

Chatbot for Tourists in Singapore Olive FoodBot

Project Report



Team Members

Edmund Leow Kwong Wei A0198458H

Ng Mei Ying A0198546L

Wilson Lum Kok Keong A0198478A



Contents

1	Exe	ecuti	ve summary	1
2	Pro	ject	objective	2
3	Kno	owle	dge acquisition	2
4	Kno	owle	dge Representation	3
5	Sys	tem	architecture	3
6	Cha	atbot	design	4
(5.1	Cha	atbot Personality	4
(5.2	Ent	ities	5
(5.3	Co	nversation flows	7
	6.3.	.1	Search for popular hawker centres	7
	6.3.	.2	Search for popular restaurants	10
	6.3.	.3	Find information on a specific eatery	16
(5.4	Fin	d information on a specific local dish	19
(5.5	Dis	cussion	21
7	Cha	allen	ges and limitations	22
,	7.1	Yel	p API	22
,	7.2	Dia	llogFlow	23
	7.2.	.1	Multiple responses with time lag not supported	23
	7.2.	.2	Lack of indication that chatbot is fetching responses	23
	7.2.	.3	Lack of control over knowledge base	23
	7.2.	.4	Unconfigurable webhook timeout	23
	7.2.	.5	Limited rich response options for platforms	24
8	Fut	ure e	enhancements	24
9	Cor	nclus	ion	24
10	Ref	eren	ces	25



1 Executive summary

Home to a diverse array of cuisines that integrate influences from food cultures of different immigrant groups who settled in Singapore and an abundance of dining establishments with a wide range of prices to fit any budget, it is little wonder that Singapore is sometimes described as a food paradise.

Not only is food an integral part of Singaporeans' way of life, it also plays a significant role in the country's tourism. Tourism receipts climbed one percent to \$27.1 billion in 2018, with food and beverage making up 9% of tourism spending [1].

To enhance the dining experience of tourists, we developed a chatbot that can assist them in their search for eateries, as well as, provide information on local dishes in Singapore.

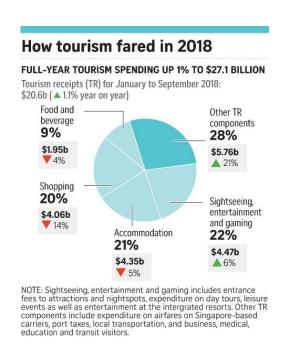


Figure 1 Tourism spending in Singapore in 2018 [1]



2 Project objective

The objective of this project is to create a chatbot that will guide tourists in Singapore in their food adventures, providing them with information on popular eateries and local food quickly and efficiently.

Sifting through copious amounts of information from numerous sources such as business review websites, blogs and food guides to understand Singapore's food culture and select dining venues can be a tedious and time-consuming affair. Thus, we developed a chatbot to facilitate tourists in their search for eateries or information on common dishes, providing concise responses instantaneously. For a more personalised experience, the searches are customised based on their food preferences and budgets.

3 Knowledge acquisition

While brainstorming for the design of the chatbot, we searched websites related to the local foodscape. Much of the information was on reviews of the best food or eateries in Singapore, with the information scattered across different websites such as business review websites, online food guides and blogs.

Web scraping of several websites was explored with mixed results. A few websites which had stable and consistent HTML tags and sub-headers tags were shortlisted but other websites were not considered either because they had messy HTML or web scraping was forbidden.

We also considered retrieving data from business review websites. A 2018 survey by ReviewTrackers found that consumers were most likely to check Google reviews before visiting a business; Yelp ranked second, followed by TripAdvisor and Facebook [2]. But Google reviews were not available for this project as access to Google My Business API should only be requested by companies directly developing an application with the API. On the other hand, detailed information in Yelp listings including rating, price bracket and address could be obtained by calling the Yelp API.



Information obtained by web scraping of a website on top eateries in Singapore was less complete than that retrieved by the API, so we decided to utilise the Yelp reviews in the search for popular eateries and obtain information on a specific eatery. In addition, two websites were used as a knowledge base to provide description of local dishes to users [3, 4].

4 Knowledge Representation

The chatbot is designed to perform the following:

- 1) Search for popular hawker centres
- 2) Search for popular restaurants
- 3) Find information on a specific eatery
- 4) Find information on local dishes

Hawker centres and restaurants are classified into separate categories as hawker centres hold a special place in Singapore's food heritage, offering multicultural food at affordable prices and are arguably where the best local dishes can be found. In contrast, restaurants tend to focus on a particular type of cuisine and have more diverse pricing levels.

5 System architecture

The chatbot is developed on DialogFlow due to its strong natural language processing (NLP) capability, ease of setup and wide integration with multiple platforms including Google Assistant, Slack and Facebook Messenger. Slack, a popular communication platform with good chatbot support, is used as the front end for our chatbot.

When a user enters a message in Slack, its API sends the message to DialogFlow. DialogFlow's NLP engine then processes the intent of the input and an appropriate response is then returned to the user. Fulfilment of intents with webhook is done with Python and Flask and the webhook is deployed on a Heroku server (See *Figure 2*).



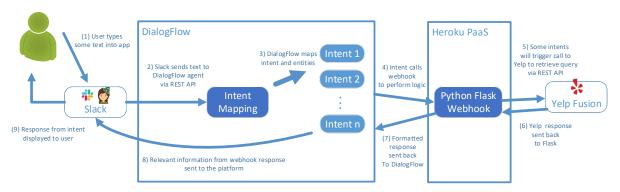


Figure 2. System Architecture for FoodBot

6 Chatbot design

This section discusses the different aspects of chatbot design on Dialogflow, including chatbot persona, entities, intents and conversation flows.

6.1 Chatbot Personality

Users might not think of our chatbot as their friend, but they will naturally interact with it in a direct, personal way if we build the chatbot's responses to anticipate a back-and-forth conversation. These personal interactions can build a greater sense of connection and loyalty to our brand than can be achieved with emails or websites or social media posts. But if our chatbot doesn't have a well-defined persona, this loyalty can easily be lost [5].

There are 12 archetypes commonly used to define a brand persona, which are functionally identical to the 12 Jungian archetypes [6]. As our target audience are tourists in search of good food, we shall stereotype them as Explorers - people who love to explore the world, are not afraid of the outdoors and keen for different experiences (in this case, food).

We want our chatbot to have a personality complementary to our target group and feel that the Magician archetype would be most appropriate – the chatbot is seen as one with special access to secret or elusive information on Singapore's foodscape and can transform users from a confused state to satisfaction and happiness. This Magician archetype is projected in



the chatbot's interactions with users. For example, when returning search results, it might start the reply with something like "\$\frac{1}{2}\text{Presto!} Here are your results!".

In coming up with a suitable name for the chatbot, we wanted something memorable and inspired by food. Finally, Olive was chosen for its nutrition and health properties. It is also associated with the Olive branch, a symbol for peace, wisdom and victory, deriving from the customs of ancient Greece [7, 8]. These will also help bring out the persona of our chatbot and enable our users identify with the chatbot.

DialogFlow's Small Talk was implemented to enhance the personality of the chatbot, with modifications made to the responses of the causal intents. A number of intents was also deleted to avoid interruptions to conversations.

6.2 Entities

DialogFlow uses entities to extract parameter values from user inputs. We created developer entities for matching custom information such as cuisine, budget and name of eatery (See *Table 1*).

Entity	Description
@budget	Price bracket of restaurant
@cuisine	Type of cuisine
@eatery	Type of eatery
@foodItem	Name of local food
@hawkerCentre	Synonyms for hawker centre
@no	Synonyms for no
@restaurant	Synonyms for restaurant



@RestaurantName	Names of eatery
@RestaurantNoPreferredLocation	Indicator for preferred location of eatery
@yes	Synonyms for yes

Table 1. Entities



Intent

Python Webhook

Master of Technology (Intelligent Systems)

6.3 Conversation flows

An intent defines a user's intention for one conversation turn and many intents can be defined and combined to handle a complete conversation. The figures and tables in the following subsections illustrate the conversation flows for different tasks.

6.3.1 Search for popular hawker centres

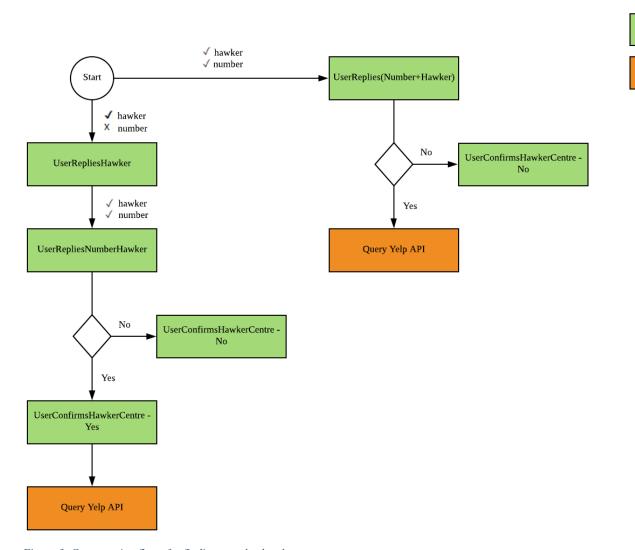


Figure 3. Conversation flows for finding popular hawker centres

A list of popular hawker centres is displayed upon the user's request, with the length of the list specified by him or her. A maximum of ten results can be returned to avoid webhook timeout as DialogFlow limits the agent's response time to five seconds. There are two types of conversation flows for this task (*Figure 3*) which has two required parameters: the term hawker centre and number of results. The conversation can be linear with parameters provided one at



a time by the user or it can be non-linear, with the user providing all required information in one input. *Table 2* shows the sample utterances and responses and *Figure 4* illustrates a conversation on this search on the platform Slack.

No	Intent	Sample	Sample	Context	Parameters
		utterence	response		
1	UserRepliesHa wkerCentre	What are the famous hawker centres in Singapore?	How many hawker centres would you like to search for? It should be between 1 to 10.	User requests to search for hawker centres.	\$hawkerCentre
2	UserRepliesNu mberHawkerCe ntre	6 is enough.	Please confirm that you're searching for 6 hawker centres.	User indicates number of results to return.	\$number
3	UserConfirmsH awkerCentre	Yes	Here are some of the most popular hawker centres in Singapore!	User confirms search request.	GetHawkerCent re

Table 2. Intents for finding popular hawker centres in linear conversation



☆ OliveFoodBot Messages About Mei Ying 9:51 PM Hello. OliveFoodBot APP 9:51 PM 👋 I am Olive. How can I help you today? 😊 Here's a couple of the things I can do: • Recommend a dining place. (eg Find popular eateries) • Get a specific restaurant's details such as its location and rating. (eg Din Tai Fung in Orchard) • Tell you about specific local food in Singapore. (eg What is Cla... See more Get eatery information Find popular restaurants Find popular hawker centres Mei Ying 9:51 PM What are the famous hawker centres in Singapore? OliveFoodBot APP 9:51 PM How many hawker centres would you like to search for? It should be between 1 to 10. Mei Ying 9:51 PM 6 is enough. OliveFoodBot APP 9:51 PM Please confirm that you're searching for 6 hawker centres. Yes Mei Ying 9:51 PM Yes OliveFoodBot APP 9:51 PM Here are some of the most popular hawker centres in Singapore! Click the restaurant name to get more info and click on address to get directions 1. Maxwell Food Centre \$ • hawkercentre 1 Kadayanallur St, Singapore 069184, Singapore

(3 kB) •

+6562255632

★ ★ ★ ★ yelp

See more

Message OliveFoodBot

Figure 4. Linear conversation on finding popular hawker centres on Slack



6.3.2 Search for popular restaurants

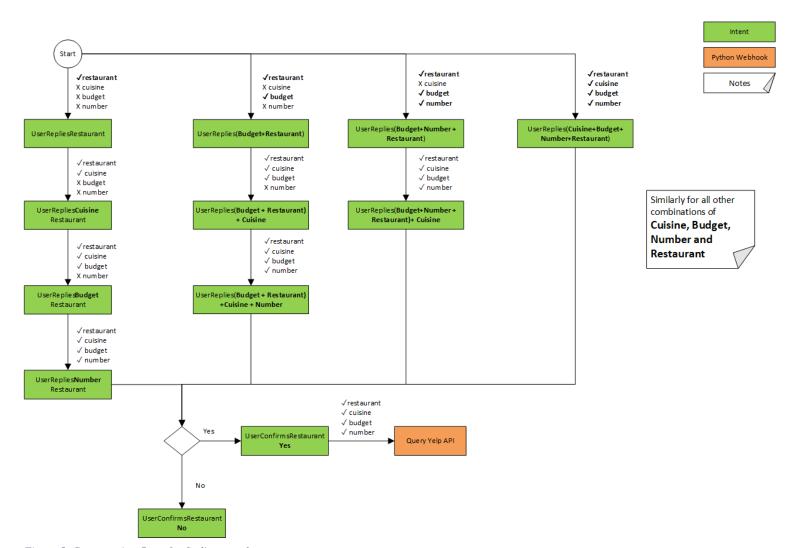


Figure 5. Conversation flows for finding popular restaurants

The second task is to search for popular restaurants. Four parameters are required: the word restaurant, type of cuisine, budget in terms of restaurant price category, and number of search results. The various ways in which the conversation can proceed depending on the order of receipt of parameters have been implemented in the chatbot.

The intents are chained together through input and output contexts and parameters are captured after intent matching. When all the required parameters have been received and the user confirms the search request, a webhook-enabled intent is invoked and the Yelp API is called, returning a list of restaurants.



No	Intent	Sample	Sample	Context	Parameters
		utterence	response		
1	UserRepliesRe staurant	I'm searching for nice restaurants.	Which cuisine do you like? For instance, do you want Mexican, French or Japanese cuisine?	User requests to search for restaurants.	\$restaurant
2	UserRepliesCu isineRestaurant	I like Japanese food.	What is your budget like? That is, do you prefer moderately priced, somewhat expensive or expensive restaurants?	User indicates cuisine type.	\$cuisine
3	UserRepliesBu dgetRestaurant	I'll like to dine at some place that's expensive.	How many restaurant are you searching for? Enter a number between 1 to 10.	User indicates price category of restaurant.	\$budget
4	UserRepliesNu mberRestauran ts	Maybe 5?	Sure. Are you searching for 5 expensive	User indicates number of	\$number



			Japanese	search results	
			restaurant?	to return.	
5	UserConfirms	Yup.	Here's a list of	User confirms	
	Restaurant		popular	search request.	
			restaurant in		
			Singapore!		

Table 3. Intents for finding popular restaurants in conversation with one parameter captured at each conversation-turn



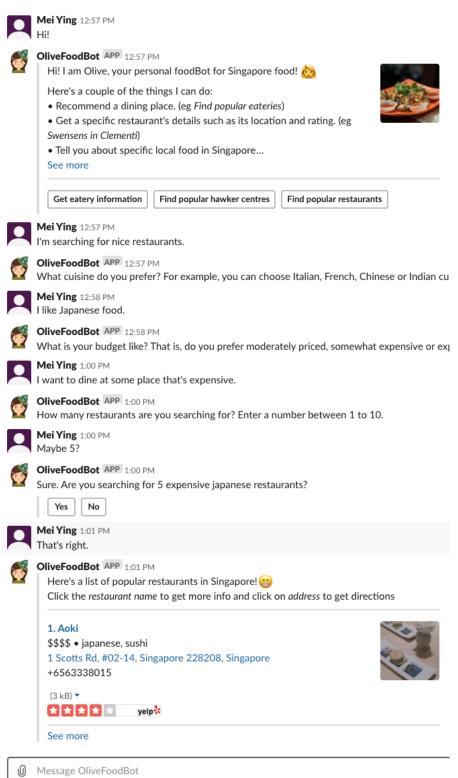


Figure 6. Conversation for finding popular restaurants on Slack with one parameter captured at each conversation turn



No	Intent	Sample	Sample	Context	Parameters
		utterance	response		
1	UserReplies(B udget+Restaur ant)	I'm searching for affordable restaurants. Can you help?	Sure. What type of cuisine do you want?	User requests to search for restaurants in a particular price category.	\$budget \$restaurant
2	UserReplies(B udget+Restaur ant)+Cuisine	I want to find Indian food.	How many restaurants do you want to find? It should be between 1 to 10.	User indicates number of search results to return.	\$cuisine
3	UserReplies(B udget+Restaur ant)+Cuisine+ Number	Please show me 8 locations.	Noted. Are you searching for 8 affordable Indian restaurants?	User confirms searchrequest.	\$number
4	UserConfirms(Budget+Restau rant+Cuisine+ Number	That's right.	Here's a list of popular restaurants in Singapore!	-	-

Table 4. Intents for finding popular restaurants with two parameters at first conversation turn



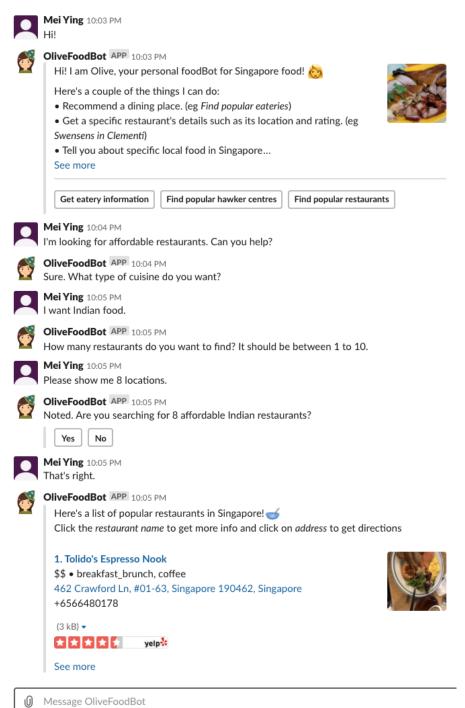


Figure 7. Conversation on searching for popular restaurants with two parameters at first conversation turn



6.3.3 Find information on a specific eatery

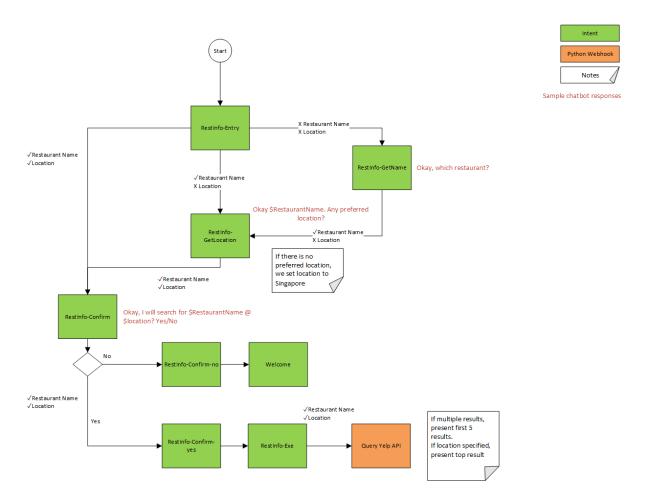


Figure 8. Conversation flows for finding information on a specific eatery

The third task is search for details of a particular eatery. Unlike in the previous two tasks, parsing of parameters and validation are done via the webhook (See Figure 8).



No	Intent	Sample	Sample	Context	Parameters
		utterance	response		
1	GetRestInfo- Location	Can you find me the details of a restaurant?	Which restaurant are you interested in?	User requests to search for details of a particular eatery,	\$budget \$restaurant
2	GetRestInfo- Location	I want to know about Candlenut.	Alright! Let's get details on Candlenut. Any preferred location?	User indicate name of eatery.	\$cuisine
3	GetRestInfo- Confirm	No.	Ok great! I will get the top results for Candlenut located in Singapore?	User replies whether there is any prefereed locationto search for.	\$number
4	GetRestInfo- Exe	Sure.	Presto! Here are your results for Candlenut located in Singapore.	User confirm search request.	-

Table 5. Intents for finding information on a specific eatery



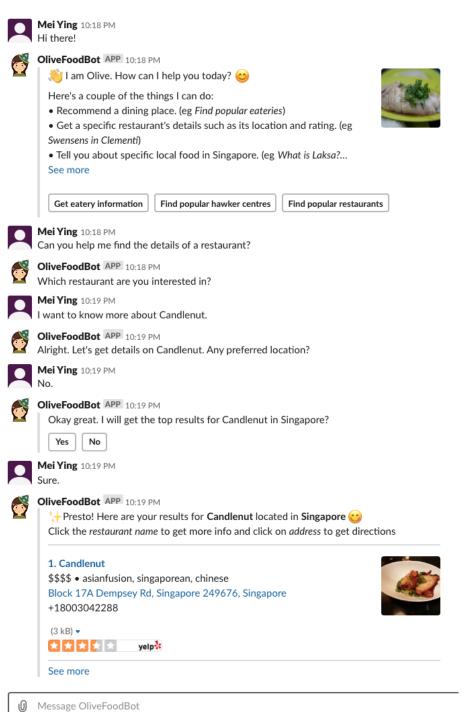


Figure 9. Conversation on finding information on a specific eatery on Slack



6.4 Find information on a specific local dish

The last task is to provide information on a local dish to the user. A single intent is created for this purpose (*Table 6*) and when it is triggered, the webhook looks up a database containing descriptions of different dishes and also calls the Yelp API to retrieve a list of eateries selling the dish. This information is then returned to the user.

No	Intent	Sample	Sample	Context	Action/Param
		utterance	response		eters
1	WhatIsDish	Can you tell	Literally fried	User requests	\$foodItem
		me what	Fujian noodles.	for information	
		Hokkien Mee		on a particular	
		is?		food or dish.	

Table 6. Intent on finding information on a specific local dish



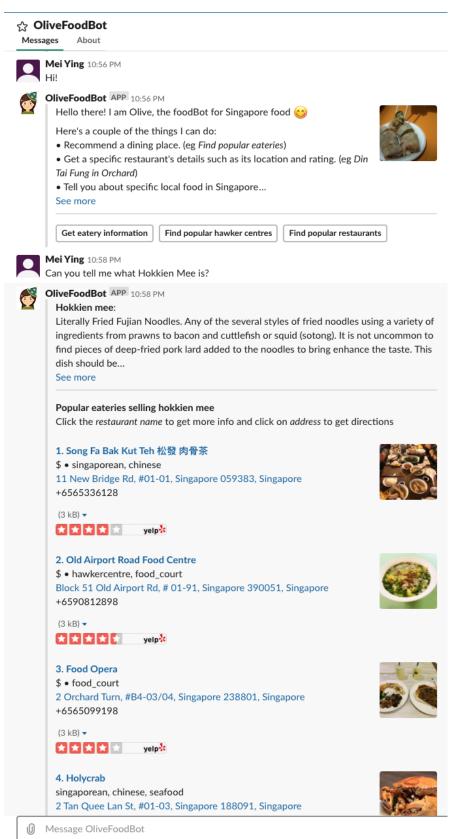


Figure 10. Conversation on finding information on specific a local dish on Slack



6.5 Discussion

We explored two different methods to determine if parameters have been captured. In the tasks to search for popular eateries, this validation is done in Dialogflow, whereas in the task to find information on a specific eatery, validation and redirection of intents are done through webhook. We found the first method to be more time-consuming as more intents have to be created, but it facilitates troubleshooting as most of the conversation is handled by the intents themselves.

It is important to ensure that conversations proceed in an expected manner. To this end, we deleted Small Talk intents that would potentially disrupt the conversations and generally kept the lifespans of output contexts to one to avoid wrong intents from being invoked.

Many of us have had not-so-pleasant experiences with automated telephone operators where we had to endure through a long series of button presses. Thus we kept button menus to a minimum in this chatbot to allow users to express themselves more freely. In addition, input contexts were not defined for the first intents of each type of conversation to enable them to initiate a new conversation at any point during the session as this made for a more natural interaction, mimicking a human conversation where people sometimes change the topic of a conversation midway. But there is a downside to omitting input contexts for these intents, as they are then at risk of being wrongly invoked. To avoid this, more training phrase were added to these intents.

The chatbot is also designed to capture multiple parameters in a single user input. For instance, when a user requests to "find restaurants that serve Mexican cuisine", the parameters restaurant and Mexican are captured, and the remaining parameters are received downstream as the chatbot prompts the user for more information (*Table 4*). It can even receive four parameters simultaneously provided enough information is provided by the user, although this might be uncommon.

Knowledge connectors are a beta feature in DialogFlow. When a user message is not matched to any conversational intents, the chatbot can parse the data sources from the knowledge base to craft an appropriate response. We explored how DialogFlow's knowledge base could complement the conversational intents. We compiled a list of FAQs introducing famous food



dishes in Singapore, while Wikipedia articles were added as knowledge base articles. Testing different confidence levels, we found that responses based on knowledge base articles tended to be better and more detailed when the confidence was increased to 0.7 from 0.1. Moreover, we found that if the user query was too general, an irrelevant response might be obtained. A confidence level of 0.1 though was found to be sufficient for accurate responses parsed from the FAQs if the questions were spot-on.

Occasionally, DialogFlow's knowledge base disrupted the conversation as the system-generated intents for the knowledge base articles were unexpectedly triggered instead of the conversational intents, despite adjusting the confidence level. Other times, responses were not parsed from the documents in the knowledge base as intended. Eventually we decided to define an intent for the FAQs instead which would return an appropriate response via the webhook based on the the list of FAQs stored in Excel.

7 Challenges and limitations

We encountered several issues with the Yelp API and DialogFlow while working on the chatbot and will elaborate on them in the following subsections.

7.1 Yelp API

A more customised search that filters eateries based on their opening hours was not possible as the Yelp API that fetches multiple results does not provide information on the opening hours or operating status at the point of query.

In addition, there is a call limit for the Yelp API of 5000 API calls per 24 hours. While it is unlikely that we will exceed this limit in this project, we have set the message "It appears that our backend is overloaded. Please try again tomorrow" to be displayed should this occur. In an actual deployed system, we could resort to web scrapping directly from the website as a stopgap measure, or utilise caching of results.



7.2 DialogFlow

7.2.1 Multiple responses with time lag not supported

A limitation of DialogFlow is that is not possible for the agent to provide multiple responses. For instance, we would like for the chatbot to respond in the likes of "Sure, please wait a while." when the user confirms a request, after which the Yelp API is queried using the webhook and the search results are returned to the user. A potential solution would be to return the above message together with the search results, but it is not ideal as the absence of a time lag between the display of the message and results makes the dialogue somewhat unnatural.

7.2.2 Lack of indication that chatbot is fetching responses

During testing, we found that it can take some time to query the Yelp API and render the responses while the user is kept waiting, sometimes wondering why the chatbot isn't responding. Having a chat bubble that says "Typing..." would help signal that the chatbot is currently processing the request.

7.2.3 Lack of control over knowledge base

DialogFlow does not have a feature to control when the system-generated intents linked to the knowledge base articles are triggered, as mentioned in an earlier section. It is also unclear what these intents actually are. If output contexts could be linked to the knowledge base like the conversational intents, we might be able to utilise the knowledge base articles in a more deterministic manner.

7.2.4 Unconfigurable webhook timeout

There is an unconfigurable five seconds timeout for webhooks in DialogFlow. The aim of this feature in Google DialogFlow is to ensure that the chatbot responds within an acceptable time limit for a natural conversation. We did tests to benchmark the duration of API calls and found them to be roughly constant at around 0.4 to 0.6 seconds. As a precaution, we defined a rule stating that the API cannot be called more than five times successively in a single response.



7.2.5 Limited rich response options for platforms

In-built rich responses for platforms were limited, so platform-specific APIs were used to build a better user experience, but these APIs have their own limitations. For example, there are branding requirements stipulated by Yelp to be fulfilled to comply with usage guidelines. However, their implementation is limited by the Slack API - the Yelp logo and rating stars images cannot be resized or positioned freely, so we had to manually resize and merge the logo and rating stars together.

8 Future enhancements

Currently, the chatbot is integrated with just one platform Slack due to time constraints, but it can be integrated with other platforms like Facebook Messenger in future to reach a wider user base. The Slack comments features can also be added for the convenience of advanced users. Implementation of Yelp API calls with GraphQL would be advantageous as it allows us to retrieve precisely what we need in a single query, reducing the risk of timeout and improving performance time. For a more personalised experience, the chatbot can query for certain user details, for example name, which can then be used during the session.

9 Conclusion

In this project, a chatbot system is developed on DialogFlow to aid tourists in their search for information on local dishes and eateries in Singapore. We defined a limited scope for the chatbot as chatbots tend to work well for specific tasks.

It has been a steep learning curve, and our team picked up valuable skills in the process, such as how to call external APIs, do web scraping, call external APIs and send rich messages in webhook responses. Most importantly, we gained practical experience in building a chatbot, seeking ways to build less hierarchical conversations and improve user experience.



10 References

- [1] T. F. Tay, "Tourist arrivals, spending in Singapore at record highs," Straits Times, 14 February 2019. [Online]. Available: https://www.straitstimes.com/singapore/tourist-arrivals-spending-at-record-highs. [Accessed 28 August 2019].
- [2] "2018 ReviewTrackers Online Reviews Survey," reviewtrackers, 2018. [Online].

 Available: https://www.reviewtrackers.com/reports/online-reviews-survey/. [Accessed 28 August 2019].
- [3] "Singapore Food Guide & Restaurant Directory Food Categories," [Online]. Available: http://www.goodfood.sg/categories.
- [4] "Singapore Food Dictionary," The Ordinary Patrons, [Online]. Available: https://ordinarypatrons.com/singapore-food-dictionary/.
- [5] V. MacMillan, "How to Develop a Chatbot Persona That Fits Your Brand," Medium, 26 April 2018. [Online]. Available: https://chatbotslife.com/how-to-develop-a-chatbot-persona-that-fits-your-brand-c48055970372.
- [6] N. Smith, "Does Your Brand Have Multiple Personality Disorder? A Look at Brand Archetypes," nvision designs, 10 November 2015. [Online]. Available: http://www.nvision-that.com/design-from-all-angles/what-is-your-brand-personality-a-look-at-brand-archetypes.
- [7] "Olive Branch," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Olive_branch.
- [8] M. Awana, "Olive Branches in Greek Mythology," Classroom Synonym, [Online].

 Available: https://classroom.synonym.com/olive-branches-greek-mythology-13999.html.