2014년 8월 전자공학회 논문지 제 51권 제 8 호
Journal of The Institute of Electronics and Information Engineers Vol. 51, NO. 8, August 2014 http://dx.doi.org/10.5573/ieie.2014.51.8.066

논문 2014-51-8-8

한국어 의존 관계 분석과 자질 집합 분할을 이용한 기계 학습의 성능 개선

(Analysis of Korean Language Parsing System and Speed Improvement of Machine Learning using Feature Module)

김 성 진*, 옥 철 영*

(Seong-Jin Kim and Cheol-Young Ock©)

요 약

최근에 한국어 의존 관계에 대한 파싱 시스템과 관련된 연구가 소프트웨어 공학자들이나 언어학자들에 의해 다양하게 연구되고 있으며, 시스템 구현은 주로 기계 학습이나 기호 주의를 사용하고 있다. 기계 학습을 사용한 방법은 한국어 문장 데이터가 매우 크기 때문에 시스템 특성상 매우 긴 학습시간을 가지며, 데이터 자체가 가지는 오류로 인하여 한정된 인식율을 가진다. 본 연구에서는 기계학습을 이용한 시스템에 대하여 학습 시간을 줄일 수 있도록 특징들을 자질 집합 모듈로 분할하여 처리하는 방법을 제안하고, 문장수와 반복횟수에 따른 인식율을 분석하였다. 설계된 시스템은 분리된 모듈과 이진 검색을 위한 정렬 기법이 사용되었다. 데이터는 세종 말뭉치로부터 추출한 후 정제된 36,090문장을 사용하였다. 학습 시간은 약 3 시간으로 줄었으며, 인식율은 10,000 문장을 50회 학습하였을 때 84.54%로 가장 높았다. 모든 학습 문장(32,481)을 10회 학습하였을 때 인식율은 82.99%이다. 결과적으로 정제된 데이터를 이용하여 시스템이 안정화될 때까지 반복하는 것이 더 효율적이었다.

Abstract

Recently a variety of study of Korean parsing system is carried out by many software engineers and linguists. The parsing system mainly uses the method of machine learning or symbol processing paradigm. But the parsing system using machine learning has long training time because the data of Korean sentence is very big. And the system shows the limited recognition rate because the data has self error. In this thesis we design system using feature module which can reduce training time and analyze the recognized rate each the number of training sentences and repetition times. The designed system uses the separated modules and sorted table for binary search. We use the refined 36,090 sentences which is extracted by Sejong Corpus. The training time is decreased about three hours and the comparison of recognized rate is the highest as 84.54% when 10,000 sentences is trained 50 times. When all training sentence(32,481) is trained 10 times, the recognition rate is 82.99%. As a result it is more efficient that the system is used the refined data and is repeated the training until it became the steady state.

Keywords : 한국어 파싱, 한국어의 의존 관계, 기계 학습, 자질 집합

* 정회원, 울산대학교, 전기공학부
(Dept. of electrical engineering, Ulsan Univ.)
© Corresponding Author(E-mail: okcy@ulsan.ac.kr)
※ 본 연구는 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [10044508, 비기호적 기법 기반 인간모사형 자가학습 지능 원천기술 개발]
접수일자: 2014년07월09일, 수정일자: 2014년07월23일
수정완료: 2014년07월31일

Ⅰ. 서 론

자연어 처리 연구는 특유의 불규칙성과 복잡성 등으로 인하여 주로 규칙 기반의 기호주의를 이용하여 처리를 하고 있다[1]. 자연어는 그 단어수가 많을뿐더러, 유사한 의미를 가지는 것들이 많기 때문에 기계학습으로

는 한계가 있었다. 따라서 단어들을 특정 물리적 기호들로 표기하고 그들 간의 상호 관계를 규정할 수 있는 지능 시스템을 구현할 수 있다는 물리적 기호 시스템 가설[2]에 기초를 둔 기호주의 관련연구들이 많이 진행되었다. 그러나 기호주의에서 처리하지 못하는 복합적인 결정이나 추론 등이 자연어 처리에 유용함으로 신경망을 이용한 연결주의에 관련된 연구들도 진행 중이다[3].

연결주의는 신경 셀 하나에 객체 하나를 일대일로 사상시키는 국소 연결주의와 하나의 객체를 여러 신경 셀들의 조합을 사용하여 표현하는 분산 연결주의로 나뉜다[1, 4]. Waltz와 Pollack는 국소 연결주의를 이용한 어휘의미, 구문, 문맥, 입력의 4층을 가지는 신경망을 구성하여 입력 단어에 대하여 구문, 의미, 문맥정보를 결정할 수 있는 시스템을 제시하였으며[5], 신경망의 퍼셉트론을 이용한 MIRA(Margin Infused Relaxed Algorithm) 등이 개발되었다[6~7].

자연어 연구에 있어서 이런 기호주의나 연결주의 방법을 이용하여 문장을 이루는 단어나 어절들 사이의 문법적 구조를 분석하는 것을 구문 분석이라 한다[8]. 구문 분석은 주로 의존 구문 분석 방법과 구구조 구문 분석 방법이 주류를 이룬다. 한국어와 같이 비교적 어순이 자유로운 언어에서는 주로 의존 구문 분석 방법을 이용하여 어절들 사이의 관계를 설명하고, 어순이 고정적인 영어권에서는 구구조 구문 분석 방법을 사용하여 분석한다. 최근에는 CoNLL-X shared task이 제안한 방법을 이용하여 영어권에서도 의존 구문 분석에 대한 연구가 진행되고 있다[9].

의존 구문 분석은 다시 결정적 의존 구문 분석 방법과 비결정적 의존 구문 분석 방법으로 나뉜다. 본 연구에서 사용된 비결정적 의존 구문 분석은 문장이 가질 수 있는 모든 의존트리 중에서 최대신장트리를 이용한 가장 높은 점수의 의존트리를 선택하는 방법이다. 따라서 문법적 관계가 가능한 모든 의존소와 지배소의 쌍으로부터 가장 높은 점수의 의존트리를 찾기 위해 전역적 학습 모델을 이용한다. 최근의 연구에서는 McDonald의 그래프 기반 의존 구문 분석을 많이 사용한다[10]. 결정적 의존 구문 분석 방법은 탐욕적 알고리즘에 기반한 방법으로 지역적 학습 모델을 사용한다. 최근의 결정적 의존 구문 분석에 영향을 많이 준 연구로는 Nivre의 전이기반 의존 구문 분석이 있다[11].

본 연구에서는 연결주의 방법 중의 하나인 기계 학습(normalized perceptron) 기법을 이용하여 한국어 의존 구문 분석을 처리하는 가장 기본적인 알고리즘을 분석하고, 이에 대한 결과를 보인다. 구성은 Ⅱ장에서 관련 연구를 소개하고, Ⅲ장에서 한국어 문법에 대하여 간단히 기술한다. Ⅳ장에서 데이터 추출과 설계된 시스템을 설명하고 Ⅴ장에서 결론을 내린다.

Ⅱ. 기본 연구

McDonald가 제안한 알고리즘은 일정한 자질 집합을 정의하고 각 어절의 의존관계마다 그 자질 집합을 만든다. 그 후, 생성된 모든 가능한 의존 관계를 이용하여 간선을 포함하는 그래프를 만들고 그 안에서 가장 점수가 높은 최대 신장 트리를 이용하여 파스트리를 결정하는 것이다[10, 12~13]. 본 논문에서도 McDonald가 제안한 방법으로 각 문장에서 만들 수 있는 모든 자질을 이용하여 신경 셀을 생성한다.

서강대에서 제안한 SKA(Sogang Korean depend-ency Analyzer) 프로그램은 세종 구문구조 말뭉치를 의존구조 말뭉치로 전환하여 학습에 사용한다. 다수의 한국어 문장들을 파일로 입력하여 그 문장들의 형태소 분석 결과와 의존구조 분석 결과를 파일로 출력해주는 기능을 제공한다[14]. 세종 구문구조 말뭉치를 사용하여 학습하였고 한국어 문장 의존구조 분석에 대해 약 85.42%의 성능을 제공한다.

임수종 등은 자질(feature)의 가중치를 학습하여 이용하는 기계학습 기반 한국어 의존 파싱의 한 기법을 제안하였다. 제안된 시스템은 SKA와 유사하게 모든 가능한 의존관계에 대하여 일정한 수의 자질들을 생성하여 구축하고, 자질마다 중요도를 나타내는 가중치를 이용하였다. 이를 위해 세종 구구조 부착 코퍼스와 ETRI 의존구조 부착 코퍼스를 이용하였다. 시스템의 성능은 세종 코퍼스에 의하여 개발하는 경우 의존관계 정확도 88.15%, ETRI 코퍼스를 이용하여 개발하는 경우 의존관계 정확도 88.06%를 보인다고 하였다. ETRI에서 사용한 코퍼스[15]에는 70개의 품사를 사용하였다고 한다[13]. 세종 코퍼스의 품사는 45개이며 3개는 분석 불능에 포함됨으로 실제로 42개의 품사를 이용하는 것이기 때문에 품사 개수에서 상당한 차이가 있다.

안광모 등은 위 연구 결과를 바탕으로 인식 결과를

높이기 위하여 기호주의 기법을 이용한 지배소 후보 집합 개념을 포함시킨 알고리즘을 제안하였다. 모든 의존소와 지배소의 관계를 전역적으로 탐색하는 일반적인 비결정적 의존 구문 분석과는 다르게 각 의존소에 대하여 문법적 관계를 가질 수 있는 지배소 후보들을 제한하여 분석의 복잡도를 감소시켰다. 그리고 한국어 구문 분석 시 일반적으로 고려되는 교착어적 특징, 지배소 후위 및 투사성(projectivity) 원칙을 반영한다. 학습데이터 및 평가데이터로는 세종 구문 분석[16] 말뭉치를 의존 구문 분석 말뭉치의 형태로 변환하여 사용하였으며, 실험 결과는 아크단위 87.52%의 정확도(accuracy)와 문장단위 34.43%의 정확도를 보였다[8].

기계학습을 이용하여 한국어 기본구(base phrase)인식의 성능을 향상시키고자 할 때, 학습 집합으로부터 획득 가능한 자질집합들 중 최적의 자질집합과, 자료부족 문제를 어떻게 완화할 수 있는 지에 대한 방법을 제안하였다. 먼저 최적의 자질집합 선택은 “점증적 유용성“이란 관점에서 자질의 적합성을 정의하고 이러한 정의에 따라 자질집합을 선택한다. 그리고 자료부족 문제 완화의 해결점을 찾기 위해 한국어의 통사적 특성을 고려한 형태소 품사체계 사용 및 선택적 어휘자질의 사용이 성능에 미치는 영향을 분석하고 결과를 제시하였다[17].

이용훈 등은 CoNLL-X의 그래프 기반 의존구문분석 방법을 한국어에 맞게 변형하고 한국어의 특성에 맞는 자질 집합을 제시하였다. 그리고 이 자질 집합을 이용한 한국어 의존구문분석 방법 또한 제안하였다. 제안된 알고리즘에서는 의존트리의 에지를 단어와 단어 간의 의존관계가 아닌 부분트리와 부분트리의 의존관계로 바라보기 위해 부분트리가 공유하고 있는 기능어 정보를 추가 자질로 사용하였다. 제안 모델을 국어정보베이스(KIBS) 말뭉치에 적용한 결과 어절 단위 정확률 88.42%의 인식율을 보인다고 하였다[18].

Ⅲ. 한국어 의존 문법

1. 어절간의 의존 관계

한국어 문법은 크게 의존 문법과 구구조 문법으로 나뉜다. 의존 문법이란 문장을 구성하는 어절과 다른 어절들 사이의 의존관계를 파악함으로서 문장을 분석하는 것이다. 구구조 문법은 여러 언어요소가 모여서 구문 요소를 만들고 여러 구문요소가 모여서 더 큰 구문 요소를 만드는 방법이다. 따라서 문장의 분석 결과는 문장 전체가 부분들로 나누어지며 각 부분은 다시 몇 개의 더 작은 부분으로 나누어지는 계층화된 구조를 가진다.

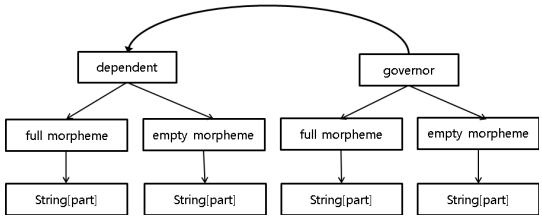

그림 1. 의존소와 지배소의 관계
Fig 1. Relation of dependent and governor.

의존 관계는 그림 1에서 보이는 바와 같이 두 어절 사이에 존재하며, 한 어절은 지배소가 되며 다른 한 어절은 의존소가 된다. 의존 문법에 의한 문장의 분석 결과는 문장 내의 가능한 모든 의존 관계들의 부분 집합이다. 통상 의존 관계에 있는 두 언어요소 중 지배소는 의미의 중심이 되는 요소가 되며, 의존소는 지배소가 갖는 의미를 보완해 주는 역할을 한다. 문장에서 가장 기본적인 틀은 의존소가 주어(체언)이고 지배소가 서술어(용언)일 때이다. 각 어절은 내용 형태소와 기능 형태소가 있으며, 기능 형태소는 생략되는 경우가 있다. 그리고 의존소의 기능형태소가 의존관계에서 중요한 역할을 한다. 본 논문에서는 세종코퍼스에서 제공하는 45개의 품사를 사용하였다. 자연어 특성상 이런 품사 분류도 특정 규칙을 가지는 것이 아니기 때문에 이와 관련된 연구도 다양하게 진행 중이다[19].

의존 문법을 이용하여 한국어 구문 분석을 하는 이유는 첫째로 어순의 자유성에 의한 어절의 위치 문제가 의존문법에서는 쉽게 해결되며, 둘째로 구성요소의 불연속성이나 구성요소의 생략 등과 같은 현상에 큰 영향을 받지 않으며 따라서 매우 견고성이 있는 파싱 방법을 구축할 수 있기 때문이다[20].

2. 데이터 추출과 분석

표 1에서와 같이 세종 코퍼스의 구구조 문법을 의존구조로 변환한 후 UTagger[21]를 이용하여 어깨번호를 부착한 데이터를 이용하여 실험 하였다. UTagger란 HMM 기반의 한국어 품사 및 동형이의어 동시 태깅시스템이며 어깨번호란 동형이의어를 명확히 구분하기 위

표 1. 원본 데이터와 정제된 데이터
Table 1. Refinement of raw data.

item	raw data	refined data
total sentence	61,553	39,300
1 phrase sentence	4,503	1,687
2 phrase sentence	2,061	1,523
more than 3 phrase	54,991	**36,090**
error type 1 : Q error (ex. Q=2[Q])	275	-
error type 2 : string*/NN	string/NN + */SW (3,488)	-
error type 3 : ex. /*/SP	//SP + */SW + //SP	-

```
example.
- 1 phrase :
    1   0   초/XPN + 미니/NNG + .../SE
- 2 phrase :
    1   2   1993/SN + //SP + 07/SN + //SP + 10/SN
    2   0   09/SN
- error sentence :
    /SS + Q=2/Q + "/SS
```

그림 2. 오류 예제
Fig 2. Example of Error.

하여 사전 등에서 사용하는 번호이다. 표에서와 같이 원본 데이터는 실험에 불필요한 데이터가 다수 존재하였으므로 이를 전처리하였다.

원본 코퍼스에는 예제와 같은 1어절과 2어절 문장이 다수 존재하며, 실험에서는 불필요한 데이터이므로 모두 삭제한다. 그리고 큰 따옴표를 처리함에 있어서, 큰 따옴표 내부의 문장들로 새로운 문장을 만들고 Q로 처리함으로서 의존관계 처리에 오류를 발생하므로 위와 같은 문장들도 모두 삭제하였다. 그 외에도 문자*/NN은 문자/NN+*/SW로 수정하였으며, /*/SP 또한 //SP + */SW + //SP로 수정하였다.

표 2. 한국어 의존 관계 거리
Table 2. Distance of dependency relation.

distance of dependency	number of phrase	rate[%]
0	57,052	8.42
1	357,693	52.80
2	91,442	13.50
3	49,177	7.26
4	29,414	4.34
5	20,116	2.97
6	14,744	2.18
7	11,052	1.63
8	8,588	1.27
9	6,661	0.98
10	5,381	0.79
≥ 10	26,139	3.86
total	677,459	100

표 3. 내용 형태소와 기능 형태소
Table 3. Example of full and empty morpheme.

원본 데이터		정제된 데이터	
34,336	580	32,301	433
통괄[NNG]	꼬[EC]	필드__01[NNG]	옵는지[EC]
하복부[NNG]	이나[JC]	풍장__01[NNG]	소냐[EF]
선경[NNG]	쌀[XR]	동소문[NNP]	바[EC]
동소문[NNP]	냐니까[EF]	해치슨[NNP]	[XPN]
삼선평[NNP]	니까요[EC]	경보__09[NNG]	즉슨[EC]
해치슨[NNP]	는구먼[EF]	야학출롱[NNP]	느니[EF]
경려[NNG]	닥[XR]	삼단__02[NNG]	다던데[EC]
적인[NNG]	리만큼[EC]	옥토약월[NNG]	다고도[EC]

표 2는 시스템 구현에 앞서 코퍼스에 있는 의존관계의 거리를 나타낸 것이다. 표에서와 같이 의존구조로 변형하였을 때, 의존 거리 0은 제일 마지막 어절을 의미하며 실제 문장 개수이다. 의존거리 1은 바로 다음 어절로 의존 관계를 가지는 경우이며, 한국어에서는 반 이상이 바로 다음 어절로의 의존관계를 가지며, 대부분의 의존 관계가 6 어절 이내(91.47%)에 있다.

표 3의 원본 데이터는 세종 코퍼스에서 추출한 데이터이고 정제된 데이터는 추출 후, 불필요한 데이터는 삭제하고 UTagger를 이용하여 어깨번호까지 부착한 데이터이다. 학습 가능한 총 문장은 57,052문장이며 이 중에서 4,503은 1어절 문장이라 제거하였다. 총 의존수는 620,407개이며 평균어절은 11.88어절이고 최대 어절은 125개이다. 그리고 표 3은 추출된 형태소의 예제이다. 말뭉치에 사용된 내용형태소(실질형태소)는 총 32,301개이고 기능형태소(문법형태소)는 433개이다. 원본데이터로부터 의존 관계 오류등 문법적으로 오류가 있는 문장을 제거한 후, 실험에는 3어절 이상 정제된 36,090문장을 이용하여 학습데이터로 32,481(90%) 문장, 평가 데이터로 3,609문장을 사용하였다.

Ⅳ. 자질 테이블의 모듈화를 이용한 기계학습 시스템과 결과 분석

1. 모듈화를 이용한 기계학습 시스템

시스템은 SKA에서 사용된 자질 생성표(표 4)를 사용하였으며, 표 5는 생성된 자질의 예이다. 생성된 자질은 F1 자질과 같이 문자열(예. 프랑스__02)을 포함하여 생성될 때는 그 수가 많아지고 셀의 길이도 길어지며,

표 4. 자질 테이블
Table 4. Feature Table.

Feature	자질	
F1	M-FIP, H-FIP	M : 의존소 H : 지배소 P : 품사 Sm : 어절의 맨 우측 형태소의 기호 Sb : 의존소의 형제 + : 바로 옆 우측 어절 - : 바로 옆 좌측 어절 C : 내용형태소 F : 기능형태소 Dl : 거리(어절수의등급) Dy : 거리(용언수) Cl : 마지막 내용형태소 Fl : 마지막 기능형태소 Sc : 어절의 컴마기호 Sq : 어절의 따옴표기호 yn : 유무 A : 어절 Sp : 어절의 괄호'[,]'기호 Eq() : 같은지 확인 Am : 형태소
F2	MFIP, HFIP	
F3	MA, HA	
F4	MClAm, HClAm	
F5	MCIP, HCIP	
F6	MFIAm, HFIAm	
F7	M+CIP, H+CIP	
F8	MSmAm, HSmAm	
F9	MScyn, HScyn	
F10	MSqyn, HSqyn	
F11	MFIP, MSmAm, HCIP	
F12	MFIAm, MFIP, MSmAm, HCIP	
F13	MFIAm, MSmAm, HClAm	
F14	MCIP, MFIP, MSmL, MSmP, HCIP	
F15	MCIP, MFIAm, MFIP, MSmAm	
F16	MClAm, MFIAm, MSmAm, HClAm	
F17	MFIAm, MSmAm, HClAm	
F18	MCIL, MCIP, MSmAm, HCIP	
F19	MScyn, HScyn	
F20	MSpyn,HSpyn	
F21	Eq(MFIAm,HFIAm)yn	
F22	Eq(MFIP,HFIP)yn	
F23	Dl(M,H)	
F24	자식 개수 차이	
F25	Eq(MFIAm,HFIAm)yn	
F26	Eq(MFIAm,HFIAm)Am	
F27	Eq(MFIP,HFIP)yn	
F28	Eq(MFIP,HFIP)P	
F29	Dl(MFIP,HFIP)	
F30	Dl(MCIP,HCIP)	
F31	Dl(MFIAm,HFIAm)	
F32	Dl(MClAm,HClAm)	

```
1    4    프랑스__02[NNP]+의[JKG]
2    4    세계적[NNG]+이[VCP]+ㄴ[ETM]
3    4    의상__01[NNG]
4    6    디자이너[NNG]
5    6    엠마누엘[NNP]
6    11   웅가로[NNP]+가[JKS]
7    8    실내[NNG]
8    9    장식__05[NNG]+용__11[XSN]
9    10   직물[NNG]
10   11   디자이너[NNG]+로[JKB]
11   0    나서[VV]+었[EP]+다[EF]+.[SF]
```

그림 3. 학습에 사용된 예제 문장
Fig. 3. Example of sentence for trining.

F3 자질과 같이 품사(예. [NNP])들만으로 생성되는 자질 집합들은 데이터 길이도 짧아지고 생성되는 자실 수도 훨씬 작다. 그리고 F32번 자질들처럼 거리나 기호를 처리하기 위한 자질들의 셀 수는 매우 작다.

SKA에서는 그림 4와 같은 구조로써 생성되는 자질을 하나의 테이블에 모두 저장한다. 사용된 시스템은 학습시간 보다는 기계학습을 이용한 한국어 파싱 시스

표 5. 자질 테이블 생성
Table 5. Creating Feature Table.

ex) 1. 프랑스__02[NNP]+의[JKG], 4. 디자이너[NNG] (* $ = 없음)
f1[!끝]$
f2[JKG]$
f3프랑스__02[NNP]+의[JKG]디자이너[NNG]
f4프랑스__02[NNP]디자이너[NNG]
f5[NNP][NNG]
f6[JKG]$
f7[NNG][NNP]
f8$$
f9[n][n]
f10[n][n]
f11[JKG]$[NNG]
f12[JKG][JKG]$[NNG]
f13[JKG]$$
f14[NNP][JKG]$[NNG]
f15[NNP][JKG][JKG]$[NNG]
f16프랑스__02[NNP][JKG]$디자이너[NNG]
f17[JKG]$디자이너[NNG]
f18[NNP]$[NNG]
f1900
f2000
f210
f220
f233
f24+
f25-
f26X
f27-
f28X
f29[JKG]$3
f30[NNP][NNG]3
f31[JKG]$3
f32프랑스__02[NNP]디자이너[NNG]3

템의 인식율에 중점을 두었기 때문에 학습시간에 대한 고려는 하지 않았다.

학습 과정에서는 자질 생성 후 특징 셀들을 저장하는 테이블에서 검색 후 없으면 마지막에 저장한다. 의존관계에 있는 어절들에 대한 가중치는 증가시키고 의존관계가 성립하지 않은 어절들에 대한 가중치는 감소시키는 방법으로 의존관계에 대하여 학습시킨다.

$$\Delta W(t+1) = W(t) \begin{cases} +1: \text{if } d_{ij} = r_{ij} \\ -1: else \end{cases} \quad (1)$$

W는 가중치를 의미하며 d_{ij}는 계산되어진 의존 관계 단어이고 r_{ij}는 정답 의존관계이다. 수식에서 가중치 변화량은 계산되어진 의존관계와 정답 의존관계가 같으면 증가, 아니면 감소한다.

인식 과정은 각 어절들의 각 feature에서 셀을 검색하여 그 가중치 값이 최대치가 되는 값을 선택한다. 따라서 특징 셀들의 개수가 8백만 개 이상으로 많아지면 검색하는데 시간이 매우 오래 걸린다.

$$answer_j = \arg\max\left(\sum_{i=1}^{32} W_{j+1}(i), \cdots, \sum_{i=1}^{32} W_n(i)\right) \quad (2)$$

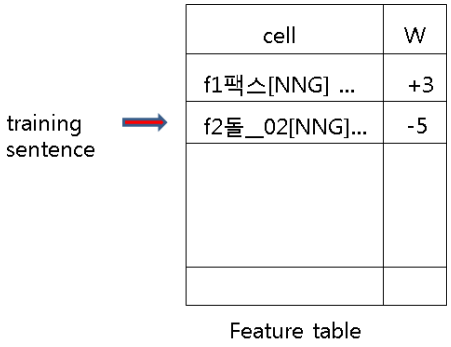

그림 4. SKA 자질 표
Fig. 4. SKA Feature Table.

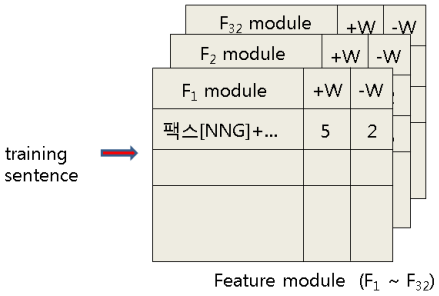

그림 5. 자질별 모듈 테이블 생성
Fig. 5. Feature Module Table.

answer$_j$는 문장에서 임의의 단어를 의미하며, 그 단어의 의존 관계는 그 다음 단어부터 문장의 끝까지 단어 중에서 가중치의 합이 제일 큰 단어가 선택된다.

이런 단점을 보완하기 위하여 수정된 시스템은 F_1에서 F_{32}까지 32개의 모듈을 만들고 생성 자질에 따라 각 모듈에 저장하는 방법을 사용하였다. 뿐만 아니라 모듈에 저장할 때 값들을 오름차순으로 정렬하고 이진 검색 방법을 사용하여 검색 속도를 높였다.

그러나 정렬된 테이블에서 이진 검색을 사용하는 것조차도 F32의 경우 2백만 개의 데이터 셀이 만들어 지기 때문에 검색 시간이 오래 걸린다. 그리고 반복 학습할 때마다 다시 검색하는 중복 과정이 포함된다. 따라서 처음에 데이터를 확장한 후 각 의존 관계에 대한 셀들의 index를 저장하는 테이블을 생성하였다.

그림 5는 모듈화 된 시스템에 대한 그림으로써 F1 모듈에 +가중치(Wpositive)와 -가중치(Wnegative)를 부여하고 의존관계에 있을 경우에는 +가중치를 증가시키고, 의존관계가 아닐 경우에는 -가중치를 증가시키는 방법으로 학습시킨다.

$$\begin{cases} \text{if } d_{ij} = r_{ij} : \Delta W_{positive}(t+1) = \Delta W_{positive}(t) + 1 \\ else \quad\quad\quad\quad : \Delta W_{negative}(t+1) = \Delta W_{negative}(t) + 1 \end{cases} \quad (3)$$

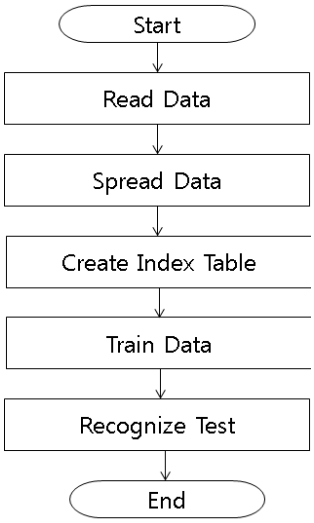

그림 6. 흐름도
Fig. 6. Flow Chart.

첫 번째 셀에는 SKA의 경우 증가치와 감소치가 섞여 있어서 +3이 되었으며, 제안된 시스템은 +가중치와 -가중치를 합하면 +3이 된다.

셀의 가중치에서 $W_{positive}$ 값이 큰 것은 정답 의존관계에 있을 경우가 많음을 의미한다.

$$W_{SKA} = W_{positive} - W_{negative} \quad (4)$$

W_{SKA}는 기존 시스템에서의 가중치이며, 결국은 제안된 시스템에서의 가중치의 합은 일치한다.

의존관계에 따른 모든 index가 구축되고 한번만 생성하면 되기 때문에 두 번째 학습부터는 검색이 불필요해진다. 결과적으로 검색시간이 매우 단축되었다.

그림 6은 시스템의 전체 흐름도이다. 먼저 데이터를 읽고 불필요한 데이터를 삭제한다. 다음으로 미리 생성 가능한 모든 의존관계를 확장하고, index 테이블을 생성한다. index 생성이 끝나면 학습 후, 인식 문장으로 테스트한다. 기존 시스템으로 테스트할 경우 약 6일(144시간)의 시간이 걸리지만, 수정된 시스템을 이용하면 index table 생성시간 162분(2.7시간), 학습 시간 15분으로 학습시간을 단축할 수 있었다. 실험은 CPU I5-4670 3.40GHz, 24GB 메모리를 가진 시스템에서 Matlab 2013a를 사용하였다.

2. 학습 문장 수에 따른 분석

표 6은 학습 문장 수에 따른 각 모듈별 생성되는 셀 수이다. 가로축은 학습 문장 수를 의미하며 세로축은

모듈별로 생성되는 셀의 개수이다. 10문장을 학습하였을 때, f1모듈에는 80개의 셀이 생성되고 f32모듈에는 528개의 셀이 생성된다. 특히 f19번과 같은 셀들에는 단지 4개의 셀만 생성되는 경우도 있다. 이와 같이 특정 모듈들은 생성되는 셀이 거의 없기 때문에 실제로 학습에 영향을 미치지 않는 모듈들도 많이 있다. 따라서 한국어의 특성에 맞는 자질 추출 테이블에 관련된 연구도 있어야 한다. 그리고 기존의 시스템은 인식에서 약 8백만 개의 셀에서 검색해야 하므로 시간이 매우 많이 걸린다.

표 7은 각 문장들을 학습한 후 인식율을 나타낸 것이다. self test란 학습 데이터를 인식시킨 결과이며, recognition test는 평가 데이터를 의미한다. 표에서와 같이 10문장만 학습함에도 불구하고 68.61%의 인식율을 보인다. 이는 한글과 시스템이 가지는 특징으로 마지막 어절은 의존관계가 없고 바로 다음 어절로 의존관계를 맺을 확률이 62%나 되기 때문이다. 그리고 시스템에서 최초 초기값으로 0으로 가중치를 부여함으로써 실제 인식결과가 학습에 의한 것인지 초기값에 의한 것인지 판단하는 것이 불분명하다.

그림 7에서는 학습된 문장의 인식율에 대한 그래프로서 학습 문장이 증가하더라도 어느 한계점 이후에는 인식율의 증가를 기대하기 어렵다. 표와 그림에서 보이듯이 만 문장을 학습하였을 때와 모든 문장들을 학습하였을 때 오히려 전자의 인식율이 더 좋다. 이는 학습 데이터 자체가 가지는 오류로 인하여 학습 되어진 셀의 가중치 값들이 추가 학습에 의하여 확산되기 때문이다. 결과적으로 학습에 사용할 문장들은 잘 정제하여 표준화된 학습 문장을 만드는 것이 인식율을 높이는 방법이라 할 수 있다.

표 6. 모듈 별 생성되는 셀 개수
Table 6. The Number of Cell of each Module.

sentence / module	10	1,000	10,000	all
f1	80	179	241	253
f2	80	175	229	241
...	...	...	...	...
f32	528	95,301	797,533	1,956,204
total	4,055	442,196	3,530,410	8,527,748

표 7. 학습된 문장과 test 문장에 대한 인식율
Table 7. Recognition rate of training sentences and test sentence.

	10	1000	10000	ALL
self test [%]	100	97.78	93.89	93.72
recognition test [%]	68.61	81.54	83.54	82.99

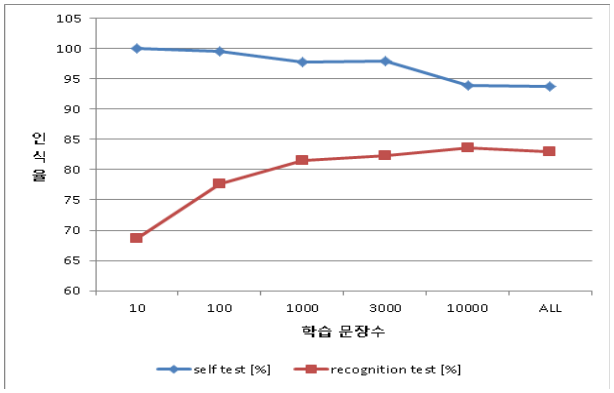

그림 7. 학습된 문장의 인식율
Fig. 7. Recognition rate of training data.

3. 학습 시 반복 횟수의 분석

그림 8은 만 문장을 50회 학습하였을 때 변화되는 셀 개수를 나타낸다. 표에서와 같이 20회 이상 학습부터 셀 변화 수가 증가($T(t) - T(t+1) < 0$) 될 수도 있으나, 다음 학습에서 더 낮은 값으로 바뀜($T(t) - (T(t+2) > 0$)으로 인해 시스템이 안정화되며, 인식율 또한 증가한다. 따라서 학습과 인식 시간의 한계 때문에 기존의 논문들이 제안하고 있는 10회 정도 횟수보다 더 많은 학습이 필요하다. 인식율 또한 10회에서는 83.54%, 30회에서 83.90%, 50회 학습에서는 84.54%를 보였다. 결과적으로 기계학습에서는 10회 정도로 학습하는 것보다 정제된 데이터를 시스템이 안정화될 때 까지 반복 학습하는 것이 더 높은 인식율을 가진다.

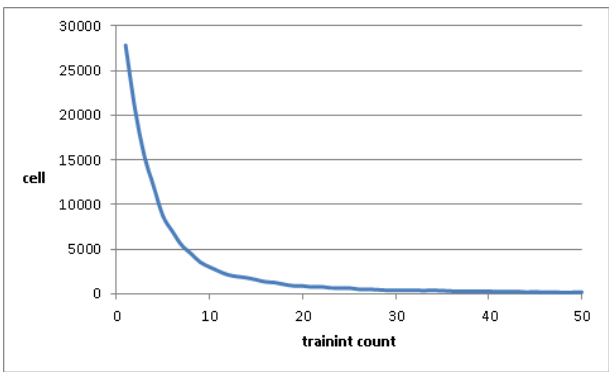

그림 8. 학습 횟수에 따른 셀 변화 수
Fig. 8. Changed cell as training count.

V. 결 론

기존 시스템은 기계 학습을 이용하여 학습 시간은 고려하지 않고 인식율에 중점을 두어 하나의 테이블에 많은 데이터를 저장하였다. 본 논문에서는 많은 데이터를 처리하기 위하여 특징 분할을 이용한 기계 학습을 통하여 한글의 의존관계를 처리하는 시스템을 설계하고 학습 문장수와 반복횟수에 따른 분석을 서술하였다. 시스템은 세종 코퍼스에서 제공된 구구조 문법을 의존구조로 바꾸고, 변환하는 과정에서 생성된 각종 오류들을 처리하였다. 그리고 세종 SKA에서 사용된 자질 집합표와 UTagger를 이용하여 어깨번호를 부착하여 실험하였다. 실험 결과 10,000 문장을 50회 반복 학습하였을 때, 가장 높은 84.54%의 인식율을 보였다.

실험에서와 같이 학습 문장이 32,481 문장으로 세배 이상 많음에도 불구하고 인식율은 오히려 82.99%로 떨어졌으며, 이는 학습 문장 자체가 가지는 오류로 인한 것이다. 결과적으로 정확히 정제된 학습문장을 반복 학습하였을 때가 인식율이 높고, 내부 테스트와 같이 학습 문장과 유사한 인식문장들의 인식율이 높았다.

최근에 발표된 각 알고리즘들은 주로 세종 코퍼스를 이용하며 정확도는 85~90%정도이다. 그러나 세종 코퍼스를 사용한다고 하더라도 구구조를 의존 구조로 바꾸고 데이터를 추출하는 방법들이 조금씩 다르기 때문에, 실제 시스템에서 사용된 데이터에 차이가 있다. 따라서 인식율 비교에 따라 어느 시스템이 더 좋다고 판단하기는 어려우며, 테스트를 위한 표준화된 데이터를 만들 필요가 있다.

향후에는 한글 특성에 맞는 자질 추출 테이블과 더 나은 인식율을 가진 시스템을 구현하기 위하여 기호주의 방법과 연계한 하이브리드 시스템에 관한 연구를 하고자 한다.

REFERENCES

[1] Geunbae Lee, “Comparison of connectionism and Symbolism in Natural Lanuage Processing”, The Journal of KIISE, pp. 1230 ~1238. 1993.
[2] Newell, A, “Physical sysbol systems”, Cognitive science, 4, pp. 135~183.
[3] Miikkulainen, R. and Dyer, M. G, “Natural Language processing with modular neural networks and distributed lexicon”, Convitive Science, 15, pp. 343~399.
[4] Hinton, G. E., McClelland, J. L., and RumelHart, D. E., “Distributed representation. - Parallel Distributed Processing: Exploratons in the Microstructure of Cognition.”, Vol I. pages 77~109, MIT Press, Cambridge, MA.
[5] Waltz, D. L., Pollack, J. B., “Massively parallel parsing”, Cognivive Science, 9. pp. 51~74.
[6] M. Collins, “Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms,” Proc. of EMNLP, 2002.
[7] Yoav Freund, Robert E. Schapire, “Large Margin Classification Using the Perceptron Algorithm”, Machine Learning Vo. 37. 277~296, 1999.
[8] Kwangmo Ahn, Younghoon Seo, “A Korean Dependency Parsing Algorithm using Sets of Head Candidates”, the journal of KIISE, Vol 41. pp. 88 ~95, 2014.
[9] S. Bucholz, E. Marsi, “CoNLL-X shared task on Multilingual Dependency Parsing”, Proc. of CoNLL, pp.149-164, 2006.
[10] R. McDonald, K. Crammar, F. Pereira, “Online Largemargin Training of Dependency Parsers,” Proc. of ACL, pp.91-98, 2005.
[11] J. Nivre, “An Efficient Algorithm for Projective Dependency Parsing,” Proc. of IWPT, pp.149-160, 2003.
[12] R. McDonald, F. Pereira, “Online Learning of approximate dependency parsing algorithms”, Proc, of EACL, 2006.
[13] Soojong Lim, Youngtae Kim, Dongyul Ra, “Korean Dependency Parsing Based on Machine Learning of Feature Weights”, the journal of KIISE, Vol 38. 4, pp. 214~223, 2011.
[14] Youngmin Park, Jungyun Seo, “Segang Korean dependency Analyzer”, competitive exhibition of 2011 Korean Information Processing System, 2011.
[15] J.H. Kim, “A Study on a Corpus Construction Tool for Machine Translation”, Research Report, Electronics and Telecommunications Research Institute (ETRI), 2012.
[16] H.G. Kim, “21st Century Sejong Project Construction of the Primary Data of the Korean Language”, Research Report NIKL 2007-01-10, National Institute of the Korean Language, 2007.
[17] Youngsook Hwang, Hoojung Chung, Soyoung Park, YoungJae Kwak, Haechang Rim,

"Improving the Performance of Korean Text Chunking by Machine Learning Approaches based on Feature Set Selection", the journal of KIISE, Vol. 29. pp. 654~668. 2002.
[18] Yonghun Lee, JongHyeok Lee, "Korean Dependency Parsing Using Online Learning", the Conference of Korea Computer Congress 2014, Vol. 37., No, 1, 2010.
[19] Myunggil Choi, Hyungwon Seo, Hongseok Kwon, Jaehoon Kim, "Detecting and correcting errors in Korean POS-tagged corpora", the Journal of KOSME, Vol. 37, pp. 227 ~235. 2013.
[20] Youngkuk Hong, Jonghyuk Hong, Geunbae Lee, "A Korean Syntactic Analyzer based on the Dependency Grammar", the Conference of KIISE, Vol. 20. pp. 781~784. 1993.
[21] Joonchoul Shin, Cheolyoung Ock, ,"A Korean Morphological Analyzer using a Pre-analyzed Partial Word-phrase Dictionary", the journal of KIISE, Vol 39, pp. 415~424, 2012.

저 자 소 개

김 성 진(정회원)
1996년 울산대학교 컴퓨터공학과 학사 졸업
1998년 울산대학교 컴퓨터공학과 석사 졸업
2009년 울산대학교 컴퓨터공학과 박사 졸업
2012년 울산대학교 전기공학부 IT 융합 전공 객원교수
<주관심분야 : 인공지능, 신경망, 자연어 처리>

옥 철 영(정회원)-교신저자
1982년 서울대학교 컴퓨터공학과 학사 졸업.
1984년 서울대학교 컴퓨터공학과 석사 졸업.
1993년 서울대학교 컴퓨터공학과 박사 졸업.
1994년 러시아 TOMSK 공과대학 교환교수
1996년 영국 GLASGOW대학교 객원교수
1984년 울산대학교 전기공학부 교수
<주관심분야 : 한국어 정보처리, 지식베이스, 기계학습, 온톨로지>