

형태소 분석기를 이용한 키워드 검색 기반 한국어 텍스트 명령 시스템

박대근¹, 이완복^{2*}

¹공주대학교 게임디자인학과 석사과정, ²공주대학교 게임디자인학과 교수

Keyword Retrieval-Based Korean Text Command System Using Morphological Analyzer

Dae-Geun Park¹, Wan-Bok Lee^{2*}

¹Student, Department of Game Design, Kongju National University

²Professor, Department of Game Design, Kongju National University

요 약 딥러닝을 기반으로 한 음성 인식 기술이 상용 제품에 적용되기 시작했지만, 음성 인식으로 분석된 텍스트를 효율적으로 처리할 방법이 없기 때문에 VR 콘텐츠에서 그 적용 예를 찾아 보기는 쉽지 않다. 본 논문은 문장의 형태소를 분석하는 형태소 분석기와 챗봇 개발에 주로 이용되는 검색 기반 모델(Retrieval-Based Model)을 활용하여 명령어를 효율적으로 인식하고 대응할 수 있는 한국어 텍스트 명령 시스템을 제안하는 것을 목적으로 한다. 실험 결과 제안한 시스템은 문자열 비교 방식과 같은 동작을 하기 위해 16%의 명령어만 필요했으며, Google Cloud Speech와 연동하였을 때 60.1%의 성공률을 보였다. 실험 결과를 통해 제안한 시스템이 문자열 비교 방식보다 효율적이라는 것을 알 수 있다.

주제어 : 형태소 분석기, 키워드 기반 모델, 한국어, 음성 인식, 명령

Abstract Based on deep learning technology, speech recognition method has began to be applied to commercial products, but it is still difficult to be used in the area of VR contents, since there is no easy and efficient way to process the recognized text after the speech recognition module. In this paper, we propose a Korean Language Command System, which can efficiently recognize and respond to Korean speech commands. The system consists of two components. One is a morphological analyzer to analyze sentence morphemes and the other is a retrieval based model which is usually used to develop a chatbot system. Experimental results shows that the proposed system requires only 16% commands to achieve the same level of performance when compared with the conventional string comparison method. Furthermore, when working with Google Cloud Speech module, it revealed 60.1% of success rate. Experimental results show that the proposed system is more efficient than the conventional string comparison method.

Key Words : Morphological Analyzer, Retrieval Based Model, Korean, Speech Recognition, Command

1. 서론

2015년, 시장 조사 업체 Digi-Capital은 [Fig 1]에서 보

이는 바와 같이 앞으로 VR시작이 폭발적 성장을 할 것이라 전망하였지만 2017년, 시장 규모 전망을 2015년 전망의 1/2 수준으로 축소하였다[1].

*This work was supported by the research grant of the Kongju National University in 2018.

*Corresponding Author : Wan-Bok Lee (wblee@kongju.ac.kr)

Received November 20, 2018

Revised January 4, 2019

Accepted February 20, 2019

Published February 28, 2019

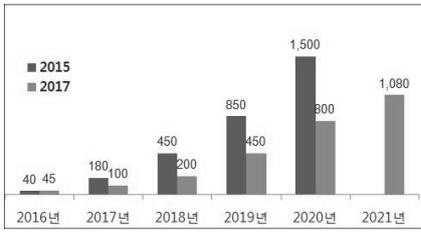


Fig. 1. Market Outlook

이유는 세 가지로 첫 번째, 일반 사용자들에게 보급화 되기엔 VR 장비들의 가격이 너무 비싸며 두 번째, VR 특유의 사이버 멀미와 두통이 사용자의 체함을 방해하고 있고 세 번째, 콘텐츠 개발이 어려워 소프트웨어의 출시가 더디다는 점이다[2,3].

첫 번째 이유와 두 번째 이유는 하드웨어의 한계로 나타나는 문제이지만 세 번째 문제가 나타나는 이유는 두 가지이다. 첫 번째, 기존 2D UX, UI를 3D로 개발할 때의 디자인적 어려움이 존재한다. 잘못된 UX, UI 디자인은 유저의 시야를 방해하고 사이버 멀미를 더욱 심하게 만들기 때문이다[4,5]. 두 번째, 컨트롤러에 존재하는 버튼 수의 한계 때문에 많은 기능을 넣기 힘들며, Fig 2에서 보이는 바와 같이 기능이 많아질수록 그것을 하기위해 거쳐야 하는 단계가 많아지기 때문에 조작 피로도는 올라가고 몰입감은 떨어진다[6,7].

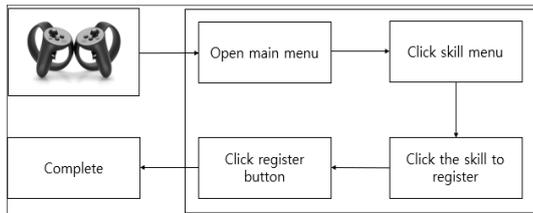


Fig. 2. The process of registering a skill

만일 음성을 통해 명령을 내릴 수 있다면 별도의 조작이 필요 없고 대화를 통해 무언가와 상호작용 한다는 느낌을 주기 때문에 조작 피로도는 내려가고 몰입감이 올라갈 것이다[8].

음성 인식은 과거부터 있어왔던 기술이지만 인식의 부정확성 때문에 상용 제품에 적용하기 힘들었다. 하지만 2017년, 4차 산업 혁명의 움직임이 본격적으로 시작되고 딥러닝을 기반으로 음성 인식 기술의 정확도가 95%가 넘어가며 본격적으로 상용 제품에 적용되기 시작하였다. 구글, 네이버 등을 비롯한 수 많은 대기업들이 저렴한

비용으로 혹은 경쟁력 강화를 위해 무료로 기술을 제공해주고 있어 기술의 진입 장벽도 상당히 낮은 상황이다 [9,10].

그럼에도 현재 국내외로 음성 인식 기술을 활용한 VR 콘텐츠를 찾아보기 힘든 것은 서버 비용, 이중 작업 문제 등 여러 가지 이유가 존재하지만 근본적으로 음성 인식으로 넘어온 텍스트를 효율적으로 처리할 방법이 없기 때문이다. 단순히 문자열 비교를 통한 행위와 명령의 1:1의 연결로는 다양한 형식의 명령에 대응하기 어렵고 수백, 수 천 가지의 명령어가 생기게 되어 유지 보수가 힘들어 질 것이다.

이에 본 논문은 문장의 형태소를 분석하는 형태소 분석기와 챗봇 개발에 이용되는 검색 기반 모델 (Retrieval-Base Model)을 이용하여 키워드 인식 (Keyword Recognition)을 통해 명령어를 효율적으로 인식하고 대응할 수 있는 한국어 텍스트 명령 시스템을 제안하는 것을 목적으로 한다.

연구는 알고리즘 개발과 실험 두 단계로 진행하며 실험은 문자열 비교 방식과 텍스트 명령 시스템을 사용하였을 때 특정한 행위들을 실행하기 위해 필요한 명령어의 개수와 명령에 대한 Action의 개수를 비교하고 구글 클라우드 스피치(Google Cloud Speech)와 텍스트 명령 시스템을 연동하였을 때 얼마나 명령이 잘 실행되는지 확인하기 위해 명령 성공 횟수를 구하였다.

2장에서 관련 연구에 대하여 서술하고 3장에서 시스템 구현에 대해 설명한 뒤 4장에서 실험 후 5장에서 결론 및 향후 과제에 대하여 이야기한다.

2. 관련 연구

2.1 구글 클라우드 스피치(Google Cloud Speech)

Google에서 개발한 Speech-to-Text 제품으로 신경망 모델이 적용되어있어 정확도가 매우 높으며 사용하기 쉬운 API 덕분에 누구나 쉽게 고성능의 Speech-to-Text 기능을 이용할 수 있다.글로벌 사용자층에 대응하여 120개 이상의 언어와 방언을 인식하며 음성 명령 및 제어 기능을 구현하고 콜센터의 오디오를 텍스트로 변환하는 등의 작업을 할 수 있고 Google의 머신러닝(Machine Learning) 기술을 사용하여 실시간 스트리밍 또는 사전 녹음 오디오를 처리할 수 있다.

2.2 형태소 분석기

입력된 문장의 품사를 분석해주는 도구로 문장 구조 (Pattern)를 토대로 미리 만들어둔 말뭉치에서 단어들을 검색하여 품사를 분석하고 필요에 따라 단어를 원형으로 되돌려준다. Twitter, KKMA, Mecab-ko 등 다양한 오픈 소스 한국어 형태소 분석기들이 존재하며, 본 연구에선 연구의 편의성을 위해 Twitter 형태소 분석기를 사용한다[11].

2.3 검색 기반 모델

검색 기반 모델(Retrieval-Base Model)은 미리 유저의 질문과 대답을 시스템에 입력해놓고 유저가 질문을 입력할 시 그 질문에 가장 가까운 대답을 내놓는 모델이다. 생성 기반 모델(Generative Model)에 비해 유연성은 떨어진다는 단점이 있지만 정확도가 높고 생성 기반 모델에 비해 개발이 쉽다는 장점이 있다. 대부분의 상용 챗봇들은 대화보다 기능 중심이기 때문에 생성 기반 모델보다 검색 기반 모델을 이용하여 개발된다. 대표적인 검색 기반 모델을 이용한 챗봇으로 IBM Watson이 존재한다[12,13].

본 연구에선 딥러닝을 이용한 문장 유사도 검색이 아닌 키워드 인식(Keyword Recognition)을 통한 키워드 검색 방식을 사용한다. 키워드 인식을 이용한 대표적인 챗봇으로는 Eliza[14]와 Alic[15]가 존재한다.

본 논문에서 제안한 한국어 텍스트 명령 시스템은 사전(Dictionary)에 Fig 3에서 보이는 바와 같이 등록된 명령을 사용자 입력, 형태소 분석, 키워드 추출, 명령 검색, 명령 실행 5단계를 걸쳐 실행한다.

3.1 명령어 데이터 설계

명령어는 Fig 4에서 보이는 바와 같이 실행자(Executor), 의도(Intent), 단어(Keyword), 행동(Action)으로 이루어져있다.

```
class CommandData:
    string executor:
    string intent:
    string keyword:
    Action actions:
```

Fig. 4. Flowchart of Korean text command system

실행자는 명령을 실행하는 주체를 나타낸다. 의도는 먹다, 자다 같은 동사와 발사, 공격과 같은 행위를 나타내는 명사를 나타낸다. 단어는 의도를 실행하는 주체나 의도를 적용해야할 대상을 나타낸다. 행위는 명령이 실행될 때 해야 할 행동을 나타낸다.

3.2 명령어 사전(Dictionary) 설계

3. 시스템 구현

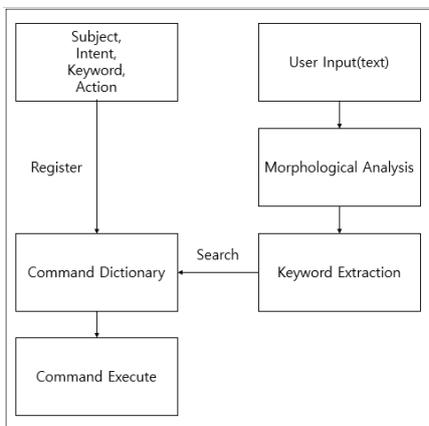


Fig. 3. Flowchart of Korean text command system

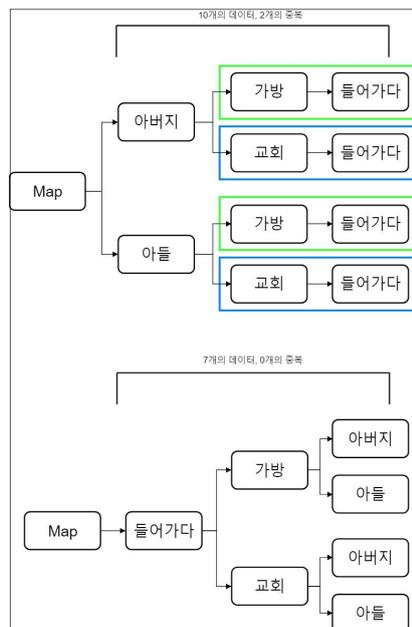


Fig. 5. Comparison of dictionary design methods

명령어 데이터를 Dictionary에 등록할 땐 Intent, Keyword, Executer순서로 Key값으로 사용하여 action을 등록한다. 이는 한국어 문장 구조의 역순으로, Fig 5에서 보이는 바와 같이 한국어 문장 순서 그대로 사전에 등록하게 되면 중복 데이터가 생겨날 수 있지만 역순으로 등록을 한다면 중복 데이터가 생겨나지 않는다.

3.3 키워드 추출

키워드 추출을 위해선 먼저 사용자에게 입력받은 명령을 형태소 분석기로 분석하고 분석된 단어 중 동사는 원형으로 되돌린다. 이를 통해 ‘열어줘’, ‘열어’, ‘열어줄래?’와 같은 다양한 종결 어미를 가진 동사들을 ‘열다’로 통일해 인식할 수 있게 된다.

분석된 단어들은 순서대로 Stack에 담는데, Fig 6에서 보이는 바와 같이 분석된 단어가 조사(PostPosition)일 경우 조사의 종류에 따라 다르게 처리를 한다.

수혜격 조사 / 목적격 조사	
을/를/에/에게/께/한테 조사 앞에 생물의 명칭이나 이름 또는 행위의 목적이 되는 존재(Whom)가 올 가능성이 높음	
관형격 조사	
의 조사 앞과 뒤의 단어가 연관성이 높음	

Fig. 6. Postpositions

관형격 조사가 나올 경우 Stack에서 단어 하나를 꺼내어 관형격 조사와 다음 단어를 합친 다음 Stack에 담는다. 이는 관형격 조사가 나올 경우 앞과 뒤의 단어가 매우 밀접한 관련이 있기 때문에 하나의 키워드로 보기 때문이다.

수혜격 조사와 목적격 조사가 나올 경우 Stack에서 단어 하나를 꺼내어 List에 담고 수혜격 조사는 버린다. 이는 수혜격 조사 혹은 목적격 조사 앞에 나오는 단어가 생물이나 사물의 명사 혹은 행위의 목적이 되는 대상일 확률이 높아 명령의 목표를 지칭하는 단어일 가능성이 높기 때문이다.

외의 조사들은 모두 버리며, 결과적으로 Fig 7에서 보이는 바와 같이 단어들의 모음인 Stack과 명령과 관련된 객체들의 명칭 모음인 Whom List 두 가지 결과물이 생긴다.

Input: 검사 스킴레톤을 공격해.

Whom	Stack
스킴레톤, NOUN	검사, NOUN 공격, NOUN 하다, VERB

Fig. 7. Keyword extraction result

3.2 명령 검색 및 실행

새로운 단어 모음 List를 만들어 Stack에서 단어들을 역순으로 옮긴다. 역순으로 옮기는 이유는 명령어 사전에 명령어 데이터를 등록할 때 한국어 문장의 역순으로 Key값을 만들었기 때문에 검색 또한 한국어 문장 구조의 역순으로 좀 더 빠른 검색을 하기 위해서이다.

그 후 단어 모음 List를 순회하며 명령어 사전에 단어가 Key값으로 존재하는지 확인한다. 단어가 사전에 Key값으로 존재한다면 해당 단어를 List에서 제거하고, 제거한 단어를 선행 Key값으로 하여 다음 Key값을 찾기 위해 다시 List를 순회한다. 이 과정을 명령어를 가져오는데 필요한 세 가지의 Key값(Intent, Keyword, Executer)을 찾을 때까지 반복하며 List를 다 순회하였음에도 Key값을 찾지 못했다면 시스템에 명령어 검색에 실패하였음을 알리고 모든 과정을 종료한다. 이 모든 과정을 순서대로 정리하면 Fig 8과 같다.

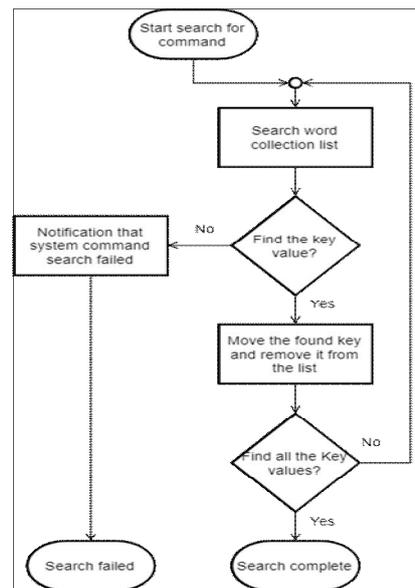


Fig. 8. Command search process flowchart

명령어 데이터의 검색을 완료했으면 단어 모음 List에 남아있는 단어들을 역순으로 하나의 문자열로 모두 합친다. 단어들을 합치는 이유는 합쳐진 단어에 액션에서 원하는 단어가 있는지 검색하기 위해서이며 단어 모음 List를 만들 때 단어들을 역순으로 넣었으므로 역순으로 합치지 않으면 ‘파이어 볼’과 같은 단어는 ‘볼파이어’로 합쳐질 것이기 때문에 단어 자체가 달라져버린다. 최종적으로 Fig 9에서 보이는 바와 같이 이름이 들어가 있는 List와 단어가 합쳐진 문자열, 명령어 세 가지의 결과물이 남게 된다.

```

Input : 검사 스킨레톤을 공격해
-----
RESULT
Whom :스켈레톤
Keyword : 하다
Command : Dictionary[검사][공격]
    
```

Fig. 9. Command Search Result

마지막으로 찾은 명령어 데이터에 존재하는 Action에 이름이 들어 있는 List와 단어가 합쳐진 문자열 넘겨주고 실행하는 것으로 모든 과정이 끝나게 되며, 넘겨받은 데이터는 Fig 10에서 보이는 바와 같이 Action 내부에서 검색에 활용된다.

```

def action(whoms, keyword):
    enemy = find_target(whoms[0])
    if enemy != null
        attack_to(enemy)
    
```

Fig. 10. Action pseudocode

4. 실험 및 결과

본 논문에서 제안하는 시스템의 효율성을 검증하기 위해 5:5 전투 시스템과 스킨, 가방, 지도, 환경옵션과 같은 기초적인 기능들을 가진 미니 게임을 만들어 두 가지의 실험을 진행하였다. 첫 번째는 “위로 이동”, “가방 열기”, “볼륨 10으로 변경”, “볼륨 20으로 변경”등 임의의 100가지의 행위를 정의하고 문자열 비교 방식과 텍스트 명령 시스템을 사용하였을 때 각각 100가지의 행위를 실행하기 위해 필요한 명령어의 개수와 명령어에 대한 Action의 개수를 비교하였으며, 두 번째는 텍스트 명령 시스템과 구글 클라우드 스피치(Google Cloud Speech)

를 연동하였을 때 얼마나 명령이 잘 실행되는지 확인하기 위해 하나의 행위당 3번씩 명령을 실행하여 명령 성공 횟수를 구하였다.

실험은 Fig 11에서 보이는 바와 같이 윈도우10 환경에서 Unity3D 2017.4.2f2버전에서 이루어졌으며 명령 시스템 구현에는 C# 언어를 사용하였다.

유니티상에선 트위터 형태소 분석기를 사용하기 어려워 유니티에서 Input Data를 파이썬 서버로 송신하고 파이썬 서버가 수신받은 Input Data를 형태소 분석하여 다시 유니티로 송신해주는 형태로 구현하였다.

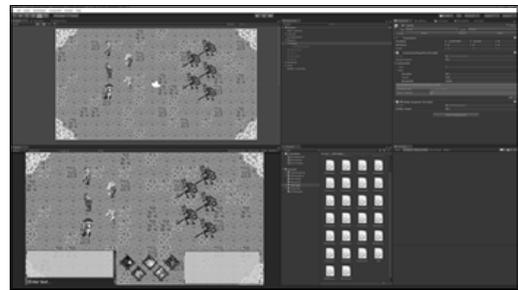


Fig. 11. Unity3D 2017.4.2.f2

첫 번째 실험 결과 Fig 11에서 보이는 바와 같이 문자열 비교 방식은 총 100개의 명령어가 필요했으며 텍스트 명령 시스템은 총 16개의 명령어만 필요했다.

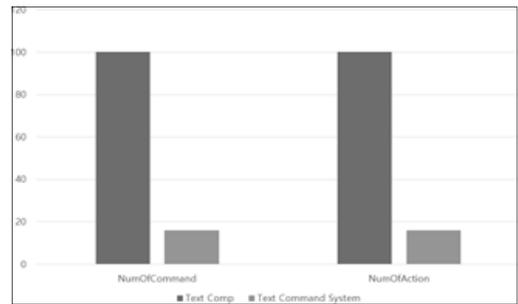


Fig. 12. Action pseudocode

문자열 비교 방식은 명령어와 Action이 1:1 대응되기 때문에 “음량 10으로 변경”, “음량 20으로 변경”과 같이 내부 수치만 달라지는 행위도 모두 각각의 명령어와 Action을 작성해야하여 행위와 명령, Action이 1:1:1이 되는 것을 볼 수 있다. 반면에 텍스트 명령 시스템은 ‘변경’과 ‘음량’이라는 키워드를 인식하고 10과 20이란 수치를 변수로 받을 수 있기 때문에 하나의 명령어와 Action으

로 처리할 수 있어 행위와 명령어, Action이 N:1이 되는 것을 알 수 있다. 때문에 텍스트 명령 시스템은 최소 문자열 비교 방식과 명령어의 개수가 같거나 최대 1개 명령어만 가지게된다. 또한 문자열 비교 방식은 한 글자라도 달라지면 인식할 수 없어 “사용해줘”, “사용해”, “사용해주세요” 같이 다양한 종결 어미에 대응하기 위해선 해당 종결 어미를 사용한 명령어를 추가해야하는데, 텍스트 명령 시스템은 핵심 키워드만 인식할 수 있으면 다양한 종결 어미의 대응은 물론, “가방 열어”, “열어 가방”과 같이 단어의 순서가 바뀌어도 정확히 인식한다. 결과적으로 텍스트 명령 시스템은 Fig 13에서 보이는 바와 같이 등록 명령어와 인식 명령어가 1:N이 된다.



Fig. 13. Recognized command and registered command

두 번째 실험 결과 총 300번의 음성 명령 중 182번의 명령이 성공하여 60.1%의 성공률을 보였다. 실험 전 70% 이상의 성공률을 보일거라 예상하였으나 그에 미치지 못하는 수치를 보였는데 이유는 2가지이다. 첫 번째, 구글 클라우드 스피치(Google Cloud Speech)를 통해 음성을 텍스트로 변환하는 과정에서 단어 보정 기능에 의해 의도한 단어와 다른 단어가 되어버리는 경우가 있다. 예를 들면, ‘태무진’이라고 말했을 때 단어가 ‘리무진’으로 보정 되어버린다. 두 번째, ‘ㅈ’와 ‘ㄱ’과 같이 비슷한 발음을 가진 모음의 구분이 어려워 ‘아래’를 의도하고 말했으나 ‘아래로’ 결과가 나오는 경우가 많았다. 그 외에 키보드로 타이핑(Typing)시에는 숫자를 아라비아 숫자 기호로 명확히 입력할 수 있었지만 구글 클라우드 스피치(Google Cloud Speech)를 사용할 경우 숫자들도 한글로 바뀌기 때문에 한글로 된 숫자가 있는지 검색하여 아라비아 숫자 기호로 변환하는 작업이 필요했다.

5. 결론 및 향후 연구 과제

본 논문은 최소한의 명령어 등록을 통해 최대한의 명령어를 인식하고자 형태소 분석기와 키워드 인식 기술을

활용하여 명령어를 효율적으로 인식하고 대응할 수 있는 한국어 텍스트 명령 시스템을 제안하였다.

제안한 시스템은 명령어를 Intend, Keyword, Excutor, Action으로 나눠 등록하고 사용자가 입력한 문장을 형태소 분석 후, 분석된 단어들을 기반으로 등록된 명령어가 있는지 검색하여 실행한다.

제안하는 시스템의 효율성 검증을 위해 100가지 행위를 정의하고 그 행위를 실행하기 위해 필요한 명령어와 Action의 개수를 문자열 비교 방식을 비교 분석하였으며, 구글 클라우드 스피치(Google Cloud Speech)와 텍스트 명령 시스템을 연동하여 명령 성공 횟수를 구하였다.

실험 결과 100가지 행위를 하기 위해 필요한 명령어가 문자열 비교 방식에 비해 상당히 줄어들어 텍스트 명령 시스템이 문자열 비교 방식보다 효율적이라는 것을 알 수 있었으며, 키워드 인식을 통해 어순이 뒤바뀌거나 종결 어미가 다른 다양한 형태의 문장도 인식할 수 있음을 알 수 있었다.

구글 클라우드 스피치(Google Cloud Speech)와 텍스트 명령 시스템의 연동에선 단어의 보정 및 비슷한 발음의 모음에 의해 명령 성공률이 다소 낮아 이에 대한 보완이 필요함을 알 수 있었으며 구글 클라우드 스피치(Google Cloud Speech)가 아라비아 숫자도 한글로 변환 시키기 때문에 변환된 한글 숫자를 다시 아라비아 숫자로 변환시키는 작업이 필요함을 알 수 있었다.

현재는 한번에 한명의 Excutor에게 한 가지의 명령밖에 내릴 수 없지만 향후 연구를 통해 한번에 여러 Excutor에게 여러 명령을 내릴 수 있도록 시스템을 개선할 필요가 있다. 또한 현재 텍스트 명령 시스템의 성능과 인식도가 형태소 분석기에 좌우되기 때문에 텍스트 명령 시스템에 최적화된 형태소 분석기의 개발이 필요하다.

REFERENCES

- [1] Hebronstar. (2018). *VR/AR Trends*. Hebronstar [Online], <http://hebronstar.com/?p=6272>.
- [2] K. U. Han & H. T. Kim. (2011). *The Cause and Solution of Cybersickness in 3D Virtual Environments*. Korean Journal of Cognitive and Biological Psychology. 23(2). 287-299.
- [3] J. Y. Jung, K. S. Cho, J. H. Choi & J. H. Choi. (2017). *Causes of Cyber Sickness of VR Contents - An Experimental Study on the Viewpoint and Movement*.

- The Korea Contents Society. 17(4). 200-208.
- [4] M. Y. Choi & S. W. Kim. (2016). *EA Study on the First Use Experience for Rapid Adaptation to HMD VR Contents -Focused on Samsung Gear VR Game Application-*. Korean Society of Basic Design & Art. 17(6). 605-616.
- [5] J. W. Park & S. K. No. (2018). *EA Study on the Structural Features of VR HMD Interface Design - Focused Oculus Home -*. The Korean Society Of Design Culture. 37. 78-87.
- [6] J. Y. Han. (2016). *Study on the Feature of Mobile HMD-Based VR Experience Contents Design*. Korean Institute of Spatial Design. 37(0). 78-87.
- [7] O. T. Kim. (2010). *The Impact of Video Game's Controller Realism on Natural Mapping, Spatial Presence, Arousal and Emotions: Using First-Person Shooting Video Games*. korean society for journalism and communication studies. 54(5). 227-253.
- [8] E. J. Hong, K. S. Cho & J. H. Choi. (2017). *Effects of Anthropomorphic Conversational Interface for Smart Home:An Experimental Study on the Voice and Chatting Interactions*. HCI Society of Korea. 12(1). 15-23.
- [9] J. H. Choi & S. H. Lee. (2017). *음Current status and implications of voice recognition AI secretary market*. Korea Association for Telecommunications Polices. 29(9). 1-37.
- [10] S. R. Jung. (2018). *Analysis of promising technologies based on artificial intelligence through patent search*. Master dissertation. Korea University. Seoul.
- [11] E. J. Park & S. Z. Cho. (2014). *KoNLPy: Korean natural language processing in Python*. HCLT. 26. 133-136.
- [12] Jack Cahn. (2017). *CHATBOT: Architecture, Design, & Development*. Senior Thesis. University of Pennsylvania. Pennsylvania.
- [13] Rob High. (2012). *The Era of Cognitive Systems: An Inside Look at IBM Watson and How it Works*. Armonk : IBM RedBooks.
- [14] J. Weizenbaum. (1996). *ELIZA-A computer program for he study of natural language communication be-tween man can machine*. Communications of the Association for Computing Machinery. 9(1). 36-46.
- [15] B. P. Schumaker, Ying Liu, Mark Ginsburg & Hsinchun Chen. (2006). *Evaluating mass knowledge ac-quisition using the ALICE chatterbot: the AZ-ALICE dialog system*. International Journal of Human-Computer Studies. 64(11). 1132-1140.

박 대 근(Park, Dae Geun)

[학생회원]



- 2018년 2월 : 호서대학교 게임학과 (공학사)
- 2018년 3월 ~ 현재 : 공주대학교 게임디자인학과 석사과정
- 관심분야 : VR, AR
- E-Mail : uemonwe@gmail.com

이 완 복(Lee, Wan Bok)

[정회원]



- 2004년 2월 : KAIST 전자전산학과 (전기 및 전자공학 전공 공학박사)
- 현재 : 공주대 게임디자인학과 교수
- 관심분야 : 게임엔진, 시뮬레이션, 이산사건시스템
- E-Mail : wblee@kongju.ac.kr