Jukebot

A Personalizeable Chatbot API for Radio Stations

Risako Owan, Esha Singh, Malik Khadar, Brett Schumacher,Ruyuan Wan, Rijul Mahajan, Maria Gini

computer science & engineering
UNIVERSITY OF MINNESOTA

UNIVERSITY OF MINNESOTA
Driven to Discover™

Introduction

What is Jukebot?

Jukebot is a chatbot API capable of answering simple questions and getting feedback from users for music stations. It was created for the 2020 PILOT Innovation Challenge, hosted by the National Association of Broadcasters (NAB). The prompt was to:

"Build an AI character that can have conversations with individual viewers, listeners or consumers. Broadcasters should be able to define and train the character's personality."

The supported domains (conversational topics) in Jukebot include Greeting, Playlist, Feedback, Help, Promotion, Weather, and Unsupported. These domains were chosen based on our conversations with KS95 (Hubbard Broadcasting), a local radio station that NAB had connected us with, and what information they provided on their website.
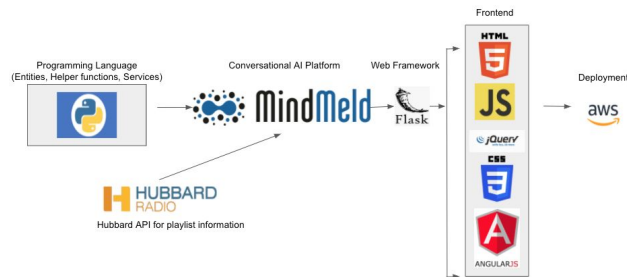
The Promotion domain is for queries about local sales and promotions, the Weather domain responds to weather queries on a city-level, and the Unsupported domain responds to queries Jukebot cannot answer, such as "What movies are in theatre right now?" or "When will Taylor Swift's next album come out?".

Seven Highlights of Jukebot

- User Friendly
- Training Data Efficient - doesn't require large datasets
- Compact Storage
- Low Latency - calculates responses with no delay
- Flexible - consists of plug and play modules
- Reliable - no unexpected outputs or grammar mistakes.
- Robust - should work for all pointers listed above.

Implementation

Technologies Leveraged

Programming Language (Entities, Helper functions, Services)

Conversational AI Platform — MindMeld

Web Framework — Flask

Frontend — HTML5, JS, jQuery, CSS3, ANGULARJS

Deployment — aws

HUBBARD RADIO

Hubbard API for playlist information

Jukebot is a Python API that utilizes publicly available Hubbard API for radio playlist information and Mindmeld, an open-source conversational agent platform, for understanding user queries and deciding how to respond. We use a list of sample queries for the training data in identifying user intent, as required by Mindmeld. In order to demo our API, we also created a Flask webpage and deployed it to AWS.
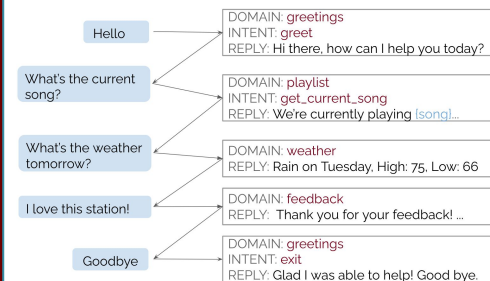
We chose to use Mindmeld in this project because it is open source, it offers advanced tools and flexibility to train and analyze custom language understanding models, and unlike cloud-based natural language processing services, it does not require training data to be uploaded to a shared cloud infrastructure.

How to Update Jukebot's Personality?

Jukebot's personality can be updated by changing the training data. Broadcasters can create example responses for some personality using their method of choice and convert/export them using a provided function and list of instructions. They may also modify the list of supported domains and intents with adequate engineering support.

Example

Conversation Example

Hello →
DOMAIN: greetings
INTENT: greet
REPLY: Hi there, how can I help you today?

What's the current song? →
DOMAIN: playlist
INTENT: get_current_song
REPLY: We're currently playing [song]...

What's the weather tomorrow? →
DOMAIN: weather
REPLY: Rain on Tuesday. High: 75, Low: 66

I love this station! →
DOMAIN: feedback
REPLY: Thank you for your feedback! ...

Goodbye →
DOMAIN: greetings
INTENT: exit
REPLY: Glad I was able to help! Good bye.

This is an example of what a conversation might look like with the default neutral personality. The left column shows what a typical user might say and the right column shows some examples of how Jukebot might respond. The domain refers to the conversation topic and the intent refers to a specific purpose. Some domains may not have intents if there is only one default intent. The weather domain returns the weather forecast for Prospect Park, Minneapolis, where the Hubbard Broadcasting headquarters are located, unless specified otherwise.

Conclusion

Ending the Project

We deployed the API in AWS and handed our work to NAB via GitLab. Additionally, we presented our work at the 2020 NAB Show virtually in October. Please feel free to contact us for the source code.