휴머노이드 구동을 위한 딥러닝 음성인식 인터페이스 개발

백지영[1], 심재훈[1], 박재흥[1]
[1] 서울대학교

Development of Interface for Humanoid Using Voice Recognition and Deep Learning

Baek Jiyeong[1], Sim Jaehoon[1], Park Jaeheung[1]
[1]Seoul National University
e-mail: jy951111@snu.ac.kr, simjeh@snu.ac.kr, park73@snu.ac.kr

요 약

과학 기술의 빠른 발전은 인간과 능동적으로 교감할 수 있는 로봇의 개발을 가능하게 하였고 다양한 로봇들이 인간 생활 속으로 들어오게 되었다. 이에 따라 로봇에 대한 전문적인 지식이나 별도의 기술 교육 없이 로봇을 구동할 수 있는 플랫폼이 필요하게 되었다. 본 논문은 Human-Robot Interaction(HRI) 분야에서 많이 사용되는 음성인식을 로봇 구동 인터페이스에 적용시켜 보고자 하였다. 사람이 음성으로 명령을 내리면 딥러닝 기반 Char CNN 알고리즘을 통해 이를 인식하고 명령에 맞는 DYROS JET 의 제어기 command 에 전달하게 된다. 본 연구는 ROS 와 연결된 V-REP 시뮬레이터에서 DYROS JET 모델을 통해 실험적으로 검증하였다.

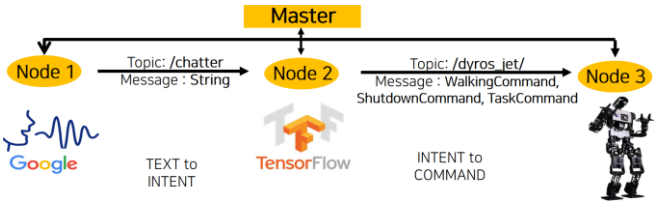
1. 서론

최근 다양한 로봇이 빠르게 인간 생활 속으로 들어오면서 인간과 능동적으로 교감할 수 있는 형태로 발전하고 있다. 음성인식과 자연어 처리는 Human-Robot Interaction (HRI) 분야에서 많이 사용되는 기술로 최근 다양하게 출시되는 AI 스피커에서 쉽게 발견할 수 있다. 로봇에 단순히 프로그램을 통해 명령을 주기 보다는 비전문가도 별도의 프로그램 스킬 없이 로봇에게 명령을 내릴 수 있는 시스템이 필요하다. 또한 기존의 햅틱스나 입는 장비와 같이 별도의 조작장비가 없어서 사용자가 작동 방법에 대한 숙지나 구동 구조에 대하여 파악할 필요가 없기 때문에 일반인도 쉽게 사용이 가능하다. 본 논문에서는 사용자의 말의 의도를 파악하는 자연어 처리를 통한 딥러닝 음성인식 인터페이스를 만들었고 이를 로봇에 적용시켰다.

본 논문에서는 음성인식(speech to text)을 위해 Google cloud의 Speech API를 사용하였다. 컴퓨터가 사람의 의도를 파악할 수 있도록 하기 위해서는 딥러닝을 통한 자연어 처리 과정이 필요하다. 이를 위해 Convolution Neural Networks(CNN), Recurrent Neural Network(RNN), Long Short Term Memory networks(LSTM) 등 많은 알고리즘이 개발되었다. 본 연구에서는 텍스트를 벡터로 임베딩하여 텍스트의 특징을 뽑아내는 Char CNN을 택하였다. Char CNN은 일반적인 다른 알고리즘과 비교하여 긍정/부정에 따른 영화리뷰 분류, 소비자 리뷰 예측 등에서 압도적 성능을 보인다 [1]. 음성인식기와 Char CNN 모델은 모두 파이썬과 텐서플로우 상에서 구현되었으며, 전체적인 프레임워크 구축을 위하여 Robot Operating System(ROS)을 사용하였다. 시뮬레이션은 ROS와 연결된 V-REP 시뮬레이터에서 DYROS JET 모델을 사용하여 진행되었다.

2. 본론

2.1 전체 시스템 구조



[그림 1] System Overview

제안하는 시스템은 크게 3가지 구성으로 이루어졌다. 구글의 음성인식기와 Char CNN 모델은 ROS package 안의 각각의 node가 되어 메시지를 주고받는다 [그림 1].

Node 1은 google speech to text api를 이용하여 구현한다. USB 마이크에 대고 말한 음성이 문장으로 인식되면 ROS 프로그램에서 이를 '/chatter'라는 이름의 토픽으로 발행하게 된다.

Node 2는 구독자이자 동시에 발행자이다. 구독할 때 토픽 이름을 마찬가지로 '/chatter'로 설정해주면 이전 프로세스에서 보내는 음성 메시지를 받을 수 있다. 이를 의미있는 형태소로 나누고 이렇게 나눈 형태소를 학습된 데이터 집합에 맞추어 Char CNN 모델로 디코딩을 수행한다. 디코딩된 결과값을 로봇 시뮬레이터의 제어기로 전달한다.

Node 3은 로봇 시뮬레이터인 V-REP과 직접 연결

되어 Node 2로부터 전달받은 command에 따라 DYROS JET를 구동할 수 있다. 프로그램을 종료시키는 Shutdown, 걸어가는 Walking, 팔을 움직일 수 있는 Task Command로 Node 3이 구성되어 있다.

2.2 자연어 처리 모델

2.2.1 데이터 전처리

데이터 전처리를 위해 한국어 정보처리를 위한 파이썬 패키지인 KoNLPy를 사용하였다. 그 중에서 Mecab 분석 엔진으로 한국어 형태소 분석을 진행하였다. 문장에서 명사, 동사 등 실질적인 의미를 가지는 형태소만 추출하여 데이터 크기를 줄이고 성능을 향상시켰다. 예를 들어 “왼 손 앞으로 뻗어”라는 문장이 들어오면 ‘왼손’, ‘앞’, ‘뻗’ 세 개의 의미 있는 형태소만 데이터로 가지게 된다.

2.2.2 Char CNN

Char CNN은 Convolutional layer(CONV)와 max-pooling layer(POOL) 를 번갈아 수행하여 최종적으로는 fully connected layer(FC)를 통해 분류를 수행하는 모델이다. [INPUT-CONV-RELU-POOL-FC] 구조로 모델을 구축할 수 있다.

INPUT 단계에서는 Word2Vector를 이용하여 문장 내의 단어들을 0~1 사이의 벡터 표현으로 맵핑하고 나열하여 2차원의 이미지 배열처럼 만든다.

CONV 레이어는 INPUT의 출력을 다중 필터를 통해 반복적으로 가중치에 따른 내적 연산을 수행하고 이때의 입/출력 볼륨은 동일하다. 입력 데이터의 특징을 추출하여 가장 핵심적인 레이어이다.

RELU 레이어는 활성화 함수로 비선형성이 적용된다. CONV와 마찬가지로 결과 볼륨의 크기를 변화시키지 않는다.

POOL 레이어는 다운샘플링을 수행하여 유일하게 볼륨이 줄어든 결과를 출력할 수 있기 때문에 공간적 크기를 줄여 오버 피팅을 제어한다.

마지막 FC 레이어는 출력 레이어로 앞에서 입력받은 다차원 텐서를 1차원으로 피는 작업을 한다. 이후 Dropout 정규화를 통해 무작위로 일부 network를 생략하여 오버 피팅을 막고 Softmax 함수를 통해 설정한 각 클래스별 확률을 나타내어 최댓값을 가지는 클래스로 예측을 수행한다.

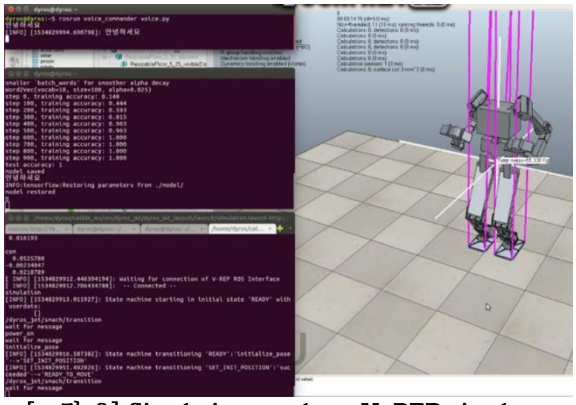
2.2.3 학습 및 실험과정

DYROS JET 구동을 위해 6종류의 명령으로 나누었고 [그림 2]와 같은 학습 데이터 집합을 주었다.

마지막으로 학습 루프를 작성한다. 데이터 배치를 반복하고 주기적으로 모델을 평가하고 가중치를 저장한다. 기본 값은 100회 당 한 차례 모델을 평가하도록 설정하였고, 이 값은 조정할 수 있다. 또한, predict 과정에서 모델의 threshold=0.9로 설정하여 예측에 포함해야 하는 신뢰도 수준을 조정하였다.

encode	decode
'멈춰', '그만', '그만 가', '그만 걸어가'	0
'앞으로 가', '앞으로 걸어가', '걸어가'	1
'왼손 들어', ' 왼손 들어봐', '왼손 위로 들어', '왼손 올려'	2
'왼손 내려', '왼손 밑으로 내려', '왼 손 아래로 내려'	3
오른손 들어', ' 오른손 들어봐', ' 오른손 위로 들어'	4
오른손 내려', ' 오른손 밑으로 내려 ' , '오른손 내려봐'	5

[그림 2] 학습 데이터 집합



[그림 3] Simulation result on V-REP simulator

3. 결론

본 논문은 사용자의 음성을 자연어 처리 및 딥러닝을 기반으로 휴머노이드 로봇에게 명령을 전달할 수 있는 음성인식 인터페이스를 제안하였다. 목표한 데이터 집합에 대해서는 높은 인식률을 보여주었다 [그림 3]. 향후 연구로 로봇에게 필요한 명령어에 맞추어 학습 데이터를 늘리고, 복잡한 명령을 학습시켜 음성으로 구동할 수 있는 명령 범위를 넓히고자 한다. 더 나아가서 일상대화와 명령을 구분하는 후속 연구가 필요하다.

후기

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2015R1A2A1A10055798).

참고문헌

[1] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv: 1408.5882, 2014.

[2] 조휘열, et al. "컨볼루션 신경망 기반 대용량 텍스트 데이터 분류 기술." 한국정보과학회 학술발표논문집, pp. 792-794, 2015.