

<http://dx.doi.org/10.7236/JIIBC.2013.13.1.199>

JIIBC 2013-1-27

모바일기반 음성인터페이스에 관한 연구

A Study on the Voice Interface for Mobile Environment

김수훈*, 안종영**

Soo-Hoon Kim, Jong-Young Ahn

요약 구글의 안드로이드 기반 음성인터페이스는 웹 어플리케이션에 국한되어 있으며 사용자층 또한 극히 드문 현실이다. 본 연구에서는 기존의 안드로이드기반 음성엔진을 사용하여 음성인터페이스가 이루어 질 수 있는 방법론을 제시하고 관련 어플리케이션을 개발하였다. 또한 안드로이드기반 어플리케이션에 있어서 일상 잡음환경에서 음성인식 어플리케이션의 음성인식 성능 즉, 음성인터페이스에 관한 환경을 연구하였으며 그에 알맞은 모바일 환경에서의 음성 인터페이스를 제시하였다.

Abstract Google's android-based voice interface is limited to the web application and the users are rare. In this paper, We suggest the method that can be done using existing android-based voice engine and develop voice application. We also study the environments of android-based voice interface and present the appropriate voice interface in mobile environment.

Key Words : Android-based, Voice Interface, Voice Recognition

1. 서론

요즘 지하철이나 버스에서 사람들이 손에 모바일 장비를 하나씩 쥐고 있는 모습은 전혀 낯설지 않다. 이러한 모바일 생활의 중심 역할을 스마트폰이 수행하고 있다. 통화 기능만을 제공하던 과거의 휴대폰과는 달리 최근의 스마트폰은 기존의 통화기능의 수준을 뛰어 넘어 디지털 장비를 통합해 가고 있다. 인터넷 접속 및 검색이 가능해지고, 사운드 및 동영상의 재생·녹화 등 다양한 멀티미디어 인터페이스를 접목 시켰다. 또한 게임과 같은 엔터테인먼트, DMB, 카메라, GPS와 이를 활용한 네비게이션 어플리케이션 등 다양한 용도로 사용하게 되었다. 그리고 스마트폰의 수행속도 또한 Quad-core CPU장착으로

클래식 펜티엄 수준의 속도를 낼 수 있으며 가격은 점점 더 저렴해지고 배터리의 효율도 점점 더 높아지고 있다. 따라서 현재 스마트폰 사용자들의 증가추세에 따라 관련 분야의 소프트웨어 수요가 그 만큼 증가할 것이다.

이에 본 논문에서는 스마트폰에 적용될 수 있는 응용 프로그램에 있어서 음성인터페이스에 관련된 내용을 연구하였다. 기존의 음성인식 방법에는 특정화자, 불특정화자를 구별하여 화자중속, 화자독립으로 나눌 수 있으며 인식단어에 따라 고립단어인식과 연속단어인식으로 나누어진다. 화자중속에는 화자인증, 핵심어인증으로 나눌 수 있다. 화자 중속은 화자 자신의 음성데이터만을 기초로 하여 비교패턴을 만들고 인식 시 화자의 패턴과 비교하여 구성된 패턴과의 일치 여부를 판단하여 인식하게

*정회원, 부천대학교 모바일통신과

**정회원, (주)유라테크 R&D 연구소

접수일자 2013년 1월 9일, 수정완료 2013년 2월 6일

게재확정일자 2013년 2월 8일

Received: 9 January 2013 / Revised: 6 February 2013 /

Accepted: 8 February 2013

*Corresponding Author: shkim@bc.ac.kr

Dept. of Mobile Telecommunications, Bucheon University, Korea

된다. 화자독립의 경우 예상되어지는 모든 화자의 나이, 지역, 연령, 성별 등 모든 화자데이터의 기초 패턴을 만들어 어느 화자가 이야기하든지 인식가능하게 하는 음성인식방법이다.

현재 음성인식에 관한 연구는 대기업을 비롯한 관련 중소기업 그리고 연구소 및 대학교에서도 연구개발 중에 있으며 외국의 경우 일부 적용된 사례도 있다. 국내에서는 한국전자통신연구원, 삼성 등 주요연구기관에서 연구하고 있으며 일부 성과도 발표되고 있다. 미국의 경우 애플사의 슈리, 차량에서는 GM-Onstar등을 비롯하여 관련 성과물들이 제품화 되고 상용화도 이루어지고 있다. 하지만 아직도 일반인들이 쉽게 사용하지 못하고 있고 음성 인식률저하 및 응용인터페이스가 익숙지 않아 사용예가 극히 드문 현실이다.^[1]

이에 본 연구에서는 기존의 안드로이드 음성엔진을 이용하여 쉽게 접근가능한 음성인터페이스 구조와 어플리케이션에 대하여 연구 하고 그 음성인식 성능을 평가하였다.

II. 본 론

1. 모바일운영체제

기존의 휴대폰은 별도의 기능은 없고 통화 기능만 제공 하였다. 제조사에서 미리 설치해 놓은 프로그램만 사용할 수 있으므로 제한적이지만 시스템에 대한 잔고장 거의 없었지만 추가적인 응용프로그램의 추가 설치가 불가능하다는 단점이 있었다. 반면 현재의 스마트폰은 크기가 휴대할 수 있는 정도의 크기고 기존의 휴대폰보다는 다소 크기가 크다고 할 수 있다. 이는 통화 기능 외에 여러 부가기능을 사용해야하는 스마트폰의 특성이라 할 수 있다. 그리고 내장된 프로세서의 성능은 구 컴퓨터 정도로 발전되어 운영체제 소프트웨어 인터페이스가 설치 가능하고 응용프로그램의 추가설치가 가능하여 별도의 이동통신사의 검수를 받을 필요가 없이 오픈마켓에서 응용프로그램을 다운받고 사용할 수가 있다. 한 장비에서 임의의 소프트웨어를 설치하고 운영하기 위해서는 시스템에서 직접 통제 관리하는 운영체제가 탑재 되어야 한다. 공통적으로 사용될 수 있는 API(Application Programming Interface)가 필요하고 일정한 규칙 하에 임의의 소프트웨어들이 조화롭게 설치 및 실행 될 수 있다. 모바일 환경에서도

상기와 같은 운영체제를 적용한다면 모바일 운영체제가 될 수 있다. 어떤 운영체제를 기반으로 하느냐에 따라서 디바이스의 전반적인 특징이 결정되며 기능이나 성능에도 많은 차이가 발생한다.

현재 많이 사용되는 모바일 운영체제는 Palm, RIM, 윈도우모바일, 심비안, iPhone, 안드로이드(Android)등이 있다.^[1] 현재는 iPhone과 안드로이드(Android)가 가장 많이 사용되는 모바일 운영체제이다.

2. 안드로이드(Android)

안드로이드의 개발은 OHA(Open Handset Alliance)로 되어 있지만 실제적인 주체는 인터넷 검색 업체인 구글이다. OHA는 개방된 모바일 환경을 위한 하드웨어, 소프트웨어 업체의 공동 연합으로서 세계적으로 유명한 30~40 여개의 업체가 가입 활동하고 있다. OHA는 비 독점 개방플랫폼을 지향하며 이에 공개 형 모바일 OS개발을 시작으로 한 프로젝트 명이 안드로이드였으며 현재의 안드로이드로 불리고 있다. 현재 독점적으로 공급되는 아이폰, 윈도우 모바일과는 서로 다른 공급 형태를 취하고 있는데 장·단점은 독점 폐쇄적인 환경은 라이선스 비용이 들지만 개발환경에 대한 본사의 지원이 분명하다는 이점이 있다. 하지만 개방형 환경에서는 라이선스 등의 비용은 들지 않지만 개발 툴에 대한 기술지원이 없고 또한 그 문제가 발생 시 책임이 없다는 단점이 있다. 안드로이드의 핵심이라고 할 수 있는 커널은 공개 운영체제인 리눅스를 그 기반으로 하고 있다. 리눅스는 꽤 오랜 기간 개발되고 사용된 커널이므로 정교한 메모리 관리, 안정적인 멀티 스레드, 보안등의 고급기능을 무료로 사용할 수 있다. 모바일 환경에 어울리지 않는 기능을 제거하여 모바일 환경에 맞게 개조한 것이다. 안드로이드는 공식적으로는 자바언어를 사용한다. 검증된 많은 라이브러리를 포함하고 있어 웬만한 기능은 별도의 외부 라이브러리를 사용할 필요가 없다. 플랫폼에 내장된 프로그램과 사용자가 만든 프로그램이 동일 한 API를 사용하므로 모든 프로그램은 동등하며 기본 제공 되는 프로그램들을 사용자가 원하는 것으로 언제든지 교체할 수 있다. 개방된 개발 환경인만큼 개발 툴과 관련된 모든 문서들이 무료로 제공된다.^[2]

그림 1에 안드로이드의 구조를 나타내었다. 가장 하위에 있는 부분은 리눅스 커널에 해당하는 부분으로 하드웨어와 직접적인 인터페이스를 담당하는 부분으로 메모

리관리, 디바이스 드라이버 등의 관리기능을 리눅스 커널이 담당한다. 리눅스커널은 다양한 디바이스드라이버를 지원한다. 커널 위에는 응용프로그램들이 공통적으로 사용하는 시스템 라이브러리가 존재한다. 2D/3D그래픽, 오디오, 비디오, SQLite 등의 라이브러리를 지원하는 계층이다. Android 런타임의 경우에는 자바코어 라이브러리와 Dalvik 가상머신으로 구성된다. 안드로이드는 자바 가상머신을 직접 사용하지 않으며 모바일 환경에 최적화한 Dalvik 가상머신을 사용한다. 라이브러리와 런타임의 위에는 Application Framework가 존재하고 이것이 안드로이드 API이며 응용프로그램은 이 API를 통해 하위 커널이나 시스템 라이브러리를 호출할 수 있다. 그리고 최상위에는 사용자의 응용프로그램이 있으며 개발자들이 만들어서 사용해야할 대상이다.^[1]

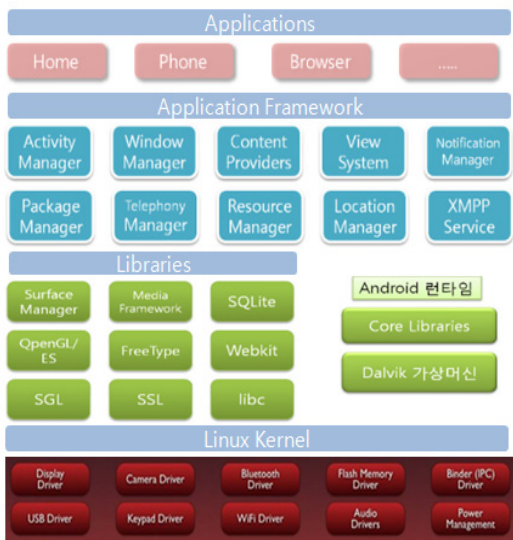


그림 1. 안드로이드 구조
Fig. 1. Android Architecture

안드로이드 어플리케이션을 개발하기 위해서는 개발 환경을 구성해야 하는데 공개되어있어서 무료로 제공받아서 사용할 수 있다. 안드로이드 기본개발언어는 자바이며 JDK를 필요로 한다. 통합 개발 환경은 이클립스를 사용한다. 그리고 ADT(Android Development Tool)를 플러그인하고 안드로이드 SDK를 설치하여 사용 한다.^[1] 실제로 폰이나 Target Board위에 올리기 전에 AVD (Android Virtual Device) 에뮬레이터가 있어 미리 실행 화면을 실행시켜 볼 수가 있다. 하지만 관련 하드웨어가

지원되지 않으면 동작 하지 않을 수도 있다.

3. 클라우드 음성인식 인터페이스

클라우드의 의미는 하늘에 떠다니는 구름(cloud)처럼 웹(web)상에 데이터·플랫폼·콘텐츠 등을 저장하는 공간과 이를 이용한 서비스를 말한다. 인터넷만 연결돼 있으면 PC나 스마트폰으로 언제 어디서나 저장된 내용을 받아 활용할 수 있다. 따라서 안드로이드 플랫폼의 음성인식엔진은 인터넷이 가능한 환경에서만 실행하여야 한다. 이는 장점이 될 수 있지만 인터넷 환경이 불안정한 모바일 기기에서는 단점으로 작용 할 수가 있다.

이러한 단점을 극복하기 위하여 본 연구에서는 안드로이드 플랫폼 음성인식 엔진을 사용하여 어플리케이션을 제작하고 그 성능을 평가 하였다. 또한, 음성인터페이스를 위한 방법론을 제시 하였다. 그림 2에서는 본 연구에서 사용 되어진 음성인식 어플리케이션 실행 아이콘을 보여주고 있다.

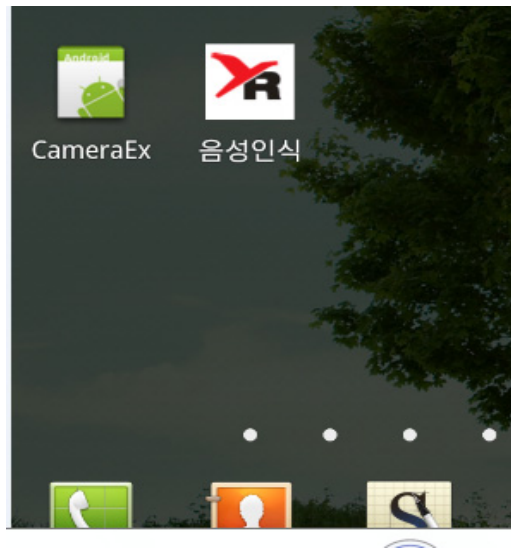


그림 2. 안드로이드 음성인식 어플리케이션 아이콘
Fig. 2. Android Voice Recognition Application Icon

그림 3과 그림 4는 본 논문에서 사용한 안드로이드 플랫폼의 음성인식엔진이다.

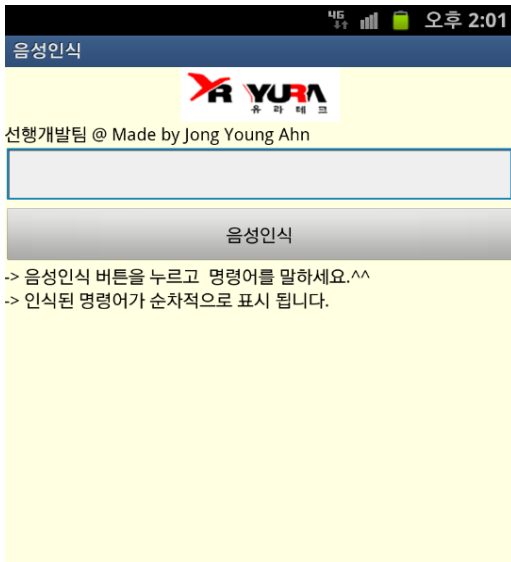


그림 3. 안드로이드 음성인식 어플리케이션
Fig. 3. Android Voice Recognition Application

에서는 인식이 떨어진다. 이는

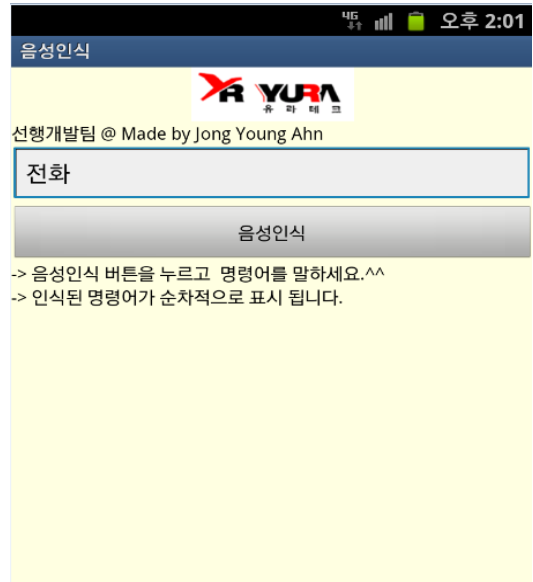


그림 5. 안드로이드 음성인식 어플리케이션
Fig. 5. Android Voice Recognition Application

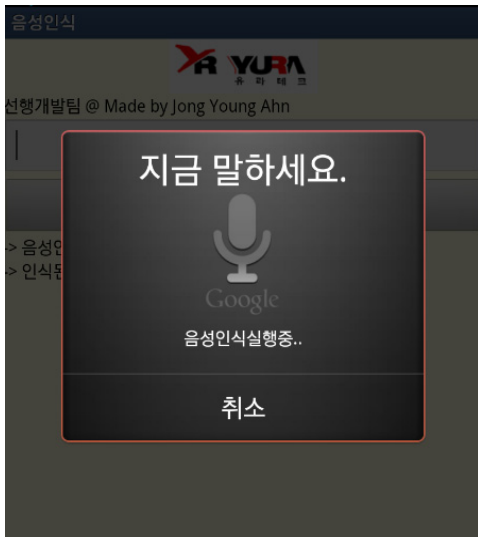


그림 4. 음성인식 실행화면
Fig. 4. Voice Recognition Execution Image

모바일 환경에서의 음성인식은 음성엔진을 클라우드 즉 웹을 통한 방법으로 인식하여 인터넷환경만 된다면 누구나 이용할 수 있는 방법이지만 실생활에서의 이용은 한계가 있다. 이동하는 모바일 기기는 이동도중 인터넷이 되지 않는 장소로 이동할 경우는 전혀 사용할 수 없다는 치명적인 단점이 있다. 그리고 여전히 생활 잡음환경

본 연구에서는 실생활에서 사용가능한 안드로이드 어플리케이션을 개발하여 실험 하였고 그림 5와 같이 인식 결과를 나타내어 그 인식성능을 평가 하였다.

다음은 실험에 사용된 인식부의 소스코드이다.^[4]

```
Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);

intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "음성
인식실행중..");

startActivityForResult(intent, REQUEST_CODE);
```

III. 실험 및 결과

실험은 모바일환경(잡음 60~80dB)에서 진행하였다. 표 1과 같이 모바일 기기에 사용 가능한 20개의 음절테이터를 사용하여 신호 대 잡음비 각 20dB, 10dB, 5dB로 실험 하였다.

표 1. 인식단어 리스트

Table 1. Recognition Word List

NO	음절
1	전화
2	예
3	아니오
4	연결
5	다음
6	이전
7	취소
8	음악재생
9	멈춤
10	정지
11	전곡재생
12	통화
13	검색
14	메뉴
15	영상통화
16	메시지
17	카메라
18	메모
19	사진
20	일정

표 2. 인식률(%)

Table 2. Recognition rate(%)

S/N 비	인식률(%)
20dB	90
10dB	60
5dB	25

표 3. 클라우드 음성엔진과 자체음성엔진 사용비교표

Table 3. Voice recognition Comparisons between Cloud and Embedded Engine

	클라우드 음성엔진	자체 음성엔진
장점	<ul style="list-style-type: none"> 엔진개발 필요 없이 사용 어플리케이션 개발이 용이함 	<ul style="list-style-type: none"> 사용지역 제한 없음 사용자 인증기능 가능(화자중속) 잡음처리 등 인식률 향상이 가능
단점	<ul style="list-style-type: none"> 인터넷 없이 사용 불가(사용지역 제한적) 잡음처리 등 인식률 향상이 불가함 	<ul style="list-style-type: none"> 엔진 개발이 필요 어플리케이션 개발이 다소 어려움

표 2에서 알 수 있듯이 조용한 환경에서는 90%의 양호한 인식률을 보였으나 실생활에 가까운 S/N 비 10dB에서는 다소 사용하기가 힘든 60%의 인식률을 보였다. 특히 신호 대 잡음비가 10dB이하에서는 인식률의 편차가 큰데 이는 아직도 주변 잡음은 인식을 수행하는데 많

은 영향을 끼친다고 할 수 있다. 특히 클라우드 음성엔진의 경우 잡음제거를 사용자가 개선하는데 한계가 있으며 표 3에서 알 수 있듯이 자체엔진사용의 경우 개발자가 잡음제거의 개선의 여지가 있는 반면 클라우드 음성엔진은 개선의 여지가 없다. 특히, 인터넷이 되지 않는 지역으로 이동시는 음성인식기능은 사용할 수가 없다는 치명적인 단점을 발견 하였다.

IV. 결론

본 논문에서는 안드로이드 플랫폼에서 클라우드 음성엔진을 사용하여 음성인식 어플리케이션을 개발하고 그 성능을 평가 하였다. 그리고 기존의 음성인식방법과 비교분석 하였다. 특히, 주변 잡음이 심한 자동차 도로 상황에서는 인식하기가 어려운 결과를 보였고 이동시 잦은 모바일 환경에서는 인터넷 연결의 문제점으로 실생활에서는 음성인터페이스의 사용이 어렵다. 또한, 주변잡음이 상대적으로 많은 지역에서는 실험화자가 다소 크게 발생해야 인식 가능한 레벨에 도달 할 수 있었으며 이는 실생활 잡음환경 하에서는 사용하기가 힘들다고 사료된다.

이에 실제로 음성인터페이스를 사용하기위해서는 인터넷망의 안정성과 더불어 잡음에 강한 음성엔진 개발이 선행 되어야만 사용자층이 보다 확대될 것이다. 현재 기존의 음성엔진 탑재 방법이 개발의 어려움은 있지만 인식률 개선을 바로 적용할 수 있어 효과적이다. 향후 음성인터페이스를 활용을 위해서는 음성인식을 통한 유·무선기기 제어의 활성화가 필요하며, 손과 발이 부자연스러운 환경에서의 음성인터페이스의 개발이 필요하다고 사료된다.

참고 문헌

- [1] J.Y..Ahn, S.B. Kim, S.H. Kim ,K.I. Her “A study on Voice Recognition using Model Adaptation HMM for Mobile Environment” The Journal of The Institute of Webcasting, Internet and Telecommunication , Vol.11 No. 3 pp.175-180. June 2011.
- [2] S.H. Kim, “Android Programming Complete Guide”, Hanbit Media Inc. 2011

[3] J.N. Woo, B.K. Lee “Android Programming”, Hanbit Media Inc. 2011
[4] FURUKAWA, Hidekazu “Android2.1 Programming

Bible”, Published by IPG inc. 2010
[5] Ed Burnette, “ Hello, Android”, ITC inc.
[6] <http://developer.android.com/sdk/index.html>

※ 이 논문은 2012년도 부천대학교 교비지원연구비에 의하여 지원된 연구의 결과임

저자 소개

김 수 훈(정회원)



- 1990년: 동아대학교 전자공학과 공학사
- 1992년: 동아대학교 전자공학과 공학석사
- 1999년: 동아대학교 전자공학과 공학박사
- 2001년 ~ 현재: 부천대학교 모바일

통신과 부교수

<주관심분야: DSP, 음성인식, 모바일콘텐츠>

안 종 영(정회원)



- 1993년: 동아대학교 전자공학과 공학사
- 1996년: 동아대학교 전자공학과 공학석사
- 2011년: 동아대학교 전자공학과 공학박사
- 1996년 ~ 2000년: 현대오토넷 전임

연구원

- 2001년 ~ 2003년: 한국폴리텍 아산캠퍼스 영상매체과 교수
 - 2004년 ~ 2006년: (주)대성전기 선임연구원
 - 현재: (주)유라테크 R&D연구소 수석연구원(부장)
- <주관심분야: 음성신호처리, 임베디드 시스템, DSP, 전장 ECU>