

# TTS 시스템을 위한 한국어 발음열 자동 생성<sup>1)</sup>

차선화, 정민화

서강대학교 전자계산학과

## Automatic Generation of Korean Pronunciation Variants for TTS System

Sunhwa Cha and Minhwa Chung

Department of Computer Science, Sogang University

{shcha,mchung}@npeng.sogang.ac.kr

### 요약

본 논문에서는 음성합성 시스템의 한 모듈로서 한국어 문자열을 음소열로 자동 변환하는 시스템을 구현하였다. 문자열을 음소열로 변환할 때에는 한국어 음운현상에 대한 체계적인 분석 과정이 필요하다. 한국어의 음운 변화 현상은 단일 형태소 내부와 여러 형태소가 결합하여 한 어절을 이루는 경우 그 형태소 경계, 그리고 어절 경계에서 서로 다른 음운 규칙이 적용된다. 따라서 어절이나 문장 등의 입력을 음소열로 변환하기 위해서는 형태소 분석, 태깅작업이 반드시 수행되어야 올바른 발음열을 유도할 수 있다. 본 논문에서 제안한 시스템은 한국어의 형태음운현상을 반영하기 위해 형태소 분석을 선행한 후, 한국어에서 빈번하게 발생하는 음운 변화 현상의 분석을 통해 정의된 음소 변동 규칙과 변이음 규칙을 선택적으로 적용하여 형태소, 어절, 언절 또는 문장 등의 다양한 형태의 입력에 대해 발음열을 생성한다. 기존의 연구에서 분리되어 있던 형태소 태깅과 변환 시스템을 통합하여 사용자 편의성을 높였으며 텍스트 기반의 형태소 분석기를 사용하기 때문에 원형이 복원되는 형태소들에 대한 처리 루틴을 두어 오류를 감소 시켰다.

### 1. 서론

음성 합성 시스템에 사용되는 문자열-발음열 변환 모듈은 문장 또는 언절 단위의 텍스트를 입력으로 받아 그에 맞는 발음열을 생성해내야 한다. 이 때 문자열에 대한 올바른 발음열을 생성하기 위해서는 해당 언어의 음운 현상에 대한 체계적이고, 정확한 분석이 필요하다. 한국어의 음운 변화는 음소의 배열과 형태소의 종류에 크게 영향을 받는다. 같은 음소의 배열이라 하더라도 그 음소열이 하나의 형태소 내부에 있는가, 형태소와 형태소가 결합하여 한 어절을 이룰 때 형태소간의 경계에 위치하는가, 또는 어절들이 결합하여 언절을 이룰 때 그 경계에 위치하는가에 따라 각기 다른 음운 변화 현상을 보여주고 있다. 예를 들어 '신고'라는 어절이 명사인 경우 형태소 내부의 'ㄴ-' 음소열은 아무런 음운 변화 규칙도 적용 받지 않아 /SHIYNGO(신고)로 발음되지만, '신+고'라는 어간과 어미의 결합인 경우 형태소 경계를 이루는 'ㄴ-' 음소열에 경음화 규칙이 적용되어 /SHIYNKKO(신포)로 발음된다. 또 '겨울 나그네(겨울라그네)'와, '너는 산을, 나는 바다를(사늘, 나는)'에서처럼 한 어절을 이루는 어절간의 경계에서는 'ㄹ-'의 음소열이 유음화되어 'ㄹ'로 변화하지만,

<sup>1)</sup> 본 연구는 한국과학재단의 핵심전문연구비 지원 (과제번호:961-0904-025-2)에 의하여 이루어졌음.

## TTS시스템을 위한 한국어 발음열 자동 생성

그렇지 않은 경우는 아무런 변화도 없다[4][5][6]. 이러한 한국어의 특징을 잘 반영하여 발음열을 생성하려면 주어진 문장을 형태소 분석하고, 올바른 형태소열로 태깅하여 그 정보를 이용해야 한다.

이러한 한국어의 형태 음운론적 특성을 반영하기 위해 [1][2][3]의 연구에서 형태소 분석된 결과를 이용하여 문자열을 발음열로 변환하는 연구가 이루어져 왔다. [3]에서는 형태소-음소열 맵핑치를 구성하여 그로부터 음운 변환 규칙인 CCV 규칙을 생성해냈다. 형태소 분석 결과를 이용하여 형태소-음소열 메타사전으로 형태소 관계에서의 음운 현상을 추정하고, 형태소 내부에서는 CCV 규칙을 적용하여 발음열을 생성해낸다. 그러나 특정 영역의 코퍼스에서 규칙을 추출했기 때문에 새로운 영역으로 확장할 때에는 수작업으로 새로운 영역의 코퍼스를 구축해야 하는 번거로움이 있다. 또한 예외 발음을 규칙으로 추출한다면, 오류를 남게 된다. 반면 [1][2]에서는 한국어의 특징을 형태음운론적으로 분석하여 그 규칙을 정리하고 각 양상별로 분류한 정보를 이용하여 발음열을 자동으로 변환하였다. 그러나 [2]의 시스템은 형태소 태거와 발음열 변환기가 서로 분리되어 있어 발음열 변환을 위해서 사용자는 먼저 형태소 태깅을 수행한 결과를 입력해야 했다. 이 변환 시스템에서는 KTS나 ETRI 태그[9]를 이용하므로 다른 태그들 사용하는 형태소 태거는 이용할 수 없다는 약점이 있다. 또한 기존의 태거들은 형태소 분석을 수행하면 텍스트에 변형되어서 나타났던 형태소들을 모두 원형으로 복원하게 되고 이로 인해, 불규칙 용언들의 어간과 어미, 그리고 일부조사는 원형이 복원된다. 원형이 복원된 형태소 태깅 결과에 음운 변화 규칙을 적용하면 잘못된 발음열을 출력하게 된다. 예를 들어 ㄷ-불규칙 용언인 '걸어서'라는 어절은 형태소 분석하면 '걷/pvg+어서/ecs'로 분석되고 여기에 음운 변화 규칙을 적용하면 /KAXDAXSAX/(걸어서)'라는 잘못된 발음열을 생성하게 된다.

이러한 문제점을 해결하기 위해 본 논문에서는 형태소 태거와 발음열 변환기를 통합하고, 태깅에 사용되었던 불규칙 정보와 불규칙 용언 활용의 규칙성을 이용해서 원형이 복원되는 형태소들에 대한 올바른 발음열을 유도할 수 있도록 처리하였다.

### 2. 한국어의 음운 변화 규칙

한국어의 음운 변화 규칙은 음운 변화가 일어나는 위치에 따라 음소 변동 규칙과 변이음 규칙으로 설명할 수 있다. 해당 음소 문맥에 의해 하나의 음소가 다른 음소로 바뀌거나, 탈락하거나, 첨가되는 양상을 규칙화한 것을 음소 변동 규칙이라 하고, 하나의 음소가 발화장에서 여러 변이음으로 실현되는 양상을 규칙화한 것을 변이음 규칙이라 한다. 음소 변동 규칙은 음운 변화가 일어나는 음소열인 음소 문맥에 적용된다.

본 시스템은 한국어의 대표적인 음소 변동 규칙중 [표 1]에 나타난 18 개를 채택하여 음소 문맥에 적용한다[1].

	규칙의 종류	세부 규칙 수
1	음절말 중화	179
2	자음군 단순화	256
3	격음화(기식음화)	21
4	연음 규칙	37
5	유음화	10
6	장애음의 비음화	34
7	유음의 비음화	19
8	구개음화	3
9	경음화	136
10	종성 ㅎ-탈락	1
11	ㄴ-첨가	30
12	전선모음화	5
13	동일 조음위치 자음 탈락	6
14	중복 자음화	6
15	변자음화	17
16	초성 ㅎ-탈락	5
17	반모음화	7
18	모음탈락	10

[표 1] 음소 변동 규칙의 분류

규칙 1번에서 11번까지는 환경이 맞으면 반드시 적용해야 하는 필수 음소 변동 규칙이고, 12번에서 16번 규칙은 음성 인식시 사전 구성을 고려하여, 발화자의 습관을 반영한 수의적인 규칙이다. 규칙 17, 18번은 용언의 어간, 어미에 적용되는 규칙이다.

각 음소 변동 규칙들은 적용되는 음소 문맥 별로 다시 세부 규칙 번호가 주어지고 이에 따라 실제 음소 문맥에 규칙이 적용된다. 음소 문맥에 따른 세부 규칙의 수는 모두 782 개로 정의된다. 이는 [1]에서 761 개로 정의

한 규칙을 실험을 통하여 재정의 한 것이다. 음절말 중화, 자음군 단순화, 격음화, 연음 규칙에서 누락되어 오류를 일으키던 음소 문맥들을 규칙으로 추가하였고, [1]에서는 초성 ㅇ-탈락의 경우 부정성 다음에 나타나는 'ㅎ'만을 탈락 대상으로 정의하였으나 유성음 다음에 나타나는 'ㅎ'에 대해서도 탈락 규칙으로 추가하였다. 또 모음 탈락 규칙에 어간 모음 --탈락 규칙을 추가했고, 형태소 내부 연음 규칙 등을 추가하여 안정화된 규칙이라고 할 수 있다.

하나의 음소는 음성 환경, 말의 속도와 스타일에 따라서 여러 가지 음가를 가지게 되는데 이러한 소리들을 변이음이라 하고, 하나의 음소가 여러 변이음으로 변화되는 현상을 규칙화한 것을 변이음 규칙이라 한다. 한국어에 존재하는 변이음에는 유성음화, 무파화, 구개음화, 마찰음화, 접근음화, 탄설음화, 반모음의 무성화, 모음의 무성화 등이 있는데[6], 본 논문에서는 이들 중 음향적으로 큰 차이를 보이는 유성음화, 무파화, 구개음화만을 고려한다.

### 3. 형태소 태거 통합과 불규칙 처리

[표 1]에서 정의한 음운 변화 규칙을 적용할 때는 형태 음운 현상을 고려해야 하므로 형태소 분석, 태깅된 결과를 이용해야 한다. [1]의 시스템은 발음열 변환기가 형태소 태깅 단계와 독립되어 있어 사용자는 먼저 외부의 형태소 태거를 통해 입력 단위를 형태소 태깅한 후 발음열 변환기에 입력해야 했다.

또한 이 시스템에서 사용하는 형태소 분석기 및 태거는 Two-level 기반이다. Two-level이란 형태소 해석 및 생성에 기저형과 표면형의 두 가지만이 사용된다는 것이다. 즉 텍스트를 형태소 분석하면 텍스트에 나타난 모든 이형태 형태소들은 하나의 기저형으로 분석된다. 따라서 이 시스템에서 사용한 태거는 형태소 분석 시 텍스트 상에 변형되어 나타났던 불규칙 용언의 어간과 어미. 그리고 일부 조사는 원형으로 복원되고, 복원된 원형에 규칙을 적용하면 잘못된 발음열을 생성하게 된다. 예를 들어 '아름다운'이란 어절을 형태소 분석하면 '아름답/paa+L./ctm'으로 분석되고, 이 것이 발음열 변환기의 입력으로 들어갈 경우 형태소 경계에서 'ㅂ ㄴ'의 음소 문맥에 장애음의 비음화 규칙이 적용되고, 그 결과는 /AA R WW M D AA M N(아름답ㄴ)이라는 잘못된 발음열을 생성하게 된다. 이러한 불규칙 용언들은

텍스트 상에서 출현 빈도가 높다.

이를 해결하기 위해서 본 논문에서 제안한 시스템에서는 형태소 태거와 변환기를 통합하여 시스템을 구현하였다. 시스템을 통합함으로써 오류를 발생시켰던 불규칙 형태소들에 대한 처리도 가능해졌다.

한국어에서 발생하는 형태소의 철자 변화는 대부분 불규칙 용언의 활용에서 비롯된다. 한국어의 불규칙 용언은 어간이나 어미의 일부가 자소가 변화하는 현상이며, 이 변화에는 일정한 규칙이 존재한다[7][8]. 따라서 형태소 분석기로 부터 얻은 불규칙 정보에서 불규칙 종류를 알아내고, 그에 따른 일정한 규칙으로 처리가 가능하다.

형태소 분석기에서는 불규칙 형태소의 원형을 복원하면서 그 정보를 유지하고 있다. 이 불규칙 정보를 발음열 변환기에 연결 시켜 불규칙 형태소들에 대한 처리를 한다.

형태소 분석 후 원형이 복원되어 추가적인 처리가 필요한 형태소들이 [표 2]에 나타나 있다.

	불규칙 종류	예
1	ㄷ-불규칙	걷+으니 → 걸으니
2	ㅂ-불규칙	곱+으니 → 고우니
3	ㅅ-불규칙	짓+으니 → 지으니
4	ㅎ-불규칙	파랗+으니 → 파라니
5	ㄹ-불규칙	이르+어 → 이르러
6	ㄹ-불규칙	흐르+어 → 흘러
7	우-불규칙	푸+어 → 퍼
8	ㄹ-탈락	알+ㅂ니다 → 압니다
9	ㄷ-불규칙	퍼+어 → 피
10	하여/해-불규칙	하+어 → 하여/해
11	와 불규칙	되+어 → 돼
12	서술격 조사-이 축약	차+아+어서 → 차여서

[표 2] 원형이 복원되는 불규칙 형태소

[표 2]에 나타난 불규칙 형태소들의 처리 규칙은 다음과 같다.

(1) ㄷ-불규칙

'ㄷ'으로 끝나는 용언이 모음으로 시작하는 어미와 결합할 때 'ㄷ'이 'ㄹ'로 변화

(2) ㅂ-불규칙

'ㅂ'으로 끝나는 용언이 모음으로 시작하는 어미와

## TTS시스템을 위한 한국어 발음열 자동 생성

결합할 때 종성 'ㄴ'이 '우'로 변화

(3) 스-불규칙

'스'로 끝나는 용언이 모음으로 시작하는 어미와 만나면 '스'가 탈락

(4) ㅇ-불규칙

'ㅇ'으로 끝나는 용언이 모음으로 시작하는 어미와 결합하면 'ㅇ'이 탈락

(5) 르-불규칙

'르'로 끝나는 용언이 '어/아'로 시작하는 어미와 결합할 때, 어미부분이 '리'로 변화

(6) 르-불규칙

'르'로 끝나는 용언이 '어/아'로 시작하는 어미와 결합할 때, 어간 발음 '르'가 탈락하고, 종성 'ㄹ'이 첨가

(7) 우-불규칙

'우'로 끝나는 용언이 '어/아'로 시작하는 어미와 결합할 때, 어간 발음 '우'가 탈락

(8) 르-탈락

'르'로 끝나는 모든 용언이 'ㄱ, ㄴ, ㄹ, 시, 오'로 시작하는 어미와 결합하면 '르'가 탈락

(9) ㄷ-불규칙

'ㄷ'로 끝나는 용언이 '어/아'로 시작하는 어미와 결합하면 어미의 '어/아'가 탈락

(10) 하여/해-불규칙

'하다'라는 원형을 지닌 모든 용언에 해당하며, '어/아'로 시작하는 어미와 결합할 때 '해', '하여'로 변화

(11) 외-불규칙

'외'로 끝나는 용언이 '어/아'로 시작하는 어미와 결합할 때 '왜'로 축약

(12) 서술격조사 아-축약

'무종성 체언+서술격조사+어/아'로 시작하는 어미 결합이 발생하는 경우에는 서술격 조사와 어미 사이에서 축약

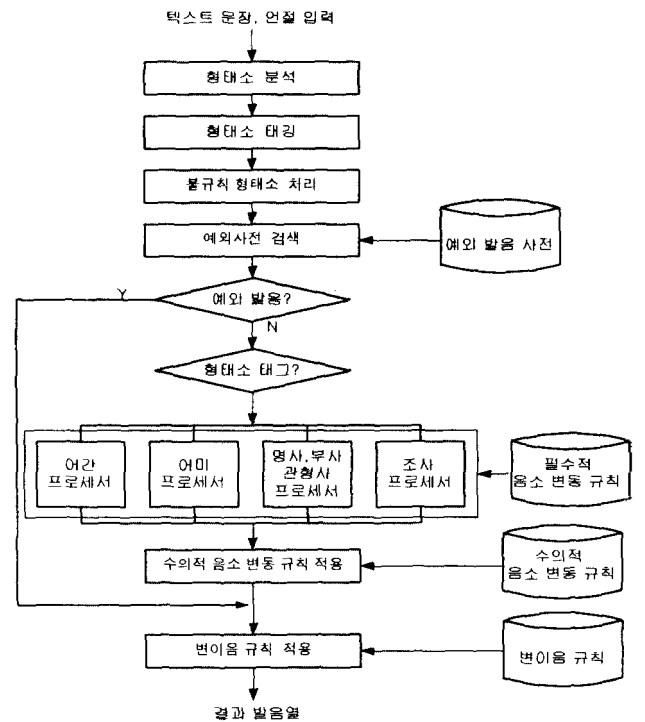
그러나 (10), (11)에 해당하는 용언들은 그 활용형인 '하여'와 '해' 또 '되어'와 '돼'가 모두 텍스트상에 표면형으로 나타나고, 형태소 분석 후 기저형은 하나이다. 이런 경우 한가지만을 출력형으로 결정하기가 곤란하다. 본 논문에서는 텍스트상에서 더 빈번하게 나타나는 축약형을 1순위로 출력한다. 이것은 실제 텍스트를 그대로

로 반영하는 것은 아니므로 오류의 원인이 되기도 한다. 이는 하나의 기저형만을 생각하는 Two-level 형태소 분석기를 이용하기 때문에 발생하는 문제이며, 이를 해결하기 위해서는 텍스트에 표면형으로 나타나는 여러 형태를 보존할 수 있는 형태소 분석기를 개발함으로써 해결할 수 있는 문제이다.

### 4. 발음열 변환 알고리즘

입력된 문장에 대한 발음열 변환 과정은 다음과 같은 순서로 진행된다.

- (1) 입력 문장을 형태소 분석, 태깅한다.
- (2) 원형이 복원된 불규칙 형태소들은 올바른 발음열을 생성할 수 있도록 처리한다.
- (3) 예외사전을 검색하여 규칙으로 설명할 수 없는 형태소들의 발음열을 획득한다.



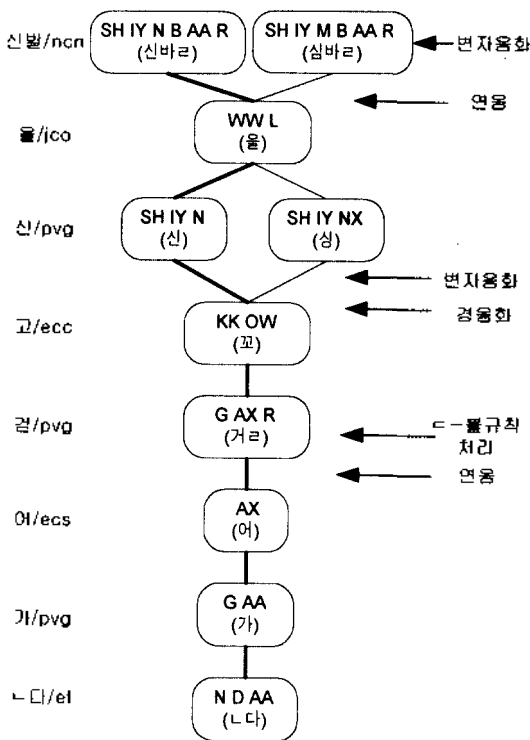
[그림 11] 발음열 변환기 구조

- (4) 형태소 범주에 따라 어간, 어미, 명사·부사·관형사, 조사로 다섯 개의 처리 루틴을 분리하여 [표 1]에 나타난 음소 변동 규칙을 적용한다.
- (5) 음소 변동 규칙이 적용된 발음열에 변이음 규칙을 적용하여 최종 발음열을 얻는다.

발음열 변환기의 전체 구조는 [그림 1]과 같다. 단계 (4)에서는 음소 변동 규칙을 적용하기 위해 음소 변동 규칙의 집합을 테이블 형태의 오토메타인 규칙들로 묶어 실제 변환기에서 보단 간편하고 체계적인 접근이 가능하도록 하였다. 규칙들은 적용 위치에 따라 형태소 내부, 형태소 경계, 어절 경계로 나누어 유지한다.

본 시스템은 인식시 발음 사전 구성을 고려하여 수의적 규칙을 적용하고, 그 결과로 표준 발음과 비표준 발음을 함께 생성하는 다중 발음열 출력 시스템이다. 그러나 합성에서는 표준 발음인 1순위 발음만을 출력하도록 한다.

'신발을 신고 걸어 간다'라는 문장 입력에 대한 처리 과정을 살펴보면, 먼저 이 문장은 '신발/ncn+음/ico 신/pvg+고/ecc 걸/pvg+어/ecs 가/pvg+ㄴ다/ef'로 형태소 분석, 태깅된다. 이 태깅 결과는 [그림 2]에 나타난 과정을 통해 규칙이 적용되어 결과를 출력한다.



[그림 2] 음소 변동 규칙의 적용 과정

'신발/ncn'은 형태소 내부에서 수의적 변자음화를 적용 받고, 형태소 경계에서 연음되어, 표준발음 'SHIYNBAAR(신바라)'과 비표준 발음 'SHIYMBAA R(심바라)'을 생성된다. '신/pvg+고/ecc'에서 '신/pvg'는 뒤에 오

는 어마 초성 'ㄱ'에 영향을 받아 수의적 변자음화가 일어나서 'SHIYN(신)'과 'SHIYNX(싱)'이, 어미 '고/ecc'는 경음화 되어 'KK(꼬)'로 생성된다. '걸/pvg'의 경우는 ㄴ-불규칙 용언이고 어미가 '어/ecs'이므로 어간 '걸'의 'ㄷ'을 'ㄹ'로 변화시키고 경계에서 연음되므로 'GAXR(거르)'이 생성된다.

[그림 2]에서 굵은 실선으로 연결한 결과가 표준 발음열이고, 나머지는 비표준 발음이다. 합성을 위해서는 표준 발음만을 출력하여 이용한다.

### 5. 실험 및 결과 분석

성능 평가를 위해, 초등학교 교과서 문장과 신문의 사회면 기사에서 발췌한 254 문장, 3,879 어절에 대한 실험 결과가 [표 3]에 나타나 있다.

총어절수	올바르게 변환된 어절 수	오류	불규칙 형태소 수
3,879	3,860	19	97

[표 3] 실험 결과

3,879 어절 중에서 19 개의 어절에서 오류가 발생했고, 이 중 11 개의 오류는 형태소 분석 오류에서 비롯되었다. '불가능/ncps+하/xsv+ㄹ/ctm 것/nbn'은 올바른 태깅이고 여기서 의존 명사 '것/nbn'은 관형형 어미에 의해 경음화 되어 '것'으로 된다. 그러나 오류가 발생한 예를 보면 '되/pvg+어야/ecs 할/ncn 것/nbn'은 '할'이 '하/pvg+ㄹ/ctm'으로 태깅되어야 하는데 명사로 잘못 태깅되었고, 따라서 경음화가 일어나지 않아 '것'이라는 오류 발음을 생성한다.

그 외의 오류 원인은 불규칙 처리 규칙 11, 12 번의 적용에 있어서 '하여/해'처럼 그 표면형으로 두 가지 다 허용되기 때문에 발생하는 오류와 '사적(사적)', '성공적(성공적)'에서처럼 '~적'과 같은 한자어 접미사가 결합할 경우 경음화가 일어나기도 하고, 그렇지 않은 경우도 있어 규칙화할 수 없는 부분에서 발생했다.

실험 텍스트에 나타난 불규칙 형태소는 모두 97 개였으며, 이는 [1]의 시스템에서는 모두 오류를 낳는 것들이다.

[1]에서 사용되었던 예외 발음 사전은 25,568 개의 어절을 포함하고 있으나, 이 중 장음 표시로 인한 발음이 상당수를 차지하고 있다. 장음 표시를 하지 않고, 본 발

## TTS시스템을 위한 한국어 발음열 자동 생성

음열 변환기 실험을 거쳐 변환기로 유도될 수 있는 발음은 모두 사전에서 제외함으로써 예외사전의 크기를 줄일 수 있었다.

	기존 예외 사전	실험 후
예외 사전 수록 어절 수	25,568	9,171

[표 4] 예외 사전 실험 결과

예외 사전에 수록된 단어 중 발음열 변환기로 생성할 수 없었던 발음은 대부분 한자어의 경음화 현상들이었다. ‘상가(상가)’, ‘평가(평가)’ 등과 같이 유성음 다음에서 경음화 현상은 규칙적이지 않아서 예외적으로 처리할 수 밖에 없다.

불규칙 처리 루틴을 추가함으로써 [1]의 시스템에서 발생했던 오류들을 감소 시켰고, 다양한 어절로 구성된 문장을 통해 실험함으로써 규칙을 안정성 있게 재정비하였다.

### 6. 결론

본 논문에서는 한국어 음운현상에 대한 체계적인 분석을 통해 문자열을 음소열로 자동 변환하는 시스템을 구현하였다. 본 논문의 시스템은 한국어의 형태소음운현상을 반영하기 위해 형태소 분석을 선행한 후, 한국어에서 빈번하게 발생하는 음운 변화 현상의 분석을 통해 정의된 음소 변동 규칙과 변이음 규칙을 선택적으로 적용하여 형태소, 어절, 언절 또는 문장 등의 다양한 형태의 입력에 대해 발음열을 생성한다. 기존의 연구에서 분리되어 있던 형태소 태거와 변환 시스템을 통합하여 사용자 편의성을 높였으며 Two-level 기반의 형태소 분석기를 사용하기 때문에 원형이 복원되는 형태소들에 대한 처리 루틴을 두어 오류를 감소 시켰다. 다양한 어절로 구성된 데이터를 통한 실험을 통해 오류를 수정하고, 규칙을 안정화 시킬 수 있었다.

본 논문에서 구현한 시스템에서 사용하고 있는 형태소 분석기는 Two-level 모델을 기반으로 하는 형태소 분석기이다. 이는 텍스트상에 여러 표면에 나타나는 형태소들을 하나의 기저형으로 복원한다. 그러나 대화체 문장들은 축약형이 많고, 형태소 원형이 복구되면 길자에 대한 발음열을 정확히 유도해내기 어려운 형태소들이 많다. 따라서 좀 더 완벽한 시스템을 구현하기 위해서

는 대화체 형태소 분석기를 필요로 한다[10].

### 참고 문헌

- [1] 전재훈, 차선화, 정민화, “음소 변동 규칙과 변이음 규칙을 이용한 한국어 음운 변화 현상의 규칙 모델”, HCI'98 학술대회 논문집, 1998.
- [2] 전재훈, “형태 음운학적 분석에 기반한 한국어 발음열 자동생성”, 서강대학교 전산학과 석사학위 논문, 1997.
- [3] 김병창, 이원일, 이근배, 이종혁, “한국어 TTS 를 위한 무제한 단어 자소열-음소열 변환”, HCI'98 학술대회 논문집, 1998.
- [4] 배주재, 국어음운론 개설, 형설출판사, 1995.
- [5] 오정란, 현대 국어 음운론, 형설출판사, 1995.
- [6] 이호영, 국어음성학, 태학사, 1996.
- [7] 남기심, 고영근, 표준 국어문법론, 탐출판사, 1997.
- [8] 이성진, “Two-level 한국어 형태소 분석기”, 한국과학기술원 전산학과 석사학위 논문, 1992.
- [9] 김재훈, 서정연, “자연언어 처리를 위한 한국어 품사 태그”, 한국과학기술원, 인공지능연구센터, CAIR-TR-94-55, 1994.
- [10] 김재훈, “음성인식과 기계번역의 통신규정 : 의사형태소 정의”, 한국전자통신연구원, TM97-1420, 1998.