인접 조건 검사에 의한 초고속 한국어 형태소 분석
(High Speed Korean Morphological Analysis based on Adjacency Condition Check)

심 광 섭 † 양 재 형 ††
(Kwangseob Shim) (Jaehyung Yang)

요 약 본 논문에서는 코드 변환 과정과 축약, 탈락, 불규칙 활용 등으로 변형된 형태소의 원형을 복원하고 분석 후보를 생성하는 등의 과정을 거치지 않고 형태소 사전에서 제공되는 인접 조건에 대한 검사만으로 형태소 분석을 하는 방법을 제안한다. 인접 조건 검사는 복잡한 연산을 하지 않고 단순한 비트 연산만으로 할 수 있기 때문에 제안된 방법은 초고속 형태소 분석기 구현에 적합하다. 본 논문에서 제안한 방법에 따라 구현된 한국어 형태소 분석기 MACH는 1.13 GHz Pentium III 개인용 컴퓨터에서 대략 5분/GB의 분석 속도를 보였으며, 분석 정확도는 99.2 %로 기존의 다른 분석기와 큰 차이가 없었다.
키워드 : 인접, 조건, 비트, 연산, 초고속, 한국어, 형태소, 분석, MACH

Abstract This paper proposes a morphological analysis method that enables morphological analysis by checking conditions between two adjacent morphemes. These conditions are fed from a dictionary. This method eliminates a code conversion module and the application of transformational rules for candidate generation. The method claims that very high speed morphological analysis is attainable through simple bit operations for adjacency condition check. MACH, an implementation of the proposed method, is a supersonic Korean morphological analyzer which is able to analyze a document of 1 GB in 5 minutes on a PC with 1.13 GHz Pentium III CPU. The analysis accuracy of MACH is 99.2 %.
Key words : Adjacency Condition, Bit Operation, High-speed, Korean Morphological Analysis, MACH

1. 서 론

과거에는 한국어 형태소 분석기가 주로 구문 분석기의 전단계 정도로 생각되었으며, 형태소 분석기는 구문 분석기에 비하여 상대적으로 처리 속도가 빠르기 때문에 형태소 분석기의 성능을 평가할 때 속도보다는 정확도가 중요한 것으로 인식되었다. 그런데 인터넷 시대에 접어들어 문서의 양이 급격하게 증가하면서 이들 문서를 컴퓨터로 처리해야 할 필요성도 증대되었다. 이에 따라 문서 처리의 가장 기본이 되는 형태소 분석을 얼마나 정확하게 할 수 있는가 하는 것뿐만 아니라 얼마나 빠르게 할 수 있는가 하는 것도 중요한 것으로 인식되기 시작하였으며, 최근 들어 고속 또는 초고속 한국어 형태소 개발을 위한 노력이 전개되고 있다[1-3].

고속 또는 초고속 한국어 형태소 분석기 개발을 위한 노력은 크게 3 가지 방향으로 전개되고 있다. 첫째, 형태소 분석 알고리즘을 개선하여 사전 탐색 회수를 줄이거나 분석 후보의 개수를 줄이는 것이다. 둘째, 사전 구조를 개선하여 사전 탐색 시간을 줄이는 것이다. 셋째, 고빈도 어절에 대한 기분석 사전을 구축하여 고빈도 어절에 대해서는 절차적인 분석 없이 사전 탐색만으로 분석을 완료하는 것이다. 형태소 분석 알고리즘을 개선하여 사전 탐색 회수 및 분석 후보를 줄이는 노력은 많은 진전을 보였다. 사전 탐색 회수 및 분석 후보를 줄이기 위하여, [4]는 음절 정보를 이용하는 방법을 제시하였으며, [5]는 음절 정보를 포함한 배제 정보를 이용하는 방법을 제시하였다. [6]은 양방향 최장 일치법에서 사전 탐색 회수를 줄이는 방법을 제시하였다. 사전 구조를 개선하여 사전 탐색 시간을 줄여 형태소 분석기의 속도를

- http://cs.sungshin.ac.kr/~shim/demo를 방문하면 MACH에 대한 추가 정보를 얻을 수 있다.

- 이 논문은 2003년도 성신여자대학교 학술연구조성비 지원에 의하여 연구되었음.

† 종신회원 : 성신여자대학교 컴퓨터정보학부 교수
shim@sungshin.ac.kr
†† 종신회원 : 강남대학교 컴퓨터미디어공학부 교수
jhyang@kns.kangnam.ac.kr
논문접수 : 2003년 2월 26일
심사완료 : 2003년 9월 18일

개선하려는 노력도 많은 진전을 보였다. [1]은 기존의 2 음절 트라이(trie)보다 약 10 % 가량 성능이 개선된 변형된 2 음절 트라이를 소개하였다. [7]은 한국어 형태소 분석에 활용할 수 있는 FST(Finite State Transducer)를 이용한 한국어 전자 사전 구조를 제안하였다. 고빈도 어절에 대한 기분석 사전을 이용할 경우 고빈도 어절에 대해서는 절차적인 분석 없이 한 번의 사전 탐색만으로 분석이 완료되므로 기분석 사전에 대한 적중률만 높다면 분석 속도를 획기적으로 개선할 수 있다. 그런데 우리말로 표현될 수 있는 어절을 모두 다 수집한다는 것은 사실상 불가능하므로 실제로는 적정 크기의 기분석 사전만으로는 분석할 수 없는 어절들도 상당히 많다[1]. 이러한 문제를 완화하기 위하여 [1]은 기본 사전과 기분석 사전을 적절하게 활용하는 하이브리드 (hybrid) 방법을 제시하였다.

전체 어절이 아닌 부분 어절에 대하여 기분석 사전 정보를 이용하는 방법이 제시된 사례도 있다. 이들 방법은 대체로 음운 변동이 심해 절차적인 방법으로 처리하기가 복잡하지만 그 수가 적어서 사전 구축이 용이한 경우에 한하여 제한적으로 활용되는 것이 대부분이었다 [8]. 이에 반하여 [2]는 부분 어절에 대한 기분석 사전을 적극적으로 활용하는 방법을 제시하였다. 특히 [2]는 부분 어절에 대한 기분석 사전을 이용하는데 그치지 않고 지금까지 한국어 형태소 분석에서 필수 불가결한 요소라고 여겨왔던 코드 변환, 원형 복원 등과 같은 과정을 과감하게 제거함으로써 고속 한국어 형태소 분석기 개발을 위한 기틀을 마련했다는 점에 주목할 만하다.

2. 인접 조건 검사에 의한 형태소 분석

형태소 분석이란 주어진 어절에서 형태소를 분리한 다음 각 형태소에 범주를 부여하는 것으로 정의할 수 있다. 이것은 계산학적으로 그다지 복잡하지 않다. 하지만 한국어의 경우 불규칙 활용, 축약, 탈락 등의 음운 현상과 음절 단위의 모아쓰기 및 이로 인한 다중 바이트 코딩으로 인해 그 과정이 타 언어 특히 구미어에 비하여 복잡한 것으로 인식되어 왔으며 그 동안 여러 가지 해결 방안이 제시되었다. 여기서는 인접 조건 검사에 의한 한국어 형태소 분석의 기본 개념에 대하여 설명하고자 한다.

2.1 음절을 경계로 한 분석

인접 조건 검사에 의한 한국어 형태소 분석에 대하여 설명하기 전에 코드 변환 및 자소 단위의 연산을 전혀 하지 않고 음절을 경계로 형태소를 분리하는 방법에 대해서 알아보겠다. 설명의 편의를 위하여 일단은 복합 명사, 접두사나 접미사에 의한 파생, 본용언과 보조용언의 결합 등과 같이 다소 복잡한 경우는 고려하지 않기로 한다. 이러한 형태를 포함하여 띄어쓰기가 잘못된 어절에 대한 분석에 대해서는 뒤에서 검토할 것이다.

한국어 어절은 크게 세 가지 방법으로 구성된다. 첫째, 단어 자체가 어절을 이루는 경우이다. 명사, 대명사, 수사 등과 같은 체언, 관형사나 부사 등과 같은 수식언, 감탄사 등과 같은 독립언 등이 이러한 경우에 해당한다[2]. 이런 경우에는 전체 어절에 대하여 사전을 한 번 탐색하는 것으로 분석이 완료된다. 둘째, 아래의 1과 같이 체언과 조사가 결합하여 어절을 이루는 경우이다.

1. (가) 학교에서 → 학교/NN + 에서/JO
   (나) 학생은 → 학생/NN + 은/JO

이 경우 형태소 분석은 음절을 경계로 체언과 조사를 인식하는 문제로 볼 수 있는데, 이상적인 경우 두 번의 사전 탐색으로 형태소 분석이 종료된다. 셋째, 다음과 같이 용언에 어미가 결합하여 어절을 이루는 경우다.

2. (가) 예쁘니까 → 예쁘/AJ + 니까/EM
   (나) 기다리다 → 기다리/VV + 다/EM
   (다) 그립다고 → 그립/AJ + 다고/EM
   (라) 믿다 → 믿/VV + 다/EM

이 경우에도 1에서와 같이 음절을 경계로 형태소 분리를 할 수 있으며 이상적인 경우 두 번의 사전 탐색으로 형태소 분석을 마칠 수 있다.

지금까지 음절을 경계로 형태소를 분리하는 방법에 대하여 살펴보았다. 이 방법에서는 음절을 경계로 형태소 분리를 하므로 자소 단위의 연산은 불필요하며, 그 결과 코드 변환 단계가 필요 없게 된다. 또한 형태소 분석 문제를 사전 탐색에 의한 단어 인식 문제로 단순화시킬 수 있다. 하지만 한국어에서는 이렇게 단순한 방법으로 형태소 분석을 할 수 없는 경우도 상당히 많다. 아래에서는 음절을 경계로 형태소 분석을 할 수 없는 것처럼 보이는 사례들에 대하여 검토하고, 이러한 경우에도 음절을 경계로 형태소 분석을 하기 위해서 어떻게 해야 하는지에 대하여 논의할 것이다.

2.2 음운 제약 조건

아무런 조건 없이 음절을 경계로 형태소 분리를 할 경우 잘못된 분석을 할 가능성이 있다. 이러한 문제는 한국어의 음운적 특성을 고려하지 않았을 때 발생한다. 다음은 이러한 이유로 인해 잘못 분석된 예이다.

1) 기분석 사전이 지나치게 커져서 주기억장치에 저장하지 못하고 보조기억장치에 저장해야 한다면 기분석 사전을 사용하는 효과가 감소하기 때문에 "적정 크기의 기분석 사전"이라는 표현을 사용했다.

2) 본 논문에서는 중분류 수준의 품사 체계를 가정하고 있으며, 각 품사는 NN(명사), NP(대명사), NU(수사), NX(의존명사), DT(관형사), AD(부사), IJ(감탄사), SY(부호), VV(동사), VX(보조동사), SV(동사화접미사), AJ(형용사), AX(보조형용사), SJ(형용사화접미사), CP(계사), JO(조사), EP(선어말어미), EM(어말어미), SN(접미사), PF(접두사) 등과 같은 기호로 표시한다.

3. (가) 먹는 → 먹/NN + 는/JO
   (나) 감는 → 감/NN + 는/JO

위의 예에서 '-는'은 종성 자음을 가지지 않는 체언과 사용되는 조사이므로 종성 자음을 가지는 '먹'이나 '감'과는 결합하지 않도록 해야 한다. 이와 같이 조사 중에는 더불어 사용되는 체언의 음운론적 환경에 제약을 가하는 경우가 있으므로, 분석시 이 점을 고려해 주어야 한다. 음운론적 환경 제약은 조사뿐만 아니라 어미에서도 발견된다. 그런데, 음운론적 환경 제약의 분포는 조사나 어미에 따라 약간의 차이가 있는데, 조사는 체언의 마지막 음절이 종성 자음을 가지는지의 여부를 제약하는 반면, 어미는 용언의 마지막 음절이 종성 자음을 가지는지, 이것이 양성 모음인지 음성 모음인지의 여부를 제약한다. 이러한 제약 조건은 형태소 사전에 반영되어야 하는데, 다음은 이러한 관점에서 만들어진 형태소 사전의 한 예이다. 여기서 […]은 표제어의 형태소 분석 결과를 표시하는 부분이며[3], @(…)은 표제어의 왼쪽에 올 수 있는 단어에 대한 음운론적 환경 제약을 기술하는 부분이다. 이제부터는 음운론적 환경 제약을 음운 제약 조건이라 부르기로 한다.

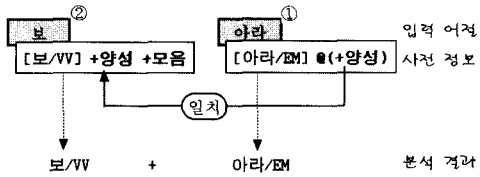
4. 에서 : ([에서/JO])
   는 : ([는/JO] @(+모음))
   은 : ([은/JO] @(+자음))
   아라 : ([아라/EM] @(+양성))
   어라 : ([어라/EM] @(+음성))

위에서 '-에서'는 아무런 음운 제약 조건도 명시되어 있지 않다. 따라서 이 단어의 왼쪽에는 임의의 단어가 아무런 제약 없이 올 수 있다[4]. '-는'은 +모음이라는 음운 제약 조건이 명시되어 있는데, 이것은 이 단어의 왼쪽에는 모음으로 끝나는 단어만 올 수 있음을 의미한다. '-은'은 +자음이라는 음운 제약 조건이 명시되어 있으며, 이것은 이 단어의 왼쪽에는 자음으로 끝나는 단어만 올 수 있음을 의미한다. '-아라'는 +양성, '-어라'는 +음성이라는 음운 제약 조건이 명시되어 있으므로 이 단어의 왼쪽에는 각각 양성 모음, 음성 모음을 가지는 단어만이 올 수 있다. 음운 제약 조건을 적용하려면 형태소 사전의 각 단어는 마지막 음절이 자음으로 끝나는지 모음으로 끝나는지, 양성 모음인지 음성 모음인지를 나타내는 음운 정보를 가지고 있어야 한다. 다음은 이러한 관점에서 만들어진 형태소 사전의 한 예이다.

5. 학교 : ([학교/NN] +모음)
   마당 : ([마당/NN] +자음)
   먹 : ([먹/VV] +음성 +자음)
   보 : ([보/VV] +양성 +모음)

인접한 두 단어에서 오른쪽 단어에 의해 제기된 음운 제약 조건을 왼쪽 단어가 만족하는지 검사함으로써 3과 같은 분석 오류가 발생하는 것을 피할 수 있다. 다음은 '보아라'를 예로 들어 인접 조건 검사만으로 형태소 분석을 하는 과정을 보인 것이다. 여기서 ①, ②는 형태소 사전을 탐색하는 순서, 즉 분석 진행 방향을 나타낸다.

6.



6에서 입력 어절에 대하여 역방향으로 사전을 탐색하면 어미 '아라'가 발견된다. 이 어미는 +양성이란 음운 제약 조건을 가지고 있으므로 이 어미의 왼쪽에는 양성 모음을 가진 용언만 올 수 있다. 추가 사전 탐색 결과 '보'가 발견되며, 이 단어는 +양성이라는 음운 정보를 가지고 있으므로 '아라'에 의해 제기된 음운 제약 조건을 충족시켜 준다. 더 이상 분석할 문자열이 없으므로 사전에서 […] 부분에 명시된 분석 결과를 가지고 와서 이들을 결합하는 것으로 형태소 분석은 종결된다.

2.3 형태 제약 조건

어미 중에는 '-ㄴ다'와 같이 자소를 포함하는 경우도 많은데, 이 경우 해당 자소는 7에서 보듯이 선행하는 용언의 제일 마지막 음절과 결합하여 어간의 형태가 변하게 된다.

7. (가) 그린다 → 그리/VV + ㄴ다/EM
   (나) 돌본다 → 돌보/VV + ㄴ다/EM

기존 연구에서는 이런 경우 어미 '-ㄴ다'의 왼쪽에 용언의 어간이 나와야 하는 것으로 보고, 이를 찾기 위한 복잡한 원형 복원 과정을 거쳐야만 했다. 그런데 시각을 좀 달리하여 '-ㄴ다'에서 자소를 제외한 '-다'를 (부분) 어미로 간주하고 이 부분 어미의 왼쪽에 용언의 어간이 아니라 어간과 자소 '-ㄴ'이 결합된 형태가 나와야 하는 것으로 본다면, 7과 같은 경우에도 코드 변환 없이 음절을 경계로 한 형태소 분석이 가능하다. 이를 위해 형태소 사전의 개념을 확대할 필요가 있다. 전통적인 형태소 사전에서는 순수한 형태소가 표제어로 등재되었지만, 확대된 형태소 사전에서는 순수한 형태소 외에 부분 어미나 용언의 어간에 자소가 결합된 형태까지 표제어로 등재된다[5].

---

3) 이 예와 같이 단순 단어의 경우에는 형태소 분석 결과를 표시할 필요가 없이 품사 정보만 기록하면 되지만, 형태소 사전에 단순 단어 외에 복합어도 등재를 할 수 있기 때문에 편의상 형태소 분석 결과를 사전에 기술하는 것으로 하였다.

4) 조사와 어미의 왼쪽에는 각각 체언과 용언이 올 수 있는데, 여기서는 체언이나 용언이라는 말 대신 '단어'라는 표현을 썼다. 그 이유는 뒤에서 설명할 '품사 제약 조건' 부분에서 분명해 지겠지만, 그 전까지는 조사 왼쪽에는 체언이, 어미 왼쪽에는 용언이 오는 것으로 이해해도 무방하다.

확대된 형태소 사전에서 각 표제어는 자신의 왼쪽에 인접하는 단어에 대한 형태론적 환경 제약을 기술할 수 있다. 이러한 제약 조건을 **형태 제약 조건**이라고 한다. 다음은 형태 제약 조건이 명시된 확대된 형태소 사전의 한 예이다. 형태 제약 조건도 음운 제약 조건과 마찬가지로 @(…) 부분에 기술한다.

8. 개 : ([ㄹ게/EM] @(+ㄹ))
   다 : ([다/EM] @(+어간 ¦ +ㅆ)) ([ㄴ다/EM] @(+ㄴ))
   세 : ([ㅁ세/EM] @(+ㅁ))
   니다 : ([ㅂ니다/EM] @(+ㅂ))

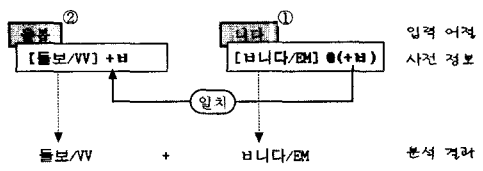
여기서 +어간, +ㄴ, +ㄹ, +ㅁ, +ㅂ, +ㅆ 등은 표제어의 왼쪽에 올 수 있는 단어의 형태를 제약하는데 사용되는 자질로, +어간은 표제어의 왼쪽에 어간만 올 수 있음을 나타내며, 나머지 자질들은 표제어의 왼쪽에 해당 자소와 결합한 형태만 올 수 있음을 나타낸다. 위에서 ¦는 OR 조건을 나타내는 기호이며, 이 기호가 사용되지 않은 경우에는 AND 조건을 나타내는 것으로 본다.

전술한 바와 같이 확대된 형태소 사전에서는 어간은 물론 ㄴ, ㄹ, ㅁ, ㅂ, ㅆ 등의 자소와 결합한 형태까지 표제어로 등재된다. 따라서 확대된 형태소 사전에 등재된 표제어는 그것이 어간인지 아니면 어떤 자소와 결합한 형태인지를 표시하는 **형태 정보**를 가지고 있어야 한다. 형태 정보는 형태 제약 조건을 표시하는데 사용된 것과 같은 자질로 나타낸다. 다음은 '**돌보다**'를 예로 들어 어간 및 어간과 자소가 결합된 형태가 확대된 형태소 사전의 표제어로 등재된 모습을 보여 준다.

9. 돌보 : ([돌보/VV] +어간 +양성 +모음)
   돌본 : ([돌보/VV] +ㄴ)
   돌볼 : ([돌보/VV] +ㄹ)
   돌봄 : ([돌보/VV] +ㅁ)
   돌봅 : ([돌보/VV] +ㅂ)
   돌봤 : ([돌보/VV + 았/EP] +ㅆ)

확대된 형태소 사전을 이용하면 '**돌보다**', '**돌본다**', '**돌봤다**', '**돌볼게**', '**돌봅세**', '**돌봅니다**' 등과 같이 어간의 형태가 변형된 어절에 대해서도 코드 변환이나 복잡한 원형 복원 규칙 적용 없이 인접 조건 검사만으로 음절을 경계로 한 형태소 분석을 할 수 있다. 다음은 '**돌봅니다**'를 예로 들어 인접 조건 검사만으로 형태소 분석을 하는 과정을 보인 것이다.

10.



5) 현대 한국어에서 어미 중에 포함되어 용언 어간의 마지막 음절과 결합할 수 있는 자소는 ㄴ, ㄹ, ㅁ, ㅂ, ㅆ 뿐이다.

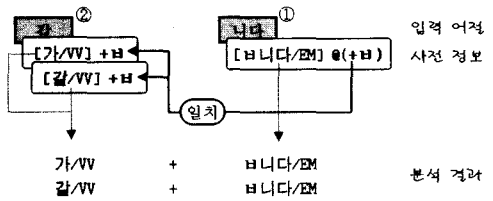
10에서 보듯이 주어진 어절에 대하여 역방향으로 사전을 탐색하면 제일 먼저 부분 어미 '니다'가 발견된다. 이 부분 어미는 +ㅂ이란 형태 제약 조건을 가지고 있으므로 어간에 자소 ㅂ이 결합한 형태만이 이 부분 어미의 왼쪽에 올 수 있다. 추가 사전 탐색 결과 '돌봅'이 발견되며 이 단어는 어간에 ㅂ이 결합되었음을 나타내는 형태 정보를 가지고 있으므로 '니다'에 의해 주어진 형태 제약 조건을 충족시킬 수 있다. '돌봅'의 왼쪽에 더 이상의 스트링이 없으므로, 사전 정보 중 […] 부분에 있는 분석 결과를 가지고 와서 이들을 결합하는 것으로 형태소 분석은 종결된다.

이번에는 '갑니다'를 예로 들어 한 어절에 대하여 여러 개의 형태소 분석 결과가 생성되는 경우에 대해서 살펴보겠다. '갑'은 용언 어간 '가-'에 자소 ㅂ이 결합한 것으로 볼 수도 있으며 용언 어간 '갈-'에 자소 ㅂ이 결합한 것으로 볼 수도 있다. 따라서 확대된 형태소 사전은 다음과 같은 내용을 포함하고 있을 것이다.

11. 가 : ([가/VV] +어간 +양성 +모음)
    갈 : ([갈/VV] +어간 +양성 +자음)
    갑 : ([가/VV] +ㅂ) ([갈/VV] +ㅂ)

입력 어절에 대하여 역방향으로 사전 탐색을 하면 부분 어미인 '니다'가 발견된다. 계속해서 사전 탐색을 하면 '갑'이 발견되는데, 이 단어는 두 개의 사전 정보를 가지고 있다. 그런데 둘 다 '니다'의 인접 조건을 만족하므로 12와 같이 두 가지 형태소 분석 결과가 생성된다.

12.



다음은 '-아'나 '-어'와 같은 매개 모음 문제에 대하여 살펴보겠다. 이들 매개 모음도 자소와 마찬가지로 용언 어간과 결합할 수 있다. 예를 들어 어간 '보-'에 매개 모음 '-아'가 결합하면 '봐'가 되며, 어간 '되-'에 매개 모음 '-어'가 결합하면 '돼'가 된다. 확대된 형태소 사전에 용언 어간뿐만 아니라 매개 모음과 결합된 형태도 등재한다면 매개 모음 문제 역시 인접 조건 검사를 통해 해결할 수 있다. 다음은 매개 모음과 결합한 형태를 확대된 형태소 사전에 등재한 예이다. 여기서 +아는 '-아'나 '-어'와 같은 매개 모음으로 시작하는 어미와 결합할 수 있음을 나타내는 자질이다. 반대로 -아는 그러한 어미와 결합할 수 없음을 나타내는 자질이다.

13. 보 : ([보/VV] +어간 +아 +양성 +모음)
    봐 : ([보/VV] -아 +양성 +모음)
    사 : ([사/VV] +어간 -아 +양성 +모음)

먹 : ([먹/VV] +어간 +아 +음성 +자음)

다음 예는 매개 모음 처리를 위해 확대된 형태소 사전에서 어미가 어떻게 정의되어 있는가를 보여준다. 이 예에서 보듯이 '-아도'나 '-어도'와 같이 매개 모음을 가지는 어미는 +아라는 자질을 가지지만, '-도'와 같이 매개 모음을 내포하지 않는 어미는 -아라는 자질을 가진다. '-고'는 '-아'나 '-어' 매개 모음을 가진 이형태가 없는 어미이므로 +아나 -아 자질을 가지지 않는다.

14. 아도 : ([아도/EM] @(+아 +양성))
어도 : ([어도/EM] @(+아 +음성))
도 : ([아도/EM] @(-아 +양성))
([어도/EM] @(-아 +음성))
고 : ([고/EM] @(+어간))

위의 13, 14와 같이 형태소 사전을 구성하면 인접 조건 검사에 의해 '보아도', '봐도', '사도', '먹어도', '보고', '사고', '먹고' 등은 올바르게 분석하지만 '봐아도', '사아도', '봐고', '먹도' 등을 동사와 어미로 분석하지는 않는다.

용언 중에는 뒤에 오는 어미에 따라 어간 일부가 변하는 불규칙 용언이 있다. 예를 들어, '듣다', '긋다', '돕다' 등은 불규칙 용언인데 특정 어미 앞에서 이들 용언은 '들어서', '그어서', '도와서'와 같이 어간의 일부가 변하게 된다. 전통적인 형태소 분석 방법에서는 이런 경우 복잡한 원형 복원 절차를 거쳐야만 하였다. 그러나, 본 논문에서 제안한 방법에서는 불규칙 용언의 경우에도 별도의 원형 복원 절차를 거치지 않고 음절을 경계로 한 인접 조건 검사만으로 형태소 분석을 할 수 있다. 인접 조건 검사만으로 불규칙 용언을 분석하려면 용언 어간뿐만 아니라 이들의 변형된 형태도 확대된 형태소 사전의 표제어로 등재하여야 한다.

15는 불규칙 용언의 어간과 변형된 형태를 확대된 형태소 사전에 등재한 예이다. '들'이나 '그'는 '들어', '그어' 등과 같이 '-아'나 '-어'로 시작하는 어미 앞에서 쓰일 수 있으므로 +아라는 자질을 가진다. 그런데, '도와도', '미워도'에서 볼 수 있듯이 ㅂ 불규칙 활용을 하는 용언은 '-아도'나 '-어도'와 결합하지 않고 '-와도'나 '-워도'와 결합한다. 이와 같이 '-아'나 '-어' 대신 '-와'나 '-워'로 시작하는 어미 앞에서만 쓰이는 변형된 용언 어간에 대해서는 +아 대신 +와라는 자질로 구분하였다.

15. 듣 : ([듣/VV] +어간 +음성)
들 : ([듣/VV] +아 +음성)
돕 : ([돕/VV] +어간 +양성)
도 : ([돕/VV] +와 +양성)
긋 : ([긋/VV] +어간 +음성)
그 : ([긋/VV] +아 +음성)

다음은 확대된 형태소 사전에서 '-아'나 '-어'로 시작하는 어미의 '-아'나 '-어' 부분을 '-와'나 '-워'로 바꾸고 형태 제약 조건의 +아 자질을 +와 자질로 바꾸어 만든 것이다. 이것은 ㅂ 불규칙 활용을 하는 용언을 분석하는데 필요한 변형된 어미이다.

16. 와도 : ([아도/EM] @(+와 +양성))
워도 : ([어도/EM] @(+와 +음성))

14-16과 같이 주어진 확대된 형태소 사전을 이용하면 '듣고', '돕고', '긋고', '들어도', '도와도', '그어도' 등과 같은 어절을 올바르게 분석할 수 있다. 여기서는 ㄷ, ㅂ, ㅅ 불규칙 용언만 예로 들었는데 다른 불규칙 용언도 마찬가지 방법으로 확대된 형태소 사전을 구성하면 된다. 또, '-아'나 '-어' 매개 모음 문제뿐만 아니라 '-으' 매개 모음 문제도 같은 방법으로 해결할 수 있다.

2.4 품사 제약 조건

조사는 체언과 함께 사용되지만 보조사 혹은 특수 조사라 하는 '-는', '-도', '-만' 등은 체언뿐만 아니라 부사와 함께 사용될 수도 있다. 또한 어미 중에는 모든 용언에 두루 쓰이는 것이 있는가 하면 동사나 형용사에 한해서 사용되는 것도 있다. 예를 들어 어미 '-거나'는 동사나 형용사 어느 것과도 결합할 수 있는 반면, 어미 '-는군요'는 동사와만 결합할 수 있으며 '-군요'는 형용사와만 결합할 수 있다. 이와 같이 조사나 어미는 자신의 왼쪽에 오는 형태소의 품사에 제약을 가하는 것을 볼 수 있는데 이러한 현상은 조사나 어미뿐만 아니라 다른 품사에서도 발견된다. 예를 들어 체언의 경우 자신의 왼쪽에 임의의 품사가 올 수 있는 것이 아니라 체언, 접두사, 관형사 또는 부호 등만이 올 수 있는 것으로 제약한다. 따라서 이것을 "임의의 형태소는 자신의 왼쪽에 오는 형태소의 품사를 제약할 수 있다"라고 일반화시킬 수 있는데, 이러한 제약을 **품사 제약 조건**이라고 한다. 다음은 품사 제약 조건이 명시된 형태소 사전의 예이다. 여기서 #(…) 부분은 품사 제약 조건을 나타낸다.

17. 에서 : ([에서/JO] #(NN NP NU NX))
은 : ([은/JO] #(NN NP NU NX AD) @(+자음))
는 : ([는/JO] #(NN NP NU NX AD) @(+모음))
거나 : ([거나/EM] #(VV VX SV AJ AX SJ EP) @(+어간 | + ㅆ))
는군요 : ([는군요/EM] #(VV VX SV EP) @(+어간))
군요 : ([군요/EM] #(AJ AX SJ EP) @(+어간 | +ㅆ))

위의 예에서 보듯이 조사 '-에서'는 명사, 대명사, 수사, 의존 명사와 같은 체언과 결합하지만, 조사 '-은'이나 '-는'은 체언은 물론 부사와도 결합하는 것으로 되어 있다. 그런데, '-은'이나 '-는'은 형태 제약 조건도 가지고 있으므로 아무 체언이나 부사와 결합할 수 있는 것이 아니라 지정된 형태 제약 조건을 만족하는 체언이나 부사와만 결합할 수 있다. 또, 어미 '-거나'는 용언이나 선어말 어미와 결합하는 반면 '-는군요'는 동사나 선어말

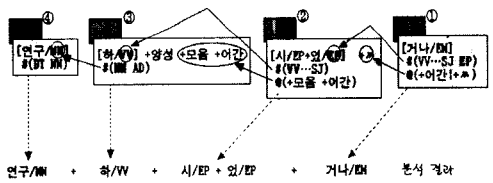
어미와, '-군요'는 형용사나 선어말 어미와 결합하는 것으로 되어 있다. 그런데 '-거나'나 '-군요'는 +어간 또는 +ㅆ 이라는 형태 제약 조건을 가지고 있으므로 이 단어의 왼쪽에는 용언 어간인 '사-', '돕-' 등이나 +ㅆ 형태 정보를 가진 '샀-', '-았-', '-었-' 등이 올 수 있다.

다음은 품사 제약 조건과 형태 및 음운 제약 조건을 모두 명시한 확대된 형태소 사전의 예이다. 이 사전을 보면 '하-'나 '했-'의 왼쪽에는 부사는 물론 명사도 올 수 있는 것으로 되어 있다. 이렇게 한 이유는 '하다'의 왼쪽에 명사가 와서 '연구하다', '실험하다'와 같이 쓰일 수도 있기 때문이다.

18. 하 : ([하/VV] #(NN AD) +어간 +양성 +모음)
    했 : ([하/VV + 았/EP] #(NN AD) +ㅆ +음성)
    시 : ([시/EP] #(VV VX SV AJ AX SJ) @(+모음 +어간) +어간)
    셨 : ([시/EP + 었/EP] #(VV VX SV AJ AX SJ) @(+모음 +어간) +ㅆ)

17, 18과 같이 주어진 형태소 사전을 이용하면 '하거나', '했거나', '하시거나', '하셨거나', '연구하시는군요', '연구하셨군요' 등과 같은 어절도 사전 탐색 및 인접 조건 검사만으로 분석할 수 있다. 19는 인접 조건 검사에 의하여 '연구하셨거나'가 분석되는 과정을 보인 것이다.

19.



위 그림에서 표시한 것과 같이 사전 탐색은 입력 어절의 오른쪽 끝에서 시작하여 역방향으로 진행된다. 따라서 인접 조건 검사는 19에서 실선 화살표를 따라 가는 방향으로 진행된다.

3. 비트 연산에 의한 형태소 분석 알고리즘

3.1 비트 연산에 의한 인접 조건 검사

확대된 형태소 사전을 이용한 분석 방법에서는 코드 변환이나 원형 복원 등과 같은 절차를 거치지 않고 오직 사전 검색과 인접 조건 검사만으로 형태소 분석을 할 수 있다. 또 이진 값으로 주어지는 인접 조건은 비트 형식으로 표현할 수 있으므로 단순한 비트 연산만으로 인접 조건 검사가 가능하다.

확대된 형태소 사전에서 @(…) 부분에 명시된 음운 형태 제약 조건의 각 요소는 이진 값을 가지므로 음운 형태 제약 조건을 그림 1과 같은 비트 형식으로 표현할 수 있다. 이 비트 형식에서 가장 왼쪽 비트는 음운 형태 제약 조건이 AND 조건인지의 여부를 나타낸다. 예를 들어 AND 조건인 @(+어간 -아 +양성)은 110100…1000으로 나타내며, OR 조건인 @(+어간 | +ㄴ)은 010010…0000으로 나타낸다. 음운 형태 제약 조건이 없는 경우에는 100000…0000으로 나타낸다.

+AND	+어간	+아	-아	+ㄴ	+ㄹ	…	+양성	+음성	+모음	+자음

그림 1 음운, 형태 제약 조건의 비트 표현

한편, 형태소 사전의 각 표제어에 대한 음운 형태 정보도 그림 1과 같은 형식에 따라 표현할 수 있다. 음운 형태 정보는 AND 조건과 같은 형식으로 표현하며 +AND 비트는 항상 1이 된다. 예를 들어 5에서 '학교'란 표제어에 주어진 음운 정보 (+모음)은 100000…0010으로 표현하며, 9에서 '돌보'란 표제어에 주어진 음운, 형태 정보 (+어간 +양성 +모음)는 110000…1010으로 표현한다.

각 품사를 하나의 비트로 표현한다면 확대된 형태소 사전의 #(…) 부분에 기술된 품사 제약 조건도 그림 2와 같은 비트 형식으로 표시할 수 있다. 예를 들어, #(NN NP NU NX AD)는 11111…0000000으로 표현할 수 있으며 #(AJ AX SJ EP)는 00000…1000111로 표현할 수 있다.

NN	NP	NU	NX	AD	…	EP	VV	VX	SV	AJ	AX	SJ

그림 2 품사 제약 조건의 비트 표현

각 표제어가 취할 수 있는 품사도 그림 2의 비트 형식에 따라 표현할 수 있다. 예를 들어, '가장'은 부사 또는 명사로 쓰일 수 있으므로 이것의 품사 정보는 10001…0000000으로 표현하며, '한'은 명사, 수사, 의존 명사 등으로 쓰일 수 있으므로 이것의 품사 정보는 10110…0000000으로 표현한다[6].

음운, 형태, 품사 제약 조건 등의 인접 조건과 음운, 형태, 품사 정보 등의 사전 정보를 위와 같은 비트 형식으로 표현한다면 간단한 비트 연산만으로 인접 조건 검사를 할 수 있다. 인접한 두 형태소에서 왼쪽 형태소를 L이라고 하고 오른쪽 형태소를 R이라고 하자. $M(R)$과 $T(R)$은 각각 R의 왼쪽에 올 수 있는 형태소에 대한 음운 형태 제약 조건과 품사 제약 조건을 나타내며, $m(L)$과 $t(L)$은 각각 형태소 L에 주어진 음운 형태 정보와 품사 정보를 나타낸다고 한다.

L이 R에 의해 주어진 음운 형태 제약 조건을 만족하는지의 여부는 다음과 같이 정의된 함수 $\mu(L, R)$를 이

6) 본 논문에서는 따로 언급하지 않았지만, 이와 같이 각 표제어의 품사를 비트로 표현할 수 있으므로 품사가 다른 체언류를 하나의 표제어로 형태소 사전에 등재할 수 있다. 예를 들어 '가장'은 ([가장/(NN AD)] +자음)으로, '한'은 ([한/(NN NU NX)] +자음)으로 형태소 사전에 등재할 수 있다. 용언류도 마찬가지 방법으로 할 수 있다. 이렇게 했을 때의 이점에 대해서는 [3]에 설명되어 있다.

용하여 판단할 수 있다. 여기서 ∧는 비트별 AND 연산을 나타낸다.

$$\mu(L, R) = m(L) \wedge M(R)$$

L이 R의 음운 형태 제약 조건을 만족하는 경우에는 $\mu(L, R)$의 값이 $M(R)$과 같거나 양수가 되며, 그렇지 않은 경우에는 $\mu(L, R)$의 값이 0 또는 음수가 된다[7]. 예를 들어, 형태소 사전이 다음과 같이 주어졌을 때,

보 : ([보/VV] +어간 +아 +양성 +모음) // m(보) = 111000…1010
봐 : ([보/VV] -아 +양성 +모음) // m(봐) = 100100…1010
아도 : ([아도/EM] @(+아 +양성)) // M(아도) = 101000…1000
도 : ([아도/EM] @(-아 +양성)) // M(도) = 100100…1000
다 : ([다/EM] @(어간|+ㅆ)) // M(다) = 010000…0000

'보아도', '봐도', '보다'에 대하여 음절을 경계로 μ 값을 계산해 보면, 각각 M(아도), M(도)와 같거나 0보다 크다. 따라서 이들 어절은 아래의 오른쪽에 나타낸 것처럼 분석할 수 있다.

μ(보,아도) = 101000…1000 = M(아도) 보/VV + 아도/EM
μ(봐,도) = 100100…1000 = M(도) 보/VV + 아도/EM
μ(보,다) = 010000…0000 > 0 보/VV + 다/EM

μ(보,도)를 계산해 보면 100000…1000으로, 이는 M(도)와 같지도 않고 0보다 크지도 않으므로 인접 조건이 맞지 않아 '보도'를 [보/VV + 아도/EM]로 분석할 수 없다.

L이 R에 의해 주어진 품사 제약 조건을 만족하는지의 여부는 다음과 같이 정의된 함수 $\tau(L, R)$로 판단할 수 있다.

$$\tau(L, R) = t(L) \wedge T(R)$$

L이 R에 의해 주어진 품사 제약 조건을 만족하는 경우에는 $\tau(L, R)$은 0 아닌 임의의 값을 취하게 된다. '바르다'는 '약을 바르다'와 같이 동사로 쓰일 수도 있고 '마음이 바르다'와 같이 형용사로도 쓰일 수 있으므로 t(바르)의 값은 00000…0100100이 된다. 어미 '-는군요' 앞에서는 선어말 어미나 동사 혹은 동사화 접미사 등이 올 수 있으므로 T(는군요)의 값은 000000…1111000이 된다. 이제, '바르는군요'에서 '바르-'가 어미 '-는군요'에 의해 주어진 품사 제약 조건을 만족하는지 검사하기 위해 τ(바르, 는군요) 값을 계산해 보면 00000…0100000이 되며 이는 0이 아니므로 어미 '-는군요' 앞에 '바르-'가 올 수 있음을 알 수 있다. 또, '바르다'는 동사로도 쓰일 수 있고 형용사로도 쓰일 수 있지만 어미 '-는군요'의 품사 제약 조건에 의해 '-는군요' 앞에 나온 '바르다'는 동사로 한정된다. 이러한 결과는 τ(바르, 는군요) 값으로부터 알 수 있다. 이와 같이 $\tau(L, R)$는 품사 제약 조건 검사뿐만 아니라 L의 품사를 한정하는 역할도 한다.

3.2 형태소 분석 알고리즘

지금까지 인접 조건 검사에 의한 한국어 형태소 분석 알고리즘의 대강을 살펴보았는데, 이것을 의사 코드로 나타내면 그림 3과 같다. 이 그림에서 analyze_word(w_{ij}, R)는 입력 어절 w_{ij}의 오른쪽에 R이 있을 때 R에 의해 주어진 인접 조건을 만족하는 w_{ij}의 분석 결과를 반환하는 함수이다. 이 함수를 처음으로 호출할 때에는 R의 값으로 ∅로 준다. r_{ij}는 입력 어절 w_{ij}에 대한 형태소 분석 결과를 저장하는 임시 변수이다. search_dict(w_{ij})는 w_{ij}의 오른쪽 끝에서 시작하여 역방향으로 확장된 형태소 사전을 탐색했을 때 발견되는 모든 사전 정보를 가져오는데, 이들은 모두 d_j에 저장된다.

```
procedure analyze_word( w_ij, R) // i < j
// T(∅) = #(NN NP NU NX SN VV … JO EM), M(∅) = @(+어말|+ㄴ|+ㄹ|+ㅁ|+아)
// +어말은 어절 끝에 올 수 있는 품사의 표제어에 주어지는 자질이다.
begin
  r_ij ← ∅
  d_j ← search_trie( w_ij)
  for ∀d^k_mj ∈ d_j do // i ≤ m < j
    if (τ( d^k_mj, R) ≠ 0 AND (μ( d^k_mj, R) > 0 OR μ( d^k_mj, R) = M(R))) then do
      if ( i = m) then do // 입력 어절의 왼쪽 끝 도달
        r_ij ← r_ij ∪ λ( d^k_mj)
      else do
        r_im ← analyze_word( w_im, d^k_mj)
        if ( r_im ≠ ∅) then do
          r_ij ← r_ij ∪ ( r_im ⊙ λ( d^k_mj))
        end_if
      end_if
    end_if
  end_for
  return r_ij
end_procedure
```

그림 3 인접 조건 검사에 의한 형태소 분석 알고리즘

입력 어절 w_{ij}에 대한 역방향 사전 탐색 결과 그림 4와 같이 m부터 j 사이의 단어가 발견되었다고 하자[8].

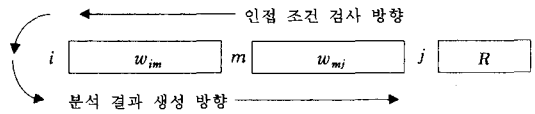


그림 4 분석 진행 방향

d_{mj}는 m부터 j 사이의 단어 w_{mj}에 대한 사전 정보 리스트를 나타내며[9], d^k_{mj}는 그 중 k 번째를 나타낸다. d^k_{mj}에 대해서 $\tau(d^k_{mj}, R)$과 $\mu(d^k_{mj}, R)$를 계산했을 때, $\tau(d^k_{mj}, R)$이 0이 아니고(즉 R에 의해 주어진 품사 제약 조건을 만족하고) $\mu(d^k_{mj}, R)$이 0보다 크거나 $M(R)$과 같으면(즉 R에 의해 주어진 음운 형태 제약 조건을 만

7) 여기서는 제일 왼쪽 비트가 부호 비트임을 가정하였다.

8) search_dict(학교에서는)를 호출하면 '는', '서는', '에서는' 등의 단어를 발견할 수 있으므로 m은 고정된 값이 아님에 주의할 필요가 있다.

9) 위의 '학교에서는'에서 '는'은 조사나 어미일 수도 있지만 '놀다'에 어미 '-ㄴ'이 결합한 형태로도 볼 수 있다. 이런 경우 한 단어에 대하여 여러 가지 다른 사전 정보가 있을 수 있다.

족하면) w_{mj}는 R에 의해 주어진 인접 조건을 충족한다. 이 경우 전체 어절 w_{ij} 중에서 w_{mj}까지는 분석이 완료된 것이므로[10], 남아있는 w_{im}에 대해서 마찬가지 방법으로 분석을 계속한다. w_{im}에 대한 인접 조건은 w_{mj}의 사전 정보인 d^k_{mj}에 의해 주어지므로 w_{im}를 분석하려면 analyze_word(w_{im}, d^k_{mj})와 같이 호출하면 된다.

그림 4에서 보듯이 입력 어절의 오른쪽 끝에서 시작하여 이웃한 두 형태소 사이의 인접 조건이 충족되는지의 여부를 검사하면서 왼쪽으로 분석을 진행하다가 입력 어절의 왼쪽 끝에 도달하게 되면 지나온 경로를 따라 다시 오른쪽으로 진행하면서 각 형태소에 대한 분석 결과를 생성하고 이를 이전 결과와 결합해 나가면서 궁극적으로는 전체 입력 어절에 대한 분석 결과를 생성한다. 각 형태소에 대한 분석 결과는 형태소 사전의 […] 부분에 명시된 내용을 그대로 가져오면 된다. 그림 3에서 λ(d^k_{mj})는 w_{mj}의 사전 정보 d^k_{mj}에서 […] 부분에 명시된 분석 결과를 나타낸다. 또, U는 분석 결과를 저장하는 변수 r_{ij}에 다른 분석 결과를 추가하는 것을 나타내는 연산자이다. 가령 r_{ij} = {[가/VV + 는/EM]}이고 λ(d^k_{mj}) = [갈/VV + 는/EM]이라면 r_{ij} ∪ λ(d^k_{mj}) = {[가/VV + 는/EM], [갈/VV + 는/EM]}가 된다. ⊙은 Cartesian 곱을 하여 형태소 분석 결과를 결합하는 것을 나타내는 연산자이다. 예를 들어, r_{ij}가 위에서 본 것과 같다면 r_{ij} ⊙ [것/NX] = {[가/VV + 는/EM + 것/NX], [갈/VV + 는/EM + 것/NX]}이 된다.

3.3 방법론 검토

본 논문에서 제안한 방법에서는 음소 단위의 연산을 위한 코드 변환과 원형 복원 등과 같은 복잡한 과정을 거칠 필요가 없다. 원형 복원 과정에서 여러 가지 원형 복원 규칙이 적용되며 이 과정에서 실제로 사전에 존재하지 않는 가짜 단어들도 분석 후보라는 이름으로 생성된다. 사전 탐색을 통해 가짜 단어가 포함된 분석 후보 중에서 진짜 단어를 찾아 낼 수는 있다. 하지만 결국은 걸러져 없어질 가짜 단어를 생성하고 또 이들의 진위를 가리기 위하여 사전을 탐색하는 과정에서 많은 시간을 낭비하게 되며 이는 분석 속도 저하라는 결과로 이어진다. 하지만 본 논문에서 제시한 방법에서는 이러한 과정이 생략되므로 분석 효율을 향상시킬 수 있다.

본 논문에서 제안한 방법은 형태소의 기본형뿐만 아니라 표층형까지 형태소 사전의 표제어로 삼는다는 점에서 [9]에서 제안한 사전 기반 형태소 분석 방법과 유사하다. 그러나 [9]에서는 음절을 경계로 한 분석을 시도하지 않았다는 점에서 본 논문에서 제안한 방법과 차이가 있다. 예를 들어, [9]에서는 '쓰', '아름다우'를 '쓰다', '아름답다'의 표층형으로 보고 형태소 사전의 표제어로 등재하였는데, 이 경우 '써서'나 '아름다운지'를 분석하려면 '써'에서 '쓰'과 'ㅓ'를 분리하고 '운'에서 '우'와 'ㄴ'을 분리하는 등의 자소 단위 연산과 이를 위한 코드 변환 과정이 필요하다. 반면, 본 논문에서 제안한 방법에서는 '쓰'와 '아름다우' 대신 '써'와 '아름다'가 형태소 사전의 표제어로 등재되며, 자소 단위 연산이나 코드 변환 없이 인접 조건 검사만으로 형태소 분석이 이루어진다.

[2]에서는 인접한 두 형태소 리스트 사이의 결합 가능성을 확인하기 위해 병합 조건 검사, 인접 조건 검사, 형태소 배열법 검사의 세 단계 과정을 순차적으로 거치는 것으로 되어 있다. 병합 조건 검사 단계에서는 인접한 형태소 리스트에 중복 형태소가 있는지의 여부를 검사하고 중복 형태소가 있는 경우 이를 제거하여 병합 리스트를 생성하는 일을 한다. 인접 조건 검사 단계에서는 형태, 음운, 품사 제약에 대한 검사를 수행하는데, 형태 제약 검사는 중복 형태소를 제거하는 절차만 없을 뿐 병합 조건 검사와 비슷한 방법으로 진행된다. 그런데, 단순히 <형태소/품사>로만 주어지는 병합 조건과 달리 형태 제약은 <형태소/품사>들의 리스트로 주어지므로 형태 제약 검사는 이 리스트를 순차적으로 따라 가면서 진행된다는 점에서 더욱 비효율적이다. 병합 조건과 인접 조건을 충족했다 하더라도 형태소 배열법에 맞지 않으면 형태소 리스트를 결합할 수 없으므로 선행 형태소의 품사 뒤에 어떤 품사가 뒤따를 수 있는지를 나타내는 품사 전이 표를 참조하여 형태소 배열법 검사를 해 주어야 한다. 반면, 본 논문에서 제안한 방법에서는 단순한 비트 연산에 의한 인접 조건 검사만으로 위에서 열거한 여러 가지 검사를 대신할 수 있다. 스트링을 비교하거나 표를 참조하거나 리스트를 순차적으로 따라 가는 등의 비효율적인 요소가 전혀 없기 때문에 빠른 시간 내에 인접 조건 검사를 마칠 수 있다.

[2]에서는 한 형태소가 하나의 품사만 가지는 것으로 보았는데, 이는 형태소 분석 과정에서 형태소의 결합 가능성 여부를 중복해서 검사하는 문제를 야기한다. 예를 들어 '칠'과 같이 명사도 될 수 있고 수사도 될 수 있는 경우, [2]에서는 [칠/NN]과 [칠/NU]로 구분하여 표현한다. 그 결과, 가령 '칠은'을 분석하려면 [칠/NN]과 [은/JO] 사이의 결합 가능성뿐만 아니라 품사 조건만 다르고 다른 조건은 동일한 [칠/NU]과 [은/JO] 사이의 결합 가능성에 대해서도 검사를 하기 때문에 비효율적이다. 본 논문에서는 한 형태소가 여러 품사를 가질 수 있는 것으로 보고 '칠'을 [칠/(NN NU)]로 나타내기 때문에 그

10) 그러나 이 단계에서는 아직 분석 결과는 생성하지 않는다. 왜냐하면 도중에 인접 조건을 충족시키지 못하는 경우가 생길 수도 있기 때문이다.

와 같은 중복 검사를 피할 수 있다.

이상에서 본 바와 같이 [2]에서 제안한 부분 어절의 기분석에 기반한 형태소 분석 방법은 음절을 경계로 분석이 이루어진다는 점에서 본 논문에서 제안한 방법과 유사하지만 형태소 사이의 결합 가능성을 검사하는 방법 등에서 차이가 있다.

형태소 분석 시 인접한 형태소 사이의 접속 여부를 결정하기 위해 각 범주별 접속 관계를 나타내는 접속 정보 표를 두고 이를 참조하면서 형태소 분석을 진행하는 것이 보통이다. 그런데 본 논문에서 제안한 방법에서는 접속 정보가 표의 형태로 한 곳에 모여 있는 것이 아니라 각 형태소 별로 분산이 되어 있기 때문에 형태소 분석 중에 접속 정보 표를 탐색하는 등의 과정이 불필요하다. 접속 정보가 분산되어 있으므로 형태소 별로 접속 정보를 달리 줄 수도 있다. 예를 들어 '먹은것', '먹는것', '먹을것', '먹는수', '먹을수' 등과 같이 흔히 볼 수 있는 띄어쓰기 오류어의 경우 20에서와 같이 '것'이나 '수' 앞에 특정 유형의 어미가 나올 수 있다는 예외적인 접속 정보를 줌으로써 분석이 가능하도록 할 수 있다[11]. 여기서 +ㄴ, +는, +ㄹ은 어미 형태를 나타내며 각각 '-은', '-는', '-을' 등의 어미에 주어진다.

20. 것 : ([것/NX] #(EM) @(+ㄴ ¦ +는 ¦ +ㄹ))
    수 : ([수/NX] #(EM) @(+는 ¦ +ㄹ))

본 논문에서 제안한 방법에서는 표제어를 무엇으로 삼든지 표제어의 왼쪽에 올 수 있는 형태소의 품사, 음운, 형태 조건을 줄 수 있기 때문에, '아름답게되어'나 '천사와같은'에서와 같이 특정 조사나 어미 뒤에 특정 단어가 동반되어 상투적으로 쓰이는 띄어쓰기 오류어에 대해서도 형태소 분석을 할 수 있다. 다음은 이러한 유형의 띄어쓰기 오류어를 분석하기 위해 형태소 사전에 추가해야 할 내용을 보인 것이다. 이러한 오류어의 분석을 위해 형태소 분석 알고리즘을 변경할 필요가 전혀 없다.

21. 와같 : ([와/JO + 같/AJ] #(NN NP NU NX) @(+모음) +어간)
    과같 : ([과/JO + 같/AJ] #(NN NP NU NX) @(+자음) +어간)
    게되 : ([게/EM + 되/VX] #(VV SV AJ SJ EP) @(+어간) +어간)

4. 형태소 사전

2 장에서 제안한 확대된 형태소 사전의 구조는 다소 복잡해 보인다. 하지만 실제 구현에서는 확대된 형태소 사전을 직접 구축하는 것이 아니라 그림 5에서 보는 바와 같이 사전 변환 프로그램을 이용하여 단순한 형태의 원시 형태소 사전으로부터 확대된 형태소 사전을 자동 생성하는 방식을 취하므로 사전 구축 작업이 그다지 어렵지 않다.

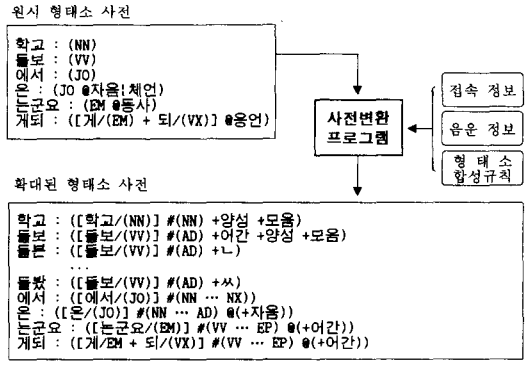


그림 5 확대된 형태소 사전 구축

사전 변환 프로그램은(전통적으로 형태소 분석기에 포함되어 있던) 접속 정보와 음운 정보 등을 참조하여 원시 형태소 사전의 각 표제어에 품사 제약 조건 및 음운, 형태 정보를 부가한다. 또, 필요한 경우 형태소 합성 규칙을 적용하여 용언의 어간과 ㄴ, ㄹ, ㅁ, ㅂ, ㅆ, ㅏ, ㅓ 등의 자소가 결합된 형태도 생성한다.

조사나 어미의 경우에는 자음으로 끝나는 체언 뒤에서 쓰이는 조사, 모음으로 끝나는 용언 뒤에서 쓰이는 어미, 임의의 동사 뒤에서 쓰이는 어미 등과 같이 각 조사나 어미가 사용될 수 있는 음운 형태적 제약에 따라 이들을 몇 가지 유형으로 분류한 다음 이러한 제약을 표시할 수 있는 @용언, @동사, @자음¦체언, @모음¦용언 등의 명칭을 사용하여 원시 형태소 사전을 만들기 때문에 원시 형태소 사전의 내용은 상당히 직관적이다. 이러한 명칭은 이후 사전 변환 프로그램에 의해 적절한 음운, 형태, 품사 제약 조건으로 변환된다.

기존 형태소 분석기에서는 분석을 용이하게 하기 위하여 복합 조사나 어미를 표제어로 형태소 사전에 등재하는 경우가 많았다[9]. 이러한 이유 때문에 [10]은 기존 형태소 분석기에서 복합 조사나 어미의 원형을 제대로 복원하지 않음을 지적하였는데, 이것은 기술적인 문제가 아니라 형태소 분석기의 용도에 따른 선택의 문제이다. 앞에서 언급한 것처럼 형태소 사전에서 […] 부분은 형태소 분석 결과를 나타내므로 22의 (가)는 형태소 분석 결과가 단위 조사나 어미로 주어져야 할 때의 사전 형식이며, (나)는 형태소 분석 결과가 단위 조사나 어미로 주어지지 않아도 될 때의 사전 형식이다.

11) 모든 명사 앞에 관형형 어미가 올 수 있다고 인접 조건을 준다면 '생각하는사람'과 같은 유형의 띄어쓰기 오류어도 분석이 가능하지만 '박찬호'를 [박차/VV + ㄴ/EM + 호/NX]와 같이 잘못 분석하는 경우가 생길 수 있다.

22. (가)에서부터는 : ([에서/JO + 부터/JO + 는/JO] …)
라고까지는 : ([라고/EM + 까지/JO + 는/JO] …)
(나)에서부터는 : ([에서부터는/JO] …)
라고까지는 : ([라고까지는/EM] …)

위의 두 가지 형식의 형태소 사전을 따로 구축해 두었다가 필요에 따라 하나를 선택해 사용할 수도 있지만, (가)를 (나)와 같은 형태로 변환하는 것은 매우 간단하므로 일단 (가)와 같은 형식의 형태소 사전을 구축해 둔 다음 형태소 분석기를 수행할 때 그대로 적재하거나 (나)와 같은 형식으로 변환한 후 적재하는 것도 가능하다.

조사나 어미 외에 다른 품사의 복합어도 표제어로 형태소 사전에 등재하는 경우가 많은데, 이 때에도 사전의 […] 부분에 형태소 분석 결과를 어떻게 명시하는가에 따라 복합어 자체를 하나의 단위로 분석 결과를 생성할 수도 있고 단어 수준의 분석 결과를 생성할 수도 있다.

5. 성능 평가 및 비교

여기서는 본 논문에서 제안한 알고리즘을 C++ 언어로 구현한 한국어 형태소 분석기 MACH (Morphological Analyzer for Contemporary Hangul)의 성능을 평가하고 그 결과를 다른 실험 결과와 비교해 보기로 한다. 이 실험에 사용된 MACH의 원시 형태소 사전은 약 61,000 개의 체언, 5,600 개의 용언, 600 개의 조사, 2,000 개의 어미, 50 개의 선어말 어미로 구성되어 있다. 사전 변환 프로그램으로 이 사전을 확대하면 약 98,000 개의 표제어를 가진 확대된 형태소 사전이 생성되는데 이 사전의 크기는 약 3 MB이다. 확대된 형태소 사전은 반음절 단위의 트라이[11]로 구성하여 주기억 장치에 적재한다.

먼저, 분석 정확도에 대한 평가를 실시하였다. 정확도 평가를 위해 한국 과학 기술원에서 개발한 대한 민국 국어 정보 베이스 평가판 0.1에 포함된 700 문서, 50.2 MB, 7.06 백만 어절의 말뭉치에서 5 개 문서를 임의로 선정한 후 여기서 10,000 어절을 무작위로 뽑아 실시하였다[12]. 평가 결과 MACH의 분석 정확도는 99.2 %로 나타났다. 이것은 [4]의 99.4 %보다는 다소 낮지만 [2]의 98.6 % 보다는 약간 높은 수치이다. 그런데 각 문헌에서 제시한 형태소 분석기의 정확도는 동일 문서를 기준으로 측정한 것도 아니고, 형태소 분석기에 따라서 분석의 정도에 차이가 있기 때문에 절대 수치의 단순 비교는 그다지 중요하지 않다고 본다[13].

다음은 평균 사전 탐색 횟수에 대한 평가를 실시하였다. 그 결과 평균 사전 탐색 횟수는 어절당 2.51 회였다. 이는 [2]의 2.31 회보다는 약간 많은 편이나 음절 정보를 이용했을 때의 5.70 회[4]나 최장 일치법에서 사전 탐색 횟수를 줄이는 기법을 적용했을 때의 4.12 회[6]에 비해서는 많이 낮은 편이다. 사전 검색 횟수는 형태소 사전의 표제어 선정 기준과 밀접한 관련이 있는데, 가령 '-었다'나 '-신다' 등을 분석할 때, [2]에서는 선어말 어미와 어말 어미를 결합한 형태를 표제어로 선정했기 때문에 사전 탐색을 한 번만 하면 되지만, MACH에서는 선어말 어미와 어말 어미를 별도의 표제어로 선정하였기 때문에 두 번의 사전 탐색이 필요하다. 이러한 구현 상의 차이가 평균 사전 탐색 횟수에 약간의 영향을 미쳤을 것으로 판단된다.

마지막으로 분석 속도에 대한 평가를 실시하였다. 이 평가는 한국 과학 기술원의 50.2 MB 전체 말뭉치를 대상으로 실시하였다. 참고로 이 말뭉치의 평균 어절 길이는 3.18 음절이다. 이 말뭉치를 1.13 GHz Pentium III의 개인용 컴퓨터(512 MB 메모리, 리눅스)에서 분석하는데 소요된 시간은 15.7 초였다. 이는 초당 약 45 만 어절을 분석할 수 있으며, 1 MB의 문서를 분석하는 데에는 약 0.3 초가, 1 GB의 문서를 분석하는 데에는 약 5 분이 걸림을 의미한다. 참고로, [2]에서 제안한 방법에 따라 구현된 형태소 분석기는 Pentium II 333 MHz의 개인용 컴퓨터에서 1 GB의 문서를 분석하는데 약 172 분이 걸린다고 했다. 또, [12]에서는 분석 배제 정보와 후절어를 이용하여 형태소 분석을 거쳐야 하는 어절의 수를 최대한 줄임으로써 고속으로 명사를 추출하는 방법을 제안하였는데, 이 방법에 따라 구현된 명사 추출기는 Pentium III 450 MHz의 개인용 컴퓨터에서 초당 4.3 만 어절을 분석한다고 했다.

위 말뭉치를 분석하는 과정에서 MACH의 주요 함수들이 어느 정도의 시간 비중을 차지하는지 분석해 보았다. 표 1은 리눅스 상에서 gprof를 사용해 분석한 결과이다. 이 표에서 analyze_sent()는 문장에 대한 형태소 분석 결과를 반환하는 함수로서 이 함수의 주요 기능은 주어진 문장을 어절 단위로 분리하여 analyze_word()를 반복 호출하고 그 결과를 취합하는 일이다. search_trie()는 트라이 사전을 탐색하여 그 결과를 반환하는 함수이다.

표 1에서 알 수 있듯이 형태소 분석과 직접적으로 관련된 두 함수가 차지하는 시간 비중은 전체의 73.59 %

12) 참고로 실험에 사용된 문서의 제목은 임상 의학의 탄생, 생활 경제 이야기, 상대성 이론, 정보 사회와 정치 과정, 나의 문화 유산 답사기이며, 이들 문서의 평균 어절 길이는 3.17 음절이다.

13) 예를 들어 복합 명사를 분해하지 않고 전체를 하나의 명사로 출력하는 형태소 분석기와 이를 단위 명사로 분해하여 출력하는 형태소 분석기가 있다고 하자. 후자의 경우 복합 명사 분해 오류로 인한 감점 요인이 추가되므로 정확도 평가시 후자가 전자에 비하여 다소 불리하다. MACH는 복합 명사를 단위 명사로 분해하여 출력하고 있다.

표 1 주요 함수별 시간 비중

함수명	시간 비중 (%)
analyze_sent()	12.65
analyze_word()	60.94
search_trie()	21.93
기타	4.48

로 사전 탐색이 차지하는 시간 비중보다 월등히 높다. 이 결과를 보면 평균 사전 탐색 횟수의 약간 높고 낮음은 형태소 분석기의 전체 성능에 그다지 큰 영향을 미치지 않는다고 할 수 있다. 뿐만 아니라 평균 사전 탐색 횟수를 낮추기 위한 노력이 사전 탐색 횟수를 낮춤으로써 얻는 시간적인 이득을 상회하여 평균 사전 탐색 횟수는 낮추었지만 분석 시간은 더 걸리는 상황이 발생할 수도 있다.

6. 결 론

본 논문에서 제시한 인접 조건 검사에 의한 형태소 분석 방법에서는 축약, 탈락, 불규칙 활용으로 변형된 형태소의 원형을 복원하기 위한 복잡한 과정을 거치지 않을뿐더러 분석 중에 무의미한 분석 후보도 생성하지 않는다. 또한 모든 연산이 음절을 경계로 이루어지므로 한글 코드 변환 과정이 불필요하다. 더구나 본 방법에서는 단순한 비트 연산에 의한 인접 조건 검사만으로 형태소 분석이 이루어지는 아주 간단한 알고리즘이므로 초고속 한국어 형태소 분석기 구현에 적합하다. 본 알고리즘에서 언어적인 요소는 모두 형태소 사전에 들어 있으므로 언어 독립적이다. 따라서 이 알고리즘은 한국어는 물론 영어와 같은 다른 언어의 형태소 분석기 구현에도 그대로 적용할 수 있다.

이 알고리즘을 구현하여 만든 한국어 형태소 분석기 MACH는 1.13 GHz Pentium III의 개인용 컴퓨터에서 대략 5분/GB의 분석 속도를 보였다. 분석 정확도는 99.2 %로서 기존의 한국어 형태소 분석기와 큰 차이가 없었다.

참 고 문 헌

[1] 김영관, 박민식, 최진석, 권혁철, "사전 성능 개선을 통한 한국어 형태소 분석기의 분석 속도 향상", 제11회 한글 및 한국어 정보처리 학술대회 논문집, pp.479~483, 1999.

[2] 양승현, 김영섭, "부분 어절의 기분석에 기반한 고속 한국어 형태소 분석 방법", 정보과학회 논문지 : 소프트웨어 및 응용, 27권, 3호, pp.290~301, 2000.

[3] Kwangseob Shim and Jaehyung Yang, "MACH : A Supersonic Korean Morphological Analyzer," Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002), pp. 939~945, 2002.

[4] 강승식, "음절 정보와 복수어 단위 정보를 이용한 한국어 형태소 분석", 서울대학교 공학박사 학위 논문, 1993.

[5] 임희석, 윤보현, 임해창, "배제 정보를 이용한 효율적인 한국어 형태소 분석기", 한국정보과학회 논문지, 제22권 제6호, pp.957~964, 1995.

[6] 최재혁, 이상조, "양방향 최장일치법에 의한 한국어 형태소 분석기에서의 사전 횟수 감소 방안", 정보과학회 논문지, 20권, 10호, pp.1497~1507, 1993.

[7] 백대호, 이호, 임해창, "Finite State Transducer를 이용한 한국어 전자 사전의 구조", 제7회 한글 및 한국어정보처리 학술발표 논문집, pp.181~187, 1995.

[8] 김재한, 옥철영, "어절 사전을 이용한 한국어 형태소 분석", 한국정보과학회 봄 학술발표 논문집, 21권 1호, pp.813~816, 1994.

[9] Hyuk-Chul Kwon, Young-Soog Chae, "A Dictionary-Based Morphological Analysis," Proc. of Natural language processing, Pacific Rim Symposium '91, Singapore, pp.178~185, 1991.

[10] 은종진, 박선영, "고성능 한국어 형태소 분석을 위한 어미 분류", 제12회 한글 및 한국어 정보처리 학술대회 논문집, pp.41~47, 2000.

[11] 김철수, 배우정, 이용석, 청강순일, "이중 배열 트라이 구조를 이용한 한국어 전자 사전의 구축", 정보과학회 논문지 (B), 23권, 1호, pp.85~94, 1996.

[12] 이도길, 류원호, 임해창, "분석 배제 정보와 후절어를 이용한 한국어 명사 추출", 제12회 한글 및 한국어 정보 처리 학술대회 논문집, pp.19~25, 2000.

심 광 섭

1986년 서울대학교 컴퓨터공학과 학사
1988년 서울대학교 컴퓨터공학과 석사
1994년 서울대학교 컴퓨터공학과 박사
1995년~현재 성신여자대학교 컴퓨터정보학부 부교수. 관심분야는 자연어처리, 한국어정보처리, 정보검색, 인공지능 등

양 재 형

1988년 서울대학교 컴퓨터공학과 학사
1990년 서울대학교 컴퓨터공학과 석사
1995년 서울대학교 컴퓨터공학과 박사
1995년~현재 강남대학교 컴퓨터미디어공학부 부교수. 관심분야는 한국어정보처리, 자연언어처리, 인공지능 등임