확장 사전 환경에서의 한국어 형태소 해석과 생성

조영환, 차희준, 김길창

한국과학기술원 전산학과

Morphological Processing in an Expanded Dictionary Environment

Young Hwan Cho, Hee Joon Cha, Gil Chang Kim

Dept. of Computer Science
Korea Advanced Institute of Science and Technology

요약

형태소 처리의 기본 원칙은 사전의 표제어를 형태소 수준으로 함으로써 사전의 크기를 줄이고, 중복되는 정보의 양을 최소화하는 것이다. 본 논문에서는 형태소 처리를 위한 여러 환경 요소들 중에서 특별히 확장된 사전 표제어를 기본으로 하는 환경을 제안한다. 확장 사전 환경은 어휘에 대한 사전 표제어와 사전 정보의 분리를 기본으로 한다. 기본 사전 표제어에 대하여 어휘의 활용형을 사전 작성의 후처리인 사전 표제어에 대한 색인구조 구성시에 자동으로 확장함으로써 용언의 불규칙 활용과 음운 축약 현상에 대처한다. 확장 사전 환경의 장점은 형태소 해석과 생성시에 필요한 불규칙 활용에 대한 처리를 사전 확장 시간으로 앞당기고, 어절의 부분문자열과 사전 표제어 간의 직접 대응성을 제공하여 여러 응응에 쉽게 적용이 가능하다는 것이다.

1장 형태소 해석에 대한 관점들

컴퓨터를 이용하여 자연어를 처리하는 데에 있어서 어휘에 대한 정보를 저장하는 구조를 사전이라 한다. 그러므로 사전의 어휘정보를 이용하기 위하여 사전을 참조하고, 적당한 정보를 어휘에 대응시키는 작업을 형태소 해석이라고 한다. 형태소 해석에 관한 언어학적 관점은 어절을 구성하는 형태소간의 문법적 결합관계를 올바르게 해석하는 것에 있다. 그러나 전산학적인 입장에서의 형태소 해석은 어절에 대한 사전

정보를 적재하기 위하여 어절을 구성하는 사전 표제어 혹은 형태소를 파악하여 적절한 사전 정보를 적재하는 것이다.

영어에는 접사들이 수가 한정되어 있고, 그들의 쓰임이 비교적 정형화 되어 있기 때문에 접사의 의미를 사전적 정보로 저장하지 않고, 접사 처리 규칙으로 처리하는 경우가 많다. 구문 해석의 경우, 어휘 자체에 대한 의미 대신에 어휘가 갖는 문법적인 범주를 대상으로 구조를 밝히는 것이므로 이들 접사에서 얻을 수 있는 구문 정보를 이용할 수 있었다. 그러나 일반적으로 접사에 의하여 의미정보 혹은 개념 정보를 얻을 수 없기 때문에 구문 해석 이상의 처리를 위해서는 영어의 경우도 사전 표제어와 정보의 확장이 불가피하다.

한국어의 경우에는 어휘의 활용이 문법적 역할과 문장의 서술양상을 표현하는 등의 역할을 하기 때문에 접사의 수가 많고, 역할 또한 다양하다. 그러므로 접사적 성격을 갖는 조사, 어미류를 사전 혹은 선언적인 저장 방식에 의하여 저장하고, 이들이 갖는 문법적인 성격 즉, 형태소가 결합하여 어절을 이루는 방식을 형태소 문법으로 정의하여야 한다. 형태소 해석의 경우는 입력 어절에 대하여 사전 표제어로 분리하고, 각 사전 표제어들 간의 문법적 적합성을 검증하여 사전 정보를 적재한다. 반대로 형태소 생성의 경우는 단위 형태소 혹은 생성을 위한 사전 엔트리의 나열에 대하여 이형태의 선택과 결합에 대한 철자 변형 처리를 거쳐 어절을 합성하게 된다.

형태소 해석에 대하여 누구나가 최초로 접하는 문제는 한글 코드 표현의 문제이다. 그리고 마지막으로 접하는 문제는 많은 경우, 가능한 여러 구조에 대하여 최상의 후보를 선택하는 문제, 즉 애매성 해소가 될 것이다. 다음은 형태소 해석의 일반적인 고려 사항들에 대하여 나열하여 본다[과기원, 92].

가. 한글 코드 : 한글에 대해서 내부에서 표현하는 코드 형태
   예) N 바이트 풀어쓰기, 3바이트 방식 등
나. 사전 표제어 색인의 방법 : 사전에 등록된 형태소에 대하여 물리적인 색인 방법
   예) TRIE 구조, B tree 구조 등
다. 사전의 분류 : 사전의 종류를 나누고 각각의 요건에 해당하는 사전 표제어를 관리하기 위하여 분류
   예) 통합 단일 사전, 품사별 사전, head와 tail사전 등
라. 사전 표제어 규정 : 복합명사, 복합조사, 접사에 의한 파생어 등에 대한 표제어 설정 여부와 형태소에 대한 활용형, 축약형에 대한 허용여부
   예) 복합형태소, 활용형 형태소 등
마. 어절 모델 : 어절을 단일의 단위로 볼 것인가, 둘 이상의 고정적인 혹은 가변적인 부분으로 나누어 고려할 것인가에 대한 모델

예) Head-Tail 모델, 고정 N 단위 모델, 가변 N 단위 모델 등

바. 어절 구성 문법 : 어절 모델에 대하여 구 구성을 표현하는 형태소 해석 문법의 표현 형태

예) Finite State Transducer, Descriptive Grammar 등

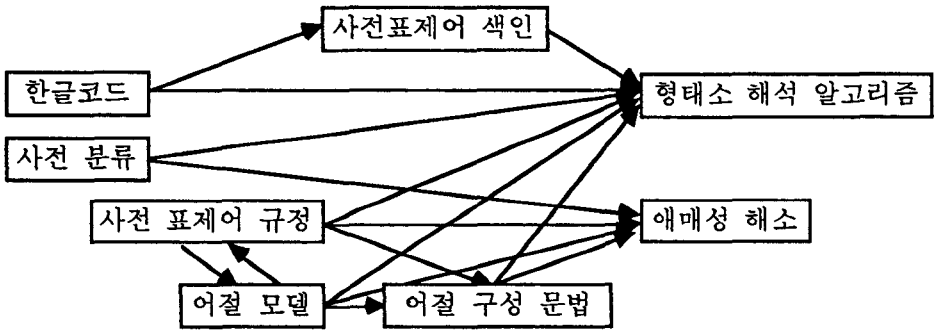
사. 형태소 해석 알고리즘 : 어절을 형태소 문법에 맞추어 Parsing하는 알고리즘

예) Chart Based, Recursive Model, Head-Tail Model 등

아. 애매성 해소 원칙 : 둘 이상의 해석이 가능할 때 이들의 순서를 부여하는 원칙

예) 무원칙, 확률모델, 최장일치 선택, 최단일치 선택 등

위와 같은 고려사항들에 대하여 다음과 같은 관계를 상호 간에 가지고 있다고 볼 수 있다.



본 논문에서 고려하는 확장 사전이란 개념은 사전 표제어의 구성 원칙의 일부가 된다고 볼 수 있다. 즉, 기본적인 표제어에 대한 정보를 저장하는 기본 사전에 대하여 특별히, 불규칙 용언과 음운축약현상에 대하여, 이들이 가질 수 있는 변형을 자동으로 확장시켜 확장 사전을 구성하는 것이다. 그러므로 기본 사전을 구성하는 노력은 기존의 방식과 동일 하면서도, 불규칙 용언 처리를 확장사전 작성시로 앞당기게 되어 전체적으로 형태소 해석 엔진에 대하여 부담을 낮추어 주는 효과를 갖도록 한다.

초기의 형태소 해석에 관하여 중요하게 여긴 것은 한국어에 대하여 형태소 해석 알고리즘을 어떻게 구성하는가 하는 것이었다[김성용, 87]. 이 분야에 대한 연구는 이제는 어느 정도 정리가 되었다고 할 수 있다. 최근의 관심의 중심은 용언의 불규칙 현상과 음운축약이었다. 문장 중에 형태소 해석의 대부분이 되는 체언류와 수식언에 해당하는 어절의 경우는 결합규칙 상에 형태적 변형을 수반하지 않는다. 용언류에 대하여는 동사 혹은 형용사 용언 어간의 불규칙적 활용과 어간과 어미 사이의 음

운 축약 현상이 비교적 복잡하여 이에 대한 효율적인 처리 방법이 중요한 문제로 보였다[강승식, 92]. 특별히 정형화된 Formalism을 한국어에 적용하여 형태소 해석을 하려는 시도도 있었다[이성진, 92].

본 논문에서는 확장 사전 환경을 기본, 사전 표제어 규격으로 인하여 영향을 받는 어절 모델, 어절 구성 문법의 표현, 형태소 해석, 생성 알고리즘에 관하여 간단하게 고찰하여 보고, 이러한 환경이 제공하는 다양한 잇점에 대하여 논하려고 한다. 그러나 본 논문에서는 우리가 적용한 형태소 해석 알고리즘이나, 어절 모델의 적합성에 대하여 논하는 것은 피하고, 확장 사전 환경을 위한 사전 시스템에 대하여 주안점을 두어 논한다.

2장 확장 사전 환경

사전 시스템은 전체적인 지식 체계상에서 사전으로 처리할 정보와 규칙 혹은 지식 베이스로 보관해야할 정보, 문법적 지식에 대한 보관에 대한 총체적인 결정에 의해 그 역할이 결정된다[과기원, 87]. 또한 시스템의 요구에 의해 언어 정보의 소스(source)에 대하여 원(original) 정보의 가공과 추가적인 정보의 생성, 정보의 형식화, 사전 인덱스의 생성 등의 일련의 처리를 통하여 확장사전으로 변형된다.

일반적으로 한국어 형태소 처리와 구문, 의미 처리에서 사용하는 정적인 지식은 사전이라는 데이타 구조로 표현한다. 사전 구조는 Key word를 검색하기 위하여 표제어 부분에 대해 색인 구조를 갖으며, Key word에 대한 지식을 저장하기 위하여 표제어에 대응하는 실제 정보 부분으로 구성되는 구조라고 볼 수 있다. 형태소 처리를 위해서는 표제어가 갖는 형태론적인 정보 즉, 형태소 문법상의 범주(category)에 대한 지식이 필요하며, 더불어 표제어의 출연 빈도, 다른 표제어와의 어울림 정도 등은 필수적이지는 않지만 유용한 정보이다. 구문적인 정보로는 구문 문법상의 범주인 품사 나눔, 품사별로 표제어가 갖는 정보 등이 있다. 의미적인 정보로는 표제어가 갖는 의미적인 속성 즉, 표제어의 출연 환경에 대한 정보가 있다. 본 논문은 특별히 형태소 처리에 관한, 즉, 어절에 대한 사전 정보의 적재에 촛점을 맞추므로 관심을 두는 정보 체계는 형태소 간의 결합 가능성에 대한 것으로 국한한다.

사전 엔트리와 정보의 정확성은 실용적인 시스템을 구축하는데 매우 중요한 요소이다. 한국어 처리 시스템에서 사전 엔트리의 구성은 형태소 수준의 단위를 기준으로 하므로 사전 엔트리에 대한 결정과 이를 인덱싱 하는 방법은 한국어 사전 엔트리의 검색에 있어서 특별한 의미를 지닌다. 사전 시스템은 사전 작성자에게는 작성과 관리가 쉽도록 하는 배려가 있어야하고, 실제의 사전 이용자에게는 효율적인 처리의 방법을 제공하여야 한다. 역으로 사전 시스템은 사전의 내용에 대한 관리가 쉽도록

고려되어야 하고, 또한 효율적인 인덱싱 방법이 요구된다.

한국어에서 대부분의 단어가 명사와 동사이고 기능어는 극히 일부이기 때문에 기능어에 대한 사전 정보를 미리 구축해 놓으면 계속적인 사전 항목의 추가는 명사와 동사에 국한된다. 기능어 사전의 문법 정보는 매우 복잡하지만 일단 기능어 사전이 구축되면, 명사의 경우는 문법 정보의 부여가 간단하고, 분야별로 사용되는 단어가 차별화되기 때문에 분야에 특정적인 명사의 계속적인 확장이 비교적 용이하다. 용언에 대한 기존의 관점은 가능한한 원형만을 사전 표제어로 등록하고, 어절에서 보이는 활용형태에 대해서는 원형 복원 처리를 거쳐 사전을 탐색하자는 것이었다.

일반적으로 어절에서 사전 표제어를 분리하는 과정은 원형의 복원 여부에 따라 나뉘게 된다. 원형복원 방법은 사전에 용언 어간에 대하여 원형만을 표제어로 하고 용언 어간의 활용형과 음운축약형에 대해서는 사전 검색을 위하여 특별한 처리를 거쳐서 용언 어간의 원형과 활용 어미의 원형을 복원하여 사전 엔트리와 비교하게 된다. 이러한 원형 복원 방법은 용언이 아닌 다수의 어절에 대하여 불필요한 처리를 할 가망성이 높고, 처리의 복잡도가 높아지지만 사전에 원형만을 저장함으로써 사전의 크기를 줄인다는 장점때문에 널리 고려되고 있다. 다음의 예는 "나는"과 "조는"에 대하여 원형 복원을 이용한 방법에서 해석하게 되는 과정이다.

나는	->	나 + 는	-> 나 + 는 {나자신, 가나다의 나}
	->	나 + 는	-> 나다 + 는 {싹이 나다, 신이 나다}
	->	나 (+ㄹ) + 는	-> 날다 + 는 {하늘을 날다}
조는	->	조 + 는	{ 사전 검색 후, 문법 성공 }
		조 (+ㄹ) + 는	{ 원형복원, 사전 검색, 문법 성공 }
		조 (+ㅂ) + 는	{ 원형복원, 사전 검색, 문법 실패 }
		조 (+ㅎ) + 는	{ 원형복원, 사전 검색, 문법 실패 }

원형 방법을 이용하면 위와 같이 원형 복원과 문법 검사 작업을 하여야 한다. 사전을 원형 형태소로만 구성하여 사전 정보의 중복과 불일치를 제거하려면 변형된 어절에서 원형의 형태소를 발견하도록 어절을 역변형시키는 형태소의 원형 복원 방법을 사용하여야 한다. 원형복원의 방법은 어미의 원형정보와 어간에 대한 불규칙 정보를 규칙화하여 처리하는 방법론이 필요하다. 이때 추가적인 원형 복원 방법의 적용이 외에도 문법 형태소의 구성이 까다롭고, 문법 형태소를 우선적으로 해석해야 하므로 오른쪽에서 왼쪽으로 해석해 나가야 한다는 알고리즘상의 제약이 발생한다.

만약 사전의 표제어가 용언의 활용형을 포함한다면 위의 경우에 대하여 다음과 같은 해석 과정이 필요하게 된다.

나는 -> {나, 나다, 날다} + 는 { 모두 성공 }
조는 -> {조, 졸다, 좁다, 좋다} + 는 { 일부 성공 }

문법적으로 변형된 형태소에 대하여 사전을 구성한다면 형태소 변형 현상은 사전 표제어의 적절한 대응에 의하여 쉽게 해결할 수가 있다. 이것은 예외적인 현상을 사전 표제어로 등록하는 것을 허용하기 용이하기 때문이다. 기본 사전을 활용형으로 작성하고 관리한다면, 사전 표제어의 관리가 그리 쉽지 않을 것이다. 반면에 형태소 해석 알고리즘을 간단하게 하고, 예외적인 현상에 대해 대처하기가 용이하기 때문에 처리상의 오류도 적게 발생시킨다. 장점은 취하고 단점은 제거하기 위하여, 본 논문에서 제안하는 확장 사전 환경은 기본 사전은 사전 표제어의 원형으로 유지하고, 실제로 형태소 처리를 위하여 사용하는 확장 사전은 활용 형태를 포함하는 확장 표제어를 자동적으로 작성하도록 기본 사전으로부터 변형, 혹은 확장하는 것이다.

3장 확장 사전 환경을 위한 사전작성도구

확장 사전 환경을 위한 사전작성도구는 크게 두개의 단계로 나뉜다. 하나는 사전에 표제어로 등록하려는 형태소에 대하여 정보를 기술하는 단계이고, 다른 하나는 기본 사전에서 색인 기능을 갖는 확장사전으로 변환하는 것이다. 여기에 기본 사전의 내용이 시스템의 전반적인 변경에 비교적 독립적으로 유지되기 위해서 기본 사전과 확장 사전 사이에 중간에 임시 사전 형태를 정의하고 임시 사전상의 프로세싱을 고려하여야 한다. 기본 사전에 심볼로 표현된 자료 형태를 숫자 형태로 자동 변경하고, 접속 범주에서 접속 번호를 자동으로 생성하며, 용언의 활용형에 대한 사전 쉬트를 자동으로 생성하고, 활용형태에 따라 접속 번호를 변경하는 등의 일련의 변형을 거쳐, 사전 엔트리에 대한 유일화 작업을 하여 사전 표제어 인덱스를 생성한다.

용언 어간에 대해서는 어간의 활용형, 축약형 표제어의 생성을 위하여 불규칙 활용에 대한 구분을 한다. 이때 사용된 구분자들은 사전작성도구에 의해서 해당하는 문법 범주로 자동 확장된다. 용언의 축약형을 허용하기 위하여 초성이 "ㅇ"으로 시작되는 어미에 대하여 축약 현상에 대응하는 어미 즉, "서/도/지/..." 등을 별개로 둔다. 용언 어간의 활용형과 음운 축약형의 처리를 위하여 다음과 같은 불규칙 현상에 대하여 활용형과 축약형의 생성과 문법 정보 자동 변환이 필요하다.

1) 동사 어간의 활용형 생성

가. "으" 불규칙 : 쓰- -> 쓰, 써 (쓰+어),
따르- -> 따르, 따라 (따르+아)

나. "ㄹ" 불규칙 : 살- -> 사-,
놀- -> 노-,
들- -> 드-

다. "ㅅ" 불규칙 : 잇- -> 이-,
낫- -> 나-,
짓- -> 지-

라. "ㄷ" 불규칙 : 일컫- -> 일컬-,
엿듣- -> 엿들-

마. "ㅂ" 불규칙 : 돕- -> 도우, 도오, 도와 (도우 + 아)

바. "우" 불규칙 : 푸- -> 프, 퍼 (푸+어),
두- -> 두, 둬 (두+어)

사. "여" 불규칙 : (부지런)하- -> (부지런)해 (부지런+하+어)

아. "러" 불규칙 : 이르- -> 이르르, 이르러 (이르+어)

자. "르" 불규칙 : 흐르- -> 흐르르, 흘러 (흐르+어),
가르- -> 가르르, 갈라 (가르+아)

2) 형용사 어간의 활용형 생성

가. "으" 불규칙 : 슬프- -> 슬프, 슬퍼 (슬프+어),
가냘프- -> 가냘프, 가냘퍼 (가냘프 + 어)

나. "ㄹ" 불규칙 : 거칠- -> 거치-,
모질- -> 모지-

다. "ㅂ" 불규칙 : 즐겁- -> 즐거우, 즐거워 (즐겁+어),
곱- -> 고우, 고오, 고와 (곱+아)

라. "ㅎ" 불규칙 : 파랗- -> 파라, 파래 (파랗+아)

마. "여" 불규칙 : (훌륭)하- -> (훌륭)해 (훌륭+하+아)

바. "러" 불규칙 : 푸르- -> 푸르르, 푸르러 (푸르+어)

사. "르" 불규칙 : 게으르- -> 게으르, 게을러 (게으르+어),
빠르- -> 빨르, 빨라 (빨르 + 아)

기본 사전에 비해서 확장 사전에는 용언의 활용형과 음운 축약형 등이 자동으로 등록되므로 이들간의 관계를 명시할 필요가 있다. 기본 사전의 작성 규칙은 다음과 같다.

1) 불규칙 활용 표제어가 아닌 사전 정보
   사전 표제어 + { 형태소 정보 } + { 문법정보 }
2) 불규칙 활용 표제어 사전 정보
   사전 표제어 + { 형태소 정보/활용정보 포함 } + { 문법정보 }
3) 예외 경우의 표제어 사전 정보
   사전 표제어 + { 형태소 정보 } + 문법정보

1)과 3)의 경우는 활용형 사전 표제어를 생성하지 않지만, 2)의 경우는 형태소 정보가 용언 어간과 형용사 어간에 대하여 활용형 표제어를 생성하고 형태소 정보도 새로 작성하게 된다. 3)의 경우는 불구동사의 활용형 등 규칙화하기 까다로운 것들을 대상으로 하는 것이다. 확장사전의 구조는 사전 표제어 인덱스 부분과 사전 쉬트 정보 부분으로 나뉘게 된다. 사전 인덱스 부분에는 사전 표제어에 대한 형태소 문법 범주들과 이에 해당하는 사전 쉬트의 포인터가, 사전 쉬트 정보 부분에는 실제 정보가 저장된다.

4장 한글 코드 체계와 음운 축약 현상 처리

확장 사전 환경에서 용언 어간의 활용형과 음운 축약형을 모두 생성하고, 이에 대한 형태소 문법을 작성하는 일은 가능하다. 그러나 음운 축약형에 있어서는 용언 어간 중에서 무종성형인 대부분의 용언 어간에 대하여 가능한 경우를 모두 사전 표제어로 하여야 하는 부담이 있다. 이러한 경우에는 음운 축약을 따로 처리하는 사전 검색 모듈을 고안할 수 있을 것이다. 한국어 형태소 처리에서는 이러한 특징을 prefix-closed 특성에 첨가하여야 한다. 음운 축약은 단모음의 용언 어간에 대하여 "ㅇ"이 초성이고, 단모음인 어미가 결합할때 어미의 초성 "ㅇ"은 탈락하고, 두개의 단모음이 합쳐져 복모음으로 합성되는 현상이다.

한국어 형태소 처리에 적합한 한글 코드 체계로 〔이성진 92〕에서 제안한 코드 체계를 사용하였다. 〔이성진 92〕의 코드는 다음과 같은 원칙을 지니고 있다.

1. 초성의 자음 중 "ㅇ"은 음가가 없으므로 표기에서 제외한다.

2. 초성과 종성의 자음을 다른 코드값으로 정의한다.

3. 복모음은 단모음의 합으로 표기한다.

위의 원칙은 모두 용언의 활용과 특히, 음운 축약의 처리에 주안점을 두어 정의한 것이다. 〔이성진 92〕의 코드체계는 다음과 같다.

단모음 :	아	어	오	우	애	에	으	이						
	a	e	o	u	8	9	_	i						
복모음 :	야	여	요	유	얘	예	워	웨	위	와	왜	외	의	
	ya	ye	yo	yu	y8	y9	we	w9	wu	wa	w8	wi	yi	
단자음 :	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ
초성 :	g	n	d	l	m	b	s		j	c	k	t	p	h
종성 :	G	N	D	L	M	B	S	*	J	C	K	T	P	H
복자음 :	ㄲ	ㅃ	ㅆ	ㅉ										
초성 :	q	f	r	v	z									
종성 :	Q			V										

확장 사전 환경을 구축하여 형태소 처리를 하는데에 있어서 위의 코드 체계는 음운 축약형의 처리가 비교적 간단하게 적용된다. 음운 축약을 분리하거나 생성해 내는 시점은 형태소 처리 알고리즘과 사전 검색 도구에서 각각 가능하다. 사전의 검색 횟수를 줄이기 위하여는 형태소 처리 알고리즘보다는 사전 검색 도구에서 복모음의 경우 추가적으로 음운 축약의 역처리를 거쳐 단모음으로 나누어 사전 검색을 하는 방법이 효율적일 것이다. 위의 〔이성진, 92〕 한글 코드는 특별히 TRIE구조를 사용한 사전 Index에 대하여 간단한 처리만을 요구한다. 다음은 "아름다와서"라는 어절에 대한 형태소 해석과 생성의 처리 예를 보인다.

사전 표제어 :	아름답	: al_MdaB	(가)
	아름다와	: al_Mdawa	(나)
	아름다오	: al_Mdaw	(다)
	아름다우	: al_Mdau	(라)
	아서	: ase	(마)
	서	: se	(바)
아름다와서 :	al_Mdawase		

1) 확장사전에 음운 축약형을 저장하는 경우

사전에 (가, 나, 라, 마, 바)가 모두 표제어로 기입된다.

이때의 해석 결과는 다음과 같다.

al_Mdawa + se = 아름다와 + 서 = 아름답다 + 아서

이때 생성의 과정은 다음과 같다.

아름답다 + 아서 = { al_MdaB, al_Mdawa, al_Mdau} + { ase, se }

= al_Mdawa + se = al_Mdawase

= 아름다와서

2) 확장사전에 음운축약형을 저장하지 않는 경우

사전에 (가, 다, 라, 마)가 표제어로 기입된다.

이때는 (다, 라)의 표제어에 대하여 음운 축약화 처리를 한다.

al_Mdaw : 음운축약에 대비한 코드 변형

al_Mdaw + ase = 아름다오 + 아서 = 아름답다 + 아서

이때 생성의 과정은 다음과 같다.

아름답다 + 아서 = {al_MdaB,al_Mdaw,al_Mdau} + {ase,ese}

= al_Mdaw + ase = al_Mdaw + ase = 아름다와서

사전 표제어에 대하여 사전작성도구가 확장 하여야 할 대상이 되는 것은 용언 어간의 활용형과 축약형의 경우이다. 용언의 어간 중에서도 동작성 명사의 용언 어간화에 의한 것을 제외하면, 순수 우리말 용언 어간의 경우인데, 이는 약 9000어휘 정도가 있다. 이들에 대하여 모두 축약형을 사전 표제어로 생성한다면, 그 양이 만만치 않을 것이다. 이러한 경우에 대하여 위의 한글 코드체계는 음운 축약 현상에 대처하는 단순한 처리 방법을 제공한다. 물론 다른 한글 코드를 사용하더라도 음운 축약 현상에 대처하는 것이 어렵지는 않을 것이다. 그러나 위의 한글 코드 체계는 음운 축약과 활용형의 처리에 대단히 민감하게 작성된 것이므로 참고할 필요가 있다고 본다.

4장 어절구조 모델과 형태소 문법

어절의 구조를 결정하는 것과, 형태소 문법을 기술하는 것, 사전의 표제어 원칙을 정하는 것은 모두 밀접한 관계를 가지고 있다. 확장 표제어를 사용하는 환경은 특별히 형태소 문법을 기술하는 것에 영향을 많이 미친다. 다음에 기술하는 좌우 접속 정보를 이용한 형태소 문법 기술 방법은 여러가지로 편리한 접근방법이다.

어절에 대한 구조 모델을 형태소 범주들 간의 어절 네트워크를 이용하여 표현하는 방법의 한계는 너무 많은 세분류가 발생함으로 네트워크의 구조가 복잡해 진다

는 것이다. 즉, 세분류들 간의 접속성을 표현한다는 것은 너무 힘든 일이다. 그러므로 형태소의 세분류를 좌접속과 우접속의 특성으로 나누어 형태소 범주를 정의하는 방법을 고려하였다. 이것이 바로 좌우 접속 모델이다[조영환, 91].

파생법과 합성법에 의한 단어의 형성을 표현하기 위한 방법으로 형태소의 좌우 접속정보를 이용할 수 있다. 이 방법의 기본 개념은 형태소를 동일한 분류 기준이 아닌 두 개의 독립적인 분류 체계에 따라 나누고, 두 분류 체계 사이의 접속성을 조사하여 이를 형태소의 접속 모델로 하자는 것이다. 형태소의 분류는 무엇의 좌측에 붙이어지는 경우에 따라 분류(좌접속정보)가 가능하고, 또 무엇의 우측에 붙이어지는 경우에 따라 분류(우접속정보)가 가능하다. 이러한 두가지 분류의 기준으로 실제 형태소를 세분류하고 좌접속정보와 우접속정보간의 접속성을 표현하는 것이 좌우접속정보표이다. 한국어 어절 구조를 Graph의 형태로 표현하여 보자. 보통의 Graph는 G = { P, E }로 P는 노드의 집합이고, E는 노드간의 연결 아크의 집합을 나타낸다. 그러나 노드를 좌측과 우측의 접속성에 따라 분류하는 방법인 좌우접속모델은 노드의 집합이 두 개의 집합으로 표현된다.

한국어 어절의 구조, 즉, 한국어 형태소 해석 문법을 나타내는 Graph를 KMG라 하면, 다음과 같은 형태의 Graph를 정의할 수 있다. 여기에서 Pl과 Pr은 각각 좌접속 분류, 우접속 분류라 한다. 연결 아크의 집합 E는 좌접속 노드와 우접속 노드의 Pair로 구성된다. 아크는 "좌접속 노드가 우접속 노드와 접속 가능함"을 의미한다.

KMG = (Pl, Pr, E)

Pl = { L1, L2, L3, L4, L5, L6, L7, L8, L9, L10 , . . . }

Pr = { R1, R2, R3, R4, R5, R6, R7, R8, R9, R10 , . . . }

E = { <L1, R11>, <L1, R13>, <L1, R325>, <L2, R12>, <L2, R14>, . . . }

이러한 구조는 실제로 노드, 실질적으로는 형태소 혹은 사전 표제어의 문법 범주, 의 크기를 줄이고, 아크, 실질적으로는 각 범주 사이의 결합성을 표현하는 문법, 의 수를 줄이는데 도움을 준다. 또한 어느 형태소 구조 - 어절이 되기전이나 혹은 여러 형태소가 모인 경우 - 라도, 그 구조를 단지 좌측접속정보와 우측접속정보만으로 어절의 구성 상태를 판단할 수 있다.

형태소 해석과 생성 모두의 경우에 대하여 위의 KMG를 이용하는 것이 가능하다. 형태소 해석의 경우에는 입력 어절에 대하여 사전 표제어 단위로 나누어진 상황에서, 문법적인 형태소 구조로서의 확인을 위하여 위의 문법이 적용된다. 문법의

적용이란, 두 사전 표제어 M_i와 M_{i+1}에 대하여 L_i 즉, M_i의 우접속 정보와 R_{i+1} 즉, M_{i+1}의 좌접속 정보의 쌍 $\langle L_i, R_{i+1} \rangle$가 좌우접속정보표 즉, 형태소 문법의 규칙으로 존재하는 지를 검증하는 것이다. 반면에 형태소 생성의 경우에는 형태소의 가능한 형태들에 대하여 어절을 이루는 것을 적당히 결정하기 위하여 위의 문법이 적용된다. 이 경우에는 형태소 각각이 동일한 의미적, 문법적 기능을 하면서 이형태 관계인 것들에 대하여 한가지를 선택하게 된다.

형태소 해석에 대한 규칙의 기술에 있어서 한글 맞춤법 표준안이나, 고교 문법에 맞추어 충실하게 해석을 하기 위하여 형태소 범주의 수를 너무 세분화 하고, 문법 규칙을 세밀하게 되면 입력어절에 대하여 처리가 불가능한 경우가 많다. 이러한 것은 우리가 일상적으로 사용하고 있는 언어 체계가 정형화된 규칙으로 나열되기 힘들기 때문인 듯 싶다. 그러므로 형태소 해석을 위한 범주 나눔과 규칙의 수를 적절한 수준으로 줄이고, 대신 출현 빈도 등을 고려한 순위메김 방법을 고려하는 것이 좋을 것 같다. 반면에 형태소 생성의 경우에는 규칙이 허술하게 되면 생성 결과에 일상적으로 받아들여지지 않는 어절이 생성되거나 의미가 곡해되는 경우가 있다. 그러므로 한국어 생성을 위한 형태소 규칙은 해석의 그것보다 훨씬 세밀하게 작성되어야 한다.

형태소 처리 알고리즘의 복잡성은 어절을 처리하는데 필요로하는 처리의 양을 수식화 함으로써 파악할 수 있다. 이러한 방법은 주로 용언의 불규칙 현상과 음운 축약 현상에 대하여, 그리고 어절을 이루는 가능한 모든 형태소 해석 구조를 발견하는데 필요한 처리의 양을 계산하는 것이다. 실질적인 입장에서 형태소 처리기의 효율성은 형태소 처리 알고리즘의 복잡성보다는 형태소 처리 알고리즘이 어절을 분석 혹은 합성하기 위하여 사전을 몇번이나 접근하는가 하는 것이 주요 평가기준이라고 보여진다. 형태소 처리기는 CPU burst하다기보다는 I/O burst 성격이 강하기 때문이다.

본 논문에서 제안하는 확장 사전 환경은 사전의 접근 횟수를 최소화 시켜준다고 할 수 있다. 형태소 처리 알고리즘이 사전의 접근 횟수를 줄여주기 위하여 확장사전 환경에서는 다음과 같은 접근 방법을 제공한다.

1) 입력어절의 부분 문자열에 의한 사전 표제어 검색 : 가능한 모든 활용형태를 사전 표제어로 색인하기 때문에, 어절의 일부분을 가능한 모든 원형으로 변형한 후에 사전을 검색하는 다중 검색을 일으키지 않는다.

2) 단일 문자열에 여러 사전의미 포함 : 기본적으로 모든 형태소 범주에 대하여 단일 사전 색인 구조를 제공한다. 범주별 사전이 나누어지는 경우 단일 문자열에 대하여 가능한 사전을 모두 검색하여야 한다.

3) TRIE구조를 이용한 Prefix closed 검색 제공 : Prefix closed 특성은

문자열 "$a_1a_2a_3...a_n$"에 대하여 "$a_1a_2a_3..a_i$"가 사전 표제어로 색인되어 있는 경우, 이들 모두를 한번의 사전 검색으로 모두 찾는 것을 의미한다. 이러한 특성을 갖는 데이타 구조가 TRIE구조이다. 확장 사전 환경을 위한 사전작성도구는 사전의 색인구조로 TRIE구조를 적용한다.

5장 결론

본 논문에서는 형태소 해석과 생성을 위하여 공학적인 접근 방법의 하나로 확장 사전 환경을 제안하였다. 이는 형태소 해석의 목적을 어절을 구성하는 형태소의 분석에 두기보다는, 사전에 등록된 정보의 적재를 위하여 사전 표제어를 찾아내는 일이라고 보는 관점에서 시작된다. 그러므로 사전 표제어를 명사, 용언 어간의 기본형, 기능어 등으로 두기보다는, 좀더 유연하게 하자는 것이다.

사전 표제어를 용언 어간의 원형으로부터 자동으로 확장이 가능한 용언의 활용형 정도로 하는 것을 보이고, 음운 축약형에 대해서는 사전 표제어로 하는 방법과 효율적인 한글 코드를 사용하는 방법에 대하여 선택적으로 적용할 수 있음을 제안하였다. 형태소 처리의 목적이 어절단위의 처리가 갖는 불합리성을 제거하기 위한 것이므로, 기본형을 표제어로 하여 기본 사전의 관리를 용이하게 하고 이를 자동으로 가공, 확장하여 활용형 표제어를 포함하는 확장 사전 환경을 제안하였다.

형태소 처리를 형태소 해석과 생성 알고리즘에 의한 엔진부의 처리 부담을 줄이고, 대신 사전 작성과 사전 검색시에 대부분의 처리를 하는 것을 기본으로 한다. 이것은 형태소 처리의 효율성에 대한 중심 척도가 사전 검색 횟수에 의해 영향을 받는다는 것을 기본 가정으로 하기 때문이다.

용언 어간의 불규칙 활용형에 대하여 사전 표제어로 하는 것은 불규칙 용언의 수가 많지 않고, 이로 인한 사전 검색의 부담이 크기 때문이었다. 그러나 음운 축약형에 대하여는 사전 표제어로 확장하는데는 문제가 있다. 그래서 본 논문에서는 음운 축약에 효율적으로 대처할 수 있는 한글 코드를 소개한 것이다. 음운 축양형에 대하여도 Prefix closed 특성을 갖는 사전 검색 알고리즘의 개발은 사용하는 한글 코드의 종류에 따라 복잡성의 정도는 있지만, 그리 어려운일은 아니다.

사전 표제어를 활용형과 음운 축약형으로 자동 확장하는 환경에서 형태소 처리를 하는 경우에 과해석과 과처리의 경우가 발생할 수 있다. 이것은 동일한 형태소 해석 혹은 생성 결과가 형태소 문법의 미비함으로 인하여 중복되게, 다른 형태의 사전 표제어의 검색 결과로 생성되는 것을 의미한다. 즉, 올바른 형태소 해석을 위해서는 어떠한 방법에서도 마찬가지이지만 더욱 사전 표제어의 관리를 철저히 하여야 하고,

형태소 문법을 조심스럽게 작성하여야 한다.

규칙의 기술과 형태소 수준의 사전 표제어만으로는 기술이 불가능한 경우에 대하여 형태소 이상의 표제어를 등록하고, 미리 해석의 결과를 사전에 기입하거나, 해당하는 구문적 정보를 기입하도록 하는 방법을 본 논문에서 제안하였다. 그러나 이러한 경우에 적용할 수 있는 형태소 해석 엔진에 대하여는 본 논문에서 제외하였다. 예외적인 현상의 처리는 실질적인 형태소 처리기에서 중요하게 다루어야 할 것으로 보인다.

〔참고논문〕

강승식, 김영택, "한국어 형태소 분석기에서 불규칙 용언의 분석 모형", 정보과학회 논문집, 1992

과기원, 한국어 처리 시스템 개발 환경의 연구, 과기처 보고서, 1987

과기원, 한국어 철자 및 띄어쓰기 교정 시스템에 관한 연구 II, 과기처 보고서, 1992

김성용, Tabluar parsing 방법과 접속 정보를 이용한 한국어 형태소 분석기, 한국과학기술원, 석사학위논문, 1987

이성진, Two-level 한국어 형태소 해석, 한국과학기술원, 석사학위논문, 1992

조영환, 서정연, "사전검색을 위한 한국어 형태소 분석", 음양학회 학술대회, 1991