한국어 형태소 분석

컴퓨터정보학부 심광섭

1. 형태론

◆ 형태소(Morpheme)

- 더 이상 분해될 수 없는 최소의 뜻 단위
  분해하면 의미를 잃어버림.

- 이형태: 하나의 형태소가 환경에 따라 모습을 달리할 때 그것들을 그 형태소의 이형태라 함.
  (예) 이/가 : 음운론적 환경으로 제약된 이형태
  (예) 었/았/였 : 형태론적 환경으로 제약된 이형태

1. 형태론

◆ 형태 자소론
- 두 형태소의 경계에서 발생하는 철자의 변형
  (예) get+s = gets, go+s = goes, try+s = tries
  (예) 막(block)+은 = 막은, 갈(grind)+은 = 간,
  가(go)+은 = 간

◆ 형태 통사론
- 어떤 형태소 배열이 올바른 어절을 형성하는가?
  (예) fragment-al, *employment-al
  (예) 동작(action)+ 하+다, *정보(information)+하+다

2001-7-25

3

2. 형태소 분석

◆ 형태소 분석
- 표층 어절을 어휘층 형태소로 분해하는 작업
  (예) "나는" →
  나/대명사 + 는/조사
  나/동사 + 는/어미
  날/동사 + 는/어미

◆ 형태소 생성
- 어휘층 형태소들로부터 표층 어절을 생성하는 작업
  (예) 분석/명사 + 하/동사화접미사 + 어/어미
  → "분석하여" / "분석해"

2001-7-25

4

2. 형태소 분석

◆ 형태소 분석의 요소
- 사전
  - ◆ 표제어(형태소) + 형태소 정보(품사, 자질)
- 문법
  - ◆ 형태 자소 규칙: 음운 현상을 표현
  - ◆ 형태 통사 규칙: 형태소 결합의 합법성을 표현
- 분석 알고리즘
  - ◆ 전처리: 한글 코드 변환, 한글 및 기호 분리
  - ◆ 형태 자소 규칙 적용: 형태 분리, 원형 복원
  - ◆ 사전 탐색: 형태소 정보 부여, 미등록 형태소 추정
  - ◆ 어절의 형태소 분석 후보 생성: 형태소 정보 결합
  - ◆ 형태 통사 규칙 적용: 비문법적인 형태소 결합 제거

2. 형태소 분석

◆ 한국어 형태소 분석 과정
- 한글 코드 변환
  - ◆ 형태소 변형을 처리하는 데는 조합형 한글 코드가 편리
  - ◆ 완성형, 조합형의 상호 변환
- 한글 및 기호 분리
  - ◆ 한글 문자열과 기타 문자열(영문자, 숫자, 기호 등)을 분리
    (예) 클린턴(clinton)은 = 클린턴+(+clinton+)+은
- 형태 분리
  - ◆ 한글 문자열을 단순 분리
    (예) 나는 = 나+는, 나+느+ㄴ, 나느+ㄴ

2. 형태소 분석

◆ 한국어 형태소 분석 과정 (계속)

- **원형 복원 (형태자소규칙 적용)**

(예) 나+는 → 나+는, 날+는
나+느+ㄴ → 나+늘+ㄴ
나느+ㄴ → 나늘+ㄴ

- **형태소 정보 부여 및 미등록 형태소 추정**

나/동사|대명사 날/동사|명사 느/비형태소 늘/동사|부사	나늘/비형태소 나느/비형태소 는/조사|어미 ㄴ/조사|어미

2. 형태소 분석

◆ 한국어 형태소 분석 과정 (계속)

- **형태소 분석 후보 생성**
  - ◆ 각 형태소 정보를 결합하여 분석 후보 생성

나/동사 + 는/조사	나/동사 + 는/어미
나/대명사 + 는/조사	나/대명사 + 는/어미
날/동사 + 는/조사	날/동사 + 는/어미
날/명사 + 는/조사	날/명사 + 는/어미
나/동사 + 늘/동사 + ㄴ/조사	나/동사 + 늘/동사 + ㄴ/어미
나/대명사 + 늘/동사 + ㄴ/조사	나/대명사 + 늘/동사 + ㄴ/어미

2. 형태소 분석

◆ 한국어 형태소 분석 과정 (계속)

– 비문법적인 형태소결합 제거 (형태통사규칙 적용)

동사+조사	나/동사 + 는/조사, 날/동사 + 는/조사, 늘/동사 + ㄴ/조사
대명사+어미	나/대명사 + 는/어미
동사+동사	나/동사 + 늘/동사
대명사+동사	나/대명사 + 늘/동사

나/동사 + 는/어미
나/대명사 + 는/조사
날/동사 + 는/어미

2.1 형태소 분석 사전

◆ 형태 자소 규칙에 의해 어절로부터 분리된 문자열을 형태소로 인식하기 위해 사전 필요

– 사전 검색 이전 : 문자열 = 형태소 후보 (미확인상태)
– 사전 검색 성공 : 문자열 = 형태소
– 사전 검색 실패 : 문자열 ≠ 형태소 (완전한 사전인 경우)

❖완전한 사전: 미등록 형태소와 정보가 존재하지 않음

2.1 형태소 분석 사전

◆ 미등록 형태소
- 미등록 형태소는 필연적으로 존재
  - ◆ 언어는 생성/발전/소멸하므로 완전한 사전을 만들 수 없음
- 미등록 형태소 문제
  - ◆ 올바른 어절에 대하여 분석 실패: 차이코프스키에게서
- 미등록 형태소 추정
  - ◆ 사전 검색 실패 문자열 = 미등록 형태소 ? / 비형태소 ?

    차이코프스키에게서

    차이코프스키에게서로서는을

2.2 형태소 분석 문법

◆ 형태소 분석 문법
- 형태소 분석을 위한 규칙의 집합
  - ◆ 형태 자소 규칙
  - ◆ 형태 통사 규칙

◆ 형태소 분석 오류
- 미분석 : 문법의 적용 범위가 좁아 합법적인 해석이 누락
- 과분석 : 문법의 적용 범위가 넓어 비합법적인 해석이 포함
- 오분석 : 문법의 정확성 부족

❖ 모든 현상을 정확하게 표현하는 문법 필요

3. 형태소 분석 방법

◆ 문자열 분리 방식 : 우좌분석

– 문자열의 우측부터 분리 (형식형태소 우선)

(예) 형태소분석하고
→ 형태소분석하+고
→ 형태소분석+하+고
→ 형태소+분석+하+고
→ 형태소+분석+하+고

– 한국어는 형식형태소가 매우 발달

– 문자열 좌측의 미등록 형태소 추정에 효과적
(예) 클린턴이

3. 형태소 분석 방법

◆ 문자열 분리 방식 : 좌우분석

– 문자열의 좌측부터 분리 (실질형태소 우선)

(예) 형태소분석하고
→ 형태소+분석하고
→ 형태소+분석+하고
→ 형태소+분석+하+고
→ 형태소+분석+하+고

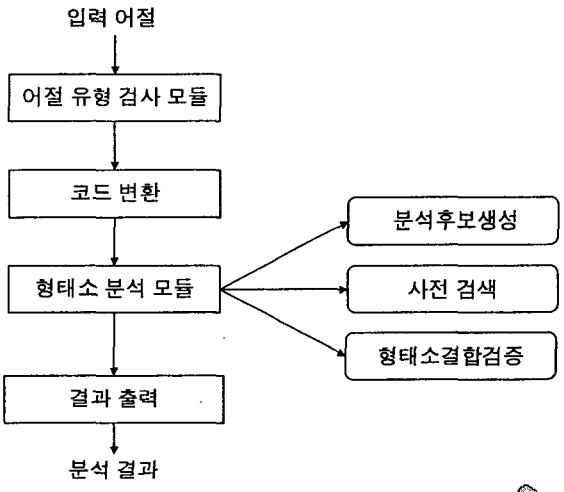
– 일반적인 문자열 판독과 동일한 방향

3. 형태소 분석 방법

◆ 문자열 분리 방식 : 양방향분석
- 우좌분석과 좌우분석을 동시에 수행
- 문자열 중간의 미등록 형태소 추정에 효과적

(예) 한글프로그램개발

3. 형태소 분석 방법



3.1 분석 후보 생성

◆ 불규칙 활용, 축약, 탈락 현상 등 형태론적 변형이 일어난 형태소의 원형을 복원

(예) 어절 "가는"의 분석 후보

(1) 가는 (독립언 후보)
(2) 가 + 는 (체언 + 조사 후보)
(3) 가 + 는 (용언 + 어미 후보)
(4) 갈 + 는 (ㄹ 탈락 후보)
(5) 가느 + ㄴ (용언 + 어미 후보)
(6) 가늘 + ㄴ (ㄹ 탈락 후보)
(7) 가늫 + ㄴ (ㅎ 변형 후보)

3.2 사전 검색

◆ 사전 검색을 통해 분석 후보 중 비형태소 제거

(1) 가는 → **가는/비형태소**
(2) 가 + 는 → 가/NN + 는/JO
(3) 가 + 는 → 가/VV + 는/EM
(4) 갈 + 는 → 갈/VV + 는/EM
(5) 가느 + ㄴ → **가느/비형태소** + ㄴ/EM
(6) 가늘 + ㄴ → 가늘/AJ + ㄴ/EM
(7) 가늫 + ㄴ → **가늫/비형태소** + ㄴ/EM

3.3 형태소 결합 검증

◆ 형태 통사 규칙을 적용하여 두 형태소 간의 문법적 적합성 검사 (접속정보표 이용)

- 어절의 시작과 끝으로 올 수 있는 형태소 지정이 필요
  - ◆ 시작 가능한 형태소 : 명사, 동사어간, 접두사, . . .
  - ◆ 끝에 올 수 있는 형태소 : 명사, 접미사, 조사, 어말어미, 수사, . .
- 형태 통사 규칙에 어긋나는 결합은 제거
  (예) 감기는
  - ◆ 감기+는 : NN+JO, VV+EM (적합)
  - ◆ 감+기+는 : VV+EF+JO (적합)
  - ◆ 감+기는 : VV+EM (적합)
  - ◆ 가+ㅁ+기+는 : VV+EM+EM+JO (부적합)
    - EM+EM은 허용되지 않음

3.4 기존 방법의 문제점

◆ 분석 후보 생성 후 부적격 후보 제거의 비효율성

- 분석 후보 생성
  원형 복원 규칙을 적용하는 과정의 비효율성
  모든 어절에 대하여 여러 가지 조건의 적용 여부를 테스트해 보아야 함

  (예) "가는" → 가는, 가+는, 갈+는, 가느+ㄴ, 가늘+ㄴ, 가늘+ㄴ

- 부적격 후보 제거
  부적격 후보의 제거를 위해 사전 검색 횟수가 증가

  → 사전 검색 횟수를 줄이는 것이 관건
  음절 정보를 이용한 사전 검색 횟수 감소 방안

3.4 기존 방법의 문제점

◆ 코드 변환으로 인한 비효율성
- 어미에 포함된 'ㄴ/ㄹ/ㅁ/ㅂ'나 '아/어' 등은 어간과 결합하여 어간의 형태를 변형시키기 때문에 원형 복원 과정에서 자소별 조작이 필요
  → 완성형 코드로 주어진 어절을 조합형으로 변환
  → 형태소 분석이 완료된 후 다시 최초의 완성형 코드로 변환

- 입력 어절 중 위와 같은 이유로 자소별 조작을 해야 하는 경우는 일부에 지나지 않으나 어절 전체를 대상으로 일괄적인 코드 변환을 실시함.

  (예) 아름다운 → 아름답 + 은

4. 인접 조건 검사에 의한 분석

◆ 음절을 경계로 한 분석
◆ 음운 제약 조건
◆ 형태 제약 조건
◆ 품사 제약 조건
◆ 인접 조건 검사
◆ 형태소 분석 사전 구축
◆ 구현 사례

4.1 음절을 경계로 한 분석

◆ 독립언 (하나의 단어가 하나의 어절을 이루는 유형)
한 번의 사전 탐색만으로 분석이 종료됨

◆ 체언 + 조사 유형
(1) 학교에서 → 학교/NN + 에서/JO
(2) 학생은 → 학생/NN + 은/JO

◆ 용언 + 어미 유형
(1) 예쁘니까 → 예쁘/AJ + 니까/EM
(2) 기다리다 → 기다리/VV + 다/EM
(3) 그립다고 → 그립/AJ + 다고/EM
(4) 믿다 → 믿/VV + 다/EM

위의 두 유형에 대해서는 두 번의 사전 탐색으로 분석이 종료됨

4.1 음절을 경계로 한 분석

◆ 효율적인 분석
음절을 경계로 형태소 분석을 한다면 자소 단위의 연산이 불필요
→ 코드 변환 과정이 불필요
→ 형태소 분석을 사전 탐색에 의한 단어 인식 문제로 단순화

(1) 학교에서 → 학교/NN + 에서/JO
(2) 예쁘니까 → 예쁘/AJ + 니까/EM
(3) 기다리다 → 기다리/VV + 다/EM

- 좌우 분석 또는 우좌 분석

4.2 음운 제약 조건

◆ 음절을 경계로 한 분석의 문제 1

(1) 먹는 → 먹/NN + 는/JO
(2) 감는 → 감/NN + 는/JO

조사나 어미 중에는 선행하는 형태소에 대하여 음운론적 환경에 대한 제약을 가하는 것들이 있다.

조사 : 마지막 음절의 종성 자음 유무
어미 : 마지막 음절의 종성 자음 유무, 모음 유형 (양성/음성)

따라서 형태소 사전에 이러한 제약 조건을 나타낼 필요가 있다.

4.2 음운 제약 조건

◆ 문제 1의 해결을 위한 사전 구조

(1) 음운 제약 조건의 표현

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

이형태를 대표형으로 나타내고자 하는 경우
어라 : ([아라/EM] @(+음성))

4.2 음운 제약 조건

◆ 문제 1의 해결을 위한 사전 구조

(2) 음운 정보의 표현

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

2001-7-25

27

4.2 음운 제약 조건

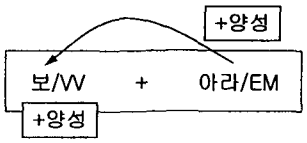
◆ 음운 제약 조건을 적용한 예

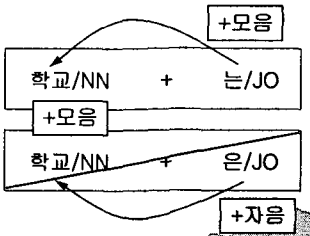
(1) 보아라

보 : ([보/VV] +양성 +모음)
아라 : ([아라/EM] @(+양성))

(2) 학교는 / 학교은

학교 : ([학교/NN] +모음)
는 : ([는/JO] @(+모음))
은 : ([은/JO] @(+자음))





2001-7-25

28

4.3 형태 제약 조건

◆ 음절을 경계로 한 분석의 문제 2

(1) 그린다 → 그리/VV + ㄴ다/EM
(2) 돌본다 → 돌보/VV + ㄴ다/EM

어미 중에는 선행하는 용언의 제일 마지막 음절과 결합하여 어간의 형태를 변형하는 것이 있다 → 원형 복원이 필요?

'돌본'을 용언으로 형태소 사전에 등재
'다'를 어미로 형태소 사전에 등재

둘 사이의 결합 관계를 잘 나타낼 수 있으면 음절을 경계로 한 분석 가능

4.3 형태 제약 조건

◆ 문제 2의 해결을 위한 사전 구조 (확대된 사전)

(1) 용언의 원형 : 가, 돌보, ...
(2) 용언 + ㄴ 결합형 : 간, 돌본, ...
(3) 용언 + ㄹ 결합형 : 갈, 돌볼, ...
(4) 용언 + ㅁ 결합형 : 감, 돌봄, ...
(5) 용언 + ㅂ 결합형 : 갑, 돌봅, ...
(6) 용언 + ㅆ 결합형 : 갔, 돌봤, ...

4.3 형태 제약 조건

◆ 형태 제약 조건의 표현 (확대된 형태소 사전)

게 : ([ㄹ게/EM] @(+ㄹ))
다 : ([다/EM] @(+어간 | +ㅆ)) ([ㄴ다/EM] @(+ㄴ))
세 : ([ㅁ세/EM] @(+ㅁ))
니다 : ([ㅂ니다/EM] @(+ㅂ))

+어간, +ㄴ, +ㄹ, +ㅁ, +ㅂ, +ㅆ 등은 표제어의 왼쪽에 올 수 있는 문자열의 형태를 제약하는데 사용되는 기호

4.3 형태 제약 조건

◆ 형태 정보의 표현 (확대된 형태소 사전)

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)

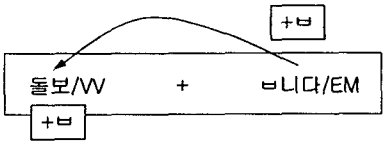
4.3 형태 제약 조건

◆ 형태 제약 조건을 적용한 예

(1) 돌봅니다

돌봅 : ([돌보/VV] +ㅂ)

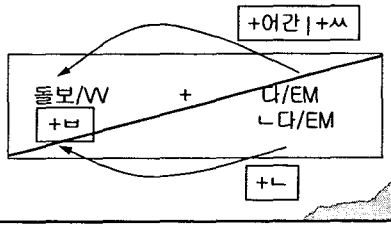
니다 : ([ㅂ니다/EM] @(+ㅂ))



(2) 돌봅다

돌봅 : ([돌보/VV] +ㅂ)

다 : ([다/EM] @(+어간 | +ㅆ)) ([ㄴ다/EM] @(+ㄴ))



4.3 형태 제약 조건

◆ 형태 제약 조건을 적용한 예 (다중 결과 생성)

(1) 갑니다

가 : ([가/VV] +어간 +양성 +모음)

갈 : ([갈/VV] +어간 +양성 +모음)

갑 : ([가/VV] +ㅂ) ([갈/VV] +ㅂ)

니다 : ([ㅂ니다/EM] @(+ㅂ))

가/VV	+	ㅂ니다/EM
갈/VV	+	ㅂ니다/EM

4.3 형태 제약 조건

◆ 형태 제약 조건을 이용한 아/어/으 문제 해결 예

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)

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

보아도, 봐도, 사도, 먹어도, 보고, 사고, 먹고, ... (분석 됨)
*봐아도, 사아도, 봐고, 먹도, ... (분석 안됨)

4.3 형태 제약 조건

◆ 형태 제약 조건을 이용한 ㄷ/ㅂ/ㅅ 불규칙 문제 해결 예

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)

듣고, 돕고, 긋고
들어도, 도와도, 그어도
...

아도 : ([아도/EM] @(+아 +양성))
어도 : ([어도/EM] @(+아 +음성))
도 : ([아도/EM] @(-아 +양성)) ([어도/EM] @(-아 +음성))
고 : ([고/EM] @(+어간))
와도 : ([아도/EM] @(+와 +양성))
워도 : ([어도/EM] @(+와 +음성))

4.4 품사 제약 조건

◆ '는/도/만'과 같은 보조사는 체언뿐만 아니라 부사와도 사용됨.

학교/NN + 는/JO	빨리 /AD + 는 /JO
학교 /NN + 도 /JO	빨리 / AD + 도 /JO
학교 /NN + 만 /JO	빨리 / AD + 만 /JO

◆ 어미 중에는 동사나 형용사에서만 사용되는 것도 있음.

아름답/AJ + 거나/EM	듣/VV + 거나/EM
아름답/AJ + 는군요/EM	듣/VV + 는군요/EM
아름답/AJ + 군요/EM	듣/VV + 군요/EM

4.4 품사 제약 조건

◆ 품사 제약 조건 표현

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))

어미 '-거나'나 '-군요'는 '사-', '아름답-' 등과 같은 어간이나 'ㅆ'과 결합된 '샀-', '-았-', '-었-', '-왔-', '-웠-' 등의 뒤에 올 수 있다.

사거나, 아름답거나, 아름답군요, 샀거나, 아름다웠이거나, ...

4.4 품사 제약 조건

◆ 품사 정보의 표현

하 : ([하/SV] #(NN) +어간 +양성 +모음)
했 : ([하/SV + 었/EP] #(NN) +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 +음성)

4.4 품사 제약 조건

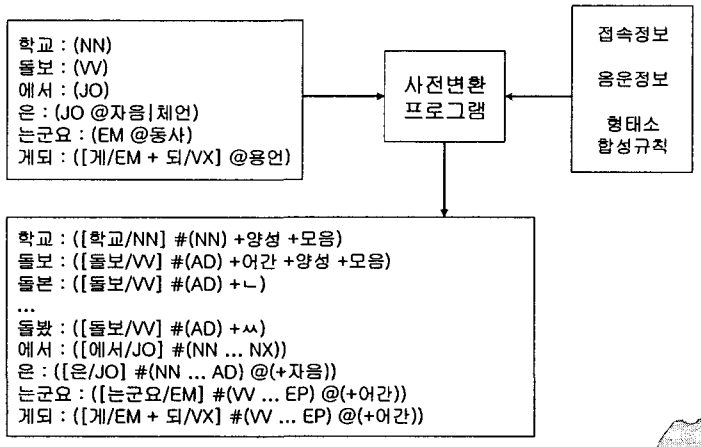
◆ 품사 제약 조건을 적용한 예

하 : ([하/SV] #(NN) +어간 +양성 +모음)
했 : ([하/SV + 었/EP] #(NN) +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 +음성)

다 : ([다/EM] @(+어간 | +ㅆ)) ([ㄴ다/EM] @(+ㄴ))
거나 : ([거나/EM] @(VV VX SV AJ AX SJ EP) @(+어간 | +ㅆ))

연구하다 : 연구/NN + 하/SV + 다/EM
연구했거나 : 연구 /NN + 하 /SV + 었/EP + 거나/EM
연구하시거나 : 연구 /NN + 하 /SV + 시/EP + 거나/EM
연구하셨거나 : 연구 /NN + 하 /SV + 시/EP + 었/EP + 거나/EM

4.5 형태소 분석 사전 구축



4.5 형태소 분석 사전 구축

- 음운/형태/품사 제약 조건을 간단하게 나타내기 위하여 다음과 같은 심볼을 사용함.

  @용언, @동사, @형용사, @자음|체언, @모음|체언, @ㅂ불|용언, @ㅂ불|동사, @ㅂ불|형용사, @아|용언, @아|동사, @아|체언, ...

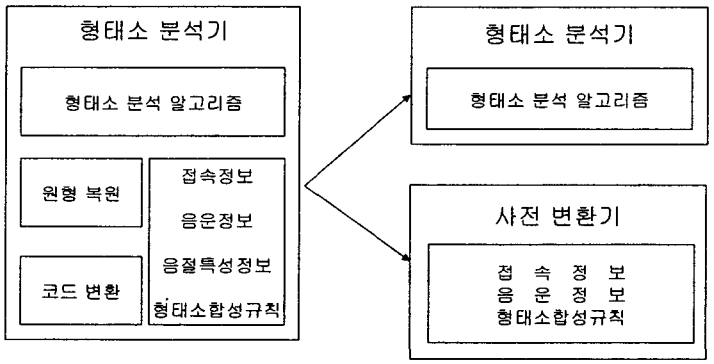
- 복합 조사/어미의 처리 방법

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

  에서부터는 : ([에서부터는/JO] ...)
  라고까지는 : ([라고까지는/EM] ...)

- 기타 다른 복합어의 처리 방법

4.6 방법론 비교



4.7 구현 사례

- **MACH (Morphological Analyzer for Contemporary Hangul)**
  - C++로 구현
  - 1.13GHz Pentium III 급의 PC에서 1GB의 문서를 분석하는데 5분이 걸리며 이는 초당 약 45만 어절을 분석하는 것임.

    N.B. 333MHz Pentium II 급의 PC에서 1GB의 문서를 분석하는데 172분이 걸림.

    N.B. 450MHz Pentium III 급의 PC에서 명사 추출기를 수행시켰을 때 초당 4.3만 어절을 분석함.

5. 품사태깅

- 형태론적 중의성
  - 하나의 표층 어절이 두 개 이상의 형태소분석결과를 가지는 경우
    (예) "나는"
    나/대명사 + 는/조사
    날/동사 + 는/어미
    나/동사 + 는/어미
- 형태론적 중의성 해소
  - 주어진 문맥 내에서는 단지 하나의 형태소 분석 결과만 가진다.
    (a) *나는* 오늘 병원에 가야 합니다.
    (b) 하늘을 *나는* 게 제 꿈입니다.
    (c) 싹이 *나는* 것을 보니 봄이 왔군요.

6. 형태소 분석의 응용 분야

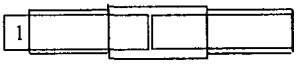
- 자연어 처리
  - 구문 분석의 하위 모듈
  - 단어 의미 중의성 해결
  - 기계 번역(Machine Translation)
- 정보 처리
  - 정보 검색
  - 자동 요약
- 전처리 혹은 후처리
  - 맞춤법(철자/띄어쓰기) 오류 검사 및 교정
  - 문자 인식 또는 음성 인식의 후처리
- 언어 정보 획득
  - 용례, 통계정보 추출

7. 주요 한글 코드

- 상용 조합형 코드
- 완성형 KSC-5601
- 유니코드 한글

7.1 상용 조합형 코드

- 한글의 한 음절을 2byte로 표현
  - 코드 당 2바이트, 첫바이트의 MSB = 1(한글), 0(아스키)
  - 5비트 초성, 5비트 중성, 5비트 종성



- 12,320 가능한 음절을 모두 표현
  - 완전한 현대 한글 음절 11,172자를 모두 표현
- 음절 구조 반영
  - 초성, 중성, 종성 정보를 코드로부터 직접 계산 가능
- 한국어 정보 처리(특히, 형태소분석)에 적합한 코드임

7.2 완성형 코드 (KSC-5601)

- 한글의 한 음절을 2byte로 표현
  - 코드 당 2바이트, 첫바이트의 MSB = 1(한글 코드임을 나타냄)
- 한글은 자주 쓰이는 2,350자를 가나다 순으로 배치

가 = 0xB0A1	겨 = 0xB0DC
각 = 0xB0A2	격 = 0xB0DD
간 = 0xB0A2	겪 = 0xB0DE
갇 = 0xB0A3	견 = 0xB0DF

- 단점
  표현할 수 없는 글자가 많고, 한글 자모의 분리가 어려움

7.3 유니코드 (Unicode)

- Unicode 2.0에서 추가된 한글 표현 방법
- 조합될 수 있는 현대 한글 모두를 가나다 순으로 정렬, 배치한 것
- 조합 가능한 글자의 수
  - 19(초성) x 21(중성) x 28(종성) = 11,172자
- 현대 한글을 모두 표현할 수 있으며 정렬에도 아무 문제가 없음
  - 고어 한글을 표현할 수 없음
- JAVA, Linux 등에서 지원