Natural Language Processing (UML602) Project Report

(B.E 3rd Year May 2018)

Virtual Assistant

Submitted by:

Abhishek Sharma – 101503008

Group: COE-1

Submitted to:

Ms. Prabhleen Juneja

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Table of Contents:

Introduction:

Today, we can ask virtual assistants like Amazon Alexa, Apple's Siri, Google Now to perform simple tasks like, "What's the weather", "Remind me to take pills in the morning", etc. in natural language. The next evolution of natural language interaction with virtual assistants is in the form of task automation such as "turn on the air conditioner whenever the temperature rises above 30 degrees Celsius", or "if there is motion on the security camera after 10pm, call Bob".

Almond is an open, crowdsourced and programmable virtual assistant that was built as part of the Open Mobile Platform project at Stanford. Central to Almond is Thingpedia, which is an open repository of different services, including Internet of Things (IoT) devices, open Web APIs and Social networks along their natural language interfaces. Thingpedia, which is an encyclopaedia for the IoT, contains information about each device along with a set of functions that correspond to each device API. Each Thingpedia entry for a function also contains a natural language annotation that captures how humans refer and interact with the device. Through the efforts of crowdsourcing, Thingpedia contains a set of 50 devices and 187 functions. The 50 devices span a variety of domains from media (newspapers, web comics), social networks (twitter, Facebook), home automation (light bulb, thermostat), communication (email, calendar), etc.

Hence, the goal of this project is to learn about NLP applications and try to about the concept behind them. This virtual assistant is not very vast as compared to the other virtual assistants made Google, Facebook, Amazon etc. But it will give brief knowledge about the NLP application used by this project. It will basically listens to the person who is operating it and do the task according to the query entered by the user.

Technologies Used and Working:

1. Speech Recognition – Google

```
def speak(audio_string):
    global i
    i = i + 1
    print(audio_string)
    tts = gTTS(text=audio_string, lang='en', slow=False)
    tts.save("audio" + str(i) + ".mp3")
    mixer.init()
    mixer.music.load("audio" + str(i) + ".mp3")
    mixer.music.play()
```

2. Text to Speech – Google

```
def my_command():
    """listens for commands"""

    r = sr.Recognizer()

    with sr.Microphone(device_index=0, chunk_size=2048, sample_rate=48000) as source:
        print('Listening...')
        r.pause_threshold = 1
        r.adjust_for_ambient_noise(source, duration=1)
        audio = r.listen(source)
```

3. Web Scrapping – Beautiful Soup

Beautiful soup is used to scarp the text from the web, like in this project it is used to scrap the text from the Wikipedia to retrieve the information about the query.

```
elif 'read' in command:
    reg_ex = re.search('read (.+)', command)
    if reg_ex:
        domain = reg_ex.group(1)
        url = 'https://en.wikipedia.org/wiki/' + domain
        source = requests.get(url).text
        soup = BeautifulSoup(source, 'lxml')
        summary = soup.find('div', class_ = 'mw-body').p.text
        webbrowser.open(url)
        speak(summary)
```

4. Json Parsing

Json is used for the parsing in the strings, like this project it is used to extract the joke from the internet when user queried for the joke.

```
elif 'joke' in command:
    res = requests.get(
        'https://icanhazdadjoke.com/',
        headers={"Accept": "application/json"}
    )
    if res.status_code == requests.codes.ok:
        speak(str(res.json()['joke']))
    else:
        speak('oops!I ran out of jokes')
```

5. Regex

Regex is used to search for a particular string in the command given by the user.

```
elif 'find' in command:
    reg_ex = re.search('find (.+)', command)
    if reg_ex:
        domain = reg_ex.group(1)
        url = 'https://www.google.co.in/search?q=' + domain
        webbrowser.open(url)
        print('Done!')
    else:
        pass

elif 'tell weather at' in command:
    reg_ex = re.search('tell weather at(.+)', command)
    if reg_ex:
        domain = reg_ex.group(1)
        url = 'http://www.intellicast.com/Local/Default.aspx?query=' + domain
        webbrowser.open(url)
        print('Done!')
    else:
        pass
```

6. OS

OS is used to interact with the personal computer via python scripts. You may shutdown, reboot, sleep, hibernate the personal computer on the basis of the action you want to perform.

```
elif 'reboot system' in command:
    os.system('reboot')
```

7. Time

Python contains the library time with from time import ctime which will tell the time on the operating system you are using.

```
elif "time" in command:
    speak(ctime())
```

References:

[1] Giovanni Campagna, Rakesh Ramesh, Silei Xu, Michael Fischer, and Monica S. Lam. Almond: The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant. In Proceedings of the 26th International World Wide Web Conference (WWW-2017), 2017.

[2] Monica S. Lam, Giovanni Campagna, Jiwon Seo, and Michael Fischer. A distributed open social platform for mobile devices. In Proceedings of the International Conference on Mobile Software Engineering and Systems, MOBILESoft '16, pages 173–174, New York, NY, USA, 2016. ACM.

[3] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In Proceedings of the 53nd Annual Meeting of the Association for Computational Linguistics (ACL-15), pages 1470–1480, 2015.

[4] Yushi Wang, Jonathan Berant, and Percy Liang. Building a semantic parser overnight. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-15), pages 1332–1342, 2015.

[5] Khai N Truong, Elaine M Huang, and Gregory D Abowd. Camp: A magnetic poetry interface for end-user programming of capture applications for the home. In International Conference on Ubiquitous Computing, pages 143–160. Springer, 2004.

[6] Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L Littman. Practical triggeraction programming in the smart home. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 803–812. ACM, 2014.

[7] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-14), pages 1415–1425, 2014.