Smart Assistant for Public Parks using IBM Watson

Introduction:

Our names are Udamya, Sarvesh and Abhishek. We know each other from the IOT internship program at 'Smart Bridge' . We all think AI as a field is very interesting and are looking forward to having a lot of professional discussions about the topic through our project work : Smart Park-Corbett National Park.

1.1 Overview

Chatbots are software applications that are built to automatically engage with messages they receive. They are designed to mimic written or spoken human speech for the purposes of simulating a conversation or interaction with a real person. Chatbots can be programmed to respond the same way each time, to respond differently to messages containing certain keywords and even to use machine learning to adapt their responses to fit the situation.

Chatbots can use lots of different mediums to respond to a message – integrating with SMS text messaging, launch a website chat window or use social messaging services across platforms like Facebook and Twitter.

There are two primary ways chatbots are offered to visitors: via web-based applications or standalone apps. Today, chatbots are used most commonly in the customer service space, assuming roles traditionally performed by living, breathing human beings such as customer support staff.

1.2 Purpose

Our Smart Park assistant can be used for Park and Tourism Management. **We have designed** a **chatbot** to automatically begin a conversation with people who interact with a certain post on our platform thus information passing with ease and convenience;

Literature Survey:

Travel chatbots offer a cleaner alternative. Online chatbots for websites and messaging apps have one thing in common — they don't create bloatware on the device. Instead, they use existing platforms like browsers or messenger apps that almost everyone has on their phone. As such, they make it possible for travelers to receive notifications, updates and promotional offers,

without downloading separate apps for each of the services they'll be using during their trip.

2.1 Existing Problem

The pandemic has changed the whole perception about technology and the value of data in planning, managing and strategizing recovery of the economy. Data has become a 'non-negotiable' thing in the current context and it is quite important to have right data of every activity .

The same goes with providing right information on national parks; a traveler will have many queries before being there and proving right up to date information is a real challenge.

2.2 Proposed Solution

The chat bot will improve the information gathering experience of travelers by providing them with more relevant suggestion while enhancing the business metrics and saving time.

Theoretical Analysis:

When planning a trip, around [84% of travelers use online searches](#) and to find the right information a travelers visit [38 sites on average](#), and [for 62% of travelers, it is hard to find the right deal.](#) At the same time, [Humley's survey said 87% of travelers](#)want to interact with a travel chatbot to find the best accommodation while saving time for the indecisive search. Moreover, [79% of them expect a travel chatbot to perform as an online travel concierge](#).

While designing our chat bot related to Corbett National Park , we have taken above datas into consideration and made a tourist friendly online information Centre.
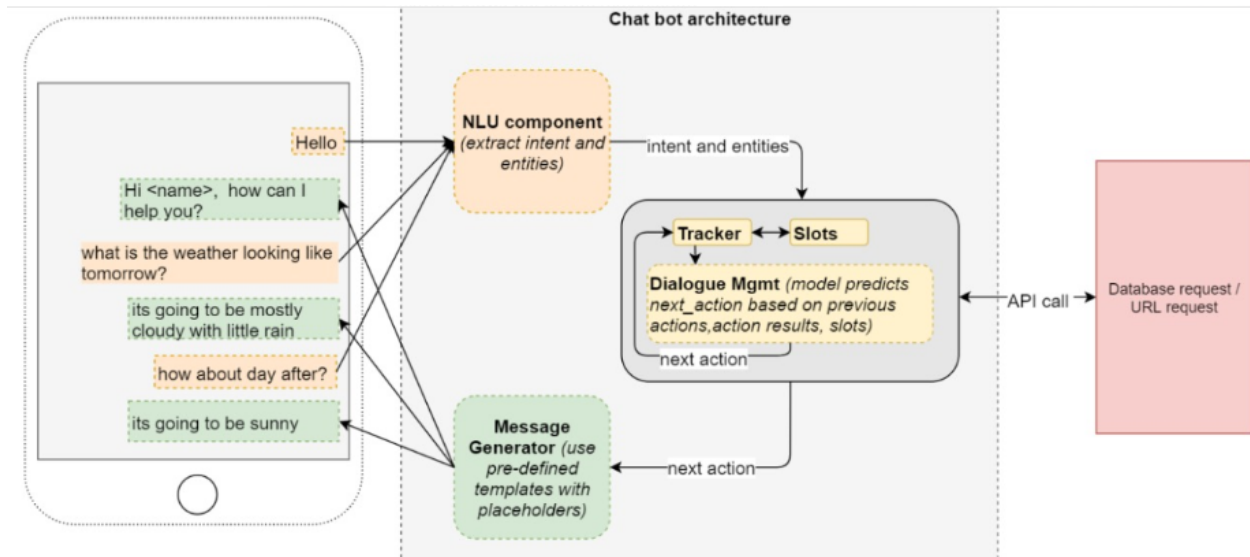
3.1 Block Diagram

Chat bot architecture

Hello

Hi <name>, how can I help you?

what is the weather looking like tomorrow?

its going to be mostly cloudy with little rain

how about day after?

its going to be sunny

NLU component *(extract intent and entities)*

intent and entities

Tracker ↔ Slots

Dialogue Mgmt *(model predicts next_action based on previous actions, action results, slots)*

next action

API call → Database request / URL request

Message Generator *(use pre-defined templates with placeholders)*

next action

Image Courtesy: Towards Data Science

3.2 Software Designing

To create our web application we have used the IBM cloud service for text to speech, speech to text and node-red services. The text to speech service is used as output, the speech to text service is used as an input. Node-red is used for creating the user interface and receiving data. The output and input services are coded using python.

Experimental Investigations

For a chatbot to be especially useful to a certain domain some criteria have to be met.

When designing an intelligent system that provides decision support one must consider the human as something outside the system, but also as an integrated system component that in the end, will ultimately determine the success or the failure of the system itself.

We made a chatbot that we used as a prototype to investigate the research questions. The chatbot was originally made for simple words. But we wanted to further use this in our project. During the design process we improved and tested the prototype. We tried to make it as helpful as we could manage within the time frames of the project by iterating multiple times.

The first participant enjoyed talking to the bot, but stressed the fact that you had to "talk like "a dummy" for it to understand what you were asking. The participant pointed out that this really would have come in handy if he didn't always know how to ask - especially if he was in a hurry. He pointed out that the prototype needs to get more features like tell the dates when park is open, or current weather ,etc

We listed a set of questions and tasks, which we asked the participants to answer and preform. We also included a few control questions to investigate the participants experience with the

chatbot and to find out if they had any suggestions for further improvement. The evaluation ended with a short talk about the experience, where we were open for any kind of feedback the evaluators could provide.

Due to time and capacity during this project we decided on including five participants acting as evaluators. The number of participants is also chosen on the basis that five participants can contribute to finding 80% of the usability flaws The evaluation was formed as a formative usability test where the goal is to look at metrics that are more qualitative than quantitative.

Flowchart
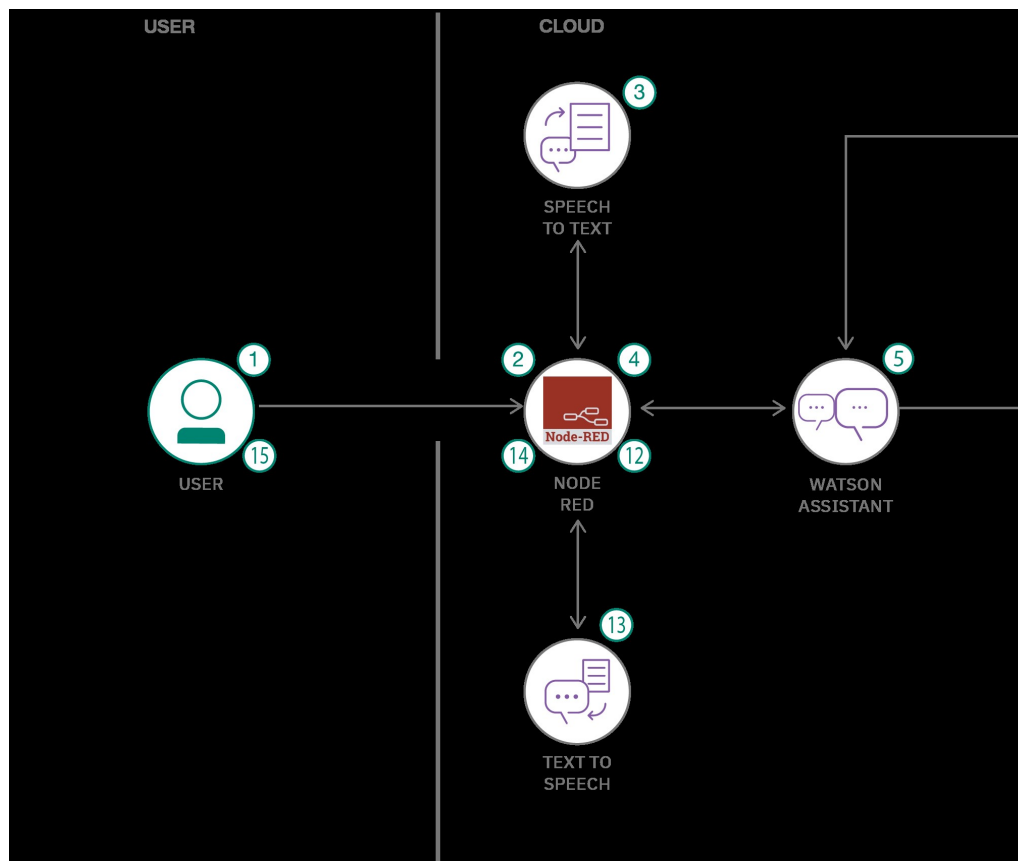
Result

The Smart Park Chatbot model demonstrates how a high degree of connectivity and exchange of information can benefit all - from nature conservationists, tourists, businesses and communities, through to park authorities and emergency services. A Smart Park could open up opportunities for new types of visitor attractions, which protect rather than spoil some of the world's most treasured landscapes.

Advantages and Disadvantages

The people that visit national park can benefit from the latest innovations in modern technology. It's about new ways of delivering conservation as well as visitor convenience and experience enhancement by having most updated information on the finger tips.

Learning and Updating – AI-based chatbots are capable of learning from interactions and updating themselves on their own. This is a big benefit when it comes to investing time in educating the executives about the same. Due to machine learning and algorithms capable of updating themselves, the need for same is eliminated while using a Chatbot.

One of the greatest **disadvantages** of chatbots is that they have been designed to handle first-level questions only. They may not be able to solve complex queries. We need to train them to converse with the questions in the right way. We also need to structure and optimize our knowledge base in a **bot**-friendly way. ZERO DECISION MAKING and POOR MEMORY are also a big issue with chatbox.

Applications

Consider applying the chatbot to the travel business, it will bring the following benefits:
- Increase engagement. As we have said, travelers are looking through various travel websites to find the best trip. At the same time, if your online travel chatbot can provide users with relevant offers.
- Reduce workload and operation costs. When a traveler is interested in a particular information, tour or hotel, one is more likely to ask a question to receive more information. FAQ chatbot can provide travelers with answers to basic questions, thus reducing both the workload on employees and customer support costs up to 30%.
- Increase sales. Travelers want to book a hotel, rent a car or pay for their ticket as quickly as possible and find long forms an unpleasant and irritating experience. Thanks to the integrated payment gateway, a Messenger travel chatbot could seamlessly get the customers through the sales funnel and close a deal with fewer interactions.
- Built community. Apart from business logic, like built-in payment and booking, a chatbot could be a handy tool for building a community by sharing their travel experience and inspiration.

- Attract new customers. Customer acquisition cost is an essential metric, Proactive travel chatbots decrease CAC and attract new customers most effectively by starting a conversation. Besides,the chatbot can send relevant information based on keywords used by the travelers

Conclusion

It's the interconnectivity of the data. The critical success factors include the degree of interconnectivity between the different devices and systems, and the interoperability of the data sources and the analytics and how they're presented to the end user. Beneficiaries could include park users, businesses, municipalities and the management authorities while also providing "improved air quality and community life, better education and generally connecting people with nature.

Future scope

Our National Parks solutions can be used in future for Wildlife Protection, Park Management and Tourist information, engagement and Safety.

1. Many travelers are going to another country searching for an authentic experience. They want to eat, entertain, and live as the locals do. The chat bot can be developed further provide travelers with local recommendations on restaurants with local cuisine, festivals, and other activities, a local insider chatbot will be the best choice. In this way, the travelers will receive relevant information without spending hours searching for the most recommended places on social networks.
2. The conservation of Park can be managed with the chatbot by recording arrival/departure of visitors; thus can prevent over-tourism.
3. Since most National Parks have no basic 3G or 4G internet coverage, we can start by covering the conservation area with a state-of-the-art communication network. This network will serve to collect data from the sensors we distribute across the park. The sensors will securely track wildlife and gate sensors protect the area. The data will be processed and presented in an easy-to-use web application. The application will provide real-time information about the location of wildlife and assimilates many other critical

elements of information, acquired from the sensors in the park. The application will provide real-time information about the location of the wildlife and other sensor information. A connected system of information would also help park employees perform many of their daily tasks more efficiently. Travelers can share data and log, for example, animal sightings or discoveries of invasive species, park rangers would be able to monitor animals and make sure vulnerable species are protected.

This will play a very important role in modern wildlife conservation.

Biblography

1. The Oxford University Online Library : www.ox.ac.uk
2. www.stanford.edu
3. Lazar, J., Feng, J. H., & Hochheiser, H. (2017). Research methods in human-computer interaction. Morgan Kaufmann.
4. Winograd, T. (1991). Thinking machines: Can there be? Are we (Vol. 200). University of California Press, Berkeley. (p.204-210)
5. Interaction design foundation . User centered design. https://www.interaction-design.org/literature/topics/user-centered-design
6. Brandtzaeg, P. B., & Folstad, A. (2017). Why people use chatbots. In I. Kompatsiaris, J. Cave, A. Satsiou, G. Carle, A. Passani, E. Kontopoulos, S. Diplaris, & D. McMillan (Eds.), Internet Science: 4th International Conference, INSCI 2017 (pp. 377-392).

Appendix

In the making of the chatbot, we thought about how the chatbot could be useful and easy to interact with. The chatbot we ended up making was a chatbot that new users could use to get simple information such as where you can get food, where you can find the room you are looking for and how to reach the park. To make the interaction more enjoyable we tried to make the conversation playful. To make the chatbot easier to use we included a lot of trigger words so that you didn't have to know the specific words to trigger the right answers. We also included a message that said "I'm sorry I'm not that smart yet, try google" with a link to google, for whenever the chatbot could not answer. While we built the chatbot we also tested it a lot, to make sure that it gave the answers it was supposed to do.

Final Code:

```python
import json
from ibm_watson import AssistantV2,SpeechToTextV1,TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from os.path import join, dirname
from playsound import playsound

authenticator =
IAMAuthenticator('pYqN3C1pQM3PdLSWPsAzfOuaLh_OBfPJLack9Mnm8_
YF')
assistant = AssistantV2(
    version='2020-04-01',
    authenticator = authenticator
    )
assistant.set_service_url('https://api.eu-gb.assistant.watson.cloud.ibm.co
m/instances/4f1e8f08-3416-4773-a53e-c4208ae518ef')

authenticator =
IAMAuthenticator('3q46ytQO4ftl2pFmunP1VoNWyK-EyAjGjOhlgmkt-GOv')
speech_to_text = SpeechToTextV1(
  authenticator=authenticator)
speech_to_text.set_service_url('https://api.eu-gb.speech-to-text.watson.clo
ud.ibm.com/instances/fe05bc36-660d-4330-bc0e-bc6c9c7a53a2')

authenticator =
IAMAuthenticator('wsDPlrLmNuqTX-zHJbxeUshfblaBOEB9zOGCzh4KX9Mu
')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
    )
```

```python
text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.clo
ud.ibm.com/instances/6265cfe4-5ac9-4783-8da6-84ca4ea07091')

def stt(file):
    with open(join(dirname(__file__), './.', file),
            'rb') as audio_file:
        a = speech_to_text.recognize(
            audio=audio_file,
            content_type='audio/mp3',
            ).get_result()
    b=a['results'][0]['alternatives'][0]['transcript']
    return b
def chatbot(text):
    response = assistant.create_session(
        assistant_id='0b8b59bb-23b1-485e-9da0-f6cccbe5e053',
        ).get_result()
    a=response['session_id']
    response1 = assistant.message(
        assistant_id='0b8b59bb-23b1-485e-9da0-f6cccbe5e053',
        session_id=a,
        input={
        'message_type': 'text',
        'text': text
    }).get_result()
    return response1['output']['generic'][0]['text']

def tts(a):
    with open('output.mp3', 'wb') as audio_file:
        audio_file.write(
            text_to_speech.synthesize(
                a,
                voice='en-US_AllisonV3Voice',
```

```
            accept='audio/mp3'
        ).get_result().content)
    #output audio file
    playsound("output.mp3")

#give input file name here
file="sampleinput.mp3"
t=stt(file)
a=chatbot(t)
tts(a)
```

Node - red Flow: