

장애인을 위한 음성 인터페이스 설계

최동욱*, 이지훈*, 문남미*
*호서대학교 컴퓨터소프트웨어학
e-mail: dwchoi95@gmail.com

Designing Voice Interface for The Disabled

Dong-Wook Choi*, Ji-Hoon Lee*, Nammee Moon*
*Dept of Computer Software, Hoseo University

요 약

IT 기술의 발달에 따라 전자기기의 이용량은 증가하였지만, 시각장애인들이나 지체 장애인들이 이용하는 데에 어려움이 있다. 따라서 본 논문에서는 Google Cloud API를 활용하여 음성으로 프로그램을 제어할 수 있는 음성 인터페이스를 제안한다. Google Cloud에서 제공하는 STT(Speech To Text)와 TTS(Text To Speech) API를 이용하여 사용자의 음성을 인식하면 텍스트로 변환된 음성이 시스템을 통해 응용 프로그램을 제어할 수 있도록 설계한다. 이 시스템은 장애인들이 전자기기를 사용하는데 많은 편리함을 줄 것으로 예상하며 나아가 장애인들뿐 아니라 비장애인들도 활용 가능할 것으로 기대한다.

1. 서론

IT 기술의 발달에 따라 컴퓨터나 스마트기기 등 전자기기의 이용량이 증가하였다. 그로 인해 소셜 네트워킹, 뉴스, 날씨 등 다양한 플랫폼의 응용 프로그램들이 출시되었다. 그러나 플랫폼들이 비장애인들을 대상으로 출시되다 보니 장애인들이 사용하기에는 어려움이 있다. 예를 들어 시각장애인의 경우 응용 프로그램의 아이콘 위치와 조작 버튼을 찾는 데에 어려움이 있다. 그래서 서비스가 제공되기가 사실상 불가능하다. 또한, 지체 장애인의 경우 볼 수는 있지만, 마우스나 키보드와 같은 인터페이스를 사용하는 것에 불편함이 있다. 이렇듯 비장애인이 장애인의 보조 기기나 프로그램을 경험할 수 있는 것과는 반대로 장애인은 비장애인이 받는 서비스를 경험하는 것에 어려움이 있다.

위의 문제들로 인해 비장애인이 기본적으로 보장받을 수 있는 일들을 장애인 또한 보장받을 수 있도록 돕는 연구 및 제품 생산이 활발하게 이루어지고 있다[1-5]. 또한, 음성인식을 기반으로 시각장애인뿐만 아니라 청각장애인의 메시지를 전송하는 시스템도 구현되었다[2].

위의 결과들을 참고하여 본 논문은 STT(Speech To Text)를 이용하여 사용자가 PC에 원하는 명령어를 내릴 수 있도록 하며, 해당 명령에 대한 응답으로 TTS(Text To Speech)를 이용하여 사용자가 소리를 통해 직관적인 인터페이스를 사용할 수 있도록 한다. 또한, 장애인이 비장애인과 공통적인 응용 프로그램을 사용함으로써 서로 간의 거리를 더 좁힐 수 있는 계기를 마련하고자 한다. 장애인과 비장애인의 의사소통을 위하여 사용하는 보조 기기는 많이 존재하지만, 본 시스템은 서로의 의사소통은 물

론, 장애로 인해 기존에 경험하기 어려운 다양한 서비스들을 경험할 수 있도록 해준다. 더불어, 인터페이스 관련으로는 음성인식과 음성 합성 기술을 통해 평소 장애인들이 PC를 조작하는 데에 느끼는 어려움을 해소할 것이다.

2. 관련연구

2-1 STT를 이용한 음성인식

STT(Speech To Text) API는 심층 신경망(Deep Neural Network, DNN)을 활용한 인공지능 학습에 기반을 둔 음성인식 기반 음성/텍스트 전환 시스템이다. STT는 기계가 마이크를 통해 사람의 음성을 수신한 후 분석을 위해 음향신호 처리의 과정을 거치게 된다. 처리 과정을 거친 음성의 특징을 추출하여 사전에 구축된 음성 모델 데이터베이스와 비교하여 유사도가 높은 것을 텍스트로 변환하게 된다[6].

2-2 TTS를 이용한 음성 합성

TTS(Text To Speech)는 음성의 음파를 기계가 자동으로 만들어 내는 음성 합성의 기술을 활용한 텍스트 문장을 자연스러운 음성으로 출력하는 시스템이다. 그러므로 TTS를 사용하면 웹서버나 메모장 등의 사용자가 원하는 내용을 음성으로 출력할 수 있게 된다.

2-3 크롤링(Crawling)

크롤링이란 무수히 많은 컴퓨터의 문서를 수집하여 검색 대상의 색인으로 포함하는 기술이다. 크롤링은 파인션 언어를 사용하며 주로 웹 크롤링을 할 때 사용된다.

2-4 Snowboy library를 이용한 시스템 호출

Snowboy는 맞춤형 Hotword 감지 엔진이다. 그리고 Snowboy는 실시간으로 실행되며 온라인이 아닌 오프라인 상태에서도 항상 실행 중이다.

2-5 명령어 종류

음성인식을 통한 명령은 크게 Execute, Read, Write, Search로 나뉜다.

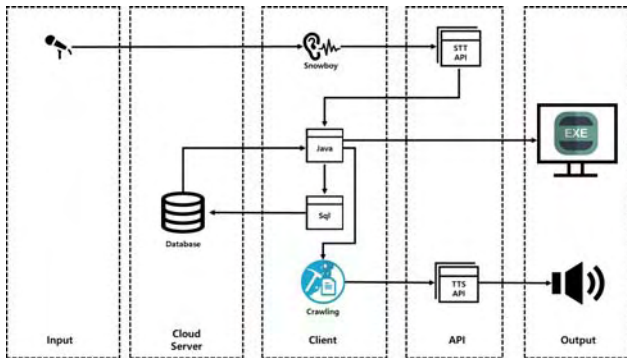
Execute: 실행 명령은 exe의 응용 프로그램들을 실행해주는 것이다.

Read: 읽기 명령은 사용자가 원하는 문서나 웹의 내용을 읽어주는 것으로 Crawling 기능과 TTS 기능을 통해 구현된다.

Write: 쓰기 명령은 SST 기능을 통해 음성을 바로 텍스트 문서로 저장해준다.

Search: 검색 명령은 STT를 통해 음성을 인식하여 텍스트로 변환한 뒤 변환된 내용을 검색엔진을 통해 검색한다.

3. 음성 인터페이스 설계



(그림 1) 시스템 구성도

(그림 1)은 제안한 음성 인터페이스의 전체 구성도이다. 본 시스템은 부 시스템과 주 시스템으로 나뉘며 부 시스템은 오프라인에서의 음성인식을 통해 주 시스템을 실행하는 것이다. 사용자가 PC에 마이크와 스피커를 연결하고 시스템을 설치한다. 시스템이 설치되면 사용자가 음성으로 Snowboy를 이용하여 주 시스템을 실행한다. 사용자가 마이크를 통해 명령하면 Google Cloud의 STT(Speech To Text) API를 통해 음성이 텍스트로 전환된다. 전환된 음성으로 Server에 있는 Database에서 명령어를 검색하여 명령어에 따라 처리해주며 처리결과를 스피커를 통해 음성으로 알려준다.

3-1 음성 인터페이스 pseudo

(그림 2)는 음성을 텍스트로 변환해주는 방식을 설명하고 있다. 마이크로 받은 음성을 헤르츠, 비트, 오디오인코딩을 통하여 audio data로 변환해준다. 변환된 audio data를 STT API로 보내주면 배열 형태의 데이터를 보내준다. 그림 그걸 처리하여 문자열로 변환한다.

(그림 3)는 (그림 2)에서 처리된 text를 조회하여 일치하는 명령어를 찾는 것이다. 실행해달라는 명령어일 경우

원하는 프로세스의 exe 파일을 실행해준다. 또한, 읽어달라는 명령어의 경우 crawling을 통해 원하는 내용을 검색해서 text의 형태로 받아온다. 일치하는 명령어가 없거나 잘 못 인식된 경우에는 (그림 2)가 재실행 되며 모든 명령어에 대한 실행 여부를 (그림 4)을 통해 스피커로 출력시켜 사용자가 인식할 수 있도록 해준다.

(그림 4)은 (그림 3)에 의해 처리된 내용을 알려주기 위하여 text를 받아와 TTS API를 통해 처리하고 처리된 데이터를 오디오 파일로 인코딩해준다. 그 후 해당 오디오 파일의 재생을 통해 스피커로 출력된다.

```

Input : speech
Output : text

speechToText
{
    audioData <- speech
    sttAPI <- audioData
    responseList <- sttAPI
    for audioList exist
    {
        result <- responseList.getResultList().get(0)
        alternative <- result.getAlternativesList().get(0)
        text <- alternative.getTranscript()
    }
    Command(text)
}

```

(그림 2) Speech To Text pseudocode

```

Input : text
Output : exefiles, crawling

Command(text)
{
    if text is execute then
        execute exefiles
        textToSpeech("exefiles is Executed")
    else if text is read then
        execute crawling
        textToSpeech("crawling is Started")
    else
        textToSpeech("Would you repeat again?")
        speechToText()
}

```

(그림 3) Command pseudocode

```

Input : text
Output : speech

textToSpeech(text)
{
    ttsAPI <- text
    response <- ttsAPI
    audioContents <- response.getAudioContent();
    fileOutputStream <- audioContents
    fileOutputStream.play()
}

```

(그림 4) Text To Speech pseudocode

4. 결론

대부분의 응용 프로세스들은 키보드나 마우스 혹은 터치를 통해 작동되므로 많은 장애가 있는 사람들이 전자기기의 응용 프로세스를 사용하는 데에 많은 불편함을 호소하고 있다. 그래서 음성인식을 통해 응용 프로세스를 제어해보고자 음성 인터페이스를 제안한다.

음성 인터페이스는 음성으로 대부분의 응용 프로세스들을 동작할 수 있게 하여 시각장애인과 지체 장애인에게 특히 더 도움을 주었다. 또한, 장애인이 비장애인의 다양한 플랫폼과 프로그램을 실질적인 주체가 되어 사용할 수 있도록 한다.

앞으로는 음성 인터페이스에 동작 인식 기능을 추가하여 언어 장애를 복합적으로 지닌 장애인 또한 서비스를 받을 수 있도록 할 계획이며 더 나아가 장애인도 비장애인처럼 전자기기를 사용하는 데 불편함이 없이 사용할 수 있는 시스템을 구현할 것이다.

ACKNOWLEDGEMENT

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2017R1A2B4008886).

참고문헌

- [1] Yeo Kyung Yoon, Sung Hoon Kim "A Study on Mobile Application UX/UI Design of Communication for the Hearing & Language Impaired." JOURNAL OF THE KOREAN SOCIETY DESIGN CULTURE, 21.2
- [2] Sung-jin Kim "Speech Recognition based Message Transmission System for the Hearing Impaired Persons" JKIIICE, Vol. 22, NO. 12
- [3] Ki-Hyung Hong "UI/UX of Voice Interface for People with Disabilities:Focusing on Spoken Command Word Recognizer and Speech Synthesizer" Telecommunications Review
- [4] Honglae Lee, Joongnam Jeon and Sukil Kim "A Development of the Braille User - Interface on PC Communication Program " 한국정보과학회 학술발표논문집, Vol. 23, No. 1A
- [5] 홍경순. "[특집]시각장애인을 위한 정보통신 보조기술과 국내 현황." 전자공학회지, 32.3
- [6] 김진태, 정훈. "음성인식 기술의 동향과 해군 정보통신 분야 적용방안." Defense & Technology, .456 (2017.2): 120-127.