User guide about the music chatbot

Chatbot subject

This chatbot is about music. You can ask him information about an artist, music and even ask him to recommend songs for you based on your library.

You can also look for the n best trending song.

This chatbot is based on the LastFM API. LastFM is a music website that tracks the listening of his users across multiple platform (Spotify, Deezer, Tidal, …) that allow him to build a strong and reliable dataset.

The bot is deployed on Heroku so you can try it yourself on this Facebook page:
https://www.facebook.com/Tryingchatbothere-106592898050484/

How to use the chatbot?

The chatbot is divided in 2 parts. The Facebook messenger platform and the recommender system. The Facebook messenger is made in TypeScript which is a typed version of JavaScript and the recommender system is made in python, so we had to make an external API to link both.

So, if you want to use the chatbot at home you will need to follow those steps:

- Get a Facebook Messenger app setup on the Facebook developers' platform, to get a page token.
- Get a LastFM API key.
- Create a .env file following the .env.example file
- You can now run the chatbot with the command "npm run start" that will run the app and the python api
- Finally register the webhook to your app with the verify token you choose at the creation of your .env file.

Your app is now up and running!

How the recommender system has been made?

Since I couldn't find any good and useable dataset online, I created a small dataset (~1000 songs) by requesting the top songs of LastFM api. With those songs I look for the tags for each one and build the data I needed. You can find the detail inside scrapper.ipynb

On the other hand, on the chatbot I made a system that allow a user to add songs on a library based on his id. Once I find the song he wanted to add, I look for the song tags and save it to a file.

Now I have all the data I need to make my recommender system; it is a content-based algorithm that will look for the tags of each songs and recommend the one that have the highest cosine score. You can find the details about recommendation inside recommender.ipynb

I then put the content of recommender.ipynb inside app.py that is a flask API that can be requested by the chatbot.

All the command you can try

Hello:

- *User*: 'Hi!'

- *Bot*: 'Hey 👋'

Bye:

- *User*: 'Bye'

- *Bot*: 'See you later! 🖐

Top songs: <n(def:n=3)> best songs from <artist_name>

- *User*: 'What are the 5 best songs from the beatles ?'

- *Bot*: 'Here are the top 5 songs from the beatles

    1 - Come Together

    2 - Let It Be

    3 - Yesterday

    4 - Help!

    5 - Here Comes the Sun'

Song search: <n(def:n=3)> songs that contain <song_name>

- *User*: 'Search 5 songs that contain help'

- *Bot*: 'Here are the 5 songs found:

    1. Help! - The Beatles

    2. Help I'm Alive - Metric

    3. Helplessness Blues - Fleet Foxes

    4. Can't Help Falling in Love - Elvis Presley

    5. Help Me Lose My Mind (feat. London Grammar) - Disclosure'

Artist search: <n(def:n=3)> artists that contain <artist_name>

- *User*: 'Search 5 artists that contain beatles'
- *Bot*: 'Here are the 5 artists found:

1. The Beatles
2. wu tang vs the beatles
3. #1 Beatles Now
4. The Beatles Tribute Band
5. The Beatles Recovered Band'

Adding a recommendation: add <song_name> - <artist_name> - <score [0-10]>

- *User*: 'add Take on Me - a ha - 10'
- *Bot*: 'Successfully added to your library 👍'

Recommendation:

- *User*: 'recommand'
- *Bot*: 'I can recommand you:

1 - Billie Jean by Michael Jackson with a confidence of 88.13052984784633%

2 - Dancing Queen by ABBA with a confidence of 74.9109503706694%

3 - Beat It by Michael Jackson with a confidence of 70.50442387827707%'

Demonstration

We've made a demonstration video that you can check out at this link : https://youtu.be/PCf6l4FWhtE