

## 음성 인식과 합성에서의 한국어 언어 처리

보이스텍 이원일

### 1. 서론

최근 몇 년간 음성 관련 시장은 그 어느 때 보다 크게 성장했고 일반인들의 인지도도 많이 높아졌다. 이동 전화에서 시작해서 가전 제품, 말로 하는 인터넷에 이르기까지 많은 서비스들이 시도되었고 계속 늘어나고 있는 추세이다. 현재 가장 크게 성공을 거둔 분야는 전화 음성 인식 분야와 음성 합성 분야라고 할 수 있다. 증권 정보 안내 시스템의 도입을 시작으로 많은 증권사와 이동 통신 업체들, 항공사 등에서 앞다투어 음성 인식과 음성 합성 시스템을 도입했고 보이스 포털 서비스까지 등장했다.

하지만, 음성 인식기와 음성 합성기 모두 다른 언어에서 볼 수 있는 정도의 성능을 얻지는 못하고 있고 성능 향상도 답보 상태에 있다. 이는 현재까지의 음성 관련 연구자들의 관심이 음성 처리에 국한되어 있었고, 음성 처리에 꼭 필요한 언어 처리는 간단하게 처리할 수 있는 것들만 사용하고 있는데 그 한 원인이 있다고 본다.

소수의 연구자들이 언어 처리와 음성 기술의 접목을 시도했었지만 빈약한 음성 자료들과 공유 자료의 부족으로 실제적인 결과를 얻는데는 실패를 했었다. 음성 분야와 언어 처리 분야의 대규모 공개 자료 구축이 진행되고 있는 지금 시점이 음성 언어 처리의 필요성을 다시 살펴보고 새로운 시도들을 이끌어 낼 수 있는 때라고 본다.

이 글에서는 음성 명령 시스템, 받아쓰기 시스템, TTS(text-to-speech) 시스템을 중심으로 현재 사용되고 있는 언어 처리를 요약하고 성능 향상과 사용자의 요구를 수용하기 위해 필요한 추가적인 요소들을 제안하도록 하겠다.

### 2. 음성 언어 처리 개요

다음 그림은 개괄적으로 본 음성 처리와 언어 처리의 기술 관련도이다.

음성과 언어 처리의 가장 하부에는 음성 DB, 말뭉치(text corpus)와 전자 사전, 언어 처리 사전이 있다. 이 기초 자료들을 기반으로 언어 모델, 문자열-발음열 변환, 형태소 분석, 품사 태깅, 끊어 읽기, 구문 분석, 형태소 생성 기술 등의 기본 언어 처리 기술들이 있고 이 기술들이 신호 처리, 단어 인식, 연속 음성 인식, 음성 합성 기술들과 결합하여 음성 명령, 받아 쓰기, TTS(text-to-speech), 오디오 인덱싱 시스템을 이루게 된다.

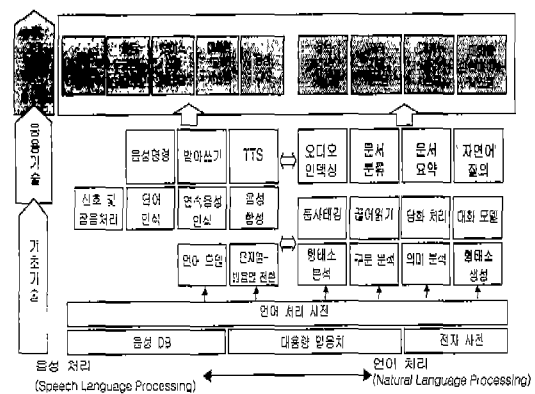


그림 1 음성 언어 처리 기술 관련도

### 3. 음성 명령 시스템

음성 인식에 대한 일반인의 관심이 늘어 가면서 말로 하는 게임, 말로 하는 인터넷, 보이스 포털 등의 다양한 서비스가 제공되기 시작했다. 이런 시스

템에서 사용하는 것이 음성으로 명령을 내리는 시스템이다. 이 시스템들은 사용자가 내릴 수 있는 명령어들을 특정한 문법의 문장들로 제한 하는데, 이런 시스템을 문법 기반 음성 인식 시스템이라고 한다. 이 때 사용하는 문법은 대부분 CFG(context-free grammar) 형태를 취하고 있다[6~11].

현재 개발되어 있는 문법 기반 음성 인식 시스템들은 200~1000개의 어휘로 만들어진 문법을 92~97% 정도의 정확도로 인식하는 정도의 성능을 보인다.

다음은 주식 거래에 사용될 수 있는 문법의 일부이다. <>는 비단말 노드를 ""은 단말 노드를 나타내고 []은 생략 가능함을 나타낸다.

- <시작> → <종목명> <숫자> "주"  
( <매도> | <매수> );
- <종목명> → "삼성전자" | "삼성전환" |  
"삼아제약" | "현대건설" ;
- <매도> → "매도" ;
- <매수> → "매수" ;

이 문법 파일을 이용하면 "삼성전자 다섯 주 매도", "삼아제약 이백 주 매수" 와 같은 음성 명령을 할 수 있다.

### 3.1 문법 개발 도구

문법 기반 음성 인식 시스템들은 보통 소프트웨어 개발 도구(SDK)와 함께 문법 개발 도구(GDK)를 제공하는데 이 문법 개발 도구에 품사 태거와 문자열-발음열 변환기가 사용된다.

앞에 보인 문법 파일에서 보면 인식 대상 단어들은 큰 따옴표로 둘러 쌓여 있는데 그 단어들이 어떻게 발음되는가의 정보는 표현되지 않았다. 언어 처리 도구들은 사용자가 입력하지 않은 이 어휘들의 발음을 자동으로 만들어주는 역할을 한다.

현재 한국어를 위한 문법 개발 지원 도구들은 대부분 품사 태거를 제거한 상태의 간단한 문자열-발음열 변환기를 포함하고 있다. 하지만 품사 정보가 없으면 정확한 발음 추정을 하는 것이 불가능하기 때문에 사용자가 직접 발음을 넣어주지 않으면 인식률에 나쁜 영향을 끼치기도 한다.

### 3.2 단어 천이 모델

문법 기반 음성 인식 시스템들은 단순한 어휘 목록을 인식하는 것이 아니고 여러 개의 어휘로 된

문장들을 인식하기 때문에 어휘와 어휘 사이의 음운 현상을 처리하는 부분이 인식기 내부에 필요하다. 어휘들 사이의 음운 현상들을 인식기 내부에서 처리하지 않으면 경계의 음소들을 제대로 인식하지 못해 인식률을 떨어뜨리게 된다.

다음은 어휘와 어휘 사이의 음운 현상의 예를 보여준다.

- "할" "것을" \*\*\*> [할] [꺼슬]
- "꽃" "한" "송이" \*\*\*> [꼬] [탄] [송이]
- "걸" "옷" \*\*\*> [거] [돋]
- "밥" "먹는다" \*\*\*> [밤] [명는다]
- "옷" "입다" \*\*\*> [온] [납따]
- "서른" "여섯" \*\*\*> [서른] [녀섯]

이런 어휘 경계의 음운 현상을 처리하는 방법 중 하나는 Fan-in / Fan-out 모델과 결합 제약을 통한 방법이다. 한 어휘의 첫번째 음소와 마지막 음소는 그 어휘의 앞뒤의 어휘에 따라 바뀌기 때문에, 그 어휘의 발음을 모두 표현한다면 Fan-in과 Fan-out을 갖는 네트워크의 형태가 된다. 다음은 명사 '닭'의 발음을 표현한 것이다.

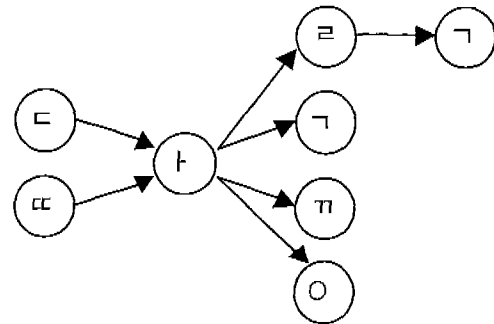


그림 2 "닭"의 발음 네트워크

하지만 이 Fan-in과 Fan-out 부분은 원래의 음소와 품사에 의해 결정되기 때문에 어휘마다 별도로 존재할 필요는 없다. 이 Fan-in / Fan-out과 그 사이에 가능한 연결을 제어하면 어휘와 어휘 사이의 음운 현상이 처리될 수 있다.

아래 그림은 "닭" 과 "고기" 사이의 경음화 현상을 보여준다. 'ㄹ' 명사의 여러 가지 변이음 중에 [ㄱ] 과 'ㄱ' 명사의 변이음 [ㄱ]이 연결 가능하다는 정보를 통해 'ㄹ'과 'ㄱ' 이 만나면 [ㄱ] [ㄱ] 으로 바뀌는 대표음화와 경음화 현상이 일어남을 보여주고 있다.

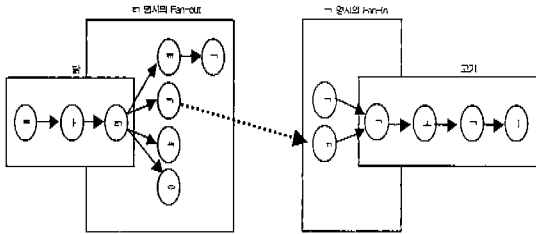


그림 3 “닭” “고기” 사이의 단어 친이

### 3.3 형태소 기반 문법

앞에서와 같이 단순한 명사류의 나열로 된 문장만 인식할 수 있는 것은 아니다. 다음과 같은 문장을 인식하고 싶다고 하자.

- “삼성전자 다섯 주를 매도해주세요”
- “삼성전자를 다섯 주 매도하고싶어요”
- “삼아제약을 오십 주 매수”
- “현대건설 다섯 주를 팔아주세요”

앞에서와 같이 여기 나온 어절들을 그대로 사용하여 문법을 만들려고 한다면 이전의 문법보다 2배 이상의 어휘를 사용하는 문법이 되고 만다. 어휘가 2배로 늘어났다는 것은 인식 속도가 느려지고 인식률이 나빠진다는 의미이다. 이 증편 거래 시스템의 예에서 실제 상장된 회사의 숫자가 1000개를 넘기 때문에 어휘가 두 배로 증가한다는 것은 인식할 수 없게 된다는 것과 같다.

한국어의 특성에 맞게 조사/어미 등을 분리시켜 문법을 만들려고 하면 조사/어미의 이형태 문제가 발생한다. 가장 쉽게 다음과 같은 문법 파일을 만들었다고 하자.

<시작>→<종목명> [“을/조사” | “를/조사”]  
 <숫자> “주” “를/조사”  
 (<매도> | <매수>) [“해주세요” | “하고싶어요”];

이 문법 파일은 인식하고자 하는 “삼성전자를 세 주 매도해주세요” 뿐만 아니라 “삼성전자를 세 주 매도해주세요”도 인식하게 된다. 문법 작성자에게 조사/어미를 골라서 사용하도록 강요하지 않으면서 지원할 수 있는 방법은 없을까? 문법 개발 도구의 전처리 또는 후처리 모듈을 통해서 지원할 수 있을 것이다.

전처리에서 해결하는 방법은 사용자에게 <을/조사> <가/조사> 등의 기본형 조사/어미를 사용하여 문법 파일을 작성하게 하고 이형태 생성기를 통

해 가능한 이형태들을 생성하고 각 이형태와 결합하는 비단말 노드들을 재분류 해주는 것이다.

예를 들어 사용자가 다음과 같은 문법 파일을 만들었다면

<시작>→<종목명> [<을/조사>] <숫자>  
 “주” [<을/조사>]  
 (<매도> | <매수>) [“해주세요” | “하고싶어요”];  
 문법 작성 도구에서는 아래와 같이 확장된 문법 파일을 만들어 주는 것이다  
 <시작>→<종목명\_유종성> [“을/조사”] <숫자>  
 “주” [“를/조사”]  
 (<매도> | <매수>) [“해주세요” | “하고싶어요”] |  
 <종목명\_무종성> “를/조사” <숫자> “주” “를/조사”  
 (<매도> | <매수>) [“해주세요” | “하고싶어요”] |  
 <종목명\_르종성> “을/조사” <숫자> “주” “를/조사”  
 (<매도> | <매수>) [“해주세요” | “하고싶어요”];

이런 도구가 지원된다면 사용자는 쉽게 문법 파일을 만들 수 있을 것이다.

후처리에서 해결하는 방법은 전처리 방법에서와 같이 간단한 문법 파일을 이용하여 인식을 하고, 인식기의 n-best 결과에 대해 접속 검사를 해서 연결될 수 없는 형태소열이 있는 결과를 지워버리는 것이다. 인식기가 인식한 결과가 다음과 같다면

1. 삼성전자 을/조사 다섯 주 매도 해주세요
  2. 삼성전관 을/조사 다섯 주 매도 해주세요
  3. 삼성건설 을/조사 다섯 주 매도 해주세요
- 1번 결과의 “삼성전자 을/조사”이 불가능한 조합이기 때문에 1번을 뺀 2, 3번만을 최종 인식 결과로 출력하면 된다.

이 두 방법 중 후처리 방법이 더 간단한 것 같지만 인식 시간과 인식률에서 전처리 방법보다 나쁜 결과를 가져올 것으로 여겨진다.

## 4. 받아쓰기 시스템

받아쓰기 시스템은 일반인들이 꿈꾸는 음성인식의 모습으로 사용자가 말하는 모든 것이 글로 받아 써지는 시스템이다. 전자 메일을 단지 말하는 것만으로 작성한다거나 보고서를 만든다거나, 녹음기로 녹음해 두었던 음성을 문서로 만든다거나 하는 것들을 가능하게 해주는 시스템이다.

받아쓰기 시스템에는 대용량 어휘 인식 기술, 화자 적응 기술, 잡음 처리 기술, 연속 음성 인식 기술, 등 음성 처리 분야의 핵심 기술이 다 필요하지

만 이 모든 기술보다 더 중요한 부분이 언어 처리 기술이다.

받아쓰기 시스템이 실제 사용되기 위해서는 6만 개 정도의 인식 단위로 일반적으로 사용하는 단어를 95% 이상 커버하고 90% 이상의 인식률을 보여야 한다. 한국어 받아쓰기 시스템의 어려운 점이 여기에 있다. 한국어에 존재하는 조사/어미류만 보더라도 복합된 형태로 10000개를 넘기 때문에 단어/어절을 기반으로 한 인식은 절대 불가능하다. 2억 5천만 어절의 말뭉치를 분석해 본 결과로는 2000만개 이상의 서로 다른 어절이 나타나고 6만개의 어휘로는 80% 정도를 커버하는데 머무른다. 일반적으로 사용하는 형태소 분석을 이용하면 200만개 이상의 어휘가 발견되고 6만개 정도의 어휘로는 90~92%를 커버하게 된다.

6만개의 어휘로 커버되지 않는 것들을 분석해 보면 복합어와 파생어류가 대부분인데 일반적으로 사용하는 형태소 분석기에서는 복합어, 파생어 등의 분석을 거의 하지 않고 있기 때문이다. 받아쓰기 시스템을 위해서는 복합어와 파생어를 가능한한 많이 분석하는 형태소 분석기를 사용해야 하고 단일 형태소에 가까운 인식 단위를 사용한 시스템을 구축해야 하는 것이다.

일반적인 받아쓰기 시스템 개발에 필요한 언어 처리 과정을 보면 다음과 같다. [2, 5]

- 말뭉치 준비

말뭉치를 수집, 구매하고 띄어쓰기, 맞춤법 등을 수정한다. 적어도 1억 어절 이상의 일반 말뭉치가 필요하고 특정 분야를 위해서는 수 천만 단위의 말뭉치가 별도로 준비되어야 한다.

- 말뭉치 표준화(TN: Text Normalization)

말뭉치에 나오는 숫자, 구두점, 기호 등을 정형화된 형태로 바꾼다. 말뭉치에 나타나는 숫자 등을 사용할 인식 단위에 맞게 풀어써야 한다. 예를 들어, "5500"은 "5000 500" 또는 "5 1000 5 100" 등과 같이 바뀌어야 한다.

- 전문 용어 목록

말뭉치에서 전문용어와 오용어를 뽑아 목록을 만든다. 특정 분야의 전문 용어 등은 미등록어로 표시되고 조사, 어미, 복합어 등이 제대로 분석되지 않을 수 있기 때문에 목록을 얻거나 말뭉치에서 미등록어를 조사하여 뽑아내야 한다.

- 언어 처리 사전 작업

말뭉치에 나오는 전문 용어와 오용어들을 품사 태거가 분석할 수 있게 언어 처리 사전에 등록한다.

- 품사 태깅

말뭉치 전체를 품사 태깅한다.

- 인식 단위 고르기

품사 태거의 결과를 실제 인식하는 단위로 바꾼다. 품사 태거의 분석 결과를 병합해서 인식 단위를 만든다. 조사/어미를 모두 나누면 언어 모델에서 context 정보를 잃게되기 때문에 적절한 복합 형태의 조사/어미를 사용하는 적절한 복합 형태를 선택하는 것이 중요하다.

- 언어 모델 추출

6만개 정도의 고빈도 어휘 목록을 뽑고 언어 모델을 만든다.

- 인식용 발음 사전 구축

고빈도 어휘 목록에 발음열 및 기타 정보를 넣는다.

- 기본 단어 목록

빈도수와 관계없이 인식되어야 하는 어휘들의 목록을 만들고 발음열 및 기타 정보를 넣는다.

- 띄어쓰기 규칙과 예외 사전

띄어쓰기 규칙을 만들고 복합 명사, 복합 동사 등 띄어쓰기 규칙의 예외 목록을 만든다.

- 고쳐쓰기(ITN: Inverse Text Normalization)

한글->숫자, 한글->한자, 한글->영어 등 고쳐쓰기 규칙을 만든다.

이 모든 과정에서 품사 정보가 필수적이다. 한국어의 1음절 형태소들이 조사, 어미, 접미사, 접두사, 수사 등과 같은 형태를 가지기 때문에 띄어쓰기, 고쳐쓰기, 언어 모델에서 품사 정보 없이는 구별되지 않는다. 다음의 예를 보자. 품사 정보 없이 인식 결과가 나온다면 띄어 쓰기와 바꿔 쓰기를 한 후에 어떤 문장이 될지 알기 어렵다.

구 월 이 십 일 일 의 일 이 다

→ 9 월이 10 일 1 의 12 다

→ 9월이21 일의1 이다

이 모든 처리가 제대로 이루어지기 위해서는 문법 기반 음성 인식기에서 지정한 품사 정보를 이용한 단어 친이 모델과 품사 정보를 사용하는 언어 모델, 품사 정보를 사용하는 띄어쓰기, 고쳐쓰기 등 품사 정보를 이용하는 형태소 단위의 시스템 구성이 필요한 것으로 보인다.

## 5. TTS(text-to-speech) 시스템

음성 합성은 일반적으로 다음과 같은 처리과정을 가지고 있다. 품사 태거를 통해 각 단어를 형태소로 분할하고 그 결과를 가지고 끊어 읽기를 어디서 할 지 결정한다. 이 끊어 읽기 정보와 품사 정보를 가지고 문자열-발음열 변환을 하고, 마지막 단계에서 품사 정보와 끊어 읽기 정보를 참조하여 적절한 합성 단위를 골라서 합친다.

### 5.1 품사 태거

TTS는 다른 어느 응용 프로그램에서 필요로 하는 것보다 정교한 품사 태거를 요구한다. 웹 문서나 전자 메일 등 미등록어가 많은 문서를 처리하기 때문에 미등록어 추정이 잘 되어야 하고, 복합어나 합성어에서 일어나는 음운 현상들을 처리하기 위해서는 복합어, 합성어도 모두 분석되는 것이 좋다. 또, 끊어 읽기를 제대로 하기 위해서는 높은 정확도가 필요하고 실시간으로 많은 수의 입력을 처리하기 때문에 속도도 빨라야 하고, 품사 태거의 크기도 작아야 한다. 이런 모든 조건을 만족하는 품사 태거는 존재하지 않는다고 할 수 있다.

현재 TTS들의 품사 태거가 가진 가장 중요한 문제는 미등록어 문제와 낮은 정확도에 있다. 많지 않은 언어 자료들을 기반으로 간단히, 빠른 속도 위주로 만들어진 품사 태거를 사용하는 경우가 대부분이기 때문에 정확도에 문제가 생기고 이것이 문자열-발음 변환과 끊어 읽기가 잘못되게 되는 것이다.

아래 두 문장은 품사 추정을 잘못하면 정확한 문자열-발음열 변환을 할 수 없는 예이다.

신을 신교 신교하려 갑니다. [신교] [신교]

영킨 실을 다시 값기는 어려운 일이다. [값끼는]

### 5.2 문자열-발음열 변환

한국어에는 음운론적 조건에 의한 음운 변이와, 형태론적 조건에 의한 음운 변이가 존재 한다[3, 5]. 음운론적 조건에 의한 음운 변이란 한 음소의 발음이 앞 뒤 음소에 의해서만 결정되는 것을 말하는 반면, 형태론적 조건에 의한 음운 변이는 현재 형태소의 품사와 앞 뒤 형태소의 품사에 따라 영향을 받는 것을 의미한다.

음운론적: 각하 [가카]

형태론적: 신/동사 + 고/어미 [신꼬]

형태론적 조건 중에서도 어려운 부분은 복합어와 파생어에서의 사이시옷, ㄴ 첨가 현상 등의 처리이다. 이 부분은 품사 태거에서 분석하지 못하는데 근본적인 원인이 있다고도 할 수 있다.

서른 여섯 되던 해의 일이다. [서른 녀섯]

아침밥을 먹고 학교에 갔다. [아침뺨]

### 5.3 끊어 읽기

끊어 읽기는 문장의 어느 곳에서 말을 멈추거나 숨을 쉬는가를 결정하는 것이다. 음성 DB의 데이터를 듣고 문장에 끊어 읽기를 표시한 끊어 읽기 DB나, 특정 문장들을 사람들에게 끊어 읽을 곳을 표시하게 하여 만든 끊어 읽기 DB를 사용하여 끊어 읽기 규칙을 뽑아내거나 통계를 얻어 사용하는 것이 일반적이다. [1, 4].

현재 판매되고 있는 TTS 시스템들을 가지고 실험을 해보면 정형화된 문장이나 단문들에서는 아주 자연스러운 끊어 읽기를 보여주는 반면, 웹 페이지나 메일 메시지 같은 미등록어가 많고 명사 명사의 나열이 많은 경우에는 아주 부자연스러운 끊어 읽기를 보이고 있다. 이것은 앞에서 말한 바와 같이 미등록어 추정을 제대로 못하는데 원인이 있다고 할 수 있다.

또 하나의 원인으로 들 수 있는 것은 대용량 끊어 읽기 DB가 없다는 것이다. 특정 화자의 실제 음성 파일에서 만든 끊어 읽기 DB가 가장 좋겠지만, 대부분 여러 화자로부터 끊어 읽을 곳을 표시하게 한 적은 DB에서 자료를 뽑아내기 때문에 예외 상황이 많이 발생하는 것으로 보인다. 이것을 위해서는 TV 뉴스 등의 특정 앵커의 음성 파일을 이용한 끊어 읽기 DB 구축 등이 필요할 것으로 보인다.

## 6. 결 론

이 글에서는 음성 처리와 언어 처리 기술이 현재 어떻게 통합되어 사용되는지 보이고 한국어 언어 처리가 음성 인식과 합성에서 특히 중요한 역할을 한다는 것을 강조했다. 한국어의 형태론적 음운론적 특이점은 향후의 연구가 형태소 단위의 품사 정보를 가진 문법, 언어 모델, 음운 현상 처리여야 함을 지적했고 음성 언어 처리를 위해서는 일반적으로 사용하는 언어 처리 도구들보다 복합어, 합성어 처리, 미등록어 처리 등의 기능이 강화되어야 한다는 점도 이야기 했다.

아직까지 한국어 음성 언어 처리 부분의 연구는 걸음마 단계라고 할 수 있다. 이 글에서 제시한 부분들 중 많은 부분이 최근에 개발되었거나 개발하고 있는 부분이다. 언어 처리를 연구하는 연구자들이 음성 언어 처리에도 관심을 기울여 현재의 기술적 한계를 뛰어넘는 결과를 많이 만들어 낼 수 있으면 좋겠다.

**참고문헌**

[1] 김선미. 한국어의 리듬 단위와 문법 구조 음성 합성에서 리듬 구현의 자연성 향상을 위한 음성 언어학적 연구, 서울대학교 박사논문, 1997.  
 [2] 김병창, 이진석, 차정원, 이근배, 이종혁. *Integrating linguistic and signal knowledge in a morpheme based speech corpus annotation tool*. Computer Processing of Oriental Language, vol. 13(4) pp.291-311 (2000. 12).  
 [3] 김병창, 이근배, 이종혁. *Hybrid grapheme to phoneme conversion for unlimited vocabularies*. submitted to ACM transactions on asian language information processing.  
 [4] 김병창, 이근배. Decision-tree based error correction for statistical phrase break prediction in Korean. Proceedings of the 18<sup>th</sup> international conference on computational linguistics (coling2000), pp1051-1055, July-Aug 2000.  
 [5] 이원일. 단일화 기반 범주 문법에 기반한 음성 한국어 처리, 박사 학위 논문, 포항공과대학 대학원, 1998.

[6] L&H ASR1600 SDK 4.0 manual, 2000.  
 [7] L&H VoiceXpress 5.0 SDK manual, 2000.  
 [8] F. Tendeau and J. Verhasselt, Proposal for an Attributed Grammar Format, report L&H-SR-2000-012, March 6, 2000.  
 [9] F. Tendeau, NLU.ppt : Powerpoint presentation that introduces the Attributed Grammar Format that is proposed for L&H BNF+, July 3, 2000.  
 [10] B. Lucas, W. Walker, and A. Hunt, ECMA Script Action Tags for JSGF, September 2, 1999.  
 [11] Microsoft Speech SDK 5.1 documentation.

**이 원 일**



1988.3~1992.2 포항공대 전산과 학사  
 1992.3~1994.2 포항공대 전산과 석사  
 1994.3~1998.8 포항공대 전산과 박사  
 1998.9~1999.12 ETRI 지식정보연구부 선임연구원  
 1999.6~1999.12 SnL 책임연구원  
 2000.1~2001.5 L&H Korca 책임연구원  
 2001.6~현재 보이스텍 책임연구원  
 1998.9~1999.12 ETRI 음성타자기 과제 언어처리 담당  
 1999.6~1999.12 IBM ViaVoice 한국어 버전 언어처리/언어 모델 담당  
 2000.1~현재 L&H, 보이스텍 하이보이스 언어처리/언어모델 담당  
 2000.2~2000.5 L&H RealSpeak TTS 언어처리 개발 참여  
 E-mail: wonilee@dreamwiz.com