

Project Jiak*Bot

**jiak - literal translation of the Hokkien verb “Eat”*

Brijesh Gupta, Chan Wesley, Kuar Kah Ling, Tham Jun Quan,
Thng Ren Jing Chris, Jiang Chen Yang Thomas



Overview

- Inspired from a common problem working class adults in Singapore's Central Business District (CBD) faced - where and what to eat for lunch
- Aims to suggest places to eat through a chatbot
 - Interactive user experience
 - More informed choice
 - Immediate responses
 - Built using Natural Language Processing (NLP) techniques

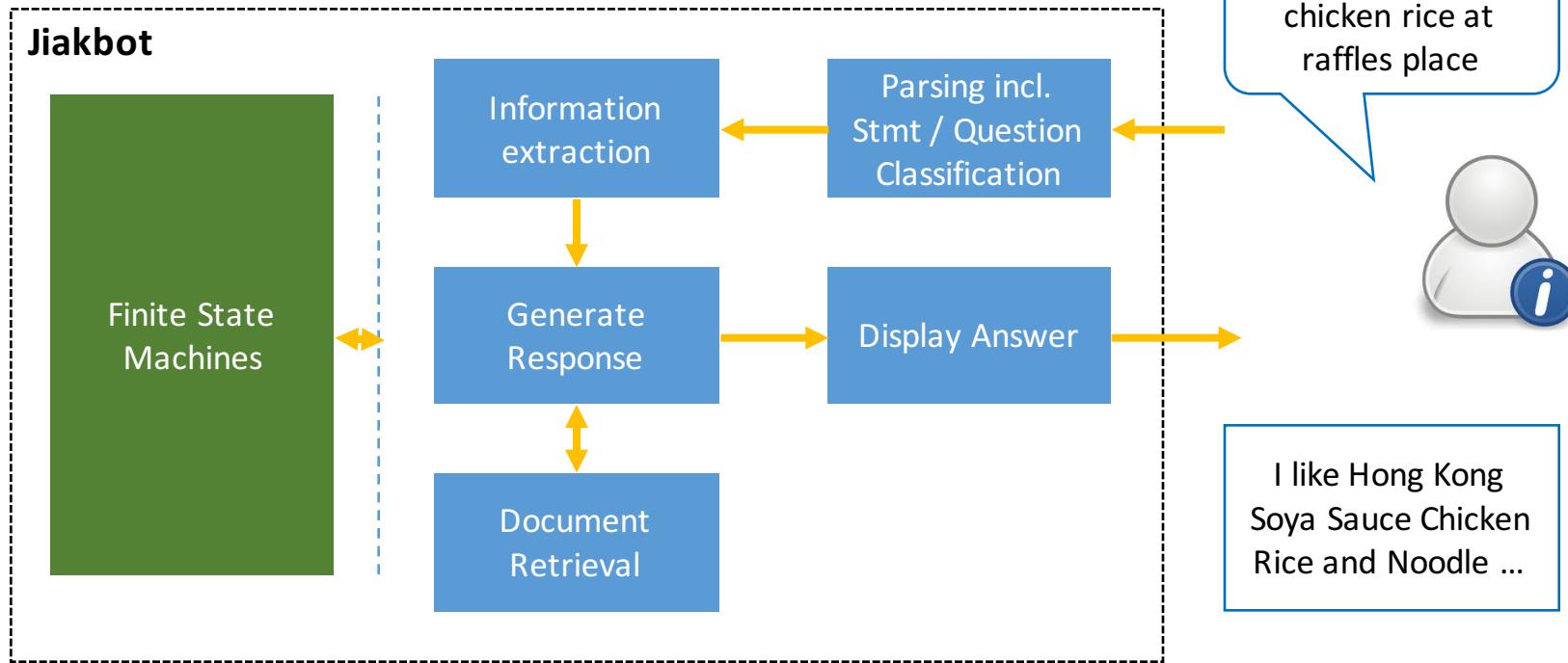


Overview

DATA

Scrape businesses data from Yelp including their reviews - Central Business District Area

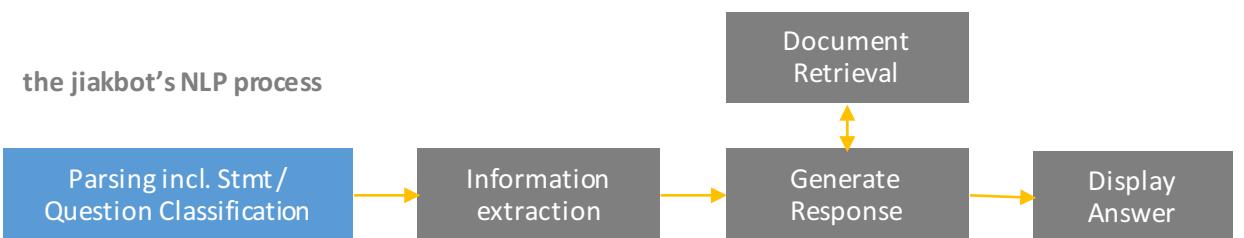
LEVERAGING THE USE OF NLP TECHNIQUES



How did we do it ... exactly?

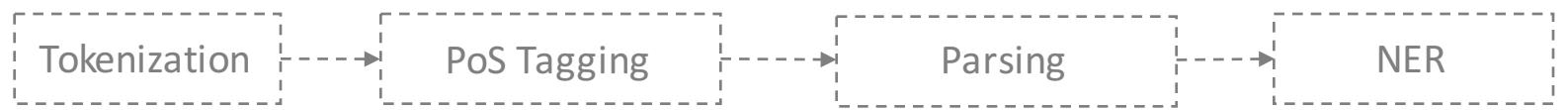
PARSING INCLUDING STATEMENT / QUESTION CLASSIFICATION

- Tokenize & part-of-speech tagging using NLTK
- Removed stopwords
- Trained question classifier to predict the type of user input i.e. statement, question, rhetorical question using Yelp data and manually constructed training data.

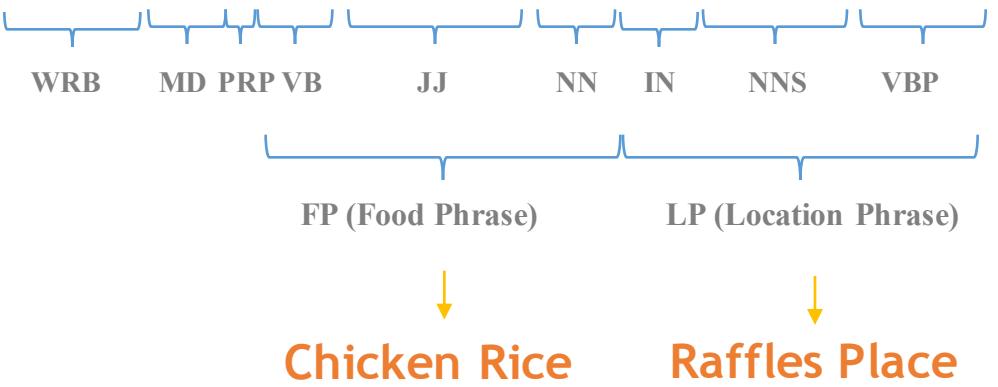


How did we do it ... exactly?

INFORMATION EXTRACTION

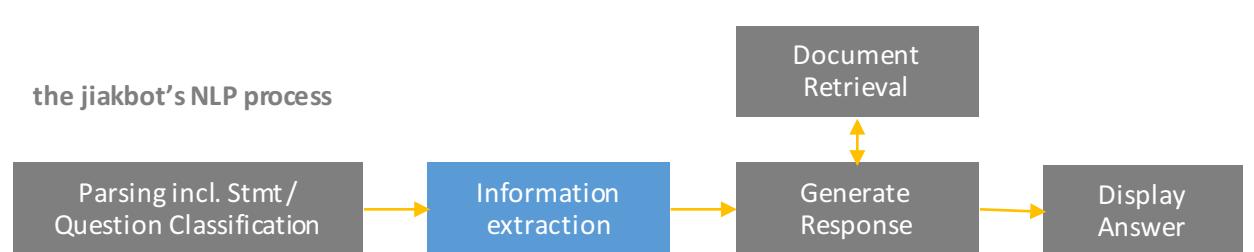


- Where can I get Chicken Rice at Raffles Place



Segmentation and
labelling at both the
Token and Chunk
levels

the jiakbot's NLP process



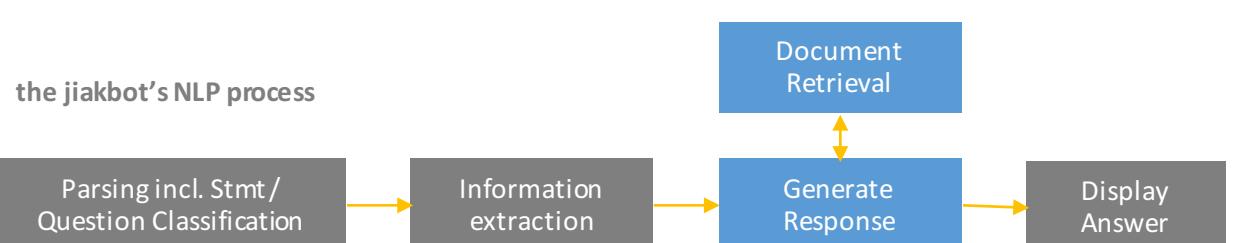
How did we do it ... exactly?

DOCUMENT RETRIEVAL

- Extract related data from database using SQL statements based on information extracted
- Compare statements in database and user input using cosine similarity and returning a relevant statement with highest similarity
- Passes output to generate a response

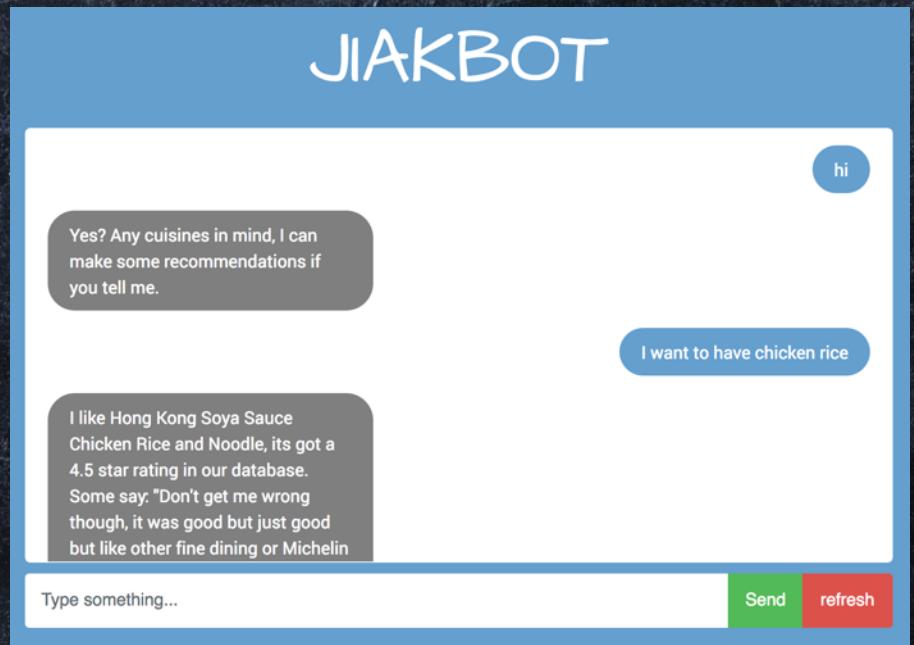
GENERATE RESPONSE

- Construct response based on user's input and document retrieved

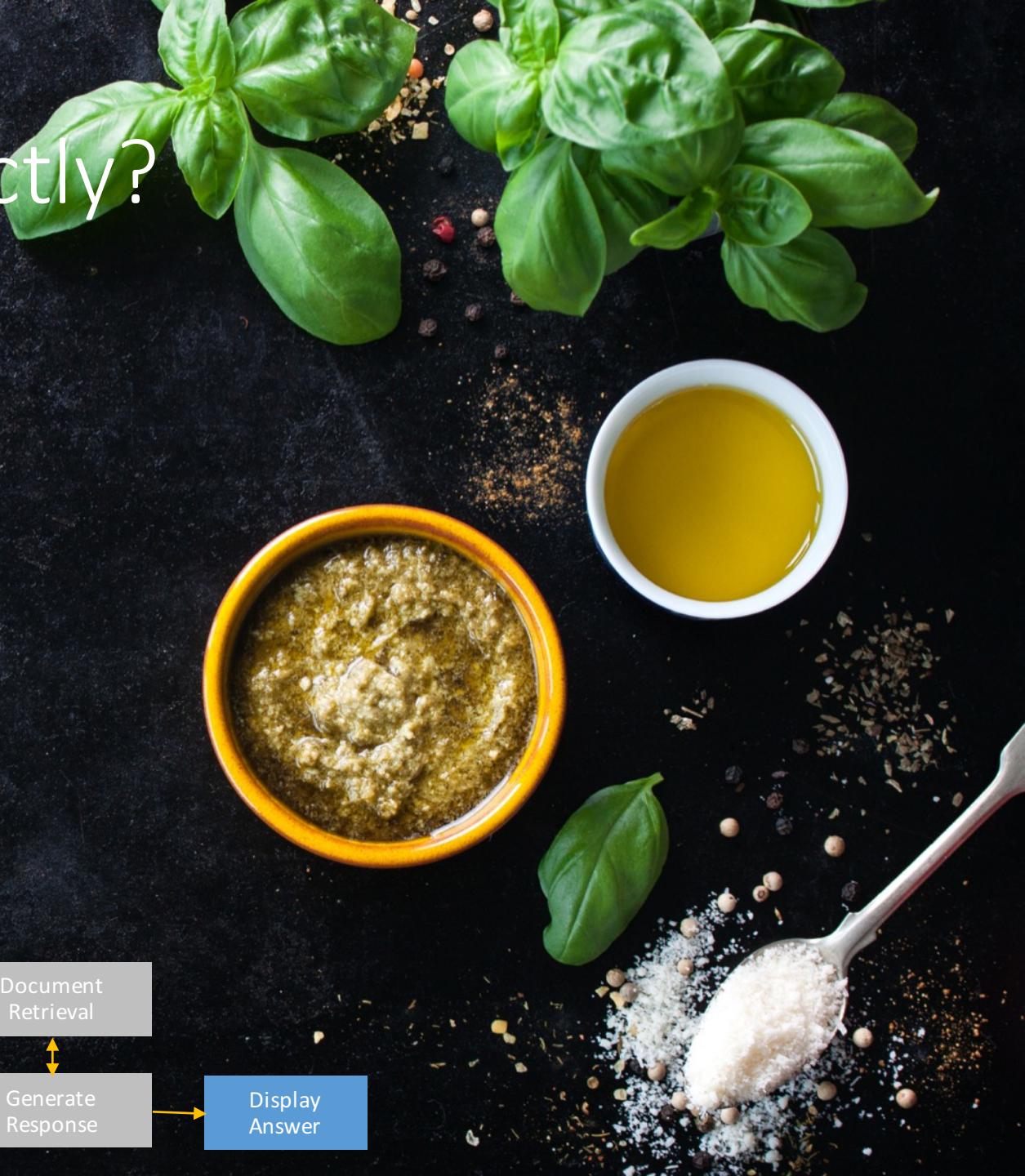
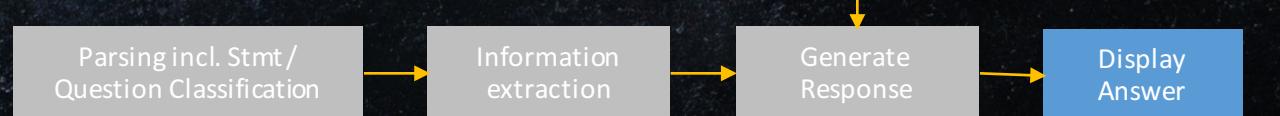


How did we do it ... exactly?

DISPLAY ANSWER



the jiakbot's NLP process



How did we do it ... exactly?

TO REMEMBER THE CONVERSATION ...

- State Machine keeps track the context of a conversation and stores current state based on user inputs and Jiakbot's responses

STATES



Demo



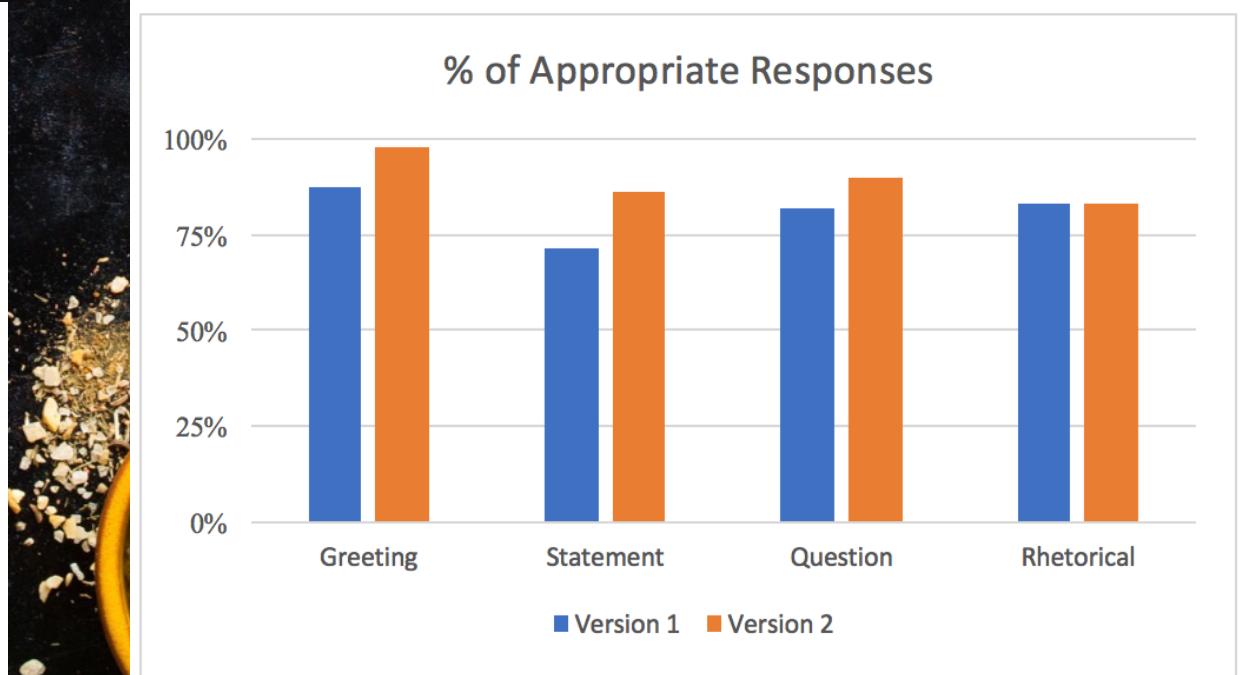
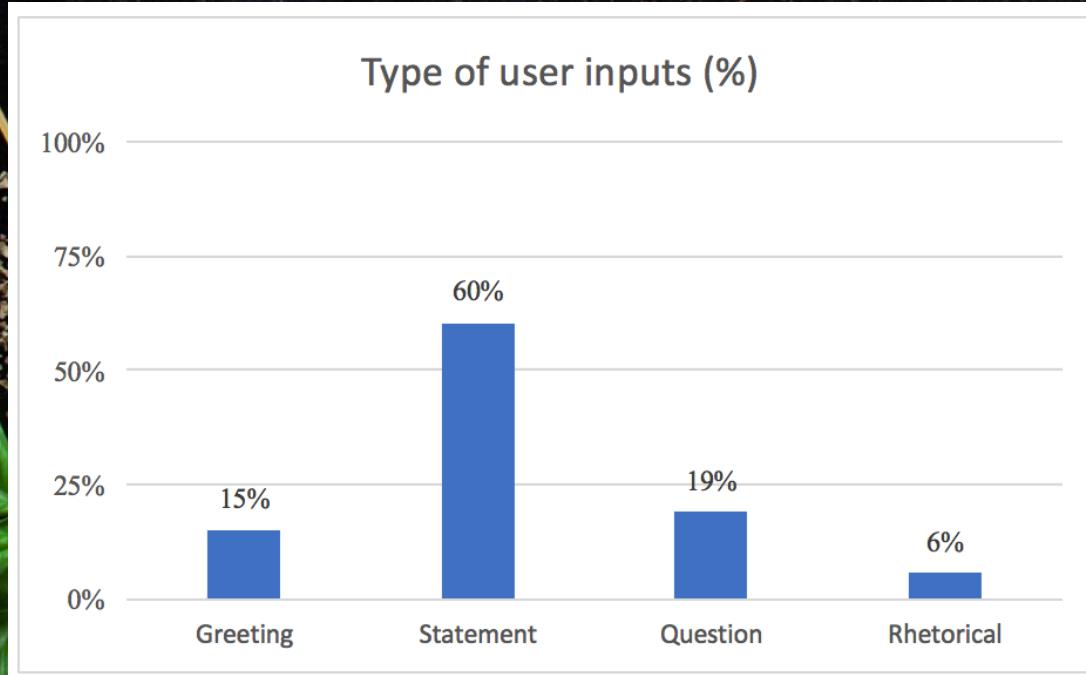
Evaluation

- Release JiakBot v1.0 for evaluation
- Prepare a list of JiakBot responses mixed with human responses for eliminating bias when evaluating
- Compile the results (Inappropriate/Appropriate)
- Identify weak areas of response (e.g. Greetings, Questions)
- Identify most asked questions and keywords
- Improve Jiakbot's vocabulary & logical design based on feedback



Evaluation

- Results based on independent evaluation by 6 raters with fixed set of statements for the bot to respond to.
- Improvement seen in all areas except rhetorical questions.



Challenges Faced

- Getting distinct topics using out-of-the-box topic modeling techniques such as LDA
- Need to explore further if complementing it with PoS tags or custom features extraction will work better
- Spelling errors, different spellings of different cuisines
- Generating an appropriate response based on the context
- Understanding user inputs in a Singapore Context. NLTK is not strong in recognising food and/or places in Singapore
- Lack of training data for question classifier
- Lack of time to conduct JiakBot use case trials and collect sufficient data for further refinement of the bot



Key Takeaways

- Real text mining problems typically involves putting several techniques together.
- Leverage on existing works and findings - For example, someone else built chatbots and done research on question classification for tweets
- Appropriate database is key - For example, food reviews for food recommendations and vocabulary for responding to users.
- Generic bag-of-words models are good as the first cut - For better results, some custom feature extraction is required
- Conversations are subjective and cannot follow traditional evaluation metrics of accuracy, recall and precision
- Improvement through feedback is an iterative process



Future Work

- Adding location features to responses
- Improve corpus and logical design
- Using conversational history and context to train the bot
- Expand geographical region coverage
- Monetize application through collaboration
- Adding localized personality (e.g. jokes, news)



Q + A