음성 인식 기반의 모바일 메신저 설계 및 구현

유상철*, 유병석*, 김유미*, 이유진*, 고훈**, 윤성현*
*백석대학교 정보통신학부
**(주)미디어젠

The Design and Implementation of the Mobile Messenger based on Voice Recognition

Sang-Chul Yu*, Byung-Seok Yu*, Yu-Mi Kim*, Yu-Jin Lee*, Hoon Koh**, Sung-Hyun Yun*
*Div. of Information & Communication Engineering, Baekseok University
**Mediazen, Inc.

요 약

음성 인식은 인간이 발성하는 음성을 컴퓨터 프로그램을 이용하여 문자 정보로 변환하는 기술이다. 음성은 사람마다 각기 다르기 때문에 인식률도 각각 차이가 나게 되어 범용 인터페이스로 사용되기에는 적합하지 않다. 하지만 최근 구글, 다음 등 대형 포털을 중심으로 서버 기반의 음성 인식 서비스가 제공되면서 사용자 인터페이스로 음성을 이용하는 것이 주요 이슈로 부각되고 있다. 카카오톡과 같은 메신저 프로그램은 네트워크를 이용하여 그룹 내의 사용자들 간에 메시지를 주고받는다. 여기에 사용되는 터치 자판은 간격이 좁아서 오타가 많이 발생하고, 긴 문장을 입력할 때 시간이 많이 걸리며, 운전 중에 사용할 경우 사고 위험이 높아지는 단점이 있다. 이러한 문제들을 해결하기 위해서는 음성 인식 인터페이스를 접목하는 것이 이상적이다. 본 논문에서는 음성 인식 기반의 스마트폰용 모바일 메신저 프로그램을 설계 및 구현하였다. 외부의 음성 인식 서버를 이용하여 음성을 인식하고, 인식된 음성을 텍스트로 변환하며, 채팅 서버를 통해 메시지를 전달한다.

1. 서론

최근 전 세계 스마트폰 이용자 수가 급증함에 따라 스마트폰 어플리케이션 사용량이 증가하고 있다. 스마트폰의 특성상 휴대성과 높은 활용도로 인해 다양한 모바일 서비스가 증가하고 있으며, 그 중 모바일을 이용한 메신저 서비스의 사용자는 스마트폰 사용자의 전체 76% 이상을 차지할 만큼 많은 사용자가 이용하고 있다[1].

모바일 기기를 이용한 메신저 서비스가 꾸준히 성장함에 따라서 터치 자판을 이용한 메시지 입력 방식에 대한 문제점이 많이 지적되고 있다. 스마트폰 자판은 자판 간격이 매우 좁아서 메시지를 입력할 때 빈번하게 오타가 발생하고, 이동 및 운전 중 사용으로 인한 사고 위험, 기기 조작에 익숙하지 않은 아동, 노인, 장애우 등 소외 계층에서 사용하기 어려운 단점이 있다.

최근의 서버 기반 음성 인식은 HMM(Hidden Markov Model) 엔진을 이용하여 서버에 접속하는 다양한 사용자의 음성 패턴을 모델화하여 인식 서비스를 제공한다. 사용자 개인차에 따른 음성 인식률 저하 문제를 해결할 수 있기 때문에 범용 사용자 인터페이스로 활용이 가능하다[4].

음성 입력의 속도는 일반적인 타자 입력보다 2~3배 빠른 수준을 보여주기 때문에 빠른 입력이 가능하다. 최근의 음성 인식 기술은 높은 인식률과, 대용량의 음성 데이터를 이용하여, 웹서비스, TV, PC, 테블릿, 자동차 산업 등 다양하고 복잡한 기능을 쉽게 이용할 수 있는 분야로 수요가 증가하고 있다.[2]

음성 인식 인터페이스를 사용하면, 기존 스마트폰의 작은 화면 속 자판 입력으로 인한 오타율을 감소시킬 수 있고, 이동 중이나 운전 중에 스마트폰을 이용하여 메신저를 사용할 경우에 화면을 보지 않고 음성으로 메시지를 입력함으로써 사고 위험을 줄일 수 있다. 다양한 스마트폰 사용자 계층을 고려함으로써, 아동, 노인, 장애우 등 폭 넓은 사용자에게 편리한 서비스를 제공할 수 있다.

본 논문에서는 음성 인식 기반의 모바일 메신저를 설계 및 구현하였다. 제안한 모바일 메신저는 기존 메신저 프로그램의 터치 자판을 이용한 입력 인터페이스에 음성 입력 인터페이스를 접목하였다.

입력된 음성은 서버의 음성 인식 모듈을 통해서 텍스트로 변환된다. 반환된 텍스트는 메신저 프로그램을 통해서 메시지로 전달된다.

메신저 프로그램에서, 클라이언트는 서버에 연결 요청을 하고, 요청이 수락되면 다른 사용자의 로그인 정보를 수신한다. 수신된 로그인 정보를 통해, 클라이언트가 다른 사용자들과 메시지를 주고받을 수 있도록 메신저 서버는 세션을 관리한다.

2 장에서는 서버 기반 음성 인식과 메신저 구조에 대해서 알아본다. 3 장에서는 제안한 음성 인식 모바일 메신저를 설계 및 구현한다. 4 장에서는 기능 및 구현 결과를 설명하고, 5 장에서 결론을 제시한다.

2. 관련 연구

2.1 서버 기반 음성 인식

음성 인식은 인간이 발성하는 음성을 이해하여 컴퓨터가 다룰 수 있는 문자나 코드 정보로 변환하는 기술이다. 일반적으로 마이크나 전화 등을 통해 얻어진 음향학적 신호를 단어나 단어 집합 또는 문장으로 변환하는 과정이며, 컴퓨터가 음향학적 신호(acoustic speech signal)를 텍스트로 매핑 시키는 과정이다. 인식된 결과는 명령이나 제어, 데이터 입력, 문서 준비 등의 응용 분야에서 사용한다.

<그림 1> 음성 인식 처리 과정

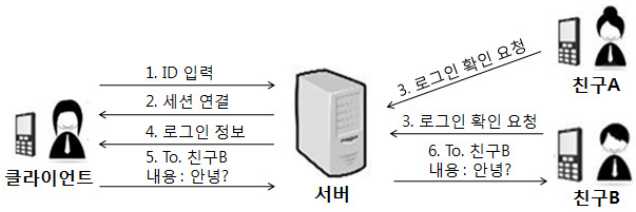
그림 1은 음성 인식 처리 과정을 보여준다. 음성 인식은 크게 전처리부와 인식부로 나눠지며 각 단계별로 작업을 수행한다.

전처리부에서는 마이크나 기기를 통해 입력된 음성이 시스템으로 들어오면 인식에 필요한 음성학적 특징 벡터를 일정 시간(보통 1/100초) 마다 추출한다. 인식부에서는 기존에 구축된 음성 모델 데이터베이스의 음성학적 정보와 전처리부에서 넘어온 특징 벡터와의 비교를 통해 인식 결과를 얻게 된다. 특징 벡터는 데이터베이스에 저장된 단어 모델, 즉 각 단어의 음성학적 특징을 음성 모델과 비교하여 높은 유사도를 나타낸 단어를 추출한다. 추출된 결과를 사용자에게 제공함으로써 음성 인식의 결과를 보여준다.

서버 기반 음성 인식은 이러한 음성 인식을 서버에서 수행하는 것인데, 사용자 개개인의 발음 차이에서 기인하는 인식률의 차이를 극복할 수 있다. HMM 엔진을 사용하여 서버에 접속하는 사용자들의 다양한 음성 패턴을 모델화하여 음성 모델 데이터베이스를 구축함으로써 사투리를 포함하여 억양이 특이한 사용자들의 인식률을 높일 수 있다. 구글, 다음, 네이버 등 대형 포털에서 서버 기반 음성 인식 서비스를 제공하고 있으며, 이러한 방식의 장점은 지도 서비스와 마찬가지로 시간이 지날수록 축적된 데이터와 경험에 의한 학습이 이루어지면서 보다 정확하고 편리한 음성 인터페이스를 많은 사람에게 제공할 수 있다는 것이다[1, 5].

2.2 메신저 프로그램 구조

서버-클라이언트 구조로 이루어진 메신저 프로그램은 클라이언트가 서버를 이용하여 다른 사용자에게 원하는 메시지를 전달한다.



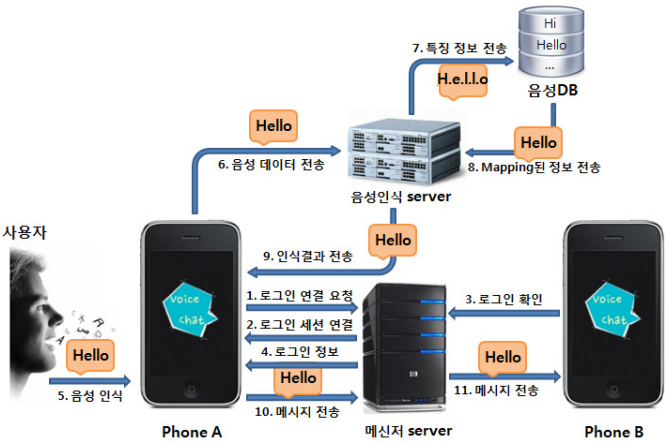
<그림 2> 메시지 전달 과정

그림 2는 메신저 프로그램을 이용하여 대화를 원하는 사용자에게 메시지를 전달하는 과정을 보여준다. 클라이언트는 메신저 프로그램을 이용하기 위하여 자신의 ID를 입력하고 서버에게 연결을 요청한다. 서버는 클라이언트로부터 전송받은 연결 신호를 확인하여 클라이언트와의 세션을 생성한다. 서버는 접속 중인 다른 사용자들의 로그인 연결 신호와 연결 완료 정보를 클라이언트에게 전송한다. 클라이언트는 현재 접속 중인 사용자의 상태를 확인하고 메시지를 보낼 사용자를 선택하여 작성한 메시지를 전송한다. 전송된 메시지는 서버로 수신되고, 서버는 클라이언트로부터 수신된 메시지를 클라이언트가 지정한 대상에게 전송한다. 서버로부터 전송된 메시지를 수신한 사용자는 메시지를 보낸 발신자와 메시지의 내용을 확인한다[3].

3. 음성 인식 모바일 메신저 설계 및 구현

3.1 모바일 음성 인식 메신저 구성도

본 논문에서는 모바일 음성 인식 메신저를 설계 및 구현하기 위하여 메신저 프로그램의 메시지 입력 인터페이스에 서버 기반의 음성 인식 처리 모듈을 접목하였다.



<그림 3> 모바일 음성 인식 메신저 구조

그림 3은 제안한 모바일 음성 인식 메신저 구조와 메시지 처리 과정을 보여준다.

Phone A는 메신저에 접속하여, 서버에게 로그인 연결을 요청한다. 서버는 요청된 연결을 확인한다. 서버는 Phone B로부터 전송받은 로그인 정보를 Phone A에게 전송한다. 사용자는 메시지 전송을 위해 기기를 통해 음성 데이터를 입력한다. 입력된 음성데이터는 음성 인식 서버로 전송되며, 서버는 사용자 음성으로부터 특징 정보(주파수)를 추출한다. 추출된 특징 정보를 음성 DB에 등록된 정보와 비교하고, 그 결과를 서버에게 알려준다. 서버는 음성 인식 결과를 Phone A에게 전송한다. Phone A는 텍스트로 변환된 음성 인식 결과를 전송 받고, 이 내용을 메신저 서버로 전송한다. 메신저 서버는 전송된 메시지를 Phone B로 중계한다.

3.2 음성 인식 모듈

서버 기반의 음성 인식은 다양한 성별, 나이, 말의 특성에 따른 내용들로 구성된 대용량의 음성 DB와 인식된 음성을 빠르게 처리하는 기술이 요구된다. 그 중 iSpeech는 다양한 플랫폼의 음성 인식 기술을 제공하고, 대용량의 음성 클라우드 텍스트를 이용하여, 높은 음성 인식률을 보여준다. 본 논문에서는 iSpeech SDK를 이용하여 아이폰용 음성 인식 처리 모듈을 개발하였다. 아이폰으로 입력된 음성 데이터를 iSpeech 서버로 전송한다. 서버는 음성 데이터로부터 특징 벡터를 추출하는 신호 처리 과정을 수행하고, 특징 벡터를 단어가 저장되어 있는 음성 데이터베이스 모델을 통해 비교 분석한다. 입력된 음성 데이터와 가장 유사한 단어를 찾아내고, 클라이언트에게 텍스트로 결과를 반환한다[7].

3.3 메신저 프로그램

메신저 서버는 SmartFoxServer를 이용하여 구현하였다. SmartFoxServer는 다양한 운영체제를 지원하며 방화벽 설정에서부터 모니터링 까지 다양한 관리 기능을 제공한다. 사용자가 원하는 대상에게 메시지를 전달 할 수 있도록 메신저의 로그인 ID를 이용하여 대상을 구분한다. 사용자는 서버와의 연결이 이루어지면 ID를 입력하고 메신저에 로그인 한다. 대상을 확인하고 메시지를 전송하면, 전송된 메시지는 서버로 전송된다. 서버는 사용자로부터 수신된 메시지를 확인하고, 확인된 메시지를 해당 수신자에게 전달한다[6].

4. 구현 결과

본 논문에서 구현한 모바일 음성 인식 메신저의 개발 환경, 구성 및 구현 결과에 대해서 기술한다.

4.1 개발 환경 및 메뉴 구성

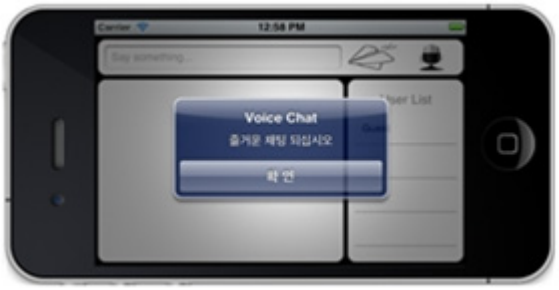
모바일 음성 인식 메신저 어플리케이션은 클라이언트 앱과 메신저 서버로 구성된다. 클라이언트 앱은 iOS 5.0 기반의 아이폰용 어플리케이션으로 Xcode 통합 개발 툴을 이용하여 Objective-C로 구현하였다. 메신저 서버는 SmartFoxServer를 사용하고, 서버는 메신저의 메시지 전송과 사용자 확인 및 채팅방 관리 기능으로 구성된다.

<그림 4> 음성 인식 메신저 메뉴

모바일 음성 인식 메신저는 그림 4와 같이 로그인, 도움말 메뉴로 구성된다. 로그인을 수행하면 채팅방이 나타난다. 채팅이 시작되면 User List를 보여주고, 음성 인식과 타자 인식을 통해 메시지를 입력할 수 있다. 입력된 메시지는 수정이 가능하며, 전송 버튼을 누르면 입력된 텍스트 메시지가 채팅방 화면에 출력된다. 도움말은 해당 어플리케이션의 간단한 도움말을 보여준다.

4.2 구현 결과

<그림 5> 메인 화면



<그림 6> 채팅 화면

그림 5는 메신저가 처음 실행되었을 때 나타나는 화면이다. 메신저를 처음 실행하면 Label을 통해 서버와의 연결 상태를 보여준다. 연결 확인 후 사용할 ID를 입력하고 로그인을 수행한다. 서버는 클라이언트로부터 메신저 접속 요청 신호를 받고, 클라이언트의 신호를 통해 적합한 사용자라면 연결을 수립한다. 그림 6은 로그인 후에 나타나는 화면이다. 우측 User List는 현재 접속한 사용자 ID를 보여준다.

<그림 7> 음성 인식 화면

그림 7은 사용자로부터 음성을 입력받기 위한 상태를 보여준다. 음성 인식 버튼을 누르면 해당 화면이 나타나고, 이 화면에서 음성을 입력하면, 입력된 음성 데이터가 서버로 전송된다. 서버에서는 전송된 음성 데이터의 신호를 처리하고, 음성 DB와 비교 분석하여, 유사도가 가장 높은 인식 결과를 다시 사용자에게 전송한다. 인식된 결과는 화면에 텍스트로 보여준다.

4.3 음성 인식 성능

본 논문에서 구현한 모바일 음성 인식 메신저는 서버 기반 음성 엔진인 iSpeech SDK를 사용하였다. 기기를 통해 인식된 음성을 서버로 전송하여 인식된 음성 데이터와 비교한다. 비교가 완료되면, 완료된 결과를 기기로 전송하고 텍스트로 출력한다.

서버 기반 음성 엔진은 HMM(Hidden Markov Model) 엔진이 많이 사용된다. 다양한 사용자의 음성 패턴을 모델화하여 인식 서비스를 제공하는 것으로, 94.8%의 인식률을 보여준다[4].

인식 내용	인식 회수	성공 회수	성공률
Hi	50	48	96 %
Good morning	50	44	88 %
I am hungry	50	42	84 %
Nice to meet you	50	39	78 %
How are you doing	50	37	74 %
Congratulations on your success	50	24	48 %

<그림 8> 음성 인식 실험 결과

그림 8은 구현된 모바일 음성 인식 메신저 어플리케이션의 음성 인식 테스트 결과 이다. 성인 남녀가 각 25번씩, 총 50번의 음성 인식을 시도하였다. 음성 인식 회수와 성공 회수를 비교하여 음성 인식 성공률을 보여준다.

실습 결과를 살펴보면 간단한 문장의 경우 높은 음성 인식률을 보여준다. 음성 인식 메시지의 내용이 길고, 발음이 어려울수록 낮은 인식률을 보여준다. 정확하지 않은 발음이나 잡음을 통해 비슷한 단어가 출력되는 경우도 있다. 원하는 단어가 아닌 비슷한 발음의 다른 단어를 보여주는 경우, 성공 회수에 포함 시키지 않았다. 사용자의 발음이 정확하거나 잡음이 적을수록 높은 성공률을 보여준다.

5. 결론

음성 인식은 다양한 산업에서 복잡하고 어려운 기능을 쉽게 이용할 수 있는 편의성을 제공한다. 스마트폰 이용자와 메신저 가입자 수의 증가로 보다 편리한 사용자 인터페이스에 대한 요구가 증대되고 있다. 다양한 사용자 계층을 수렴하고 자판 입력이 불가능한 제한적인 상황을 극복하기 위해서 음성 인식 인터페이스의 접목은 매우 적합한 해결책으로 보인다.

본 논문에서는 사용자의 음성으로 입력한 메시지를 주고받을 수 있는 모바일 음성 인식 메신저를 설계 및 구현하였다. iSpeech API를 이용하여 음성 인터페이스를 설계하고, SmartFoxServer를 사용하여 메신저 서버를 설계 및 구현하였다.

Acknowledgement

* 이 논문은 2012년도 정보통신산업진흥원 IT멘토링 팀프로젝트 지원사업으로 수행된 연구임
* 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(지역대학우수과학자사업, No. 2012-0004515)

참고문헌

[1] Eric Thelen, Stefan Besling, US6487534, Nov, 26 ,2002
[2] 이윤근, “음성인터페이스 기술 개요 및 스마트폰 환경에서의 서비스 동향”, 한국통신학회지29(4), 2012.3, 3-9
[3] Al-Sultany, Maozhen Li, Jan, Al-Raweshidy "Intelligent mobile messaging", Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on (2011), pp.2671 - 2674
[4] Bou-Ghazale, S.E, “HMM-based stressed speech modeling with application to improved synthesis and recognition of isolated speech under stress”, Speech and Audio Processing, IEEE Transactions on, pp.201 - 216, May 1998
[5] Bo cui, Tongze Xue, "Design and realization of an intelligent access control system based on voice recognition", Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on (2009), pp.229 - 232
[6] http://docs2x.smartfoxserver.com/
[7] http://www.ispeech.org/developer, ISSpeechRecognition Class Reference