
오픈소스기반의 지능형 개인 도움시스템(IPA) 개발방법 연구

김길현* · 김영길*

*아주대학교

A Study on the Intelligent Personal Assistant Development Method Base on the Open Source

Kil-hyun Kim* · Young-kil Kim**

*Ajou University

E-mail : remnant24c@ajou.ac.kr

요 약

최근 시리(siri)와 같이 사람의 말을 인식하고 대답해주는 서비스를 스마트폰 혹은 웹서비스로 제공하고 있다. 이러한 지능형 처리를 위해서는 음성을 받아 드리고 웹상의 빅데이터를 검색하고 구문으로 분석, 정확도 부여등의 구현이 필요하다. 본 논문에서는 공개된 소스를 기반으로 하여 음성인식하는 ASR(Automatic Speech Recognition), 질문에 대한 내용을 데이터로 분석하고 응답을 만드는 QAS (Question Answering System), 결과를 음성으로 전달하는 TTS(Text to Speech) 로 구분하고 하나의 시스템으로 구현하는 연구와 분야별 적용될 수 있는 방법을 제안하고자 한다.

ABSTRACT

The latest the siri and like this is offering services that recognize and respond to words in the smartphone or web services. In order to handle intelligently these voices, It needs to search big data in the cloud and requires the implementation of parsing context accuracy given.

In this paper, I would like to propose the study on the intelligent personal assistant development method base on the Open source with ASR(Automatic Speech Recognition), QAS(Question Answering System) and TTS(Text To Speech).

키워드

IPA, Intelligent, Assistant, Opensource

1. 서 론

음성인식의 경우 1952년 미국 통신업체 에이티엔티(AT&T) 벨연구소(Bell Laboratories)의 음성으로 말하는 숫자 인식 시스템 '오드레이(Audrey)'의 첫 시작이 되었다. 이후 IBM의 '슈박스(shoebox)'를 통해 16개의 영어 단어를 인식할 수 있는 장비를 개발하게 된다. 70년대 미 국방성 산하에 국방첨단연구사업국(DARPA)을 통해 음성 이해연구(Speech Understanding Research)가 진행된다. 이것을 1세대, 2세대라고 보게 된다. [1]

2000년 후반이 되면서 휴대용 장비들의 등장

사람 말을 인식할 수 있는 것이 일반인의 생활 속으로 들어오게 된다. 현재 2016년 시점에서 가장 직접적인 것이 바로 핸드폰으로 자신의 집으로 전화를 걸고, 아이폰 Siri를 통해 오늘 날씨를 안내받을 수 있는 기술이다.

Facebook가 2015년 1월 인공지능 HW, SW를 오픈소스화한 것을 시작으로 볼 수 있다. [2]

위와 같이 한 기업에서 시작한 것도 있지만, 이러한 많은 것이 한 번에 일어난 것이 아니라 대학과 연구소와 각종 작은 분야로 나누어져 분야별로 음성인식, 인공지능, TTS, ASR, QAS 등의 시스템으로 개별적으로 개발되었다. 위와 같은 발

전된 기술들을 하나로 통합하여 Sirius라는 오픈 소스 프로그램도 나오게 된다. (2015년 Lucida로 명칭 변경)[3] 그러나 Sirius 소스 역시 Linux에서 자동 설치 스크립트로 만들어 놓았지만, 모듈별 연동에 대한 부분으로 인해 쉽게 수정 및 컴파일 이 되지 않았다.

이 논문에서는 단계별 오픈소스를 개별 컴파일 시키고 Text 파일을 통해 연동하였으며 분야별로 적용될 수 있는 부분을 제안하였다.

II. 본 론

가. ASR(Automatic Speech Recognition) - Pocketsphinx

Sphinx는 Open Source Speech Recognition Engines 으로 CMU Sphinx Group은 DARPA (Defense Advanced Research Projects Agency)의 재정지원으로, 음성인식과 관련 대화 시스템, 음성합성과 같은 분야에서 툴과 application 개발을 공개하고 있다.

이에 따른 Sphinx Toolkit은 음성인식 기능의 지속적인 개선으로 2000년대 후반 동안 카네기 멜런 대학 (CMU)에서 만들어졌다. [4]

Pocketsphinx 설치 이전에 기타 패키지들이 필요하다. 최신의 소스는 git을 통해 내려받는다.

```
$ sudo apt-get install git automake libtool
bison python-dev swig make pkg-config
$ git clone
https://github.com/cmusphinx/sphinxbase.git
```

1) Sphinxbase (CMU Sphinx common Libraries)

이 패키지는 CMU Sphinxtrainer 와 모든 Sphinx decoders(Sphinx-II, Sphinx-III, and Pocketsphinx) 그리고 오디오 파일과 음성 특징을 다루는 일반적 기본 라이브러리가 포함되어 있다.

```
[5]
$ ./autogen.sh
$ ./configure
--prefix=/my/own/installation/directory
&& make && make install
```

2) Pocketsphinx

Sphinxbase 패키지가 설치되었다면 위의 절차와 동일하게 작업진행 할 수 있다.

라이브러리 경로를 환경변수를 지정하여 준다. (libpocketsphinx.so.3)

3) 동작을 통해 확인

```
$ pocketsphinx_continuous -inmic yes
-hmm Directory containing acoustic model files.
```

나. QAS(Question Answering System) - OpenEphyra

질의-응답 시스템(QAS)은 사용자가 발화한 것을 자연어 처리(Natural Language Processing)하여 질의로 만들고 각종 문서나 웹 정보를 검색하여 사용자에게 답변에 대한 리스트를 보여주고 원하는 것을 선택할 수 있게 해주는 시스템이다.

현재 웹상에 소스가 공개된 것은 YodaQA[6], START[7], OSQA[8], QANUS[9] 등이 있다. 그러나 실제로 C나 Java로 구현할 수 있는 Opensource 인 OpenEphyra 구현을 진행하였다.

1) Indri

OpenEphyra 내에는 Indri이라는 문서파일의 구문분석하여 검색할 수 있는 검색엔진이 포함되어 있다. Indri는 질의 Inquiry query를 기반으로 잘 디자인된 질의 언어이다. [10] 이것은 단순한 키워드와 복잡한 질의를 처리할 수 있다. 구조화되지 않은 문서도 가능하지만, 검색엔진은 state-of-the-art text search rich structured query language 제공한다.

본 연구에서는 Wikipedia Dump 11GB 자료를 검색하기 위해서 사용하고 있다.

Indri 5.5 이상 버전 컴파일을 위해서 Java를 이용할 경우 --enable-java 옵션과 JDK 설치된 경로를 옵션으로 주어야 한다.

```
$ ./configure --enable-java
--with-javahome=/usr/lib/jvm/java-8-oracle
$ make
실행이 되면 libindri_jni.so 파일이 만들어진다.
이 라이브러리를 OpenEphyra/lib/search 에 복사한다.
```

2) OpenEphyra

OpenEphyra 는 최소의 QAS 오픈 프레임워크이다. 이것은 자연어 지문을 웹과 다른 소스원으로부터 정확한 답을 얻는다. 이 프레임워크는 TREC 평가에 참여한 CMU's Ephyra System의 많은 알고리즘을 적용하고 있다. [11]

SourceForge 사이트에서 최신 소스(0.1.2)를 내려받는다. 소스를 컴파일하기 위해서는 Eclipse를 사용하여 ant 실행하기를 권장하고 있다.

이클립스에서 "File->New->Project->Java->Java project existing ant buildfile" 선택하고 소스 상에 build.xml 를 선택한다. ant build를 통해 실제 build를 할 수 있다.

외부에서 만들어진 라이브러리를 LIB_LIBRARY_PATH에 연결해 주고 컴파일을 위해서 이클립스와 외부 jar를 연결해준다. 실행 시 CLASS PATH를 맞추어 준다면 실행은 가능하다. 그러나 현재 0.1.2 버전의 OpenEphyra 에서는 Indri 검색과 BingAPI를 검색할 수가 없다.

BingKM.java를 Version2에 맞게 자료형 구조체

를 만들어 주어야 한다. 그러나 인터넷상에 나온 자료와 실제 자료를 API를 받아온 응답 구조체는 다르다는 것을 알 수 있었다. SearchResult 기본 String 결과이지만, Web은 다시 배열로 되어 있다. 그렇기에 다시 새로운 구조(class SearchResult_Web)를 만들어 준다. [12]

indriKM.java 경우 Indri 인덱싱에서 생성될 때 [p] section이 없어 졌기 때문에 제거해주고 combine 질의를 주게 되면 인덱싱된 자료에서 검색된 정보를 얻을 수 있다.
qs = "#combine(" + qs + ")";

다. TTS(Text To Speech) - eSpeak

eSpeak는 "formant synthesis" 방법을 사용한다. 이것은 작은 용량에도 불구하고 많은 언어 지원이 가능하게 한다. 프로그램에서 자국의 언어를 지원하지 않을 경우 프로그램을 수정하여 발음을 추가할 할 수 있다. [13] 명확하게 말하고 빨리 말하기도 가능하다. 그러나 사람처럼 자연스러운 발음은 아니다.

이 논문에서는 Linux 기반으로 설치를 진행한다. 먼저 오디오 라이브러리를 설치하여 문제소지가 있을 것을 제거한다.

- (1) apt-get install libportaudio-dev
- (2) 임베디드용(arm)이 아니라면 make 하여 빌드 후 생성되는 시킬 수 있다.
- (3.1) 문자열로 실행
\$ espeak-ng "Thank you."
- (3.2) text 파일 저장된 내용 실행
\$ espeak-ng -f filename

라. 프로그램 연동 방법

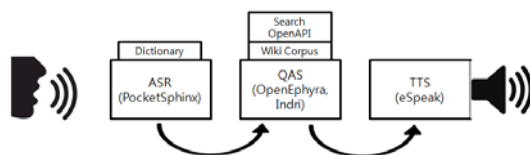


그림 1. 연동을 위한 기본 블록도

사용자의 말하는 것을 ASR을 통해 입력을 받으며 QAS 시스템에서 자연어를 분석 및 질의문을 생성하고 질의문을 통해 Wiki, Web 자료를 검색을 통해 답변을 찾는다. 마지막으로 TTS를 이용하여 결과를 음성으로 알려준다.

부분마다 처리되는 것을 어떻게 서로 전달할 것인가를 생각해보아야 한다. 이것을 구현하는 방법으로 아래와 같은 방법이 있을 것이다.

- (1) Text 파일을 통해 문자열 공유

- (2) JNI 이용한 파라미터 문자열 공유
 - (3) sqlite, mysql 과 같은 DB를 이용한 공유
- 이 논문에서는 Text 파일을 이용하여 연동 가능성 유무를 확인하려 하였다.

사용자 음성을 마이크에 "Where is the Seoul" 이라고 할 경우, ASR은 음성신호를 전달받아 사전(dictionary)정보를 통해 오류를 최소화하게 된다. 그리고 in.txt 파일을 생성한다. in.txt 내용은 아래 예문이 들어간다.

(예문) Where is the Seoul

OpenEphyra 는 in.txt 파일을 스레드(Thread)로 파일의 변화를 감지한다. 변화가 감지되면 QAS은 고유명사 인식기술(WordNet)을 이용하여 Place 등으로 구분한다. [14] 그에 대응하는 질문 질의(Query)로 생성하여 웹과 Wiki 자료를 검색한다. (본 구현에는 BingAPI 와 Wiki dump가 이용되었다)

Where is the Seoul 에 생성된 질의어는 아래와 같다.

- seoul (seoul OR "capital of South Korea")
- "the seoul is in"
- "the seoul is at"

표 1. Where is the seoul 검색결과

순위	Bing API	Wiki dump
1	South Korea Score : 0.984	Seoul Score : 3099
2	Myeongdong Score : 0.42	South Korea Score : 1609
3	Hanja Score : 0.056	Korea Score : 1319
4	Nom Hangeul Score : 0.056	Korean Score : 648
5	US Score : 0.028	Japan Score : 592

위와 같이 차이가 나기 때문에 점수가 높거나 우선순위가 높다고 해서 그것을 답변으로 취해서는 안 될 것이다. 우선 ASR에서 정확하게 인식하여 문자열로 만드는 것이 첫 번째 핵심이다. 다음으로 정확한 질의어를 생성하는 것이다. 이에 대한 문제는 다른 연구에서도 제시된 바 있다. 그러므로 좀 더 정확한 질문을 생성하여 답변을 만드는 방식을 제시하고 있다. [15][16]

검색된 자료를 out.txt 파일로 만들고 이것을 eSpeak가 읽어서 스피커로 출력하게 된다. (본 구현에서는 최우선 순위의 내용을 전달한다.)

마. 분야별 인식률을 높이려는 방법 및 Control 기능 추가 제안

사용자의 음성을 받아드리는 것은 라. 부분과 같다. 그러나 Classify 프로그램이 새롭게 추가된다. 이 프로그램은 들어온 Text 정보를 분석하여 제어(Control)가 필요한 명령인지 혹은 유하원칙(5W1H)의 답변을 요구하는 명령어인지 구분하여

답변을 요구하는 질문이라면 in.txt 파일을 생성하고 QAS가 in.txt 파일을 스레드로 감시하며 읽어 드리게 된다. 이후 처리는 라. 에서 구현된 방법과 같다.

Classify Program에서 제어명령어가 들어오게 되면 Text 파일은 control.txt 파일을 생성하고 Control Program은 스레드를 통해 그 파일을 읽는다.

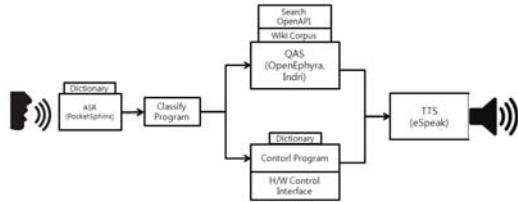


그림 2. Control 기능이 추가된 블록도

읽어 드린 파일은 사전정보를 통해 분야별 제어에 맞는 장비단어인지 필터링(filtering)한다.

예로 가정집에서 하드웨어를 제어할 경우, "Happy, TV On"이라고 말할 수 있다.

표 2. Control 을 위한 프로토콜

시작알림	장비 종류	행동
Happy	TV	On

제어하고자 하는 Hardware는 여러 종류가 될 수 있고 그 안에 여러 회사 제품이 있을 수 있으므로 Interface로 만들어 사용하게 한다.

Home Network, Factory Automation, Hospital 등 특정 분야로 나눌 수 있게 되며 TV와 같이 중분류로 다시 나눌 수 있다. 또한, Samsung, LG, GE 등 소분류로 나누어질 것이다. 그러나 중분류에서 이미 행동은 같기 때문에 같은 Interface를 제공하여 On이라는 명령어를 내리게 되면 켜지는 동작을 하게 된다. 즉, 리눅스의 VFS(Virtual File System)와 유사한 계층(Layer)을 두어 On 명령어로 동작한다. 장비는 다르지만, 동작하는 명령어는 같다는 것이다.

Hardware Control Interface의 경우 모듈 타입으로 개발하여 다양한 장치를 연결하고 제거할 수 있도록 개발하는 것을 제안한다.

III. 결 론

이 논문에서는 깊은 차원의 자연어처리 이론, 필터처리 알고리즘을 다루고 있지 않다. 다만 기술기반이 오픈되어 있다는 전제하에 프로그램적으로 어떻게 구현할 것 기술되어 있으며 각 프로그램 간에 어떻게 연결할지 고민하였다. 그리고 차후 더 발전할 수 있도록 분야별 인식률을 높이고 하드웨어 제어 부분도 연동할 수 있도록 방법

을 제안하였다.

Pocketshpinx, OpenEphyra, eSpeak 등과 같은 오픈소스의 자료는 여러 사람이 개발하기 때문에 내부적인 함수가 바뀌어 동작이 되지 않을 수 있다. 그러므로 버전이 높아져도 문제없이 출력물을 토대로 이를 연동하는 것으로 구현하였다.

git 혹은 소스 압축파일을 내려받아 컴파일시키도 문제없이 개별 파일 혹은 라이브러리만 바꾸어 구현 시키는 것이 하나의 목적이 된다. 또한, H/W Control 분야로 발전 가능성을 제안하여 현재 가정용으로 대답만 하는 시스템이 아니라 IPA(Intelligent Personal Assistant)가 분야별 기능을 할 수 있도록 기술을 제안하였다.

참고문헌

- [1] 한국 콘텐츠 진흥원, 음성인식 기술의 동향과 전망, 문화기술(CT) 심층리포트, 2011.11
- [2] 이은옥, 문형돈, 인공지능 오픈소스 현황과 시사점, 한국통신학회 하계종합학술 발표회, 정보통신기술진흥센터, 2016
- [3] <http://sirius.clarity-lab.org/>
- [4] <http://cmusphinx.sourceforge.net/wiki/tutorialpocketsphinx>
- [5] <https://github.com/cmusphinx/sphinxbase>
- [6] <http://live.ailao.eu/>
- [7] <http://start.csail.mit.edu/index.php>
- [8] <http://www.dzonesoftware.com/products/open-source-question-answer-software>
- [9] <http://www.qanus.com/>
- [10] <http://lemurproject.org/lemur/IndriQueryLanguage.php>
- [11] <https://mu.lti.cs.cmu.edu/trac/Ephyra/wiki/OpenEphyra>
- [12] <https://msdn.microsoft.com/en-US/library/mt707570.aspx#composite>
- [13] Richard Pronk, Adding Japanese language synthesis support to the eSpeak system, University of Amsterdam, 2013
- [15] Joseph Chang, Minimally Supervised Question Classification and Answering based on WordNet and Wikipedia, Yuan Ze University
- [15] 김민경, 질의문 자동생성방식의 질의응답시스템의 설계 및 구축, 서울시립대학교 대학원, 2009
- [16] 임경민, 질의응답시스템을 위한 질문유형 추출 및 질의 변환방법, 한양대학교 대학원, 2007