

SongBot – Ask me anything about any song.

IU NLP 2020 Course Project

Abstract

SongBot is an attempt to create a chatbot that lets user explore songs and/or their related information by querying in an everyday conversational language. Users write to the bot with a query related to songs and the bot tries to break down that query and map the pieces to certain intents and entities. With those intents and entities in hand, the bot will call the corresponding action handler if any and retrieve required pieces of information with which to respond.

1 Credits

This document serves as a report to the project task our team undertook as part of the Natural Language Processing course offered at Innopolis University in 2020. Our team consists of three members: Syed Muhammad Zuhair Abbas, Kusal KC, and Eugene Murawjow.

2 Introduction

Chatbots are becoming one of the trending topics in the recent years. We, humans, have always preferred communication using our everyday language rather than with some made-up, counterintuitive rules and syntaxes. Chatbots, also known as “conversational agents”, are software applications specifically designed to mimic human-like written or spoken speech for the purpose of simulating a conversation or interaction with a real person.

Songs have become one of the important parts of human life. We can now listen to songs using various modern devices anywhere, anytime. However, getting information about a particular song is not always straightforward. Sometimes the user will have to navigate through pages of search engines to find the right information. In addition, some users often miss the right set of keywords that would have facilitated in finding the required song & information. Our attempt with this chatbot is to address this gap and create an agent that will take queries in a conversational, loosely typed human language and help get the required information. For the time being, the bot we have created is in preliminary stage which can detect intent and entities and respond to simple queries and is subject to future works for an extensive user experience.

3 Methodology & Implementation

We provide a list of examples and the chatbot will learn to adapt to variations of those examples. Given the example: “Who sang the song sugar?”, we decide that the action needed is *get_singer* with parameters { *song*: “sugar” }. The bot should also recognize basic elements of conversation like greetings and do the basic talks, e.g. when the user says “Goodbye”, the bot should respond with something along the lines of “Bye, see you later”.

3.1 Training the Chatbot

We train one extractor for each entity type. First, we create datasets for every entity. We then go to extract the sentences that contain references to each entity type we want to train an extractor for. Next thing we need to build is a mapping between intents and extractors. We don't want to apply all the extractors all the time. We want to apply an extractor only to the intents that contain samples with such entities. And, finally, we need to actually train the extractor.

We trained on a small dataset containing both generic and specific intents.

Sample dataset with generic intents:

```
'greetings': [
  ("Hello!", {}),
  ("Hi!", {}),
  ("How are you?", {}),
  ("Hi There!", {}),
  ("Hello there!!!", {}),
  ("Ello!", {}),
  ("Hey!", {}),
  ("Hello mate, how are you?", {}),
  ("Good morning!", {}),
  ("mornin'", {}),
],

'thanks': [
  ("Thanks dude!", {}),
  ("Thank you!", {}),
  ("Thanks", {}),
  ("awesome, thanks for your help", {}),
  ("Thank you so much!", {}),
  ("Cheers!", {}),
  ("Cheers, thanks", {}),
  ("many thanks", {}),
],

'bye': [
  ("Bye!", {}),
  ("Bye Bye!", {}),
```

Samples with song-specific intents:

```
'what-singer': [
  ("Who sang Sugar?", {'song': 'Sugar'}),
  ("Who sang Bohemian Rhapsody?", {'song': 'bohemian rhapsody'}),
  ("Who sang Halo?", {'song': 'Halo'}),
  ("Who is the singer of We Are One?", {'song': 'We Are One'}),
  ("The singer of Cold is?", {'song': 'Cold'}),
  ("singer of Moves Like Jagger is?", {'song': 'Moves Like Jagger'}),
  ("Who sang Killer Queen?", {'song': 'Killer Queen'}),
  ("Who is the singer of Senorita?", {'song': 'Senorita'}),
],

'what-release-year': [
  ("In what year was We are the Champions released?", {'song': 'We are the Champions'}),
  ("In what year was Hello released?", {'song': 'Hello'}),
  ("When was Somebody to Love released?", {'song': 'Somebody to Love'}),
  ("When was Time is Running Out released?", {'song': 'Time is Running Out'}),
  ("When was Gangam Style produced?", {'song': 'Gangam Style'}),
  ("What's the release year for Attention?", {'song': 'Attention'}),
  ("What's the release year for Numa Numa?", {'song': 'Numa Numa'}),
],
```

```

'get-similar-songs': [
    ("What are some similar songs to Memories", {'song': 'Memories'}),
    ("What are some similar song to Welcome to the Jungle?", {'song': 'Welcome to the Jungle'}),
    ("Can I get some similar song to Baby?", {'song': 'Baby'}),
    ("I want more song like Home Sweet Home ?", {'song': 'Home Sweet Home'}),
    ("I want more song like Not Afraid ?", {'song': 'Not Afraid'}),
    ("I want more song similar to Last Christmas ?", {'song': 'Last Christmas'}),
    ("I want to know more song similar to Complicated ?", {'song': 'Complicated'}),
    ("Get me more song like Numb ?", {'song': 'Numb'}),
    ("What are some song similar to In the End ?", {'song': 'In the End'}),
],

'get-latest': [
    ("What are some new songs?", {}),
    ("What are some new released songs?", {}),
    ("What are some new releases?", {}),
    ("Can I get some new songs?", {}),
    ("What's new?", {}),
    ("What are some new and cool songs?", {}),
    ("Tell me about some new songs?", {}),
    ("Tell me about new releases?", {}),
],

```

We trained extractor for each entity type with a Logistic Regression Classifier and the training results were as below (keep in mind, our training dataset is really, really small):

```

1 bot = Chatbot()
2 bot.train(TRAINING_SET)

```

Accuracy on training set: 0.9519230769230769
[[('Who', 'WP'), 'O'], (('sang', 'VBD'), 'O'), (('Sugar', 'NNP'), 'B-song'), (('?', '.', 'O')], [(['Who', 'WP'), 'O'],

3.2 Integration with WolframAlpha

Once the user query has been broken down into intents and entities, we need to handle them to get required information. These are done by action handlers. The action handlers themselves are based on directing specific intents to corresponding API calls that will return us the desired information. For development purposes, we chose WolframAlpha client python package to act as the API provider for us.

The choice of WolframAlpha was also based on the fact that we could not find an appropriate songs database that would have all required information for us; this was due to the copyright issues of the songs as we found. The use of this package is pretty simple. We simply direct the entities extracted from user query as the request data to the client package. This will respond with results if any that we further process to display to the user.

4 Results

The chatbot can recognize and respond to simple queries as you can see below. Let us see it action.

For greetings:



```
1 while True:
2     line = input("You: ")
3     print("SongBot:", bot.tell(line).strip(), "\n")
```

```
... You: hello
    > Intent: greetings
    SongBot: Hello yourself!
```

```
You: good morning
> Intent: greetings
SongBot: Hi!
```

```
You: bye
> Intent: bye
SongBot: Goodbye
```

For other basic stuffs:

```
You: you're an idiot
> Intent: rude
SongBot: That's pretty rude of you. Come back with some manners ...
```

```
You: how are you?
> Intent: chit-chat
SongBot: I'm fine, thanks for asking ...
```

```
You: tell me about yourself
> Intent: ask-info
SongBot: I come from the future!
```

For song specific stuffs:

```
You: Hi
> Intent: greetings
SongBot: Hi there!
```

```
You: Who sang Moon?
> Intent: what-singer
> Entities: {'song': 'Moon'}
SongBot: Björk
```

```
You: Who worked in Baby?
> Intent: song-team
SongBot: Glee Cast, Justin Bieber, Relient K
```

```
You: Thanks
> Intent: thanks
SongBot: Come back anytime
```

```
You: In what year was Hello released?
> Intent: what-release-year
SongBot: 2012
```

```
You: Can I get some similar song to Halo?
> Intent: get-similar-songs
SongBot: Say My Name, Formation, Telephone, Irreplaceable, Bootylicious, Drunk in Love, XO, Listen, Survivor
```

```
You: bye
> Intent: bye
SongBot: Goodbye to you!
```

For queries that are out of scope of the knowledge base:

You: can you tell me about life?

> Intent: ask-info

SongBot: I know a looot of songs, try me!

You: what is life

> Intent: None

SongBot: Sorry, I can't help you out with that ...

You: how to download this song

> Intent: get-similar-songs

SongBot: Can not understand you.

You: which are the best movies

> Intent: None

SongBot: Sorry, I can't help you out with that ...

You: which singers are the best

> Intent: None

SongBot: Sorry, I can't help you out with that ...

5 Difficulties & Limitations

One of the most significant difficulties currently is the lack of training examples which is why we have small training dataset used for now. Again, it turns out that songs datasets are restricted by copyright policies to be distributed which is why they are not available freely.

Another limitation is that the bot currently cannot recognize compound/complex queries consisting of multiple intents and/or entities. This is intentional as we only wanted to focus on addressing simple query case so as to better understand how these work under the hood.

6 Conclusion & Future Work

As we can see, the chatbot works and responds to user queries ranging from generic talk case to song specific cases. In human conversation, there are endless variations with which some intents can be expressed. This opens up the possibility to include more variations of the intents so the extractors can better understand the query and separate intent and entities out of it. Also, we humans tend to query with multiple intents naturally. The future work could focus on capturing these multiple intents present in compound queries, so that the user can be rewarded with a better, more complete response. Another future work is to deploy the bot on its own or in an existing platform (e.g., Telegram), which was not possible at this point due to time constraints.

Reference

Matt Schlicht. 2016. The Complete Beginner's Guide To Chatbots. <https://chatbotsmagazine.com/the-complete-beginner-s-guide-to-chatbots-8280b7b906ca>.

Amit Bhardwaj. 2020. Simple Chat-bot for recommending Netflix movies and TV -shows <https://chatbot-sliffe.com/simple-chat-bot-for-recommending-netflix-movies-and-tv-shows-bd411f7827eb>.

Nlpforhackers.io. 2018. Building Chatbots. <https://nlpforhackers.io/chatbots-introduction/>

Mina Park, Milam Aikem, and Laura Salvador. 2018. How do Humans Interact with Chatbots?: An Analysis of Transcripts. *International journal of management & information technology*, 14:3338-3350