제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003.5)

문장분석용 통합 사용자 인터페이스 ISAAC의 개선

김곤*, 김민찬*, 배재학*, 유해영**, 이종혁***
*울산대학교 컴퓨터·정보통신공학부
**단국대학교 정보컴퓨터학부
***포항공과대학교 전자컴퓨터공학부 컴퓨터공학과
*e-mail:{gonkim, tomatuli, jhjbae}@ulsan.ac.kr,
**yoohy@dankook.ac.kr, ***jhlee@postech.ac.kr

Improvement of ISAAC (An Integrated User Interface for Sentence Analysis)

Gon Kim*, Min-Chan Kim*, Jae-Hak J. Bae*, Hae-Young Yoo**, Jong-Hyeok Lee***
*School of Computer Engineering and Information Technology, University of Ulsan
**Division of Information and Computer Science, DanKook University
***Dept. of Computer Science & Engineering, Pohang University of Science and Technology

요 약

문장분석은 문장의 의미를 파악하기 위한 작업이다. 문장분석에는 문장 구성성분에 대한 종합적인 정보를 필요로 한다. 문장분석을 위해서는 다양한 언어학적 도구와 자원이 필요하다. 가용 도구와 자원은 대부분 독립적으로 개발·축적된 것들이다. 이러한 도구와 자원을 이용하여 문장분석 정보들을 단계적으로 관리하고 처리하기에는 어려움이 있다. 이를 위해 본 논문에서는 문장분석용 통합 사용자 인터페이스 ISAAC을 개선하여 구문분석의 성공률과 그 정보의 상호보완성을 높이고자 하였다.

1. 서론

문장분석에는 다양한 언어학적 도구와 자원들이 필요하다. 구문분석기나 유의어사전 등이 대표적인 예이다. 이러한 도구와 자원들은 독립적으로 개발되고 축적된 것들이다. 이들을 활용하는 문장분석 작업은 자동적인 처리가 가능한 부분들도 있으나, 사람이 수작업으로 해결해야 할 부분도 있다. 이는 문장분석에 필요한 정보들을 관리하고 처리하기에 어려움이 있음을 의미한다. 따라서, 문장분석의 효율을 높이기 위해서는 도구와 자원을 통합하고, 이들의 정보를 정확하고 체계적으로 관리해줄 종합적인 인터페이스가 필요하다.

이에 본 논문에서는 사용자 중심의 인터페이스[12]를 바탕으로, 문장분석용 통합 사용자 인터페이스 ISAAC(An Interface for Sentence Analysis and Abstraction with Cogitation)[3]의 기능을 개선하였다. ISAAC은 문장분석시 필요한 언어학적 도구와 자원들을 통합하여 종합적인 관리가 가능하게 한다. 본 논문에서 문장분석을 위해 활용하는 언어학적 도구와 자원은 (1) 구문분석기를 통해 얻은 문장의 통사구조, (2) Roget 시소러스(Roget Thesaurus)[4]의 범주정보, (3) 이야기를 이해하기 위한 온톨러지 OfN(Ontology for Narratives)[1] 정보 등이다.

2. 활용한 도구와 자원

문장분석용 통합 사용자 인터페이스 ISAAC이 활용하는 도구와 자원은 다음과 같다: (1) 구문분석기로 LGPI+[3]와 MINIPAR+, (2) 유의어 사전으로 Roget 시소러스, (3) 이야기 이해를 위한 온톨러지로 OfN, (4) 말뭉치 자원으로는 DearAbby[7] 상담문을 활용한다.

2.1 구문분석기

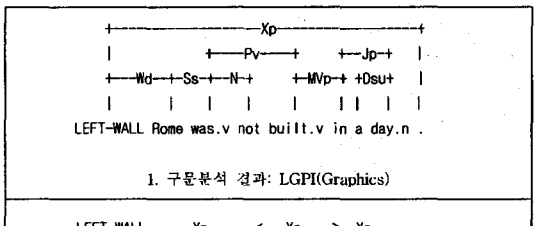
구문분석기로는 LGPI(Link Grammar Parser Interface[6])를 확장한 LGPI+에 MINIPAR(A

Minimalist Parser[10])를 확장한 MINIPAR+를 추가하였다. 이는 구문분석의 성공률과 분석정보에 대한 상호보완성을 높이기 위함이다.

LGPI는 Link Grammar Parser[5]에 대한 SWI-Prolog API를 제공한다. 입력문장에 대한 LGP 구문분석 결과는, 표식 고리(Labeled Link)의 집합으로 문장의 통사구조가 표현된다. 표식 고리는 한 쌍의 단어를 연결함과 아울러 그것들의 문법적인 기능을 표시한다.

MINIPAR[10]는 적용범위가 넓고 매우 효율적인 영어 구문분석기로, 초당 300단어를 처리한다. SUSANNE Corpus[8]로 평가하여 MINIPAR는 88%의 정확도와 80%의 재현율을 보인다[10, 11].

다음 문장을 생각해 보자: *Rome was not built in a day.* 이에 대한 구문분석 결과는 그림 1과 같다.



1. 구문분석 결과: LGPI(Graphics)

	LEFT-WALL	Xp	<—Xp—>	Xp	.
(m)	LEFT-WALL	Wd	<—Wd—>	Wd	Rome
(m)	Rome	Ss	<—Ss—>	Ss	was.v
(m)	was.v	Pv	<—Pv—>	Pv	built.v
(m)	was.v	N	<—N—>	N	not
(m)	built.v	MV	<—MVp—>	MVp	in
(m)	in	J	<—Jp—>	Jp	day.n
(m)	a	Ds	<—Dsu—>	D*u	day.n
	.	RW	<—RW—>	RW	RIGHT-WALL

2. 구문분석 결과: LGPI(Links)

```
(
E0  (()     fin C   *       )
1   (Rome   ~ N     4       s       (gov "build in"))
2   (was    be be   4       be      (gov "build in"))
3   (not    ~ A     2       neg     (gov be))
4   (built  "build in" V    E0      i       (gov fin))
5   (in     ~ U     4       lex-mod (gov "build in"))
E2  (()     Rome N  4       obj     (gov "build in")    (antecedent 1))
6   (a      ~ Det   7       det     (gov day))
7   (day    ~ N     4       mod     (gov "build in"))
)
```

3. 구문분석 결과: MINIPAR(Dependency)

그림 1 예문의 구문분석 결과: LGPI, MINIPAR

그림 1에서 보는바와 같이, LGPI와 MINIPAR의 구문분석 결과는 문장구성성분들간의 관계를 표식고리로써 나타내고, 각 단어의 품사정보를 보여주고 있다. 그러나, 그림 1의 구문분석 결과는 기계가 가독하기에는 무리가 있다. 이를 기계가독형으로 변경하기 위해 입·출력 알고리즘을 적용하고, 관련 함수를 재작성 하여 LGPI와 MINIPAR를 확장한 LGPI+와 MINIPAR+를 구현하였다. 그림 2는 예문에 대한 LGPI+와 MINIPAR+의 구문분석 결과이다.

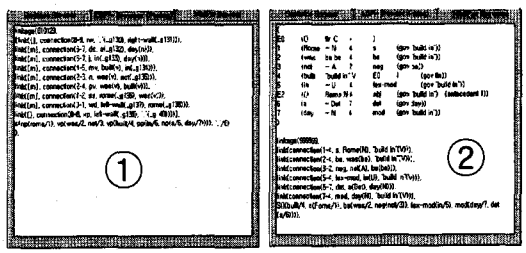


그림 2 예문의 구문분석 결과: LGPI+, MINIPAR+

2.2 Roget 시소러스와 OfN

Roget 시소러스는 총 6개의 의미 분류에 기초한 강(Class)으로 구성되며 각 분류는 하부에 부(Division), 과(Section) 등의 계층구조로 세분화 되어있다. 각 계층은 저마다의 표제정보를 가지고 있으며 계층구조의 말단에는 총 1044개의 범주가 존재한다. 각 범주에는 품사별 유의어 목록이 나열되어 있다.

문장에서 중요정보를 분별하고 이야기를 이해하기 위한 온톨러지 OfN은 Roget 시소러스를 심중사전(Lexicon)으로 삼아 이를 재구성하여 얻은 것이다. OfN은 다음의 7가지 범주로 구성된다: 등장인물(Character), 심상(Affect State), 사건(Event), 상태(State), 시간과 공간의 변화(Delta-{Time, Space}), 담화표지(Discourse Marker). 설정된 OfN을 구축하기 위해서 먼저 Roget 시소러스의 범주를 심상, 시간과 공간, 사건, 그리고 상태 등으로 재편성하였다. 등장인물 유형에 속하는 어휘들은 고유명사 자원[8]을 이용하여 선정하였다. 담화표지의 경우는 수사구조의 연구결과[7]를 활용하였다. 이와는 달리 시공의 변화는 구문분석 후 문장 구성성분간의 상호작용에 의하여 확인되는 유형인 바, 그 기본유형은 시간과 공간이다.

2.3 어형변화처리

문장의 통사구조 분석을 통하여 문장을 구성하는 어휘들을 분리한다. 문장의 어휘들은 주로 원형이 아닌 복수형, 과거형, 불규칙 동사와 같은 변형어휘 형태로 나타난다. 이러한 변형어휘들은 어형변화 처

리를 통하여 원형어휘를 찾는다. 원형어휘는 Roget 시소러스의 색인정보를 검색하는데 사용된다. 어형변화처리는 문장구성성분에 대한 OfN범주를 확인하기 위한 작업이다.

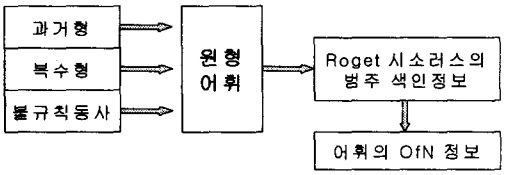


그림 3 어형변화 처리

2.4 이야기 말뭉치: DearAbby

ISAAC은 말뭉치 자원으로 DearAbby[7]에서 발췌한 상담문을 이용한다. DearAbby는 전 세계적으로 상당히 잘 알려져 있는 인생상담 기고란의 이름이다. 상담 이야기는 건강, 금전, 애정, 가족갈등, 대인관계, 학교생활, 인생진로 등에 관한 내용으로, 글의 유형은 사적인 경험들이다.

3. 문장분석 인터페이스 통합

ISAAC의 문장분석 처리과정은 (1) 문장의 의미파악을 위한 '문장의 통사구조 분석단계', (2) Roget 시소러스와 OfN 검색을 위한 '단어의 원형어휘 판별단계', (3) 원형어휘에 대한 'Roget 시소러스 범주정보 추출단계', (4) OfN 범주정보 추출단계'로 나누어진다. 아래의 그림 4는 단계별 문장분석 처리과정을 보여준다.

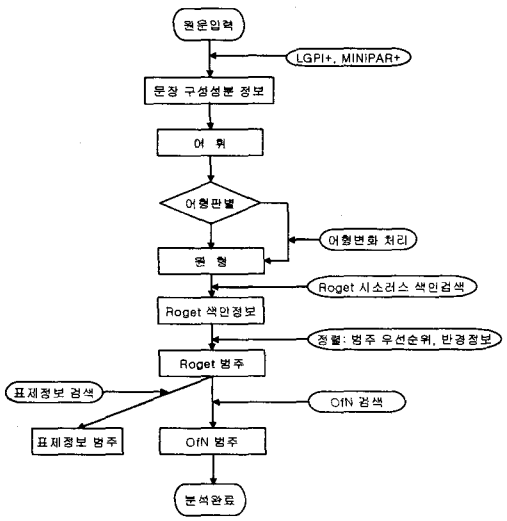


그림 4 문장분석 처리과정

그림 5는 문장분석 처리과정을 고려하여 구현한 통합사용자 인터페이스의 모습이다. 그림 5에서 ① 은 'Dear Abby'의 상담문 중에서 입력받은 한 문장이며, ②는 ①문장에 대한 한글 번역문이다. ③은 입력문장의 LGPI 구문분석 결과이다. 여기에서는 문장의 통사구조를 나타낸다. ④는 Roget 시소러스의 범주 참조정보이며, ⑤는 ④에서 나열된 범주들을 참조정보를 이용한 방식의 OfN 재분류 정보이다.

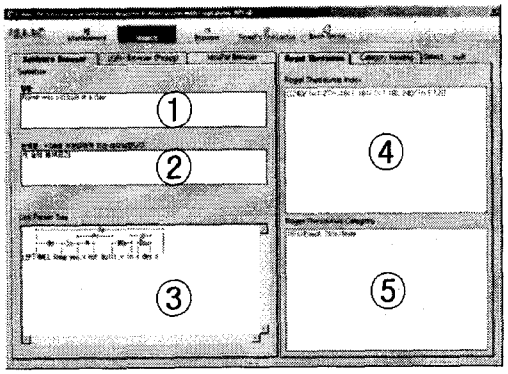


그림 5 ISAAC: 문장분석용 통합 사용자 인터페이스

4. 통합 인터페이스 평가

표 1은 ISAAC의 효율성을 평가하기 위하여 문장분석 작업을 수작업으로 했을 때와 ISAAC을 활용한 경우를 비교분석 한 결과이다. 평가를 위한 말뭉치로는 DearAbby를 사용하였다.

분석과정		단계별 분석과정	수작업	ISAAC	비고
문장의 통사 구조 분석	1	구문분석	○	○	1~3단계는 ISAAC 에서 단일과정으로 처리
	2	분석과정 오류	○		
	3	구문분석 확장	○		
	4	어휘 정보 기록	○	●	
어형 판별	5	원형어휘 검색	○	○	ISAAC 변형어휘사전으로 각 어휘사전을 통합.
	6	과거형 어휘 검색	○	●	
	7	복수형 어휘 검색	○		
	8	불규칙 어휘 검색	○		
	9	(원형어휘 재검색)	○		
	10	원형어휘 검색 및 기록	○	●	
Roget 시소러스	11	Roget 시소러스 검색	○	○	Roget 시소러스와 OfN의 검색, 추출과정 기록은 ISAAC에서 실시간 처리
	12	범주정보 추출 및 기록	○	●	
OfN	13	각 범주에 대한 OfN 검색	○	○	
	14	OfN범주정보 기록	○	●	

표 1 ISAAC의 효율성 평가(●: 수작업에서 사라진 단계)

문장의 통사구조 분석단계는 문장을 의미적으로 분석하기 위한 선행단계로 문장을 구성하는 어휘들

의 정보를 얻고 문장 구성성분들 간의 문법적인 관계를 파악하는 과정이다. 이를 위해, 구문분석기인 LGP와 LGPI+를 이용한다. LGPI+의 구문분석 결과는 기계가독형으로 문장의 통사구조를 나타낸다. 이 과정에서 오류가 발생할 경우에는 LGP의 결과를 토대로 문장을 정규화하거나 LPG에 수록된 어형사전을 참조하여 어형변화 처리과정을 거친다. 이러한 문장정규화나 어형변화 작업은 수작업으로 했을 때 시간소요가 가장 많은 부분이며, 분석자의 판단에 따른 오류가 존재할 가능성이 높은 단계이다.

단어의 원형어휘 판별단계는 Roget 시소러스정보를 검색, 추출하기 위한 과정이다. 이를 위해 과거형, 복수형, 불규칙 어휘 사전을 구축하고 해당 단어의 원형어휘를 검색한다. 검색된 원형어휘로 Roget 시소러스 범주정보를 얻고, 이를 통해 OfN 범주정보를 추출한다.

표 1에서 확인되는 바, ISAAC을 활용한 문장분석은 각 단계별 작업에서 얻어지는 정보의 기록 여부에 따라 단계별 작업 중 반복적이거나 수동적인 작업을 줄일 수 있다. 문장분석 과정은 선행단계의 결과를 다음단계에서 필요로 한다. 수작업의 경우에는, 이러한 정보들을 수기나 파일의 형태로 기록·참조하게 된다. 이 과정에서 인적오류로 인하여 불확실한 정보의 기록, 정보의 손실, 판단의 오류 등을 야기할 수 있다. 이로 인하여 획득한 정보의 신뢰도가 낮아질 수 있다. 이러한 점은 문장분석의 모든 단계에 영향을 미치게 된다.

ISAAC을 통한 문장분석은 수작업에서 발생할 수 있는 오류를 줄이고 반복적인 검색을 실시간으로 처리한다. 또한 유의어 사전 및 어휘 사전을 이용하여 Roget 시소러스 범주정보나 OfN 범주정보, 어휘의 원형 등 관련 정보들을 데이터베이스를 활용하여 효율적으로 처리한다. ISAAC은 이러한 개별적으로 존재하는 언어학적 도구와 자원들을 통합하여 문장분석에 활용할 수 있도록 사용자 편의성을 제공한다.

4 결론

본 논문에서는 문장분석용 통합 인터페이스 ISAAC을 개선하여 구문분석의 성공률과 문장 분석 정보의 상호보완성을 높이고자 하였다. 이를 위해 구문분석기 LGPI와 MINIPAR를 확장하여 적용하였다. 또한, 구문분석가의 정확성과 효율성을 높이기 위해 사용자 중심의 인터페이스를 적극 반영하였다.

ISAAC은 문장분석과정에서 문장구성성분들에 대한 정보들을 시각적이면서도 체계적으로 관리하도록 한다. 또한 개별적으로 존재하는 언어학적 자원들의 모든 기능들을 유지하면서 편리한 사용자 인터페이스를 제공한다.

이러한 통합환경은 기존의 수작업과는 달리 시간절감, 인적오류방지, 자원재활용 등의 장점을 가지고 있다. 이를 확인하기 위하여 문장분석을 수작업으로 한 경우와 ISAAC을 이용한 경우를 비교해 보았다. 그 결과, ISAAC을 활용하였을 때 반복적인 작업을 자동화하여 소요시간을 줄일 수 있을 뿐만 아니라, 단계별 정보의 처리 및 가공에 있어 오류를 없애고, 분석 결과의 정확성과 효율성을 높일 수 있음을 알 수 있었다.

참고문헌

[1] 양재군, 배재학. "온톨로지 정보를 이용한 범주 재편성: Roget 시소러스의 경우." 한국정보처리학회, 제9권, 제1호, pp.515-518, 2002.

[2] 배재학. "언어학적인 방법론을 취하는 자동 문서요약에 대한 연구." 공학 연구논문집, 제 29권 2호, pp.351-363, 울산대학교, 1998.

[3] 김명수, 김민찬, 배재학. "문장분석에 활용할 종합적인 사용자 인터페이스." 한국정보처리학회, 제9권, 제1호, pp.535-538, 2002.

[4] Roget's Thesaurus. http://promo.net/cgi-promo/pg/t9.cgi?entry=22&full=yes&ftpsite=ftp://ibiblio.org/pub/docs/books/gutenberg/.

[5] Link Grammar. http://www.link.cs.cmu.edu/link/.

[6] SWI-Prolog. http://www.swi-prolog.org/.

[7] DearAbby. http://www.dearabby.com/.

[8] Proper Names Wordlist. http://clr.nmsu.edu/cgi-bin/Tools/CLR/clrcat#I4.

[9] SUSANNE Corpus. http://www.grsampson.net/.

[10] MINIPAR. http://www.cs.ualberta.ca/~lindek/.

[11] D. Lin, "Dependency-based Evaluation of MINIPAR", *In Workshop on the Evaluation of Parsing Systems*, Granada, Spain, May, 1998.

[12] B. A. Myers, "Why are Human-computer Interfaces Difficult to Design and Implement?", Technical Report CS-93-183, Carnegie Mellon University, School of Computer Science, July 1993.