한국어 동사와 명사 관용구 인식 알고리즘

이 호 석
호서대학교 공과대학 뉴미디어학과
hslee@office.hoseo.ac.kr

A recognition algorithm of Korean verb and noun idiomatic phrases

Ho Suk Lee
Dept. of New Media, College of Engineering, Hoseo University

요 약

본 논문은 한국어 관용구 인식 알고리즘에 대하여 논의한다. 다음(daum) 전자 사전에는 관용구의 의미를, “두 개 이상의 단어로 이루어져 있으면서, 그 단어들의 의미만으로는 전체 의미를 알 수 없는, 특수한 의미를 나타내는 어구” 라고 설명되어 있다. 한국어 관용구의 길이는 2글자 ~ 4글자인 경우가 많으며 그 이상인 경우도 있다. 대부분의 관용구는 일반 사전에 동사와 명사를 기준으로 분류되어 있으며, 품사 표시나 구절 표시 없이 어절의 문자열 형태로만 표현되어 나타난다. 본 논문에서는 전자 사전에 품사 표시나 구절 표시 없이 어절 문자열 형태로 저장되어 있는 한국어 관용구를 입력 문장에서 인식하는 관용구 인식 알고리즘에 대하여 논의한다. 그리고 연어 인식과 명사의 의미 속성 처리에 대하여서도 논의한다.

주제어: 한국어 관용구, 전자 사전, 관용구 인식 알고리즘, 구문 처리기 구조

1. 서 론

대부분의 한국어 관용구는 일반 사전이나 관용구 사전에 동사와 명사를 기준으로 분류되어 있다. 관용구는 품사 표시나 구절 표시 없이 뜻풀이와 함께 어절 문자열 형태로 표현되어 있다. 관용구는 2글자 ~ 4글자인 경우가 많다.

다음은 뜻풀이를 제외한 일부 동사 관용구의 예이다[1].

[가하다]
메스를 가하다
박차를 가하다

[감다]
눈을 딱 감다
눈을 감아 주다

[구르다]
굴러 온 호박

[흔들다]
고개를 흔들다

다음은 일부 명사 관용구의 예이다[1].

[가죽]
뼈와 가죽뿐이다

[감초]
약방의 감초

[값]
같은 값이면 다홍치마

[떡]
그림의 떡
누워서 떡 먹기

[힘]
목에 힘을 주다
어깨에 힘을 주다

이 예들을 보면, 관용구는 동사 혹은 명사를 기준으로 분류되어 있으며 어절 문자열 형태로만 표현되어 있는 것을 알 수 있다. 동사 [구르다]의 경우를 보면, “굴러 온 호박”이라는 명사구가 관용구로 등록되어 있는 것을 알 수 있으며, 명사 [힘]의 경우를 보면, “목에 힘을 주다”, “어깨에 힘을 주다” 등의 동사구가 관용구로 등록되어 있는 것을 알 수 있다. 또한 관용구가 등록되어 있는 표제어에도 일정한 규칙이 있다고 할 수는 없다. 예를 들면, 단어 “호박”에는 “호박이 굴렀다”, “호박이 굴러 떨어졌다”등의 동사구들이 관용구로 등록되어 있다.

이 밖에도 “그도 그럴 것이”, “바꿔 말하면”, “뿐만 아니라”, “예를 들면, "할 수 있다”, “말 할 것도 없이” 등도 관용구로 간주할 수 있다[2].

따라서, 한국어 관용구는 전자 사전에 품사 표시나 구절

본 연구는 2009년도 한국전자통신연구원의 위탁과제연구비를 받아 수행되었다. (과제번호:2009-0126)

표시가 없고 구성에 큰 제한이 없이 어절 문자열 형태로 저장할 수 있다.

관용구에 대한 조사와 연구는 주로 기계 번역을 위하여 진행되었다[3]~[12]. 문헌 [3]~[10]에서는 영한기계번역의 관점에서 영어 관용어에 대하여 논의하였다. 문헌 [11][12]에서는 한영기계번역의 관점에서 한국어 관용구 인식에 대하여 논의하였다. 문헌 [11]에서는 어휘모호성을 해결하지 않은 상태에서 관용구 인식을 시도하였다. 직접적인 패턴 매치(pattern matching) 방법을 사용하지 않고 기본형 일치를 시도한 다음에 성공한 관용구 항목들에 대하여 접사 일치를 시도하여 인식하려고 하였다. 또한 분산값(dispersion value)이라는 것을 정의하여 관용구의 근접성을 계산하여 관용구를 선택하였다. 그러나 한국어 관용구 자체의 인식부터 논의할 필요가 있다. 문헌 [12]는 한국어 구문 분석의 모호성을 해결하기 위하여 관용구 인식을 시도하였다. 관용구 인식은 문헌 [11]의 방법을 참조하여 시도하였다.

문헌 [13]은 모든 문자의 코드를 한글을 표현하는 2바이트 조합형 코드로 정규화 하여 패턴 매치를 수행하는 알고리즘을 제시하였다. 문헌 [14]는 n바이트 한글 코드, 2바이트 완성형 코드, 2바이트 조합형 코드, 3바이트 한글 코드에 대하여 영어를 위하여 개발된 여러 패턴 매치 알고리즘을 적용하고 실험 결과를 제시하였다. 이 연구들은 매우 기초적인 연구라고 할 수 있으며, 관용어 인식으로 발전하지는 않았다. 문헌 [15]는 Visual C++ MFC CString 데이터 타이프에 대하여 한글 문자열 처리를 위한 클래스 함수 구현에 대하여 논의하였다. 문헌 [16]은 데이터베이스 SQL 연산자 LIKE에서 한글 음절을 초성, 중성, 종성 단위까지 구분하여 패턴 매치하는 문자열 패턴 매치 알고리즘을 논의하였다.

문헌 [17]의 논문들은 대부분 언어학적인 측면에서 한국어 구문 분석에 대하여 논의하였다. 1장에서는 영어를 위하여 개발된 HPSG(Head-driven Phrase Structure Grammar)를 사용하여 한국어 구문 분석을 시도하였다. HPSG가 한국어 구문 표현에 적합한지는 논의할 필요가 있다. 5장의 146쪽에는 한국어 관용구 사전에 대한 설명으로 “핵어 표제어와 숙어 표제어의 상대적 위치를 ‘선행, 후행’의 탐색 방향으로 나누어 표시한다고” 논의하였으나 관용구 인식은 논의하지 않았다. 8장에서는 의존문법에 의한 한국어 구문 분석기를 논의하였으며 관용구와 연어 인식은 다루지 않았다. 문헌 [18]은 한국어 관용구의 저장, 전산 처리, 그리고 의미에 대하여 논의하였으며 XML 을 사용하여 관용구를 표현하고 저장하였다.

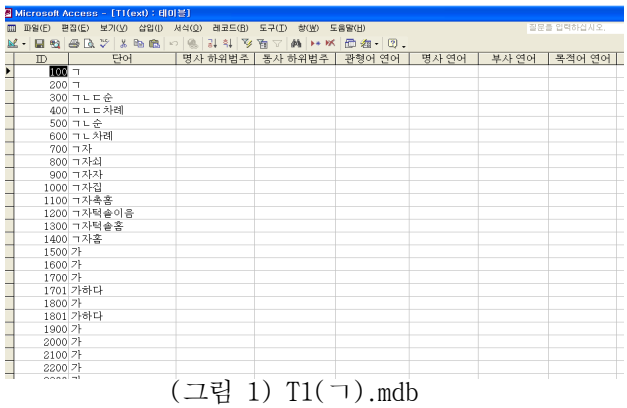
본 논문에서는 한국어 관용구를 어절 문자열 형태로 전자 사전에 저장하고, 이들 관용구를 입력 문장에서 그대로 패턴 매치 하여 인식하는 한국어 관용구 인식 알고리즘에 대하여 논의한다. 구문 분석기로 입력되는 입력 문장에서 어휘중의성 문제는 해결되어 있다고 가정한다[23].

2. 본 론

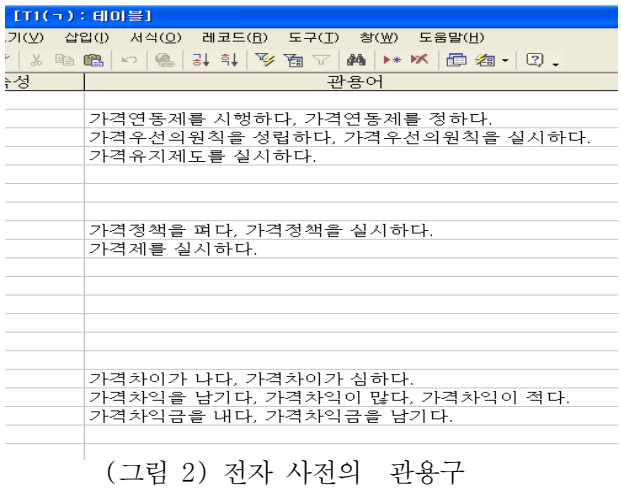
2.1 전자 사전

전자 사전은 마이크로소프트 엑세스 MDB를 사용하고 있으며 관용구는 해당 표제 단어의 관용구 필드(field)에 어절 문자열 형태로 저장된다. 전자 사전은 국립국어원의 표준국어대사전 초판을 참조하여 구성하였다[19]. 다음 (그림 1)은 전자 사전의 모양을 보여 준다. (그림 2)는 전자 사전에 저장되어 있는 어절 문자열 형태의 관용구를 보여 준다. 우선, 전자 사전의 표제어에 수록되어 있는 정보는 다음과 같다[23]. ID는 일련번호이고 단어는 표제어를 의미한다.

ID, 단어, 명사 하위범주, 동사 하위범주, 관형어 연어, 명사 연어, 부사 연어, 목적어 연어, 어미 연어, 명사 의미 속성, 목적어 의미 속성, 관용어, 품사(17가지 품사)



(그림 1) T1(ㄱ).mdb



(그림 2) 전자 사전의 관용구

구문 처리기는 우선 전자 사전으로부터 정보를 읽어서 사전 정보를 저장하는 자료 구조에 저장한다. 전자 사전으로부터 정보를 읽을 때, 전자 사전의 필드에 정보가 저장되어 있는 경우도 있고, 정보가 저장되어 있지 않은 경우도 있다. 정보가 저장되어 있는 경우는 정보를 읽으면서 적절하게 처리를 하여 자료 구조에 저장하고, 정보가 저장되어 있지 않는 경우는 0으로 기록해 둔다. 또한 현재는 동일한 단어는 5개까지 읽고 처리하여 자료 구조에 저장할 수 있도록 하였다.

2.2 관용구 인식 알고리즘

관용구는 기본적으로 단어 패턴 매치(words pattern matching)에 의하여 인식한다. 그러나 입력 문장에서 관용구를 직접 패턴 매치 하는 것은 매우 비효율적이다. 따라서 패턴 매치를 용이하게 수행할 수 있도록 관용구의 특징에 대한 정보가 필요하다. 이러한 정보에는 전체 관용구의 길이, 해당 어절의 관용구 내부에서의 위치, 그리고 해당 어절의 위치에 따른 문자열 패턴 매치의 방향 등이 있다. 문자열 패턴 매치의 방향에는 전(forward)방향, 후(backward)방향, 그리고 양(bidirectional)방향이 있다. 전방향이라는 것은 해당 어절이 제일 왼쪽에 위치하기 때문에 왼쪽에서 오른쪽으로 관용구 패턴 매치를 수행한다는 의미이고, 후방향이라는 것은 해당 어절이 가장 오른쪽에 위치하기 때문에 오른쪽에서 왼쪽으로 관용구 패턴 매치를 수행한다는 의미이다. 양방향이라는 것은 해당 어절이 관용구의 내부에 위치하기 때문에 양쪽으로 패턴 매치를 수행해야 한다는 의미이다. 관용구 “같은 값이면 다홍치마” 가 양방향의 경우에 해당된다. 길이 정보와 방향 정보는 해당 단어의 관용구를 전자 사전으로부터 읽고 처리하여 내부의 자료 구조에 저장하는 과정에서 구할 수 있다. 사전에서 관용구를 읽어서 처리할 때에는, 어절의 실제 문자열과 형태소 분석된 어절의 어간을 동시에 사용한다. 관용구가 없는 경우에는 길이 정보가 0으로 나타난다.

관용구 인식 알고리즘은 미리 파악된 이러한 정보들을 바탕으로 문장에서 관용구를 효율적으로 패턴 매치하여 인식할 수가 있다. 관용구 인식의 결과는 관용구 시작 노드와 마지막 노드가 해당 단어에 기록되는 형태로 나타난다.

관용구 인식을 C 코드로 나타내면 다음과 같다.

```
struct idom {
      int   dir;
      int   tlen;
      int   alen;
      char  idiom[ ];
};

void idiom_recognition(dir, idiom) {
   if(dir==forward)
      pat_match(forward, tlen, 0, idiom);
   else
   if(dir==backward)
      pat_match(backward, tlen, 0, idiom);
   else
   if(dir==bidirectional)
      pat_match(bidirectional, tlen, alen, idiom);
}
```

2.3 연어 인식

연어 인식은 구문 처리가 완료된 다음에 수행한다. 전자 사전으로부터 읽어서 자료 구조에 저장되어 있는 연어 정보를 구문 처리 결과인 의존 구조를 탐색하면서 연어 인식을 시도한다. 연어 인식 결과는 연어 시작 노드와 마지막 노드의 인덱스가 해당 단어에 기록되어 나타난다.

단어 “바닥”을 예를 들면, 단어 “바닥” 이 뒤에 나타나는 “마루 바닥”, “시장 바닥” 등의 연어가 있고, “바닥” 이 앞에 나타나는 “바닥 장세” 등의 연어가 있다[20]. 연어 정보는 구분되어 전자 사전에 저장된다.

2.4 의미 속성 처리

의미 속성 처리는 명사들의 의미 속성을 비교하고 처리하여 명사들의 수식 범위를 결정하는 것이다. 다음 문장을 예로 들면,

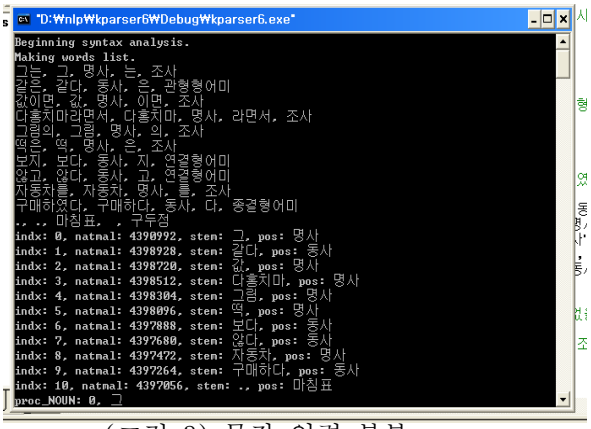
(예 1) 나는 고기와 밥을 먹었다.
(예 2) 나는 영수와 밥을 먹었다.

명사 “나”, “고기”, “밥”의 의미 속성을 비교하여, (예 1)에서는 “고기와 밥을” 연결하고, (예 2)에서는 “나는 영수와” 를 연결하는 것이다. 전자 사전에 의미 속성은 외연, 내포 등으로 구분되어 저장된다. 의미 속성은 “+음식물”, “+사람”, “-여성” 등의 형태이다[21].

구문 처리 과정에서 명사 스택에 저장되어 있는 명사구들을 별도의 자료 구조로 옮긴 다음에 명사들의 의미 속성을 사전에서 읽어서 서로 비교하면서 처리한다. 전자 사전에 해당 단어의 의미 속성 정보가 없는 경우에는 의미 속성 처리를 수행하지 않는다.

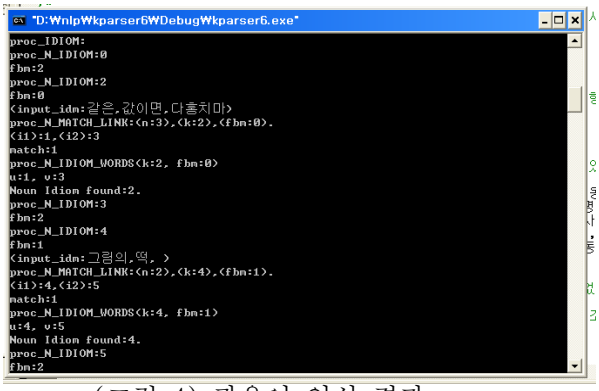
2.5 구문 분석 처리 결과

다음 (그림 3)에서부터 (그림 7)까지는 문장 “그는 같은 값이면 다홍치마라면서 그림의 떡은 보지 않고 자동차를 구매하였다.”에 대한 입력과 구문 처리 결과들이다. 이 문장은 두 개의 관용어 “같은 값이면 다홍치마”와 “그림의 떡”을 포함하고 있다. 우선 (그림 3)은 문장 입력을 보여준다.



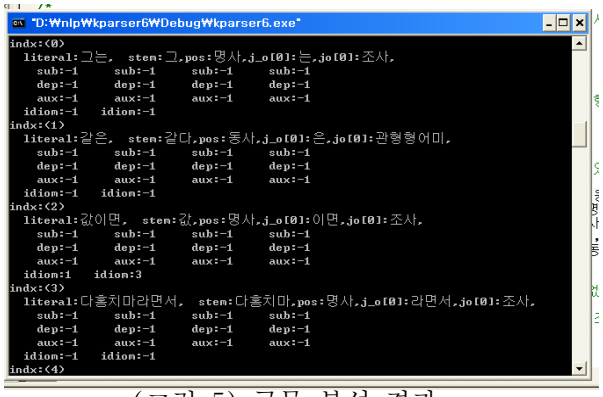
(그림 3) 문장 입력 부분

다음 (그림 4)는 관용어가 인식된 결과를 보여준다. (그림 5)부터 (그림 7)까지는 (그림 3)의 입력 문장에 대한 구문 처리 결과이다. 하나의 그림에 구문 처리 결과를 모두 나타낼 수가 없어서 그림을 분할하여 나타내었다.

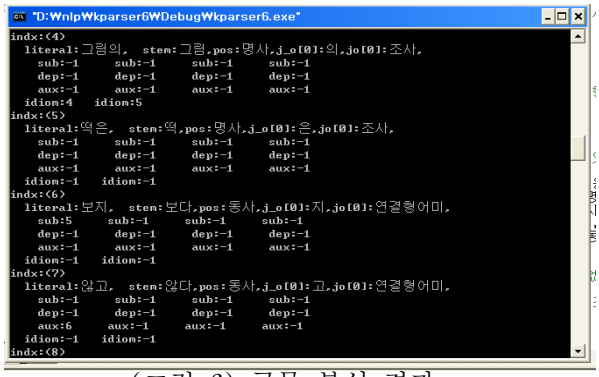


(그림 4) 관용어 인식 결과

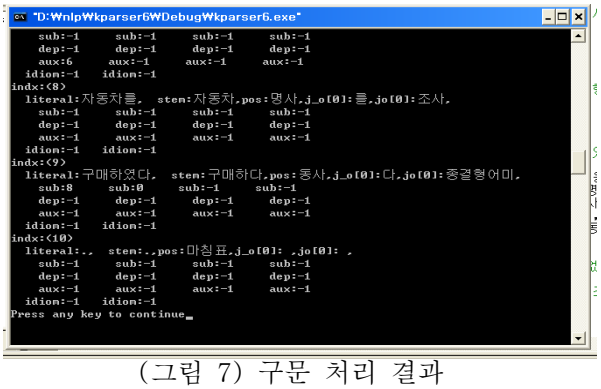
(그림 4)를 보면 관용구 "같은 값이면 다홍치마"와 "그림의 떡"이 인식되었다는 표시가 <input_idm:같은, 값이면, 다홍치마>와 <input_idm: 그림의, 떡,> 의 형태로 나타나 있다.



(그림 5) 구문 분석 결과



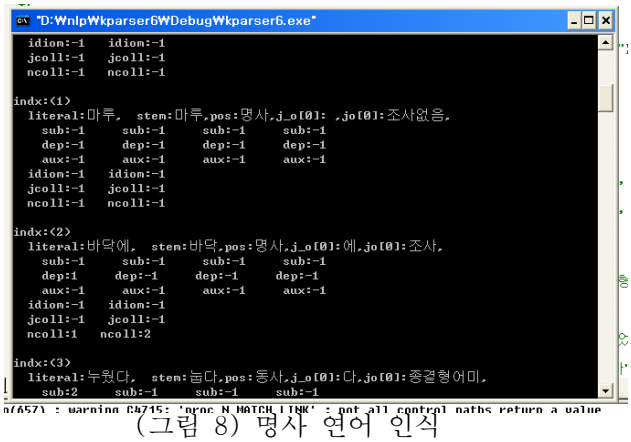
(그림 6) 구문 분석 결과



(그림 7) 구문 처리 결과

(그림 5)를 보면 indx:<2> 노드 "값이면" 아래에 idiom:1 idiom:3 이라는 표시가 있는데, 이것은 (1)번 노드부터 (3)번 노드까지가 "값이"와 관련된 관용어 "같은 값이면 다홍치마" 라는 의미이다. (그림 6)에도 indx:<4> 노드 "그림의" 아래에 idiom:4 idiom:5 라는 표시가 있는데, 이것도 (4)번 노드부터 (5)번 노드까지가 "그림의 떡"이라는 관용구라는 의미이다.

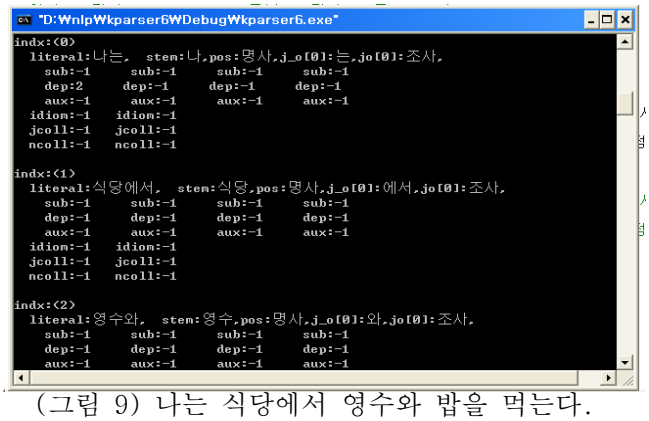
다음 (그림 8)은 명사 연어 인식에 대한 예이다. (그림 8)은 "그는 마루 바닥에 누웠다" 문장에 존재하는 명사 연어 "마루 바닥"을 인식한 실험 결과를 보여 준다.



(그림 8) 명사 연어 인식

어절 "바닥에"에 ncoll:1 ncoll:2가 보인다. 이것은 노드 1부터 노드 2까지가 "바닥"의 연어라는 의미이다. 이 경우는 명사 연어 "마루 바닥"을 의미한다.

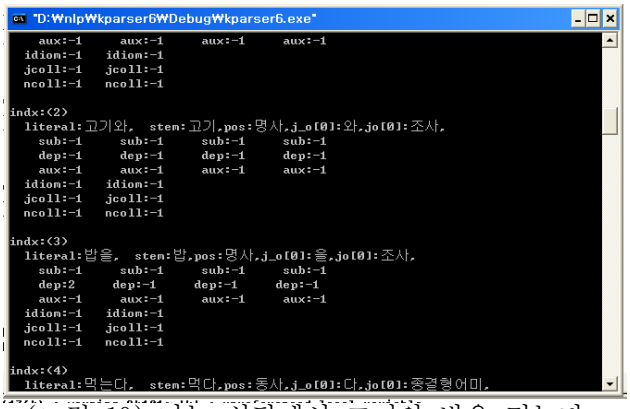
다음 (그림 9)와 (그림 10)은 의미 속성에 의한 구문 처리 결과이다. 두 개의 문장 "나는 식당에서 영수와 밥을 먹는다."와 "나는 식당에서 고기와 밥을 먹는다."를 제시하여, 단어 "영수"와 "고기"의 의미 속성에 따라서 구문 분석이 다르게 이루어진 결과를 보여 준다. 여기서 "영수"의 의미 속성은 "+사람"이고 "고기"의 의미 속성은 "+음식물"이다.



(그림 9) 나는 식당에서 영수와 밥을 먹는다.

(그림 9)를 보면, "나는" 노드에서 dep:2를 볼 수 있다. 이것은 노드 <0> "나는" 노드가 노드 <1> "식당에

서” 노드 다음에 있는 노드 <2> “영수와” 노드와 관련이 된다는 의미이다.

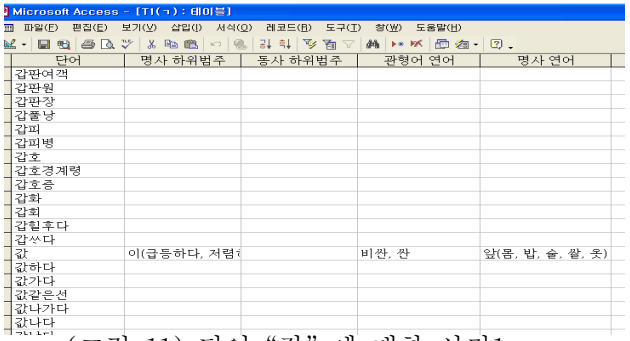


(그림 10) 나는 식당에서 고기와 밥을 먹는다.

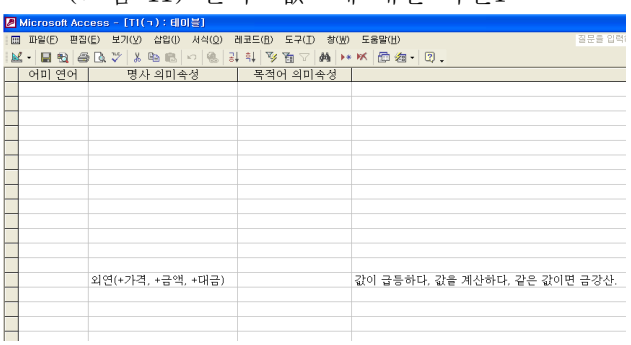
(그림 10)을 보면 노드 <3>“밥을” 노드에서 dep:2를 볼 수 있다. 이것은 노드 <3>“밥을” 노드가 노드 <2>“고기와” 노드와 관련이 있다는 의미이다.

구문 분석 결과에서 sub는 하위범주를 나타내고, dep는 단어들 간의 연결 관계를 나타낸다. dep을 사용하여 연어 관계도 나타낸다. aux는 보조 용언(보조 동사, 보조 형용사) 관계를 나타낸다. “-1” 값은 정보가 없다는 것을 의미하고 양수 값은 관계가 있는 노드의 인덱스 값을 의미한다.

다음은 또 다른 문장 “그는 같은 값이면 금강산을 구입하였다.”에 대한 처리 결과이다. 이 문장에서 관용어는 “같은 값이면 금강산”으로 가정하였다.

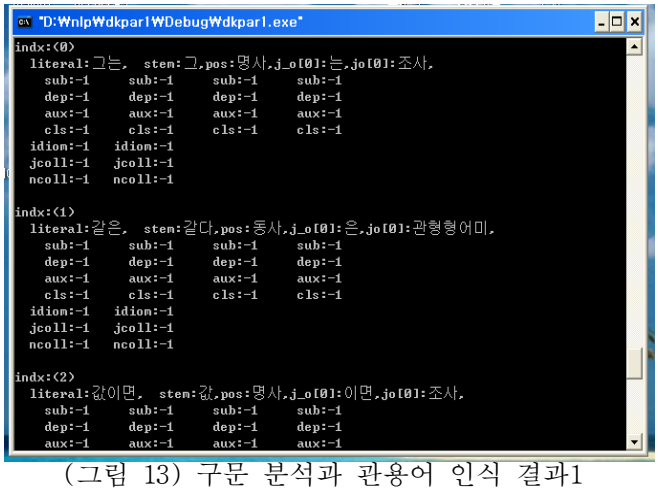


(그림 11) 단어 “값”에 대한 사전1

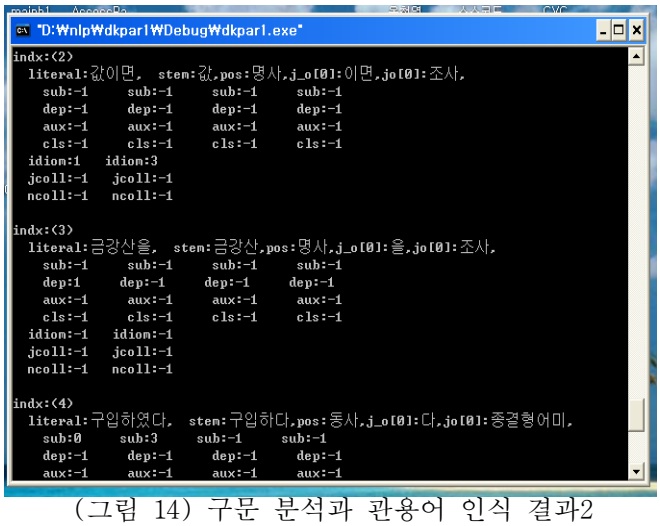


(그림 12) 단어 “값”에 대한 사전2

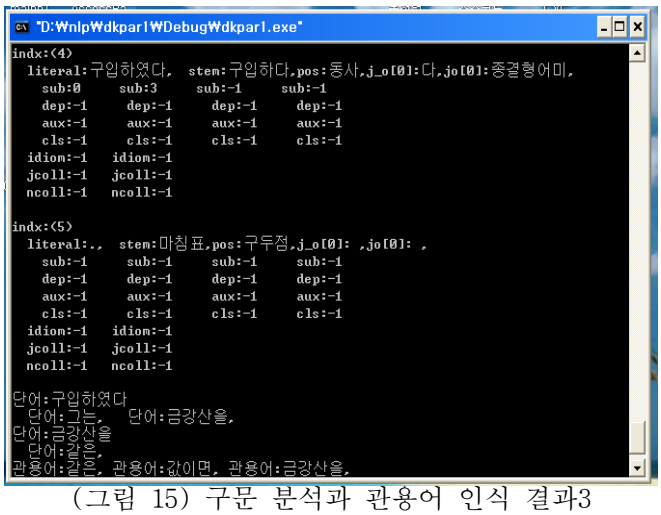
구문 분석과 관용어 인식 결과는 다음과 같다.



(그림 13) 구문 분석과 관용어 인식 결과1



(그림 14) 구문 분석과 관용어 인식 결과2



(그림 15) 구문 분석과 관용어 인식 결과3

단어 “값”에 대한 사전은 (그림 11)과 (그림 12)로 분할하여 나타내었다. 구문 분석과 관용어 인식 결과는 (그림 13) ~ (그림 15)로 분할하여 나타내었다. (그림 14) indx:<2> 노드를 보면 idiom:1 idiom:3 으로 표시가 되어 있다. 이것은 노

드 1부터 노드 3까지가 관용구라는 것을 의미한다. 여기서는 “같은 값이면 금강산”을 의미한다. 또한 (그림 15)의 indx:<4> 노드를 보면 sub:0 sub:3을 볼 수 있다. 이것은 동사 “구입하였다”가 노드 0과 노드 3을 하위범주로 택하였다는 것을 의미한다. 노드 0은 단어 “그는" 을 나타내고 노드 3은 단어 ” 금강산을“ 나타낸다. 이 실험 결과들은 문장에서 관용구 인식과 하위범주 결합이 성공적으로 처리되었다는 것을 의미한다.

2.6 구문 처리기 구조

구문 처리기는 크게 두 부분으로 구성된다. 하나는 구문 처리를 담당하는 부분이고, 다른 부분은 전자 사전으로부터 정보를 읽고 적절한 처리를 수행한 다음에 전자 사전 자료 구조에 저장하는 부분이다.

구문 처리기는 단어 문법 개념에 의하여 구성되었다. 단어 문법에 대하여서는 참고 문헌 [22][23]에 자세히 논의되어 있다. 구문 처리기는 크게 명사를 처리하는 부분, 동사를 처리하는 부분, 관용구를 인식하는 부분, 그리고 연어를 인식하는 부분으로 구성된다. 명사를 처리하는 부분에서는 명사, 복합 명사, 조사 등을 처리한다. 동사를 처리하는 부분에서는 동사 하위범주, 명사 하위범주, 명사 의미 속성, 그리고 보조 용언 등을 처리한다. 관용구를 인식하는 부분에서는 입력 문장을 왼쪽에서 오른쪽으로 읽으면서 관용구 인식을 위한 패턴 매치를 수행한다. 연어를 인식하는 부분에서는 구문 처리가 완료된 결과에 대하여 의존 관계 링크를 조사하면서 연어 인식을 수행한다.

전자 사전으로부터 정보를 읽어 들이는 부분은 입력 문장을 구성하는 단어에 대한 레코드를 엑세스 MDB로부터 읽어 들인다. 레코드를 구성하는 필드들을 읽으면서 각 필드에 저장된 정보를 적절하게 처리하여 전자 사전을 위한 자료 구조에 저장한다. 전자 사전의 정보들은 구문 처리를 수행하는 과정에서 활용된다. 관용구의 경우에는 이 과정에서 관용구 인식에 필요한 정보를 계산하여 자료 구조에 저장한다. 현재 관용구의 단어 문자열 길이는 5개까지 허용하고 있다.

3. 결 론

본 논문에서는 한국어 관용구에 대한 설명과 관용구의 인식 방법을 논의하였다. 관용구는 3가지의 방향을 고려하여 어절 문자열 패턴 매치 알고리즘을 수행하여 인식한다. 관용구 인식에 필요한 정보는 관용구를 전자 사전으로부터 읽어 들이는 과정에서 모두 구할 수 있다. 그 밖에도 연어 인식 방법과 의미 속성 처리 방법 등도 논의하였다. 그리고 구문 처리 실험 결과를 제시하여 관용구 인식 결과, 연어 인식 결과, 그리고 의미 속성 처리 결과를 보였으며 구문 처리기의 전체 구조도 설명하였다.

앞으로의 연구로는 전자 사전의 내용을 점검하고 확충하는 것이다. 또한 관용구 인식 알고리즘도 많은 한국어 관용구를 대상으로 실험을 수행하여 인식률을 향상시키도록 노력하는 것이다.

이러한 연구의 목적은 새로운 방법으로 한국어를 처리하는 노력을 계속 수행하여 한국어 정보 처리에 더욱 적합한 방법을 찾는 것이다. 그리고 새로운 방법을 많은 한국어 응용에 적용할 수 있는 기회를 제공하여 한국어 응용의 저변을 확대시키고 한국어에 대한 인식을 전반적으로 향상시키는 것이다.

참고 문헌

[1] 박영준, 최경봉, 관용어 사전, 태학사, 2007.
[2] 임홍빈 외, 한국어 구문 분석 방법론, 한국문화사, 2003.
[3] 이호석 외, “영어-한글(한국어) 기계 번역 시스템의 설계 및 구현,” 1984년도 한국정보과학회 가을 학술발표대회논문집, 제13권, 제2호(하),pp.83~90, 1984.
[4] 윤성희 외, “기계번역을 위한 자연 언어의 숙어적 분석,” 한국정보과학회 논문지, 제 20권, 제 8호, pp.1148 ~ 1158, 1993,8.
[5] 윤성희 외, “관용적 표현의 대응관계에 기반한 영어-한국어 기계 번역,” 제 5회 한글 및 한국어 정보처리 학술발표대회 논문집, 1993.
[6] 정한민 외, “효율적인 영한 번역을 위한 복합 단위 인식기 설계,” 1996년도 한국정보과학회 가을 학술발표대회 논문집, 제 23권, 제 2호, 1996.
[7] 정한민 외, “부분 파싱을 이용한 복합 단위 인식,” 1997년도 한국정보과학회 가을 학술발표대회 논문집, 제 24권, 제 2호, 1997.
[8] 전현경 외, “영한기계번역에서의 단일화 문법에 기반한 복합단위형태소 주도의 관용어 처리,” 1998년도 한국정보과학회 봄 학술발표대회 논문집, 제 25권, 제 1호, 1998.
[9] 이호석 외, “영한 변환사전 생성을 위한 말뭉치에 기반한 연어와 관용어의 자동 추출,” 한국정보과학회 논문지 제21권 제11호, pp. 2110 ~ 2117, 1994,11.
[10] 전현경 외, “영한기계번역에서의 단일화문법에 기반 한 복합단위형태소 주도의 관용어 처리,” 한국정보과학회 1998년도 봄 학술발표논문집 제25권 제1호(B), pp. 387 ~ 389, 1998,4.
[11] 이하규 외, “기계 번역을 위한 한국어 숙어의 표현 및 인식,” 한국정보과학회 논문지, 제21권, 제 1호, pp.139 ~ 149, 1994,1.
[12] 양재형 외, “다중 지식원을 이용한 한국어의 분석,” 한국정보과학회 논문지, 제 21권, 제 7호, pp.1324 ~ 1332, 1994,7.
[13] 이영진, 윤지희, “한국어 텍스트를 위한 pattern matching 알고리즘의 개발,” 한국정보과학회 봄 학술발표논문집, 제 17권, 제1호, 1990.
[14] 정부금, 이광수, “한글 텍스트를 위한 패턴 매칭 알고리즘에 관한 연구,” 한국정보과학회 가을 학술발표논문집, 제18권, 제2호, 1991.
[15] 윤지현, 변정용, “한글 문자열 처리를 위한 클래스 라이브러리,” 한국정보과학회 봄 학술발표논문집, 제26권, 제1호, 1999.
[16] 박성철 외, “연산자 LIKE의 새로운 한글 탐색 패턴,” 한국정보과학회: 데이터베이스, 제34권, 제3호, 2007,6.
[17] 김종복 외, 한국어 정보화와 구문분석, 월인, 2004.
[18] 이동혁, 한국어 관용 표현의 정보화와 전산 처리, 도서출판 역락, 2007.
[19] 국립국어원, 표준국어대사전 초판, 2000.
[20] 김하수 외, 한국어 연어사전, 커뮤니케이션북스, 2007.
[21] 홍사만, 국어의미분석론, 한국문화사, 2008.
[22] 이호석, “한국어 파싱을 위한 새로운 알고리즘 설계,” 한국정보과학회 2008 가을 학술발표논문집, 제35권 제2호(C), pp. 192 ~ 197, 2008,10.
[23] 이호석, “하위범주, 첨가소, 의미 속성을 활용한 한국어 구문 분석,” 한국정보과학회 2009 한국컴퓨터종합학술대회 논문집, 제36권, 제1호(C), pp. 346 ~ 351, 2009,6.