제20회 한국정보처리학회 추계학술발표대회 논문집 제10권 제2호 (2003. 11)

정규표현을 이용한 연속 및 불연속 복합단위 인식기

여상화*, 서정연**
*경인여자대학 컴퓨터정보기술학부
**서강대학교 전자계산학과
e-mail : shyuh@kic.ac.kr;sjy@ccs.sogang.ac.kr

An Interrupted and Uninterrupted Compound Unit Recognizer using Regular Expression

Sanghwa Yuh*, Jungyun Seo**
*Div. of Computer Information Technology, Kyung-In Women's College
**Dept. of Computer Science, Sogang University

요 약

기계번역 시스템에서 복합단위 처리는 원문의 분석 부담을 줄이고 조합적으로 대역문의 의미를 생성하지 못하는 원문의 처리를 위해 필수적이다. 본 논문에서는 정규표현(Regular Expression)을 이용하여 영어의 연속(Non-Interrupted) 및 불연속(Interrupted) 복합 단위를 인식하는 복합단위 인식기를 제안한다. 제안된 방법은, 기존에 trie 와 같은 index 의 갱신 과정이 불필요하므로, 다수의 작업자에 의해 복합단위 사전을 동시에 구축하는 경우에, 한 작업자의 결과가 실시간으로 다른 작업자의 작업에 반영되는 장점이 있으며, 복합단위 인식에 있어 정규 표현을 이용함으로써 복합단위 인식기의 성능을 선언적으로 향상시킬 수 있다. 번역 실행시의 고속 탐색을 위해서는 전체 복합단위로부터 FSA(Finite State Automata) 를 자동으로 구축하여 빠른 속도로 인식 가능하도록 하였다.

1. 서론

규칙기반의 전통적인 자동 번역 시스템은 그 의미가 조합적(Compositional)으로 이루어 지지 않는 언어 표현에는 매우 취약하다. 이러한 유형에는 Collocation, 관용어(Idiom), 숙어, 고정표현(Frozen expression 또는 Fixed Expression) 등이 있다[1][2].

관용어: *Let' s call it a day; 오늘은 이만 끝냅시다*
I'm through with you. 당신과는 절교입니다.,
숙어:*He is, **as it were**, a swindler. 그는 **말하자면** 사기꾼이다)*
고정표현:*How is it going?*

이러한 표현들은 여러 성분들이 모여 각 성분의 의미로부터 조합해 낼 수 없으며, 문법적으로는 하나의 어휘로 대체 가능하다. 이들은 구문 분석 앞 단계에서 인식함으로써, 구문 분석의 부담을 줄여주며, 자연스러운 대역문을 생성하기가 용이하다. 또한, 번역 시스템을 특정 도메인(Domain)에 특화 시키는 경우, 튜닝이 용이하다는 장점이 있다.

따라서, 이들 표현들은 자동번역에서 **복합단위(Compound Unit; 이하 CU)**라는 용어를 사용하며 숙어(Idiom), 고정표현(Frozen Expression) 등을 포함하여, 기계적인 처리가 어려운, 조합적이지 않은 표현을 모두 지칭하는, 보다 포괄적인 개념으로 사용한다. 복합단위에는 복합단위 성분들 사이에 다른 단어나 구(Phrase)가 삽입되는 **불연속(Interrupted)** 표현과 다른 성분이 삽입되지 않는 **연속(Non-Interrupted)**표현이 있다[13].

이러한 중요성으로 인해 대규모 복합단위 사전의 구축은 기계번역 시스템의 필수적인 것으로 인식되어

왔고, 대규모 말뭉치(Corpus)로부터 통계적인 방법으로 자동으로 구축하는 방법들이 연구되었다[9][13].

자동번역 용 사전은 다수의 작업자에 의해 대량의 사전을 단기간에 구축하는 번역지식 구축 환경이 필수적이며, 이를 위해서는 다수의 작업자의 결과가 실시간으로 DB 에 저장되고, 번역시스템에 즉각적으로 반영되어야 한다. 또한, 번역 실행시간에는 빠른 속도로 복합단위 후보들을 인식해야 한다.

따라서, 본 논문에서는 정규 표현(Regular Expression)을 이용하여 연속 및 불연속 복합단위를 인식하고, 작업결과가 실시간으로 반영되도록 하며, 기존의 번역 사전과 통합되어 기술하도록 하며, 복합단위 표제어도 단일 표제어와 유사한 방법으로 기술하도록 하여 대량 구축이 용이하도록 하며, 가독성을 높여 쉽게 수정이 가능하도록 하였다. 번역기가 단독으로 수행되는 경우에는 번역 사전을 스캐닝하여 전체 복합단위를 추출하고 이들 전체를 하나의 유한상태 오토마타(Finite State Automata; FSA)로 구축하여 빠른 검색이 이루어 지도록 하였다.

2. 기존의 연구

[12]에서는 먼저 복합단위 형식 중의 고정 단어를 순서대로 문장 단어 열과 비교하여 인식 성공 가능성을 판별한 후, 이어서 성공 가능성이 판별되면 문장의 지역적 구간에서 복합단위 변수 메타 기호들에 지정될 부분 구조를 찾는다. 부분 구조의 분석에는 활성 챠트(Active Chart)를 사용하며 메타기호가 지정한 문법 범주에 의한 하향식 분석을 병행한다.

[1][3][4][5][6]은 번역기를 위하여 이종(Heterogeneous)노드를 가진 trie 를 이용하여 고속의 연속 및 불연속 복합 단위 인식기를 제안하였다. 또한 RTN(Recursive Transition Network)을 이용한 부분 구문분석기(Partial Parser)를 사용하여 가변성분에 대한 구문적인 제약 검사를 동시에 수행하도록 하여 인식의 정확률(precision)을 높였다[2]. Trie 를 이용한 방법은, 복합단위 표제어의 추가/삭제/변경이 이루어지는 경우, 전체 trie 가 메모리에 상주해야 하므로 대규모 복합단위 사전을 이용하는 경우 메모리 부담이 크며 복합단위 사전이 번역용 사전과 별도로 관리되고, 복합단위어를 기술하기가 용이하지 않은 단점이 있었다.

[10]은 유한상태변환기의 일종인 Head Automata 를 이용하여 한영 번역에서의 날짜, 시간 표현과 같은 제한된 고정표현 인식과 변환 방법을 제안하였다. 이 방법은 가변성분[1]을 포함하는 불연속 고정표현의 인식과 같은 범용적 용도로는 부적당하다

3. 복합 단위 인식

그림 3.1 은 본 논문에서 제안하는 복합단위 인식기의 구성도이다. 복합단위 인식기는 품사 태거[8]와 구문분석기[11] 사이에 위치하며, 입력 문장에서 사용된 연속 및 불연속 복합단위를 인식하여 구문분석기에 제공한다. 복합단위 인식기는 후보 인식과정과 최적 후보 선정의 두 단계로 이루어진다. 인식된 다수의 복합단위 후보 중에서 최적의 후보를 선정하는 것은 구문분석기에 의해 결정된다. 복합단위 인식기는 지역적인 문맥정보만을 사용할 수 밖에 없으며, 구문 분석 전단계에서 최적후보를 결정짓는 다면, 잘못된 후보를 결정지을 수 있고, 이로 인해 이후의 구문분석 실패를 야기시켜 번역 실패라는 치명적인 결과를 초래할 수 있다. 따라서, 본 논문에서는 Full Parsing 의 결과 만들어지는 구문 트리(Parse Tree)의 단말(Terminal)에 나타나는 복합단위가 최종적으로 선택된 복합단위가 되도록 한다. 구문분석기는 Bottom-up Best-First Parser 를 이용한다[11].

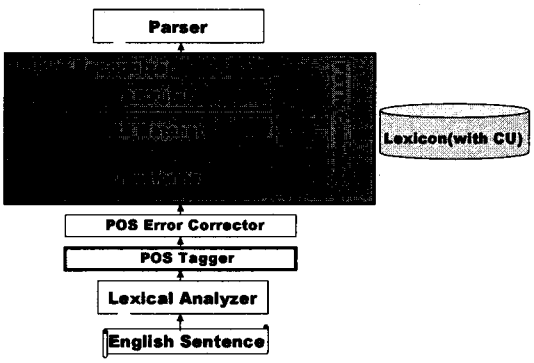

그림 3.1 복합단위 인식기의 구성도

복합단위 인식기는 가변 성분을 포함하는지의 여부에 따라, **연속 복합단위** 인식과 **불연속 복합단위** 인식의 과정을 수행한다.

3.1 연속 복합단위 인식

품사 태거와 규칙기반의 품사 태깅오류 교정기[8]를 거친 어휘 분석결과는 먼저, 정규 표현 매칭을 위한 형태로 변형된다. 즉, 입력 문장의 각 단어는, 활용형, 원형(root form), 품사 정보를 갖는 문자열로 변형되며, 이때, 어휘 중의성과 품사 중의성으로 인해 하나 이상의 정보가 기술되기도 한다. 예를 들어, 입력 문장이 " I saw a doctor." 인 경우, 다음과 같은 형태로 변형된다[2].

{0}_i_:ROOT_i:POS_PRON_
{1}_saw_:ROOT_see_:POS_VERB_
{2}_a_:ROOT_a_:POS_DET_
{3}_doctor_:ROOT_doctor_:POS_NOUN_ "

복합단위 후보는, 단위어 중에서 첫번째 나오는 내용어(Content Word)의 원형을 Key 로 하여 번역 사

[1] 가변성분은 특정 어휘, 특정 품사, 특정 구문의 제약 정보를 가질 수도 있다.

[2] 실제로는 약속된 기호들로 표현하여 문자열의 길이를 줄인다(예: :R(원형정보), :P(품사정보)

전에 수록되며, 다양한 품사로 사용되는 복합단위의 예는 다음과 같다.

[look_on_with@VERB]
[look_in@NOUN]
[in_front_of@PREP]
[in_full@ADV
[a_lot_further_down@PREP]
[a_lot_of@ADJ]
[a_lot_of@DET]

복합단위는 단일어와 마찬가지의 문법적인 역할을 하므로, 이들에 대한 품사정보도 부여된다(예:@VERB, @PREP, @NOUN, @ADV).

"see" 를 key 로 하여 번역 사전을 참조하면, 다수의 복합단위 후보들이 검색되며, 이들 각 후보들은 입력 문장과의 패턴매칭(Pattern Matching)을 통해 복합단위 후보로 인식된다. 예를 들어, " 진찰받다" 의 뜻을 가진 "see a doctor" 의 경우, 번역 사전에는 "[see_a_doctor@VERB]"의 형태로 기록되어 있으며, 패턴 매칭을 위해 다음과 같이 정규 표현으로 변환된다.

{[0-9]+} [A-Za-z\: _]*_see_[A-Za-z\: _]+
{[0-9]+} [A-Za-z\: _]*_a_[A-Za-z\: _]+
{[0-9]+} [A-Za-z\: _]*_doctor_[A-Za-z\: _]+

정규 표현으로 변형된 각 복합단위는 입력 문장과의 고속의 패턴 매칭을 위해, 유한상태 오토마타(Finite State Automata)로 Compile 된다[7].

정규 표현에 의해 복합 단위로 인식된 후보는 하나 이상이 될 수 있으며, 이들은 인식 과정에서 최적 후보를 결정하지 않고, 구문분석 단계에서 잘못된 후보의 Filtering 과 최적 후보를 선택하게 된다. 구문분석기에 의해 복합단위 후보 중에서 최적의 후보가 선택되면, 이후의 변환 과정을 위해 번역 사전을 탐색하여 변환정보를 올리게 된다. 변환 사전 탐색을 위한 Key 는 복합단위 자체가 된다. 예를 들어, "a_lot_further_down@PREP"를 Key 로 하여 변환사전을 탐색하면 다음과 같은 변환 정보를 얻게 된다.

[(KPOS JOSA) (KROOT 아래로_훨씬_더_멀리)
(KGCODE D05D05)]

가변 성분을 포함하는 불연속 복합단위는 가변 성분에 대한 문법 제약 정보와 함께 변환 정보를 포함한다. 다음은, " take_#1:~_at_#2:~_word@VERB " 의 번역 사전 정보이다.

{복합단위 [#1:NP #2:DETP] }
{변환
[(ETYPE I0) (KPOS VERB) (SEM DURATIVE)
(KROOT #2의_말을_믿) (KCODE N00V04)] }

3.2 불연속 복합단위 인식

불연속 복합단위는, 단위어 성분 중에 가변적인 성분을 포함하는 것으로, 특정 어휘, 특정 품사, 특정 구문 또는 임의적인 가변 성분이 포함되는 복합단위이다. 불연속 복합단위의 예는 다음과 같다.

(1) [take_#1:PRON_#2:ADV_to@VERB]
(2) [take_effect_over_#1:NUM_month@VERB]
(3) [take_to_#1:DETP_legs@VERB]
(4) [take_#1:~_at_#2:~_word@VERB]

위의 예에서, (1)과 (2)은 특정 품사(PRON:대명사, NUM:수사)의 가변 어휘를 가지는 예이고 (3)은, 특정 성분(DETP:관형사구)의 가변 성분을 가지는 예이며, (4)는 두 개의 가변 성분(#1 과 #2)을 가지는 것으로, 각 성분에 대한 제약이 복수 개인 경우이다. 가변 성분에 대한 제약이 여러 개인 경우에는 사전을 통해 가능한 가변성분의 문법 정보를 획득하도록 하여, 동일한 복합단위의 중복 인식을 방지하도록 한다. (4)의 경우, 사전에는 "**{복합단위 [#1:NP #2:DETP] }**" 같이 가변 성분의 문법 정보를 수록한다.

특정 어휘 또는 특정 품사를 포함하는 불연속 복합단위는, 연속 복합단위와 유사한 방법으로 인식 가능하다. 그러나, 특정 구문 분석을 가진 경우와, 임의적인 구문성분을 가지는 경우에는, 가변 성분에 대한 부분적인 구문분석이 이루어져야 한다. 기존의 방법에서는 변형된 trie 를 RTN(Recursive Transition Network)과 같이 구성하고 별도의 CFG 문법을 사용하여 부분 파싱을 하거나[2], 구문분석기의 해석 문법을 사용하여 부분 파싱을 수행하고 수행 과정에 시도한 작업을 Chart 에 기록하도록 하여 중복을 배제하도록 하였다[12].

첫번째 방법은 복합단위 인식을 위한 별도의 문법이 필요하고, 구문분석 과정에서 가변성분들에 대한 중복 분석이 발생하는 단점이 있다.

본 논문에서는, 가변성분은 다음과 같이 임의 개의 어휘에 매칭이 가능하도록 정규 표현으로 변환하여 패턴매칭을 시도한다.

"({[0-9]+}[!-z|~()]+)+"

가변요소를 포함한 복합 단위 인식이 성공한 경우, 해당 가변요소의 구성성분의 시작 위치와 끝위치, 그리고 제시된 구문 정보를 Chart 에 기록한다. 구문분석기는 분석과정에서, 복합단위 인식기에 의해 기록된 정보를 이용하여 구문분석을 수행한다. 따라서, 가변요소에 대한 부분 분석은 구문 분석 과정에서 자연스럽게 이루어지게 된다.

4. 복합단위 FSA

지식 구축 환경이 아닌 번역기가 단독으로 수행하는 경우에는(runtime), 사전에서 검색된 복합단위들을 순차적으로 매칭하는 것이 아니라, 전체 복합단위들로 이루어진 하나의 FSA 를 이용하여 매칭하는 것이 필요하다.

대규모 FSA 구축을 위해 상용 컴파일러 제작도구를 사용하며, 번역 사전이 자주 변경되는 특성을 고려하여 DLL(Dynamic Link Library) 형태로 제작한다.

5. 구현 및 실험

본 논문에서 제안한 복합단위 인식기는 WindowsXP 상에서 VisualC++6.0, 컴파일러 제작 도구인 ParserGenerator[15][16], 그리고 Regular Expression Library[7]를 이용하여 구현되었다.

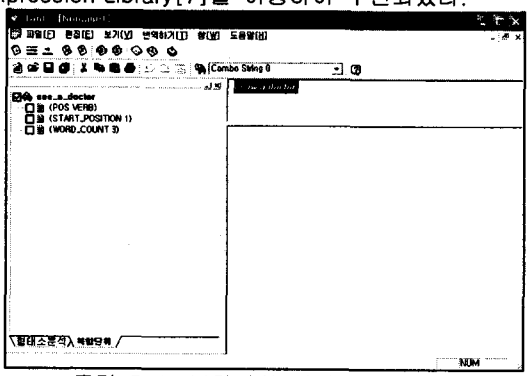

그림 5-1. 복합단위 인식기 실행 화면

복합단위 인식기는 영어 품사 태거[8]의 결과를 입력 받아 복합단위 인식을 수행하며, 인식 결과를 구문분석기를 위한 Chart 에 기록한다. 구문 분석기는 Chart 에 기록된, 복합단위 인식 결과를 이용하여 Bottom-up Best-First Chart Parsing 을 수행하며, Full Parsing 과정을 통해, 복합단위 후보의 Filtering 과 최적 후보 선택을 하게 된다. Full Parsing 을 통한 최적 후보 선정은, 지역적인 문맥을 사용하는 경우의 오류 발생을 방지하여 잘못된 후보 선택으로 인한 번역 실패의 가능성을 없애준다.

6. 결론 및 향후 연구

본 논문에서는 정규표현을 이용하여 영어의 연속 및 불연속 복합단위를 인식하는 복합단위 인식기를 제안한다. 복합단위 인식에 있어 정규 표현을 이용함으로써 복합단위 인식기의 성능을 선언적으로 향상시킬 수 있으며, 단일 표제어와 동일한 방식으로 기술할 수 있도록 함으로써, 기존의 번역용 사전에 통합 기술하도록 하고, 복합단위어 기술 시 가독성(Readability)을 높여 대량 작업을 용이하게 하였다.

제안된 방법은 다수의 작업자에 의해 대량의 번역 지식이 구축되는 환경 하에서, 작업자의 결과가 실시간으로 다른 작업자에 반영되도록 하는 장점이 있다. 또한, 번역 실행시의 고속 탐색을 위해 전체 복합단위로부터 FSA를 자동으로 구축하도록 하였다.

향후 연구내용은, 구조 변환기능을 겸하는 Bottom-Up Best-First Parser[11]와의 통합을 통해 단순화된 자동번역기를 실험 제작하고 그 유용성을 확인하는 것이다.

참고문헌

[1] Hanmin Jung, Taewan Kim, and Sankyu Park, "A Pattern-based Approach Using Compound Unit Recognition and Its Hybridization with Rule-based Translation," *Computational Intelligence*, Vol. 15, pp.114-127, 1999.

[2] Hanmin Jung, Sanghwa Yuh, Taewan Kim, and Dong-In Park, "Syntactic Verifier as a Filter to Compound Unit Recognizer," PACLIC1998, 1998.

[3] Hanmin Jung, Sung-Kwon Choi, Chul-Min Sim, Sanhwa Yuh, Taewan Kim, and Dong-In Park, "Toward Hybrid Translation: Focusing on the Pattern-based Compound Unit information in Syntactic Analysis," IASTED1998, 1998.

[4] Hanmin Jung, Sanghwa Yuh, Taewan Kim, and Dong-In Park, "Compound Unit Recognization for Efficient English-Korean Translation," ACH-ALLC1997, 1997.

[5] Hanmin Jung, Sanghwa Yuh, Taewan Kim, and Dong-In Park, "Compound Unit Recognizer for Pattern-Based Approach to Mutilingual Machine Translation," PACLING1997, 1997.

[6] Hanmin Jung, Sanghwa Yuh, Taewan Kim, and Dong-In Park, "Efficient Compound Unit for Heterogeneous Types," NLPRS1997, 1997.

[7]Regexp, http://bazar.conectiva.com.br/~niemeyer/projects/regexx/

[8] Sanghwa Yuh, *et al.*, "NeuTag: A Hybrid Neural Network English Tagger with Pre-Fail Softener," ICCPOL'99, 1999.

[9] Youngkil Kim, Hanmin Jung, Sanghwa Yuh, Taewan Kim, ByungRae Ryu, and Sangkyu Park, "Automatic Extraction of Korean Idiomatic Patterns for Korean-English Machine Translation System," ICCPOL1999, 1999.

[10]박준식, 최기선, "한영 기계번역에서의 효율적인 구문분석과 번역을 위한 유한상태변환기 기반 전처리기의 설계 및 구현," 제 11 회 한글 및 한국어 정보처리 학술대회 논문집, pp.128-134, 1999.

[11]여상화, 서정연, "구조변환을 겸한 영어 구문분석기", 2003 년 봄 한국정보과학회 학술발표논문집(B), pp.507-509, 2003.

[12]윤성희, *영어-한국어 기계번역을 위한 숙어 기반의 효율적 문장 분석*, 서울대 컴퓨터공학과 박사학위 청구논문, 1993.

[13]S.Ikehara, S. Shirai, and H Uchino, "A Statistical Method for Extracting Uninterrupted and Interrupted Collocations from Very Large Corpora," Proceedings of COLING, pp.574-579, 1996.

[14] F. Smadja, "Retrieving Collocatios from Text:Xtract," *Computational Linguistics*, Vol.19, No.9, pp.143-177, 1993.

[15]Bumble-Bee Software, http://www.bumblebeesoftware.com/

[16]John R.Levine, Tony Mason, and Doug Brown, *Lex & Yacc*, O'Reilly & Associates, Inc, 1992