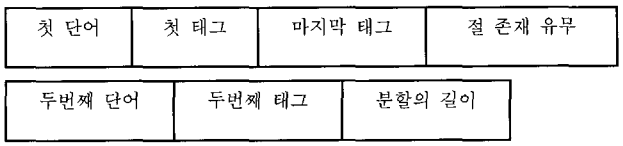
Penn Treebank에서 사용하는 구문 범주와 영한 번역 시스템에서 사용하는 구문 범주가 일치하지 않기 때문에 구문 범주 변환 과정이 필요하다. 그리고 Penn Treebank에는 "SBAR-*" 유형의 구문 범주 값이 존재하는데 이들을 영한 번역 시스템에서 사용하는 구문 범주로 변환하여야 한다. <표 1>은 Penn Treebank에서 사용한 구문 범주와 영한 번역 시스템에서 사용하는 구문 범주간의 대응 관계를 보여준다.

〈표 1〉 구문 범주 대응 관계

Penn Treebank 구문 범주	대응 구문 범주	비고
NP	NP	
VP	VP	PASTP, PRESP 포함
ADJP	AJP	
ADVP	AVP	
PP	PP	
WHADVP	SUBCL	
SBAR-TMP	SUBCL	
SBAR-ADV	SUBCL	
SBAR-PRP	SUBCL	
SBAR-1	RELCL	
SBAR-2	무시	
SBAR-CLR	무시	
SBAR-MNR	SUBCL	
SBAR	RELCL	"that"으로 시작
	SUBCL	다른 경우

3.3 특성 설계

구문 범주를 결정하는데 고려되는 요소들을 추출하여 학습을 위한 특성(feature)을 설계하였다. 본 논문에서는 7가지 요소들이 구문 범주 예측에 필요하다고 판단하여 (그림 5)와 같이 특성을 설계하였다.



(그림 5) 구문 범주 예측을 위한 특성

분할의 첫 단어와 두번째 단어, 그들의 품사 태그, 분할의 마지막 품사 태그, 분할의 길이, 그리고 분할이 절을 될 수 있다는 것을 알려주는 단서[3] 등 7가지 요소를 3.2절에서 설명한 구문 범주별 예제에서 추출하고 목표 구문 범주를 추가하여 구문 범주 예측 함수와 규칙을 획득하기 위한 학습 데이터를 구성한다.

3.4 학습 데이터와 구문 범주 예측 규칙과 함수 생성

본 논문에서는 결정 트리와 신경망 등의 기계 학습 방법과 품사 태그 트라이그램의 확률에 의한 통계적인 방법을 구문 범주 예측 함수와 규칙의 생성에 이용하였다. 본 절에서는 각 방법을 위한 학습 데이터 구성에 대해서 설명한다. (그림 6)은 구문 범주별 예제로부터 결정 트리와 신경망 학습을 위한 학습 데이터를 생성하는 예를 보여준다[4].

아래의 그림에서 (a)는 Penn Treebank에서 추출한 구문 범주별 예제 3가지를 보여주고 있으며 (b)와 (c)는 각각 이들 예제로부터 생성한 결정 트리 학습을 위한 3개의 데이터와 신경망 학습을 위한 3개의 데이터이다. 각 데이터는 (그림 5)의 7개의 필드에 대응하는 7개의 요소와 목표 구문 범주로 구성된다. 단어에 해당하는 첫번째와 다섯번째 요소는

3) 절의 주 동사(main verb)가 될 수 있는 품사 태그인 MD, VBD, VBZ, VBP와 절을 이끌 수 있는 관계 대명사 품사 태그인 WDT, WP의 존재 유무를 나타낸다.

4) 그림 (b)에서 '*NONE*'은 해당 필드의 값이 존재하지 않음을 의미한다.

Pierre/NNP Vinken/NNP NP
61/CD years/NNS old/JJ AJP
will/MD join/VB the/DT board/NN as/IN a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD VP

(a)

2, NNP, NNP, *NONE*, 2, NNP, 2, NP	2	13	13	35	2	13	2	0
0, CD, JJ, *NONE*, 4139, NNS, 3, AJP	0	1	6	35	4139	12	3	3
4085, MD, CD, MD, 1920, VB, 10, VP	4085	10	1	10	1920	25	10	1

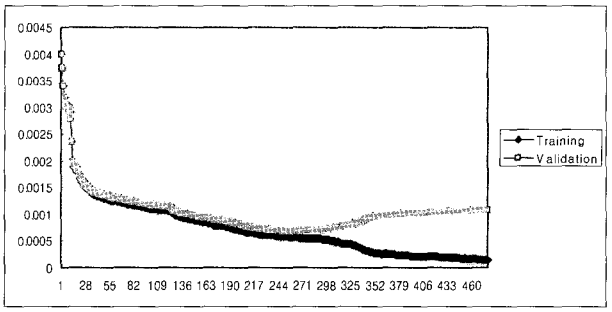
(b) (c)

(그림 6) 결정 트리와 신경망 학습을 위한 학습 데이터 생성의 예

수치 값(numerical value)으로 변환하였다. 이를 위해 Wall Street Journal에 나타나는 모든 단어를 추출하여 정렬(sort)한 후 각 단어의 정수 값을 결정하였다. 고유 명사, 관사, 수치 값에 대해서는 하나의 정수 값을 할당하였다. (그림 6)의 (b)의 첫번째 데이터는 [2, NNP, NNP, *NONE*, 2, NNP, 2, NP]인데 이는 첫 단어가 고유 명사이며, 첫 태그와 마지막 태그는 NNP이고, 분할 내에 절(clause)이 존재하지 않고, 두번째 단어가 고유 명사이고 두번째 태그가 NNP이며, 분할의 길이가 2인 NP(명사구) 분할이라는 것을 나타낸다. 결정 트리 학습 방법은 심볼 값(symbolic value)을 입력으로 사용할 수 있기 때문에 예제로부터 생성된 데이터 (b)를 바로 학습에 사용할 수 있으나 신경망은 수치 값을 입력으로 해야 하므로 심볼 값을 정수 값으로 변환하여 (c)와 같은 데이터를 생성하였다. 즉, 품사 태그와 구문 범주 심볼을 정수 값으로 변환하였다. (c)의 첫번째 데이터는 [2 13 13 35 2 13 2 0]인데 이는 NNP→ 13, *NONE*→35, NP→0으로 변환한 결과이다. 22,112개의 학습 데이터를 생성하여 결정 트리와 신경망 학습을 하였다.

본 논문에서는 구문 범주 예측 문제를 분류 문제(classification problem)으로 간주하고 분류 문제에 널리 활용된 기계 학습(machine learning) 방법인 결정 트리 학습을 이용하여 규칙을 생성하고자 하였다. 결정 트리 학습 결과 초기에 146,373개의 노드로 구성되는 트리가 생성되었으며 트리의 가지치기(tree pruning) 과정을 통해 최종적으로 37,132개의 노드를 가지는 트리를 생성하였다. 이 트리를 이용하여 학습 데이터의 155개의 구문 범주 예측 규칙을 생성하였다. 구문 범주 예측 과정에서 특정 단어의 역할을 고려하기 위하여 단어를 특성에 포함시켰으며 단어가 약 4500개가 존재하기 때문에 결정 트리가 매우 많은 노드로 구성되었다. 단어를 특성에 포함시키지 않는다면 수천개의 노드로 구성된 초기 트리가 생성되고 이후에 수백개의 노드를 가지는 트리로 변환될 수 있다.

또 널리 알려진 기계 학습 방법인 신경망 학습도 분류 문제에 적용될 수 있으므로 구문 범주 예측을 위한 신경망을 생성하였다. 결정 트리 학습에서 사용한 같은 데이터를 이용하였는데 신경망 학습에서는 22,112개 중 3,000개를 검증 데이터(validation data)로 나머지 19,112개를 학습 데이터로 사용하였다. Early stopping을 사용하여 검증 데이터의 에러가 증가되는 시점에서 학습을 종료시켰으며 학습 알고리즘은 scaled conjugate gradient를 사용하였다 [9]. 신경망은 2개의 은닉 계층(hidden layer)으로 구성되며 출력 층(output layer)은 7개의 뉴런(neuron)으로 이루어진다. 다양한 조합에 대해 신경망 학습 후 검증 에러(validation error)가 가장 작은 모델을 선정하였다. 신경망의 은닉 계층에는 *tansig*(즉, hyper tangent 함수)를 사용하였으며 출력 층에는 *sig*(즉, sigmoid 함수)를 사용하였다[5]. (그림 7)은 신경망의 학습 과정에서 학습 에러와 검증 에러의 감소 모습을 보이고 있다. 세로축의 값은 에러 제곱의 평균(mean squared error: MSE)값이며 가로축은 epoch의 수를 나타낸다. 261 epoch에서 검증 에러가 최소가 되었으며 이 때 생성된 모델을 구문 범주 예측을 위한 모델로 선정하였다.



(그림 7) 신경망 학습 과정

품사 태그 트라이 그램의 확률을 이용하는 방법에서는 분할의 첫번째 태그, 두번째 태그, 마지막 태그 정보만을 이용하여 은닉 마르코프 모델(hidden Markov model)의 접근법을 이용하였다. 즉 목표 구문 범주 C가 주어지면 출력단에는 품사 태그 트라이그램이 나타난다는 가정에서 출발하여 구문 범주 예측의 목표는 태그 트라이그램이 나타났을 때 C를 결정하는 것이고 가장 가능성이 높은 구문 범주 C'는 다음과 같이 구할 수 있다.

$$C' = \arg\max_{C} \Pr(C \mid trigram) = \arg\max_{C} \frac{\Pr(C)\Pr(trigram \mid C)}{\Pr(trigram)} \quad \text{(식 1)}$$

5) [첫번째 hidden layer, 두번째 hidden layer, 출력층 뉴런의 수] = [15 10 7]의 신경망 구조가 가장 에러가 작아서 이 구조의 신경망을 이용하여 구문 범주 예측 함수를 생성하였다.

식 (1)에서 분모에 해당하는 것은 동일한 분할에 대해서는 항상 같은 값을 가지므로 실제 구문 범주 결정에는 아무런 영향을 주지 못하므로 다음의 식을 이용하여 주어진 분할의 구문 범주를 결정한다.

$$C' = \arg\max_{C} \Pr(C)\Pr(trigram \mid C) \quad \text{(식 2)}$$

본 논문에서는 (그림 6)의 (a)와 같은 예제에서 $Pr(C)$와 $Pr(trigram|C)$를 계산하였으며 주어진 분할에 대해서 식 (2)를 이용하여 모든 가능한 구문 범주에 대해 확률 값을 계산하여 가장 높은 확률을 가지는 구문 범주를 분할의 구문 범주로 결정하였다.

4. 실 험

4.1 데이터 생성

본 논문에서는 미국의 펜실바이나 대학에서 구축한 구문 구조가 태그된 말뭉치(parsed corpus)를 이용하여 구문 범주 예측을 위한 데이터를 생성하였다. Wall Street Journal에서 추출한 문장의 구문 분석 결과가 태그된 말뭉치에서 쉼표로 구분되는 절이나 구를 추출하여 구문 범주 예측을 위한 특성을 고안하였다. 3장에서 설명한 방법들을 위한 학습 데이터 생성을 위하여 Wall Street Journal의 427,180 단어로 구성된 19,697 문장의 구문 분석된 말뭉치로부터 추출한 학습 데이터의 구문 범주별 분포는 <표 2>와 같다.

〈표 2〉 학습 데이터의 구문 범주별 분포

구문 범주	데이터 개수
NP	14,091
VP	1,499
ADJP	3,806
ADVP	334
PP	1,585
SUBCL	751
RELCL	46
합계	22,112

본 논문에서는 구문 범주 예측 성능을 평가하기 위한 테스트 데이터와 구문 범주 예측이 번역의 품질(translation quality) 향상에 기여하는 정도를 평가하기 위한 테스트 데이터를 구축하였다. 학습 데이터와 같은 Wall Street Journal, 그리고 Brown 말뭉치, IBM 매뉴얼, ECTB 말뭉치 등의 구문 분석된 말뭉치에서 테스트 데이터를 추출하였다. 그리고 번역 품질 향상 평가를 위해 컴퓨터 영역, 정치, Wall Street Journal, 그리고 고등학교 교과서에서 문장을 추출하였다. <표 3>, <표 4>는 테스트 데이터에 대한 통계를 보여준다.

〈표 3〉 구문 범주 예측 성능 평가를 위한 테스트 데이터

영역	문장 개수	데이터 개수
WSJ	4,000	4,626
Brown	4,001	3,821
IBM	4,404	1,403
ECTB	3,825	4,834
합계	16,230	14,684

〈표 4〉 번역 품질 향상 평가를 위한 테스트 데이터

영역	문장 개수
WSJ	202
컴퓨터	58
정치	49
교과서	60
합계	369

4.2 구문 범주 예측 방법의 평가

3장에서 설명한 것처럼 본 논문에서는 결정 트리, 신경망, 품사 태그 트라이그램 등을 이용하여 구문 범주 예측을 위한 규칙과 함수를 생성하였다. 각 방법의 성능은 예측의 정확률(accuracy)로 평가하였다. 정확률은 테스트 데이터의 수와 예측이 적중한 데이터 수의 비로 정의된다. <표 5>는 각 방법의 구문 범주 예측 정확성에 관한 평가 결과를 보여준다.

〈표 5〉 구문 범주 예측 방법의 성능 평가[6)]

	결정 트리	트라이그램 확률	전문가에 의한 규칙	신경망
WSJ	4486/4626 **(97%)**	4274/4626 (92%)	4065/4626 (88%)	4408/4626 (95%)
Brown	3603/3821 **(94%)**	3367/3821 (88%)	3171/3821 (83%)	3478/3821 (91%)
IBM	1331/1403 **(95%)**	1096/1403 (78%)	1193/1403 (85%)	1211/1403 (86%)
ECTB	4671/4834 **(97%)**	4353/4834 (90%)	4359/4834 (90%)	4623/4834 (96%)
평균	14091/14684 **(96%)**	13090/14684 (89.1%)	12788/14684 (87.1%)	13720/14684 (93.4%)

<표 5>에서 각 테스트 데이터 영역마다 가장 좋은 성능을 나타내는 정확률은 진한 글씨로 표시하였다. 실험 결과 결정 트리가 생성한 규칙에 의한 구문 범주 예측 방법이 모든 영역에서 가장 좋은 결과를 보여주고 있다. 트라이그램 확률을 이용한 방법은 구현이 간단하다는 장점을 지니면서도 높은 정확률을 보이고 있다. 전문가가 구축한 규칙은 사람의 노력을 가장 많이 필요로 하면서도 현재 성능이 가장 낮은데 이는 사람에 의한 규칙의 한계 때문이라 할 수 있다. 신경망에 의한 예측 함수도 어느 정도 높은 성능을 보이고 있지만 결정 트리에 의한 방법에 비해서는 약간 성능이 떨어진다고 할 수 있다.

6) 예측 성공 수/테스트 데이터 수, (정확률, %).

결정 트리가 생성한 구문 범주 예측 규칙은 해석이 용이하고 오류 검증을 통해 성능 개선이 용이하다는 추가의 장점을 가진다. 이에 비해 트라이그램이나 신경망에 의한 방법들은 어떤 방식으로 특정한 결과가 도출되는지 해석할 수 없으며 성능 개선을 위해서 추가의 데이터를 이용한 학습을 해야 하는데 이를 통해서 만족할만한 성능의 개선이 가능한지 판단할 수 없다는 문제를 지닌다.

4.3 구문 범주 예측에 의한 번역 품질 개선의 평가

<표 5>에서 보듯이 결정 트리가 가장 좋은 성능을 보였으므로 결정 트리가 생성한 규칙을 영어 구문 분석기에 통합하여 구문 범주 예측에 의한 번역 품질 향상을 평가하였다[7]. 결정 트리 학습 방법에 의해서 155개의 규칙을 얻었으며[8] <표 6>은 각 목표 구문 범주별 규칙의 수를 보여준다.

〈표 6〉 목표 구문 범주별 예측 규칙의 개수

목표 구문 범주	규칙 개수
NP	35
VP	14
ADJP	16
ADVP	34
PP	34
SUBCL	21
RELCL	1
합계	155

번역 품질의 평가를 위해 7명이 평가에 참여하였으며 100점 만점으로 번역문을 평가하였다. 구문 범주 예측을 적용한 경우와 그렇지 않은 경우 번역이 다른 문장에 대해서만 평가를 하였다. 구문 범주 예측이 얼마나 많은 문장의 번역 품질 향상에 도움을 주었는지 판단하기 위해 <표 7>에서는 구문 범주 예측 규칙의 적용률(coverage)을 보여주며 <표 8>은 두 가지 경우에 번역 결과가 다른 문장의 수, 즉 번역 품질 개선 평가를 위한 문장의 수를 영역별로 보여준다.

〈표 7〉 구문 범주 예측 규칙의 적용률[9]

영역	문장 분할 개수	구문 범주가 예측된 분할 개수
WSJ	427	250 (58.55%)
컴퓨터	169	112 (66.27%)
정치	153	101 (66.01%)
교과서	150	85 (56.67%)
합계	899	548 (60.96%)

〈표 8〉 번역 품질 개선 평가에 사용된 문장의 수

영역	문장 개수	전체 문장과의 비율 (%)
WSJ	38	18.8
컴퓨터	27	46.6
정치	14	28.6
교과서	8	13.3
합계	87	23.6

<표 7>에서 볼 수 있듯이 쉼표로 분리되는 문장 분할의 약 61%에 대해서 구문 범주가 예측됨을 알 수 있다. 구문 범주 예측 규칙은 문장(SENT) 범주에 대한 예측을 하지 않고 있으며 한 문장이 문장 범주에 해당하는 하나의 주절(main clause)가지고 있다고 가정할 때 구문 범주 예측이 되지 않은 나머지 39%는 전체 문장의 개수(369개) 만큼에 해당한다고 볼 수 있다. 따라서 예측 대상이 되는 거의 모든 분할에 대해서 구문 범주 예측이 이루어지고 있다고 판단되며 이는 결정 트리가 생성한 규칙이 실제 응용에서 높은 적용률을 보인다고 할 수 있다.

<표 8>을 통해 약 24%의 문장에 대해 구문 범주 예측에 의해 다른 번역 결과가 생성되었음을 알 수 있다. 이는 현재 예측된 구문 범주를 유용하게 이용할 수 있는 체제가 구문 분석기에서 확립되지 않으며 단지 구문 범주 예측 기능만을 현재의 구문 분석기에 추가하여 문장 분석을 하였기 때문이라 할 수 있다. 이 문제는 앞으로의 과제로서 구문 분석기의 성능 향상에 중요한 영향을 미칠 수 있을 것이다.

<표 9>는 평가자에 의한 평가 결과를 평가자가 번역 결과에 준 평균 점수 값으로 보여준다. Wall Street Journal의 문장에 대해서는 약 4점, 컴퓨터 영역의 문장에 대해서는 약 3점 정도 구문 범주 예측을 이용한 번역이 높은 점수를 보였으며, 정치와 교과서 영역의 문장에 대해서는 각각 1.4점, 2점 정도의 차이를 보였다. 영역마다 약간씩의 차이는 있으나 모든 영역에서 구문 범주 예측을 이용한 경우에 번역의 품질이 개선됨을 알 수 있다. 비록 품질 개선이 큰 폭으로 이루어지지 않았다고 할 수 있지만 기계번역의 다른 과정에서 유발되는 문제에 의한 영향[10]도 반영되어 번역 결과에 오류가

〈표 9〉 구문 범주 예측에 의한 번역 품질 향상 평가 결과[11]

	WSJ		컴퓨터		정치		교과서	
	NO-CP	CP	NO-CP	CP	NO-CP	CP	NO-CP	CP
평가자 1	61.05	66.97	60.93	63.93	63.57	68.57	65	68.36
평가자 2	81.58	84.21	79.59	84.63	85.81	88.43	75	76.88
평가자 3	84.87	87.89	83.52	86.11	87.5	87.5	91.88	89.38
평가자 4	77.89	82.5	78.89	82.41	74.29	80.36	80	85
평가자 5	80	84.34	85.19	89.44	90.71	89.29	90.62	91.25
평가자 6	73.79	77.16	77.78	79.48	72.64	72.14	82.38	84.38
평가자 7	76.45	79.61	79.26	79.63	78.57	78.57	81.88	85.62
평균	76.52	80.38	77.88	80.80	79	80.41	80.96	82.95

7) 통합과정에서 결정 트리가 생성한 규칙은 UPenn의 품사 태그 집합으로 표현되었는데 이를 구문 분석기가 사용하는 품사와 자질 정보로 변환하여 구문 범주 예측 규칙을 변환하여 기존의 구문 분석기와 통합하였다.

8) 결정 트리에 의한 디폴트 목표 구문 범주(default target category)는 명사구(NP)인데 이는 구문 분석기에 적합하지 않으므로 디폴트 목표 구문 범주를 사용하지 않도록 규칙을 수정하여 구문 분석기에 통합시켰다. 즉, 디폴트 값을 사용하는 경우에는 구문 범주를 예측하지 않도록 수정하였다.

9) 적용률은 구문 범주 예측의 단위가 되는 문장 분할의 개수와 구문 범주가 예측된 분할 개수의 비로 표현된다.

10) 전치사의 번역 문제, 단어의 대역어 선정 문제, 긴 문장 생성에서의 어순 문제에 의해 번역문에 오류가 포함될 수 있다.

11) 표에서 "NO-CP"는 구문 범주 예측(category prediction)을 사용하지 않은 경우이고 "CP"는 구문 범주 예측을 사용한 경우이다.

포함될 수도 있기 때문에 구문 범주 예측이 번역 품질 향상에 기여를 하지 못했다고 할 수 없다. 또한 위에서 언급했듯이 구문 범주 예측 결과를 활용하는 구문 분석기 내에서의 알고리즘을 고안함으로써 번역 품질을 보다 많이 개선할 수 있을 것으로 판단된다.

5. 결 론

본 논문에서는 영한 기계번역에서 구문 분석의 정확성 향상을 통한 번역 품질의 개선을 위해 구문 범주 예측을 제안하였으며 데이터를 이용한 통계적, 기계학습적 방법에 의해 예측 규칙을 생성하는 방안을 제시하였다. 구문 분석의 효율성 제고를 위한 문장 분할을 적용한 구문 분석에서 각각의 분할은 독립적으로 분석된다. 이들 분할의 구문 범주를 분석 이전에 예측하여 보다 정확한 분할의 분석 결과를 선택하게 하고 입력 문장의 구문 분석 정확도를 높임으로써 결과적으로 번역 품질을 향상시키는 것이 본 논문에서 제안한 구문 범주 예측의 목표이다.

펜실바니아 대학에서 구축한 Penn Treebank를 사용하여 학습 데이터를 생성하였으며, 결정 트리, 품사 태그 트라이그램 확률, 신경망 등의 방법으로 구문 범주 예측 규칙과 함수를 생성하였다. 실험 결과 결정 트리가 생성한 규칙이 가장 높은 예측의 정확도를 보였으며 따라서 이를 구문 분석기에 통합하여 번역 품질 개선 평가를 하였다. 문장 분석 과정에서 구문 범주 예측은 높은 적용률을 보였으며 다양한 영역의 문장에 대해서 모두 번역 품질의 개선에 기여하였음을 실험 결과를 통해 확인하였다.

앞으로의 과제로서 결정 트리가 생성한 구문 범주 예측 규칙의 정확도 향상을 위해 전문가가 구축한 규칙을 활용할 필요가 있다. 예측 에러의 분석을 통해 예측 규칙의 정확도를 개선할 수 있을 것으로 판단된다. 이는 통계적인 방법과 규칙 기반 방법을 접목하는 하이브리드(hybrid) 방법의 일종이라 할 수 있다. 그리고 4장에서 언급하였듯이, 번역 품질의 보다 많은 개선을 위해서는 구문 범주 예측의 결과를 활용하는 알고리즘을 고안해야 할 것이다. 또한 구문 범주 예측은 구문 분석 과정에서 적용될 규칙들을 여과(filter) 할 수 있게 함으로써 구문 분석의 시간/공간 면에서의 복잡도를 낮추는 데도 기여할 것이다. 구문 분석 예측은 쉼표로 분리되는 분할들로 이루어진 긴 문장의 구문 분석 방법에 대한 새로운 방법을 연구하는 데도 기여할 것으로 기대된다.

참 고 문 헌

[1] 김성동, "효율적인 영어 구문 분석을 위한 최대 엔트로피 모델에 의한 문장 분할" 한국정보과학회 논문지. 제32권 제5호, pp. 385-395, 2005.

[2] Sung-Dong Kim, Byuong-Tak Zhang, Yung Taek Kim, "Learning-based Intrasentence Segmentation for Efficient Translation of Long Sentences," Journal of Machine Translation. Vol.16, No.3, pp.151-174, 2001.

[3] 김성동, 김영택, "효율적인 영어 구문 분석을 위한 문장 분할", 한국정보과학회 논문지. 제24권 제8호, pp.884-890, 1997.

[4] Abney, Steven. 'Parsing by Chunks,' Principle-Based Parsing, Robert Berwick, Steven Abney and Carol Tenny (eds). Kluwer Academic Publishers, pp.257-279, 1991.

[5] Abney, Steven, 'Chunks and Dependencies : Bringing Processing Evidence to Bear on Syntax. Computational Linguistics and the Foundations of Linguistic Theory,' Jennifer Cole, Georgia M. Green and Jerry L. Morgan (eds). *CSLI Publications*, pp.145-164, 1995.

[6] Abney, Steven, "Partial Parsing via Finite-State Cascades," In Proceedings of ESSLLI Workshop on Robust Parsing Workshop, Praque, 1996.

[7] Sung Dong Kim and Yung Taek Kim, "Sentence Analysis using Pattern Matching in English-Korean Machine Translation," In Proceedings of 1995 International Conference on Computer Processing on Oriental Languages, pp.199-206, 1995.

[8] 양승현, "영한 기계번역을 위한 언어 스타일의 변환," 서울대학교 대학원 박사학위 논문. 1997.

[9] Martin Fodslette Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, Vol.6, pp.525-533, 1993.

김 성 동

e-mail : sdkim@hansung.ac.kr

1991년 서울대학교 컴퓨터공학과(학사)

1993년 서울대학교 컴퓨터공학과(석사)

1999년 서울대학교 컴퓨터공학과(박사)

1999년~2001년 서울대학교 컴퓨터신기술 공동연구소 특별연구원

2001년~현재 한성대학교 컴퓨터공학과 조교수

관심분야 : 기계번역, 자연언어처리, 데이터마이닝, 기계학습, 인공지능, 금융공학