



SG TRAVELBot

Your Smart Companion

By : PULIANI VIDUR, DONAL NGO, LI JINGMENG, JIANG YANNI, GONG YIFEI



Background

- - - - - x

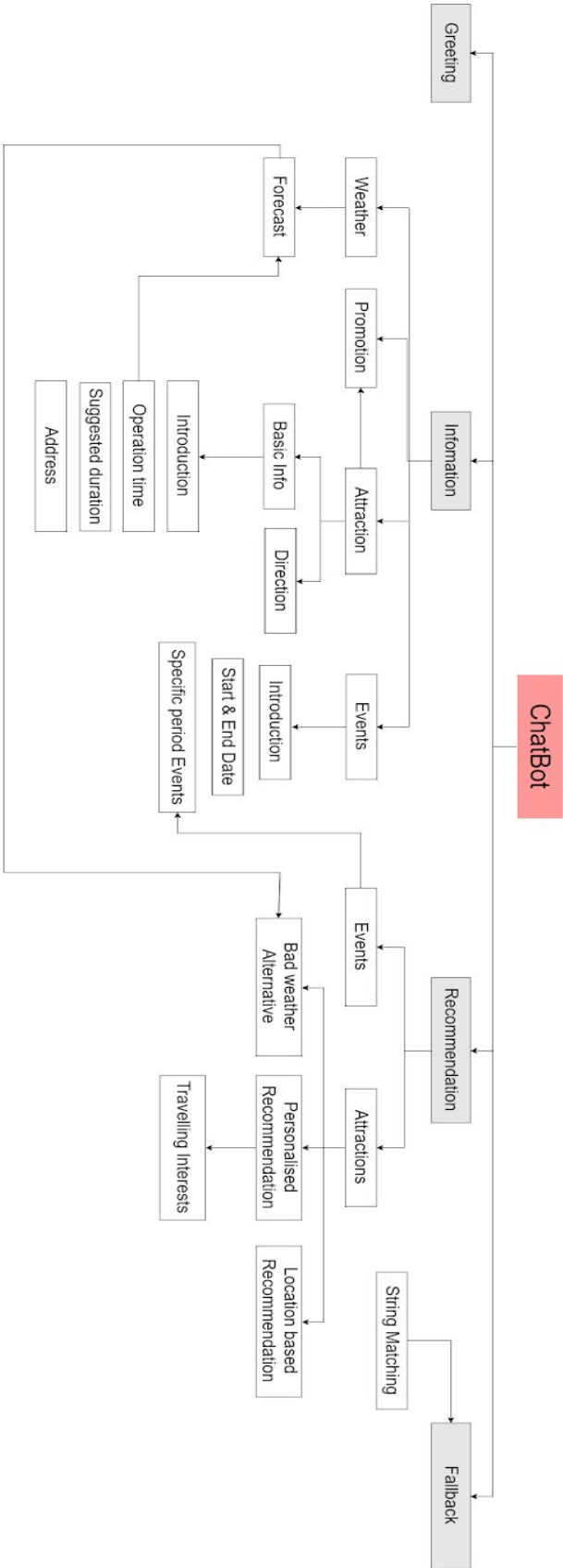
There exist a number of web and mobile applications such as **TripAdvisor** and **Klook** to assist tourists in planning their holiday to Singapore. The SG Travel Bot developed in this project assimilates data from these travel websites to provide services that can simplify the travel planning process and serve as the perfect travel companion for tourists in Singapore. This chat bot differentiates itself by providing personalised intelligent recommendations and information about attractions, events and weather through a single easy-to-use interface.

The SG Travel Bot has the following capabilities:

- Location based and personalised attraction recommendations.
- Retrieving information about attractions, promotions, events and weather advisory.
- Generating intelligent alternative attractions to visit using association mining in case of bad weather forecast
- Handle incorrect and incomplete attraction names or keywords to detect user's true intent

Theses capabilities have been implemented as intents in DialogFlow which have been described in more detail in the subsequent section.

Inference Diagram



RecomTypeIntent

This intent is used to detect users' preference between attending events and visiting attractions and accordingly return relevant recommendations for the user.

Slots & Entities

TypeThings:

This entity is used to distinguish whether the user prefer attractions or events recommendations. The value of it is 'attraction' or 'event'.

Recommendation Intent:

This intent is used to detect the user's intention for travel recommendations in the city of Singapore.

Slots & Entities

The following three slots are required to be captured before recommendations can be returned:

Traveller Type:

This slot can take one of the following entity values – Solo, Couple, Family, Friends and Business.

Travel Interest or category:

This slot captures the user's interests such as Nature, Museum, etc.

Location: This slot is used to gather the location area that the chatbot user is interested in. This is an optional slot and is assumed to cover the entire area of Singapore if the user does not provide any location related information. Additionally, it can also detect values such as 'nearby', 'around me' to capture the user's intent to search near their current location.

In case the Traveller Type and Travel Interest slots are not captured in the initial user request, follow-up questions are asked using fulfillment to capture the required slots. The user's responses to these follow-up questions are handled using DialogFlow 'Context', which allows certain follow-up intents to be triggered only when the context is active. These follow-up intents are described in more detail in the subsequent section. After capturing the slots from user, the knowledge base is queried to retrieve recommendations. The request is forwarded to a google maps API in case it is unable to retrieve recommendations from the constructed knowledge base.

Follow- up Intents

Intents coupled with 'Context' objects are useful in creating a conversation with the user that is connected across multiple requests and responses. Here context objects are used to trigger certain intents only when requesting for the missing slots. The parent Recommendation Intent activates the 'recommendationContext'. This context has a lifespan of 5 requests, after which these follow-up intents will not be activated. The follow-up intents used here are enumerated below:

Traveller Type Intent:

Intent to capture user's response about their traveller type

Category Intent:

Intent to capture user's response about their travel interests

User Info

Intent to detect user's response to a request for the current location of their device. This intent uses a built-in context 'Google Assistant Permission' provided by 'Actions By Google' for detecting user responses to permission requests.

Knowledge Base

The attraction recommendation knowledge base has been constructed by scraping attraction information and visitor reviews from [TripAdvisor](#) . The knowledge base is accessible through a series of APIs created using Spring Boot framework. These APIs query the scraped and transformed knowledge base from an embedded H2 database. The API used to retrieve recommendations from the constructed knowledge base is *GET {ipAddress}/recommendations*.

In case the constructed knowledge base is unable to find relevant recommendations, the request is fulfilled using the Google Maps API,

gmaps.places_nearby(location=geolocate,rank_by='distance',type=typename). This API returns several recommendations based on the category and location, and the results sort by distance.

Fulfilment

The fulfilment API tracks the slots that have been captured in the current conversation and accordingly returns follow-up questions to capture the required slots for travel recommendations. After all the slots have been successfully captured in a conversation, an API request is sent to Knowledge Base to get recommendations for the provided parameters. The recommendations are combined with direction links from Google Maps API, which is then formatted as a Browse Carousel using DialogFlow's built-in rich responses.

Get Attraction Info Intent:

This intent is used to get basic information of the attraction. Users can get useful information before and during their travels easily and quickly.

Slots & Entities

Slots:

- **Introduction:**
 1. Information Type
This slot is used to understand what type of information the user is asking for. This slot can take one of the entity values in InfoType Entity.
 2. Attraction name
This slot is used to identify the attraction that user is interested in. String Matching Algorithm is used when the input is not accurate.
 - **Operation time:**
 1. Information Type
 2. Attraction name
 3. Time(Optional)
This slot is used to capture when the user is going to visit the attraction. It can automatically transfer the fuzzy phrasing of date and time mentioned by the user into this format: YYYY-MM-DD[T]HH:mm:ss (SGT) and return the operation time on specified date. If that date is within four days, the handler will check whether this attraction is Outdoor Activity. If so, the handler will call the weather function to check whether that day will be raining and return the kindly remind.
When a user asks directly about operation time without mentioning a specific date or time, the intent will return the operation time table of attraction.
 - **Suggested Travel Duration:**
 1. Information Type
 2. Attraction name
 - **Address:**
 1. Information Type
 2. Attraction name
-

- **Direction:**

1. Information Type
2. Attraction name
3. Origin(Optional)

If the Origin is not specified, this function will use the geolocation of the user device as Origin.

4. Destination

InfoType Entity:

This entity is used to define types of basic information users can ask about an attraction.

Values of this entity are: introduction, operation time, suggested travel duration, address, direction and weather.

Knowledge Base

Google Maps API

This API is used to get address, operation time, geolocation and direction of attractions.

The knowledge base is accessible through a series of APIs created using Spring Boot framework. These APIs query the scraped and transformed knowledge base from an embedded H2 database. The list of APIs available to query the knowledge base are briefly described below:

Tourist Attraction Basic Information(introduction and suggested travel duration):

GET {ipAddress}/attractions{attractionId}

Fulfilment

The fulfilment API tracks the slots that have been captured in the current conversation. An API request is sent to Knowledge Base to get recommendations for the provided parameters. The recommendations are combined with direction links from Google Maps API, which is then formatted as a Browse Carousel using DialogFlow's built-in rich responses and suggestion chips.

Introduction:

This entity can get a brief introduction of attraction from TripAdvisor and a photo from Google Maps API.

Operation Time:

This entity can get the operation time by the user. If the date is not assigned, the handler will show the operation timetable. If the exact date is assigned and within four days, the handler will check whether this attraction is 'Outdoor Activities'. If so, this intent handler will call the 'weather' utility to check whether there will be raining on that day and get a kind reminder to the user.

Suggested Travel Duration:

This entity can get the approximate suggested travel duration of attraction from TripAdvisor .

Address:

This entity can get the address of attraction from Google Maps API.

Direction:

This entity can get the approximate travelling time with public traffic. The user can redirect to the Google Maps url to get the direction information in detail. This entity will use the origin and destination to get directions from Google Maps API. If user only assign the destination, this entity will use the geolocation of the user as the origin.

Promotions Intent

This intent is used to provide related promotions based on the attractions provided by users.

Slots & Entities

Attraction/Keyword:

It validates the attraction name using a String Matching Algorithm to find promotions similar to the user input.

Fulfilment

The fulfilment API tracks the slot for attraction names and returns list of promotion and related pictures to the users. The fulfilment also returns a set of guided suggestions to guide users to other information related to the attractions.

The API uses the Promotions Matching Algorithm as a knowledge base to find related attractions based on the captions of promotion provided by Klook website.

Knowledge Base

Promotions Matching Algorithm

Since an API is not available for Klook website, the Promotion Matching Algorithm is used as knowledge based to return the Promotions with matching names in the promotion captions in Klook Promotions Data. This will allow the chatbot to provide correct promotions when searching for a promotion for a particular attraction. This algorithm can work hand in hand with String Matching Algorithm.

Methodology

Algorithm is written on top of Fuzzy Wuzzy Algorithm to provide a heuristics and eliminate non-related candidate promotions before a brute force window search is performed. The brute force is done by initiating a search window which slides through the promotion caption to look for an exact match of a particular attraction to promotions. Given the example below, the algorithm will be able to find all 4 attractions in colored text within the promotion caption below:

[Sale] **Universal Studios Singapore** and **Skyline Luge/S.E.A. Aquarium** + **Adventure Cove Waterpark**/SIM Card Combo Packages

GetEventIntent

This intent is used to detect users' intention to get to know more detailed information about different kinds of events in Singapore and show the accessible events in a given period

Slots & Entities

Date period:

This entity is used to capture the date period that visitors want to know what events will happen. It is always a period of time, such as October, next month, etc.

Event name:

This entity is used to capture the name, keywords or part of the name of the events.

Interrogative:

This entity is used to distinguish different types of questions (basically, when and what) and help to get more accurate response.

Fulfillment

Events information:

This function is used to provide a list of events in a specific period that it helps visitors to decide when to travel if they want to join some events.

The fulfillment API tracks the slot for event names or keywords and returns a basic card of the event including its name, description, image and website link to the users.

Events list:

This function is used to provide detailed information about a specific event.

The fulfillment API tracks the slot for date period and returns a browsing carousel of events which happens during that specific period and each event shows its name, description and image.

Events duration:

This function is used to provide the start date and end date of a specific event.

The fulfillment API tracks the slot for event names or keywords and returns the start date and end date of the event.

Knowledge Base

For events information part:

The knowledge base is accessible through a series of APIs from The Tourism Information & Services Hub (TIH). This website includes lots of events information and it is the most suitable one we found to use as the events' part of knowledge base.

The list of APIs used in the project to query the knowledge base are briefly described below:

Events detailed information

<https://tih-api.stb.gov.sg/content/v1/event/search?keyword={eventskeyword} &apikey={key}>

Different types of events

<https://tih-api.stb.gov.sg/content/v1/event/search?keyword={type}&sortBy=type&apikey={key}>

Events' image

[https://tih.stb.gov.sg/bin/GetMediaByUuid?uuid={EventsImgUuid}&fileType=Medium%2520Thumbnail
&mediaType=image](https://tih.stb.gov.sg/bin/GetMediaByUuid?uuid={EventsImgUuid}&fileType=Medium%2520Thumbnail&mediaType=image)

EventDurationIntent

This intent is used to detect users' intention to get to know what events will happen during a specific time period after the user shows he wants to get some events recommendation.

Slots & Entities

Date period

This entity is used to capture the date period that visitors want to know what events will happen. It is always a period of time, such as October, next month, etc.

Fulfillment

This function is used to provide a list of events in a specific period that it helps visitors to decide when to travel if they want to join some events.

The fulfillment API tracks the slot for event names or keywords and returns a basic card of the event including its name, description, image and website link to the users.

Weather Info Intent

This intent is used to detect users' intention to visit an attraction in Singapore when asking for attraction information, and provide weather advice for them. For example, If a user asked about the opening time of an outdoor attraction on a specific day, the chatbot would check if it would be rainy that day. If it would be rainy, then the chatbot would remind the user about the weather and provide some alternative attractions. Also, this intent is used to provide answers when users ask directly about weather in Singapore.

Slots & Entities

Entities

InfoType Entity

As in Get Attraction Info Intent, this entity is used to define types of basic information users can ask about an attraction. Values of this entity are: introduction, operation time, suggested travel duration, address and direction.

Slots

Attraction Name

This slot is used to identify the attraction that user is interested in. String Matching Algorithm is used when the input is not accurate.

Time

This slot is used to capture when the user is going to visit the attraction. It can automatically transfer the fuzzy phrasing of date and time mentioned by the user into this format:

YYYY-MM-DD[T]HH:mm:ss (SGT). When a user asks directly about weather without mentioning a specific date or time, the intent will fill in the time slot with current date and time.

Information Type

This slot is used to understand what type of information the user is asking for. This slot can take one of the entity values in InfoType Entity. Only when the slot value is operation time will this intent be activated, as it can indicate a strong intention of user to visit a particular attraction.

Knowledge Base

APIs

Current weather API fromOpenWeather

This API is used to get current weather information. The response of this API includes current weather description,temperature, humidity, pressure and so on. Only the weather description part is needed.

4-day Weather Forecast API from Data.gov.sg

This API is used to get weather forecast information in four days. The response of this API includes weather forecast information in four days. With time slot successfully captures the datetime, the weather forecast information can easily be extracted from the API response.

Google Maps API

This API is used to get nearby attractions for generating alternatives.

Generating Alternatives

Methodology:

There are three conditions to be considered when deciding which attraction is to be recommended as alternatives: location, association, and similarity of categories. Every attraction in database will be assigned a score, which is calculated based on the three conditions. For location, the chatbot will search for places near the asked attraction within the radius of three kilometres through Google Maps API, and match the results with attractions in database. Any Matched attraction will get a higher score. For association, the chatbot will search for a matching association rule and give a higher score to the matched right hand side attraction. For similarity, as each of the attractions belongs to several categories, the chatbot will calculate the number of overlapping categories between the asked attractions and every other attraction in database. The more overlapping categories with the asked attraction, the higher score the attraction will get.

Association Rules

Association between attractions is extracted based on the reviews on [TripAdvisor](#). The review data is transformed into the format of 'transactions', that is, every tourist is a transaction, and the attractions they had been to are the groceries of the transactions. Apriori algorithm is used to extract the association rules. One example of the association rules is as below:

"lhs": "River Safari ", "rhs": "Singapore Zoo"

This association rule means that if a person wants to visit River Safari, then this person will also want to visit Singapore Zoo.

Fulfillment

The fulfillment API tracks the slots from input questions and returns the corresponding answers. If the user asks for operation time of an outdoor attraction and mentions a specific time, the fulfillment will request for weather information and generate alternatives if necessary.

Default Fallback Intent

This intent is matched when the users input does not match any of the intents created. It acts as a last layer to guide users back to main context of the Chatbot. Through testing, it is found that Dialogflow would not be able to handle input directly targeting a certain objective. Therefore, a fulfillment is used to enhance cognitive abilities, which provides suggestions directed to intents when users key a single attraction name, partial attraction name or keywords without any matching intents.

Fulfilment

The fulfillment API tracks the when intent falls under a Fallback intent. The full text is parsed into a custom made string matching algorithm as a knowledge base which returns location information the best possible matches based on the Scrapped Attraction's name and Keywords.

In situations where there are more than one recommended attraction returned, the API will return a Browsing Carousel of attractions and suggestion tabs for users to select from then return a basic card with basic information and promotions available for the locations and also Suggestion Tabs to guide users about the information leading to other intents in the API.

Knowledge Base

String Matching Algorithm

The String Matching Algorithm is a custom algorithm specially designed as a knowledge base on top of the scraped data from TripAdvisor. It serves as an enhancement to the cognitive ability of Dialogflow to “interpret” the inputs of users when a non-exact input is detected in an intention slot or string and returns the most similar result.

Methodology

Names and keywords are simultaneously broken down into feature vector using One-Hot encoding method. Each attraction is assigned a 2D vector space with an array of Name features and Keywords features creating a search space stored in the working memory of the Fulfillment API.

By using the input features by the user in comparison with TripAdvisor attraction name and keywords, it returns the “Nearest Neighbors” based on the calculation of Bray-Curtis Distance with the pre-defined vectors. A penalty is also implemented to space attractions without a known feature to

provide a much more accurate output. With exact matches (Bray-Curtis Distance= 0) it return only one result to the user.

Use Cases

It is useful when the user does not remember an attraction name in any particular order or any form; this algorithm will be able to return results using partial wordings (eg. Bay Gardens (input) => Gardens by the Bay (output)) or keywords (eg. War (input) => Battlebox, Fort Siloso (output)). This is also useful when any intent is not met, where user text can be processed to match similar strings or attraction to guide user back to the preloaded intent of the chatbot.

Search Intent

Returns a list of attractions with basic information and promotions (if any) matches when user searches for keywords or partial attraction names. Guides users to the next action regarding the attraction.

Slots

Attraction name or keyword

Uses attraction name and keywords to find information of attraction. This slot value is stored to provide information to the next intent.

Fulfillment

The identified slots is parsed into a custom made string matching algorithm as a knowledge base which returns location information the best possible matches based on the Scrapped Attraction's name and Keywords.

In situations where there are more than one recommended attraction returned, the API will return a Browsing Carousel of attractions and suggestion tabs for users to select from then return a basic card with basic information and promotions available for the locations and also Suggestion Tabs to guide users about the information leading to other intents in the API.

Future Improvements

- ❖ Engineer an ETL pipeline to update the knowledge base at regular intervals
 - ❖ Use crowd information from Google to suggest non-peak hours for better experience
 - ❖ Personalize travel with Google Data Analytics to provide better information.
 - Filtering out visited attractions
 - Hotel Location and provide traffic conditions back to the hotel
-

Contributions

Creators	ID	Contributions	Email
Puliani Vidur	A0198492L	Tripadvisor Knowledge Base Recommendation, Database API, Category, TravellerType and UserInfo Intent	E0402033@u.nus.edu
Ngo Jinze Donal	A0198487A	Tripadvisor, Klook, String Matching, Promotion Matching Knowledge Base. Promotions, Fallback and Search Intent	E0402028@u.nus.edu
Jiang Yanni	A0201097M	Data Cleaning, Attraction Info Intent, Project Integration and Troubleshooting	E0409752@u.nus.edu
Li Jingmeng	A0198484J	Data Cleaning, Weather Info Intent, Alternatives Knowledge Base, Project Integration and Troubleshooting	E0402025@u.nus.edu
Gong Yifei	A0198495E	Event Knowledge Base,GetEvent, RecomType, EventDuration Intent, Recommendation,Troubleshooting and video	E0402036@u.nus.edu
