말뭉치 기반 부분 어절 기분석 사전의 구축과 형태소 분석

신준철○, 옥철영
울산대학교, 전기공학부
ducksjc@hotmail.com, okcy@ulsan.ac.kr

Construction of Partial Word Morpheme Dictionary based on Tagged Corpus and Korean Morphological Analysis

Joon-Choul Shin○, Cheol-young Ock
Dept. of Electrical Engineering, Ulsan University,

요 약

기존의 말뭉치 기반 한국어 형태소 분석 방법은 대용량의 어절 기분석 사전을 사용하여 분석하고, 그 사전에 없는 어절은 코드 변환, 형태소 분리, 원형 복원 규칙 적용 등을 거치는 복잡한 분석 방법을 통해 후보들을 생성했다. 이 복잡한 분석 방법은 제작과 유지보수, 실행 관점 모두에서 효율적이지 못하며 정확률을 낮추고 속도를 느리게 하는 요인이 된다. 이런 문제를 해결하기 위해 부분 어절의 기분석 사전을 구축하여 사용하는 방법이 연구되었다. 본 논문에서는 대용량의 분석 말뭉치를 통해 부분 어절의 기분석 사전을 구축하고 형태소 분석에 사용하는 방법을 제안한다. 세종 말뭉치로 실험한 결과 재현율이 99.05%였으며, 품사 및 동형이의어 태깅 정확률은 96.76%였다.

주제어: 한국어, 형태소 분석기, 말뭉치, 부분 어절 기분석 사전

1. 서론

한국어 형태소 분석기는 기계 번역, 정보 검색 등 한국어를 처리하는 정보 기술 분야에서의 필요성 때문에 오래 기간에 걸쳐 많은 연구가 이루어졌다. 그러나 여전히 속도와 정확률 그리고 유지보수 측면에서 새로운 접근방법이 필요하다.

기존의 알려진 한국어 형태소 분석 방법들은 크게 전분석과 무분석 2가지로 나뉜다[1]. 전분석 방법이란 입력 어절에서 형태소의 원형을 복구해 가면서 하나씩 분리하는 방법으로 테이블 파싱법[2], 최장 일치법[3], Two-Level 분석법[4] 등이 있다. 이 방법은 복잡한 원형 복구 과정을 위해 조합형으로 코드변환이 필요하고 다양한 중간 과정을 만들어내어 사전 검색횟수가 많아지고 과분석되는 단점이 있다.

전분석의 이런 단점들을 개선하기 위해 사전 검색 횟수를 줄이는 방법이나 중간 분석결과의 조합수를 최소화하는 등의 시도[5, 6]가 있었다. 그리고 미리 통합형태소를 만들어 사용하여 원형복구 과정을 일부 생략하도록 하는 시도[7]가 있었다.

무분석 방법이란 원형 복구 전, 어절에 있는 표층형의 전체 또는 부분을 사전에 미리 등록하고, 형태소 분석시에는 표층형으로 검색하여 미리 분석된 내용을 조합하는 것이다. 빈번하게 나타나지만 처리가 곤란한 준말을 어절 그대로 등록하거나 말뭉치에서 모든 어절을 등록한 기분석 사전 Full Word Morpheme Dictionary(FWD)이 대표적이다. 그러나 현실적으로 FWD가 모든 입력 어절을 처리할 수는 없다. 이런 문제를 해결하기 위해 부분 어절의 기분석을 등록한 Partial Word Morpheme Dictionary(PWD)를 이용할 수 있다. 통합형태소를 이용한 [7]의 연구가 PWD에 포함될 수 있다. 이 외에 PWD를 구축 사용하는 연구로 형태소 분석기 개발 환경 MADE[8]가 있었고, 형태소 사전을 통해 구축하고 사용하는 연구[1]가 있었다. 일단 구축된 FWD 또는 PWD가 있다면 무분석 처리는 간단한 알고리즘으로 구현 가능하다.

무분석은 전분석보다 비교적 최근에 연구된 것들로 더 큰 사전용량을 필요로 하지만 컴퓨터 기술의 향상으로 주 기억 장치에 모두 저장이 가능하며, 실행 시에는 처리 속도가 빠르고 사용자가 사전을 수정/추가하면서 유지보수를 쉽게 할 수 있기 때문에 특정 도메인의 최적화도 쉽게 할 수 있다.

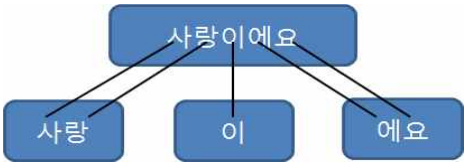
그러나 PWD는 구축이 간단하지 않다는 문제가 있다. [1]은 형태소 사전에 등록된 형태소에 음운 변화를 일으켜 표층형을 만들어 PWD를 구축하였다. 이와 다르게 본 논문에서는 말뭉치를 통해 FWD를 구축할 때 PWD를 같이 구축하는 방법을 제안하며 여기에 적합한 형태소 분석 방법도 같이 제안한다.

2. 학습 데이터 구축

분석된 말뭉치를 통해서 PWD와 형태소 위치 적합성를 구축하는 방법을 제안한다.

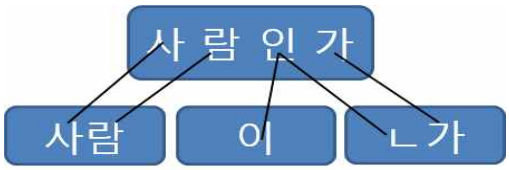
2.1 표층형-형태소의 음절 연결

부분 어절의 표층형과 원형을 추출하기 위해서는 표층형과 원형간의 음절 단위의 연결 정보가 필요하다. [그림 1]은 한 예로 전체 어절 '사랑이에요'를 3개의 형태소로 분해한 것이다. 부분 어절인 '이에요'에 대해서 기분석 내용을 구축할 때"이+에요"가 되는 것을 연결 정보를 통해 알 수 있다.



[그림 1] 음절 연결의 예 '사랑이에요'

[그림 1]의 경우 음절 연결이 쉽지만 음운 변화가 발생한 경우 복잡해진다. [그림 2]는 음운 변화가 일어난 예로 표층형에 원형과 다른 음절이 나타나고 음절의 총



[그림 2] 음절 연결의 예 '사람인가'

길이도 다르다.

음운 변화가 일어나면 항상 원형의 길이가 표층형의 길이와 같거나 더 길어지게 된다. [그림 2]에서는 '인'이 "이+ㄴ"으로 분리되면서 글자가 변하고 하나가 더 생겨났다. 일반적으로 음운 변화가 일어나면 이처럼 글자가 변하고 길이가 하나 늘어나기 때문에 좌에서 우측으로 연결을 하면서 글자가 다른 경우 [그림 2]의 '인'처럼 이중 연결을 시도한다. 반대의 경우는 단일 연결이라고 한다. 이 방법으로 대부분의 음운 변화 문제를 해결할 수 있다.

그러나 한국어의 음운 변화에는 많은 불규칙이 있기 때문에 위와 같은 방법에는 다양한 예외가 발생한다. 예를 들어 "들어 - 듣+어"와 같이 ㄷ불규칙이 있다. 이런 경우 단일 연결하도록 예외처리 해야 된다.

또 다른 예로 'ㄹ불규칙'의 경우 "이르러 : 이르+어", "갈라 : 가르+아"에서 어미'러'와 '어'를 같은 글자로 인식하도록 하여 단일 연결이 되도록 예외처리를 해야 한다. 이와 비슷한 패턴의 음운 변화들을 모두 처리하면 말뭉치의 거의 모든 어절에서 음절 연결 정보를 구축할 수 있게 된다.

이 외에 남는 경우에는 이중 연결이 아닌 삼중 연결이 필요한 준말이 있다. 예를 들어"뭘 - 무엇+ㄹ"이 그렇다. 이렇게 완전히 예외적인 경우에는 음절 연결과 부분 어절 학습을 포기하고 전체 어절을 그대로 학습한다.

정리하면 좌측 끝에서 오른쪽 방향으로 이동하면서 음운 변화를 예외처리하면 연결 정보를 자동으로 구축하는 모듈을 제작할 수 있다.

2.2 부분 어절 추출

[그림 1]에 경우 총 6가지의 부분 어절을 추출할 수 있으며 이 것이 <표 1>에 나타나있다.

<표 1> 추출 가능한 부분 어절들

번호	표층형	원형
1	사랑이에요	사랑+이+에요
2	사랑이	사랑+이
3	사랑	사랑
4	이에요	이+에요
5	에요	에요
6	이	이

천만 어절이 넘는 세종말뭉치를 통해 사전을 구축할 경우, 실제로 사용될 수 있는 대부분의 조사와 어미 그리고 보조 용언의 조합(어절의 꼬리)을 추출할 수 있을 것으로 가정한다.

이렇게 구축된 부분어절을 이용하여, 형태소 분석시에 어절의 꼬리를 분석하기 위해 두 가지 이상의 부분 어절을 조합할 필요 없이 한 번에 찾아낼 것이다. [그림 1]을 예로 들면, '이에요'를 분석하기 위해 '이'와 '에요'를 조합하는 것이 아니라, 한번에 '이에요'를 검색하여 해결한다. 따라서 '이'만을 따로 학습할 필요가 없다. 정리하면 [표 1]에서 6번과 같이 어절의 좌측 끝이나 우측 끝을 포함하지 않는 부분 어절은 학습하지 않는다.

[그림 2]의 '인'처럼 이중 연결된 음절은 특별 처리가 필요하다. 연구[9]에서는 'ㄴ가'에 대한 표층형으로 '가'만을 사용하지만, 본 논문에서는 '인가'를 사용할 것을 제안한다. 단, '인'의 좌측 연결 부분이 존재했음을 따로 기억하기 위해 '*인가'의 형태로 저장한다. 이런 방식으로 [그림 2]에서 모든 부분 어절을 추출하면 <표 2>가 된다. 원형 '사람+이'의 표층형은 '사람인*'으로, 좌측이 '*인'으로 시작하는 '*인가'와 조합될 수 있고, 둘의 원형을 연결하여 '사람+이+ㄴ가'을 만들 수 있다.

<표 2> 부분 어절의 표층형 추출 예

번호	표층형	원형
1	사람인가	사람+이+ㄴ가
2	사람인*	사람+이
3	사람	사람
4	인가	이+ㄴ가
5	*인가	ㄴ가

2.3 동사 음운 변화 추가 등록

'~하다'류 동사들은 모두 "~했다, ~한다" 등으로 변하면서 '하'는 다른 글자로 바뀔 수 있다. 말뭉치에서 모든 '~하다'류 동사들이 모두 가능한 음운 변화를 나타낼 것을 기대하기는 힘들기 때문에 직접 등록할 필요가 있다. 이 예가 <표 3>에 나타나있다. 간단하게 음운 변화된 마지막 글자를 표층형에서 수정하여 등록하면 된다.

표층형 '수영했*'은 "*했다, *했었다, *했었겠다" 등과 조합될 수 있다. 이와 비슷하게 '수영한*'은 "*한다. *한다면, *한다던가"등과 조합될 수 있다.

이와 같은 방법으로 모든 동사들의 음운 변화를 등록하면 된다.

<표 3> 동사 음운 변화 추가 등록

번호	표층형	원형
1	수영하	수영하
2	수영했*	수영하
3	수영한*	수영하
4	수영함*	수영하

2.4 형태소 위치 적합성 학습

하나의 어절이 두개 이상의 부분 어절로 분석될 경우 각 부분 어절의 원형이 서로 연결될 수 있는지, 또는 각각 좌측이나 우측에 위치해도 되는지를 검사해야 하며 이는 문법적인 요소를 통해 가능하다[1]. 본 논문에서는 PWD를 통한 분석으로 나온 후보들에 점수를 평가하여 상위 몇 개만 출력하는 방법을 제안한다.

두 원형의 연결 적합성은 연결될 두 형태소가 말뭉치에서 나타나는 형태를 수집하여 계산할 수 있다. 본 논문에서는 다음의 수식을 사용하였다.

$$connection(m1,m2) = p(right(m2)|left(m1))$$

수식 1 인접한 두 형태소의 연결 강도

[수식 1]에서 m1과 m2는 형태소이며 connection은 두 형태소의 연결 강도를 나타낸다. right(m2)와 left(m1)는 인접한 두 형태소에서 각각 우측과 좌측에 해당 형태소가 나타날 확률이다. 정리하면 connection(m1, m2)는 좌측에 m1이 나타났을 때, 우측에 m2가 나타날 확률이다.

$$firstMorpheme(m) = first(tag(m))/totalWord$$

수식 2 형태소 m이 처음에 위치할 적합성

[수식 2]의 firstMorpheme(m)은 형태소 m이 어절의 처음 나타나도 되는지를 계산한다. tag(m)은 형태소 m의 태그를 뜻한다. 어절의 처음에 올 수 있는 명사 형태소의 종류가 매우 많기 때문에 적합성을 계산하기 위해 태그정보만을 이용하는 것이다. first()는 태그가 어절의 처음에 나타난 횟수이며, totalWord는 말뭉치에서 총 어절수다. 정리하면 firstMorpheme(m)은 해당 형태소 m의 태그가 말뭉치에서 어절의 처음으로 나타난 횟수를 모든 어절의 수로 나눈 값을 반환한다.

$$endMorpheme(m) = p(end(m)|p(m))$$

수식 3 형태소 m이 마지막에 위치할 적합성

[수식 3]의 endMorphmem(m)은 형태소 m이 어절의 마지막에 위치해도 되는지를 계산한다. end(m)는 형태소 m이 어절의 마지막일 확률이며 p(m)은 형태소 m이 나타날 확률이다. 정리하면 endMorpheme(m)은 형태소 m이 나타났을 때, 그 형태소가 어절의 마지막일 확률이다.

3. 형태소 분석 방법

3.1 분해와 검색 순서

2장의 방법대로 PWD를 구축하면, 어절의 꼬리부는 한번에 검색이 가능하다. 따라서 어절의 좌측에 오는 부분이 복합명사처럼 특별한 경우가 아닌 경우에 어절 전체를 좌우 2개로 나눈 뒤 검색하여 분석할 수 있다.

어절의 좌측에 위치하는 실질형태소들은 형식형태소에 비해 길이가 길고 분리될 경우 다른 의미를 가진 명사가 검색되기 쉽기 때문에 우측에서부터 우선적으로 분리를 시작한다. 예를 들어 '사람인가'에 경우 '사-람인가' 보다는 '사람인-가'를 먼저 시도한다. 즉, 모든 조합 방법을 찾기 보다는 분리 지점을 우측 끝에서 좌측으로 이동시켜가면서 분석하다가 정답을 찾았다고 판단될 때 중단한다.

실제로는 <표 2>의 '사람인*'과 '*인가'이 조합될 수 있는 것처럼, 음운 변화로 인해 *가 붙은 표층형끼리 조합될 수 있기 때문에 검색 순서는 <표 4>와 같다. <표 4>의 4번과 5번은 같은 분석 결과를 만들 것이며, 둘 다 정답이다. 이렇게 이 방법은 약간의 중복 후보를 만들 수 있으나 백트래킹의 문제는 일어나지 않는다.

<표 7> 분해와 검색 순서

순서	좌	우
1	전체 검색 - 사람인가	
2	사람인가*	*가
3	사람인	가
4	사람인*	*인가
5	사람	인가
6	사람*	*람인가
7	사	람인가
8	사*	*사람인가

조합 가능한 모든 방법을 찾기 보다는 실질형태소가 2개로 분리되는 오류를 일으키는 7, 8번이 되기 전에 우선적으로 만들어진 분석 후보들 안에서 정답을 찾는 것이 좋다. 따라서 분석 후보를 만드는 검색 조합이 일정 개수 성공하면 분석을 중단한다.

3.2 분석 후보 점수 계산

FWD을 구축할 때에는 각 해석 후보마다 말뭉치에서 나타나는 빈도를 저장하여 반환하고, 이 정보는 태깅 과정에서 매우 유용하게 사용될 수 있다. PWD를 통한 분석에서는 이런 빈도 값은 없으나 점수를 계산하여 빈도로서 사용할 수 있다.

분석 후보의 점수를 계산하기 위해서 기본적으로 좌측과 우측의 빈도를 사용할 수 있으며 2.4절의 3가지 수식을 사용할 수 있다. 간단하게 이 5가지 값을 모두 곱하여 점수를 계산한다.

[수식 4]에 m1은 분석 후보의 좌측 형태소, m2는 우측 형태소이며 편의상 2개의 형태소로만 구성된 경우를 예로든 것이다. freq(m)은 형태소(또는 부분 어절 분석)m의 빈도수다.

$$score(m1+m2) = freq(m1) \times freq(m2) \times connection(m1,m2) \times firstMorpheme(m1) \times endMorpheme(m2)$$

수식 4 분석 후보의 점수 계산식

이 점수는 태깅 단계에서 과분석으로 생긴 후보를 선택하지 않게 하기 위해 일부 후보를 삭제하는 용도로 사용할 수 있다.

3.3 복합명사 예외처리

복합명사가 어절에 포함된 경우 3.1의 방법으로는 분석 후보를 만들어내지 못할 수가 있다. 만약 만들어 낸다 하더라도 정답이 아닐 수 있으며 이런 경우 3.2의 계산법에 따라 점수가 지나치게 낮게 평가될 수 있다. 즉, 후보가 생성되지 못했거나, 점수가 임계점 이하일 경우 복합명사 처리를 시도한다.

복합명사처리는 매우 복잡한 문제로, 어절에서 어느 부분이 복합명사인가를 추측하는 것과 어떤 복합명사인지 분석하는 방법으로 나눌 수 있다. 본 논문에서는 간단하게 PWD를 사용한 분석 방법의 확장으로 복합명사를 처리하였다.

복합명사는 실질형태소이기 때문에 어절의 좌측에 위치한다. 예를 들어 '한국어처리인가'에 경우 먼저 우측의 '인가'를 찾고, 좌측의 '한국어처리'를 복합명사로 간주하여 분리할 수 있다. 이때 '인가'에 해당하는 우측은 이미 3.1의 방법을 통해 검색을 모두 마친 것을 그대로 사용하면 된다.

그렇게 복합명사 부분을 분리했다면, 좌측 부분을 다시 2개로 분리하여 PWD 검색을 시도한다. 이 때는 <표 4>와는 달리 중앙에서부터 분리한다. (<표 5> 참조)

검색에 성공한 경우 3.2와 비슷하게 점수를 계산한다. 좌측과 우측의 빈도를 곱하고 둘 사이의 connection() 값을 곱하여 계산한다. 이렇게 구해진 점수는 전체 어절의 좌측 빈도로 사용하여 전체 어절의 점수를 계산할 때 사용한다.

<표 9> 복합명사 분해와 검색 순서

순서	좌	우
1	한국어	처리
2	한국	어처리
3	한국어처	리
4	한	국어처리

3.1에서 분석을 중단하는 것과 마찬가지로, 모든 조합 가능한 방법을 시도하기보다는 일정 개수 조합에 성공하면 분석을 중단하는 것이 좋다.

4. 실험과 결과

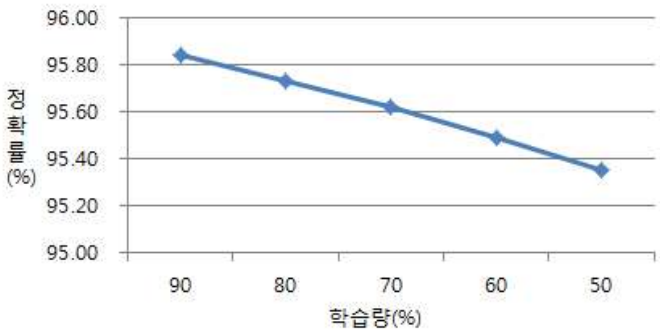
4.1 실험 환경

2장의 학습 데이터를 구축하기 위해 세종 말뭉치를 사용하였으며 전체 어절 수는 약 1,100만 어절이다. 세종말뭉치에 출현하지 않은 어휘들을 위해 추가적으로 표준국어대사전의 어휘를 포함하였다. 형태소 분석을 위해 세종 태그셋과 동형이의어 수준의 의미 번호를 사용하였으며, 태그와 동형이의어가 모두 같아야 정답으로 인정하였다.

실험용 학습 데이터를 구축하기 위해 말뭉치를 문장 단위로 구성한 뒤 10 문장 단위로 50~90%를 학습하고 나머지 10%를 테스트셋으로 남겨 실험에 사용하였다. 학습한 정보를 담은 File Data Base는 간단한 key/value 구조를 가진 다중 Hash Table을 사용하였다.

속도를 테스트하기 위한 컴퓨터는 모든 사전을 주 기억 장치에 저장하기 위한 충분한 용량을 가지고 있으며 CPU는 i7 860(2.8GHz)이고 단일 스레드만을 사용하였다. 실험 결과 110만 어절을 분석하는 동안 약 23초를 소요하였다.

태깅 정확률을 검사하기 위해서는 분석 후보 중 하나를 선택하는 태깅 모듈이 필요하며 본 실험에서는 은닉 마르코프 모델[9, 10]을 사용하여 어절 원형간의 전이 확률이 가장 높은 후보를 결정하였으며, 원형간의 전이 확률이 없는 경우에는 원형을 차등적으로 더 일반화(추상화) 시켜서 전이 확률을 계산하였다.



[그림 3] 학습량별 정확률

4.2 실험 결과

말뭉치의 90%를 학습한 뒤 형태소 분석기 자체의 재현율을 측정하기 위해 나머지 말뭉치 10%에서 기호가 없는 어절들 91만개를 분석하였다. 형태소 분석 후보에서 정답이 존재하는지를 측정하여 재현율을 구했으며 실험 결과 약 99.05%의 재현율이 나왔다.

학습량 별로 태깅 정확률을 측정한 결과가 [그림 3]이다. 90%를 학습한 상태에 정확률은 95.84%이고, 50% 학습과 정확률 차이는 약 0.49%로 크지 않다. 90%를 학습했을 때 테스트셋에서 기호를 제외하면 정확률은 96.76%로 기호를 포함했을 때보다 더 높다. 이 차이의 주원인은 말뭉치의 기호 태그 오류이며 일부는 프로그램의 오류다.

4.3 학습 데이터 크기

학습량에 따른 학습 데이터의 크기가 <표 6>에 나타나 있다. 50%에서 90%까지 학습량이 증가하면서 데이터 크기는 269MB에서 321MB로 약 19% 상승하였다.

<표 10> 학습량에 따른 학습 데이터 크기

학습량(%)	90	80	70	60	50
크기(MB)	321	303	294	279	269

4.4 오류 분석

오류의 대부분은 정답이 복합명사이거나 표준국어대사전에 등록되지 않은 명사 형태소를 포함하는 경우가 많았다. 그리고 정답 자체가 오류를 가진 경우도 많았다. 일부는 재현에 성공했으나 과분석 오류를 방지하기 위해 삭제하는 부분에 의해 삭제되었다.

오류 유형의 표본 검사를 위해 오류 중에서 FWD에서 전체 어절이 검색되지 않았고, 기호를 포함하지 않는 경우 50개를 무작위로 선택하였다. 이 중에 정답 자체에 오류가 없으며 미등록어가 아닌 경우는 14개로 28%였으며, 14개 중에서 6개는 복합명사 또는 접두사, 접미사가 포함되어서 오류가 발생한 것이다. 직접적인 오류의 원인은 정답이 될 조합이 형태소간의 connection 계산에 의해 낮은 점수가 나와 후보에서 삭제된 것이다.

5. 결론

오랫동안 연구된 한국어 형태소 분석 분야에서 비교적 최근에 연구된 무분석 방법은 형태소 분석 단계에서 알고리즘이 간단하지만 사전 구축이 힘들다는 단점이 있었다. 본 논문에서는 이를 해결하기 위해 세종형태의미 말뭉치를 통해 사전을 구축하는 방법을 제안하였고, 실험결과 초당 약 4만 8천 어절을 분석하는 빠른 속도와 형태소의 의미번호를 포함하고도 99.05%의 높은 재현율을 보였다. 그리고 형태소 분석의 마지막 단계인 태깅에서 정확률이 96.76%였다. 일부 말뭉치의 오류가 있기 때문에 실제 재현율과 정확률은 더 높을 것이다.

무분석 방법은 붙여 쓰기 오류에도 안정적인 분석을 하였으며, 분석이 힘든 보조용언도 잘 처리하였고, 복합명사처리에서 일부는 조사와 붙어 있는 명사를 한 번에 분석함으로 복합명사 처리 모듈까지 가지 않기도 했다.

말뭉치를 기반으로 부분 어절 기분석 사전을 구축하여 실제로 사용되는 형식 형태소 조합의 대부분이 사전에 등록되었고, 자주 출현하는 붙여쓰기의 대상들이 등록되었다. 이런 특징 때문에 더 유연하고 인간적인 분석이 가능해져 높은 정확률과 빠른 속도가 나타난 것으로 분석된다.

제안하는 사전 구축 방법은 다양한 부분 어절 정보를 수집하기 때문에 학습 데이터의 용량이 300MB보다 크다. 이것은 기존의 알려진 모든 형태소 분석기들 보다 많은 용량이 필요하다는 의미지만 현재의 컴퓨터 기술로는 시스템의 주 기억 공간에 사전 전부를 쉽게 저장할 수 있다. 그리고 사전의 용량이 학습량에 따라 증가하는 폭이 크지 않으며 재현율이 높기 때문에 사전의 크기는 문제가 되지 않을 것이다.

무분석 방법은 그 핵심이 되는 사전을 수정함으로써 유지보수가 쉽다고 알려져 있다[1, 8]. 본 논문이 제안하는 PWD의 구조도 단순하기 때문에 유지보수가 쉽지만, 근본적으로 말뭉치를 학습하는 방법을 사용하기 때문에 말뭉치를 관리하는 더 쉬운 방법으로 유지보수를 할 수 있다. 만약 미등록어를 등록하겠다면 말뭉치에 해당 단어를 추가하면 된다. 이때는 단순히 해당 형태소의 표층형과 원형을 등록하면 된다.

태깅 실험 결과 일부 분석 오류는 학습 말뭉치의 오류로 인한 것 이었다. 즉, 본 논문이 제안하는 말뭉치 기반의 학습 방법은, 말뭉치의 오류로 인해 정확률이 낮아질 수 있다는 단점을 가지고 있다.

나머지 오류들은 대부분 미등록어나 복합명사로 인한 문제였다. 이를 해결하기 위해서는 더 많은 명사를 등록해야 하지만 현실적으로 계속 생겨나는 명사를 말뭉치를 통해 등록하는 것은 어려우며, 사용자 사전을 추가하는 방법을 차후 연구해볼 필요가 있다. 또한 복합명사를 더 정확하게 분석하기 위해서 복합명사 처리 모듈의 수정이 필요하며, 명사간의 connection을 계산하기 위해 어휘망 등 추가적인 언어 자원을 이용하는 연구를 해야 할 것이다.

참고 문헌

[1] 양승현, 김영섬, "부분 어절의 기분석에 기반한 고속 한국어 형태소 분석 방법", 정보과학회논문지 : 소프트웨어 및 응용, 제27권, pp. 290-301, 2003.3

[2] 김성용, 최기선, 김길창, “Tabular Parsing 방법과 접속 정보를 이용한 한국어 형태소 분석기”, 한국정보과학회 춘계 인공지능발표논문집, pp. 133--147, 1987.

[3] 최재혁, 이상조, “양방향 최장일치법을 이용한 한국어 형 태소 분석기”, 한국정보과학회 봄 학술발표논문집, Vol.20, No. 1, pp. 769--772, 1993.

[4] 한용기, 이근용, 이기오, 이용석, "한국어 형태소 분석에서 확장된 최장 일치법을 이용한 의사 투-레벨 모델", 한글 및 한국어 정보처리 학술대회, pp. 491-496, 1999.10

[5] 최재혁, 이상조, “양방향 최장일치법을 이용한 한국어 형태소 분석기에서의 사전 검색 횟수 감소 방안”, 한국정보과학회 논문지, Vol. 20, No. 10, 1993.

[6] 강승식, 김영택, “사전 정보에 기반한 효율적인 한국어형태소 분석기의 설계 및 구현”, 한국정보과학회 봄 학술발표논문집, Vol. 18, No. 1, 1991.

[7] 김재한, 옥철영, “통합형태소를 이용한 한국어 형태소 분석기”, 한국정보과학회 가을 학술발표논문집, Vol 21, No. 2, pp. 653--656, 1994.

[8] 심광섭, "MADE : 형태소 분석기 개발 환경", 인터넷정보학회논문지, 제8권 제4호, pp. 159-171 2007.8

[9] 신중호, 한영석, 박영찬, 최기선, "어절구조를 반영한 은닉 마르코프 모델을 이용한 한국어 품사태깅", 한글 및 한국어정보처리 학술대회, pp. 389-394, 1994.10

[10] 김동명, “HMM을 이용한 한국어 품사·동형이의어

동시 태깅 시스템", 울산대학교 일반대학원 석사학
위논문, 2009