SMU
SINGAPORE MANAGEMENT UNIVERSITY

School of
Information Systems

Master of IT and Business (Analytics)

ISSS609 Text Analytics and Applications

Project Report: JiakBot

Submitted by: Group 2
Brijesh Gupta
Chan Wesley
Jiang Chen Yang Thomas
Kuar Kah Ling
Tham Jun Quan
Thng Ren Jing Chris

24 April 2017

INTRODUCTION

Overview

The project was inspired from a common problem working class adults in Singapore's Central Business District (CBD) faced - where and what to eat for lunch. Daily, the working crowd in CBD look forward to their meal times to recharge and perhaps to catch up with fellow colleagues. However, given the numerous food choices and a lack of time to research on where and what to eat, most people have difficulty deciding their meals. Some have resorted to generating random places to eat using Microsoft Excel.

This project aims to suggest places to eat through a chatbot, named “JiakBot”, in order to provide an interactive user experience and to enable the user to make a more informed choice. JiakBot is built using Natural Language Processing (NLP) techniques.

Motivation

Chatbots, through Natural Language Processing (NLP), is an attractive method of human-computer interaction due to the interactivity as compared to traditional mediums such as a Frequently Asked Questions (FAQ) section on a website. It has been used in a variety of domains such as environment and sustainable development [1], education [2] and other commercial purposes [3].

However, chatbots are not widely used in Singapore for food and restaurants recommendations, making it an obvious primary choice for us to develop one which is specific to Singapore. Furthermore, creating relevant and appropriate response leveraging on various NLP techniques is a tough but critical component to a successful chatbot. Hence, we decided to take on the challenge of developing one.

There are also several use cases or scenarios which we think will be immediately useful. They are:

- Tying up with business to provide exposure as well as a source of revenue (e.g. incentives/discounts)
- Letting users to have a quick and easy access to information about food places in their vicinity
- Letting users subscribe to this service to receive information about trending restaurants based on their preferences communicated to the chatbot
- Providing some entertainment (localized humour) when choosing where to eat
- Providing other information about food location by complementing current data with other data sources such as Google visitor information (e.g. popular time, live visits, duration, etc.)
- Generating revenue through advertisements within app

METHODOLOGY

Data Preparation

Yelp, a website which publishes crowd-sourced reviews about local business, is our primary source of data. We used both the Yelp API and web scraping to extract all businesses data, including reviews, for restaurants in City Hall, Raffles Place, Bras Basah and Dhoby Ghaut. The locations were chosen primarily because they represent the Central Business District of Singapore.

The number of records for each entity is provided below:

Businesses	Reviews	Statements	Cuisines	Foods	Places
2,590	17,235	171,906	1,740	1,088	985

Businesses - Contains the names of the businesses together with their categories (e.g. the type of cuisines they serve)

Reviews - Contains the reviews of the businesses in free-text.

Statements - Contains the statements tokenized from the reviews' text. The statements are sometimes returned as part of the response to the user to provide a more "human" touch.

Cuisines - Types of cuisines, for example, Japanese, Chinese, Indian.

Foods - Types of food, for example, Burgers, Coffee, Noodles

Places - Types of places, for example, Cafes, Bars, Bistros

SQLite3 was used to build the knowledge base of the chatbot using reviews and businesses data extracted (i.e. database).

JiakBot’s Question Answering Process

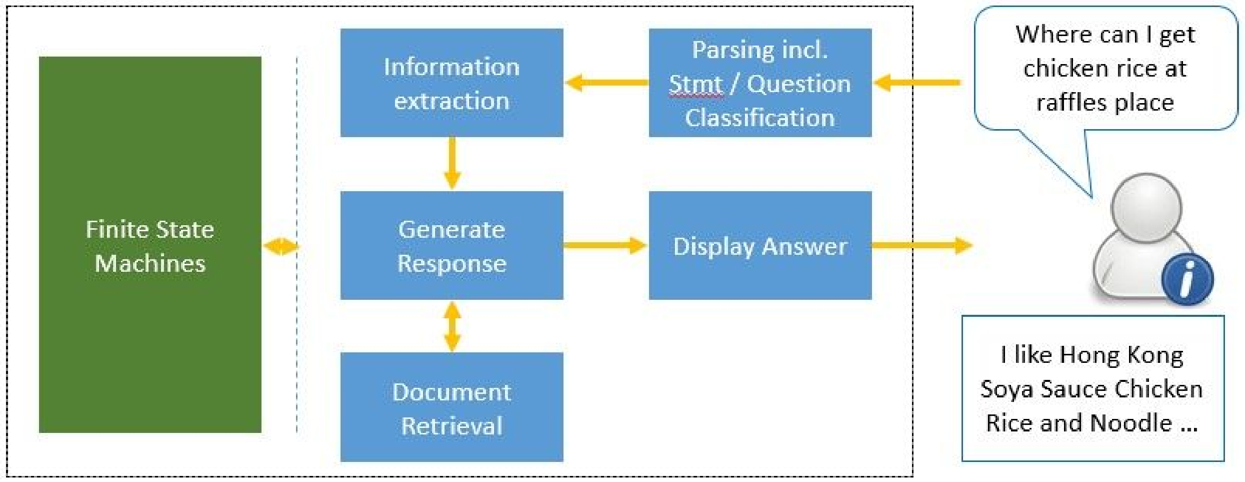

Finite State Machines
Information extraction
Parsing incl. Stmt / Question Classification
Where can I get chicken rice at raffles place
Generate Response
Display Answer
Document Retrieval
I like Hong Kong Soya Sauce Chicken Rice and Noodle ...

Figure 1: Overview of JiakBot’s Tasks

When a user enters an input, several tasks are performed on the input. It will first be parsed and classified as either a statement, question or rhetorical. Then, Information Extraction is performed to detect food, location and cuisine (i.e. states) so that their states can be tracked by the Finite State Machine and that the chatbot knows what to search for in its database.

Using the information extracted, Document Retrieval from the database is executed to get the list of restaurants most relevant to the user's input. Additional information such as a statement from a review to the user is also provided to help them make their choice. Combining the results from Information Extraction, Document Retrieval and current state, a response is generated and displayed to the user.

The Finite State Machine keeps track of the context of a conversation and stores current state based on user inputs and JiakBot’s responses. Context is a way for JiakBot to keep track of what was said previously and being able to take this into account when selecting its next response. JiakBot always starts at State #1 (Understood Nothing). When user enters an input, the state may either change to one of the other five states or remain at #1. (**Figure 2**)

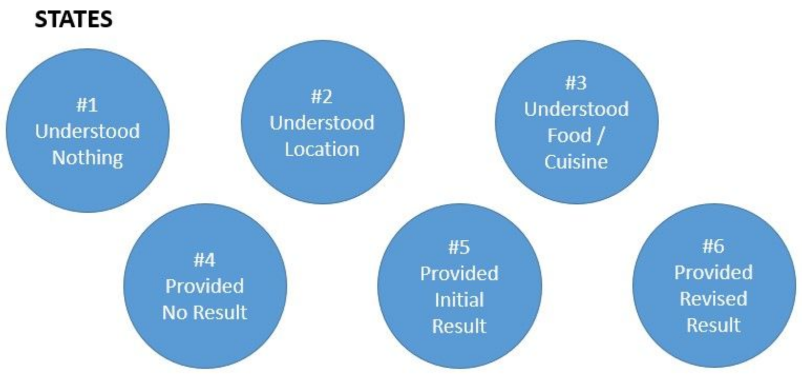

STATES
#1 Understood Nothing
#2 Understood Location
#3 Understood Food / Cuisine
#4 Provided No Result
#5 Provided Initial Result
#6 Provided Revised Result

Figure 2: The 6 States of Jiakbot

For example, when the user enters “What can I eat in raffles place?”, location is detected and the current state is updated from #1 to #2 (Understood Location). If “Where can I get chicken rice” is entered instead, food is detected and the current state is updated to #3 (Understood Food/Cuisine). If both location and food are detected, JiakBot will update the state to #2.

Depending on the result from Document Retrieval, a relevant result may be found. If no relevant result is found, the state is updated to #4 (Provided No Result) and JiakBot will inform user that no result was found and request for an alternative search. If a result is found, the state is updated to #5 (Provided Initial Result) and feedback is sought on the recommendation. If the user is pleased with the recommendation i.e. Yes, the state reverts to #1. If the recommendation was not useful i.e. No, the state is updated to #6 (Provided Revised Result) and an alternative recommendation is provided. Subsequently, the state reverts to #1.

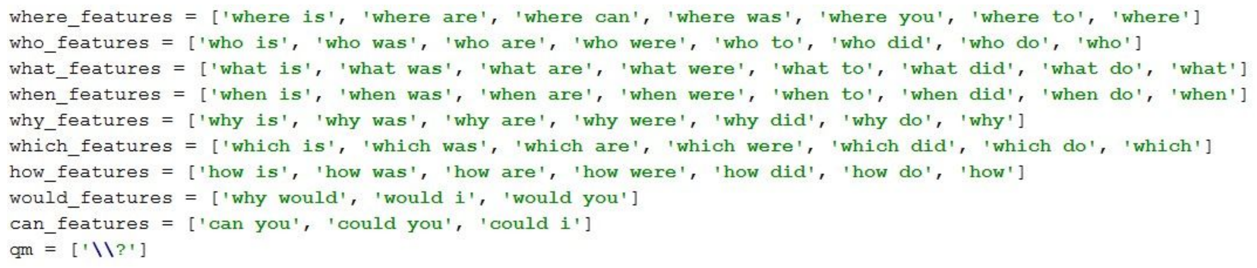
Natural Language Processing Tasks

Parsing

Each statement entered by user is tokenized and part-of-speech tagged using NLTK. Stopwords and punctuations are removed to produce a cleansed text. The outcome is a parsed Python dictionary of tokens, cleansed text, verbs, adverbs, nouns, adjectives and pronouns.

Statement / Question Classification

In order to detect user inputs which contain questions, we modified an approach used by Li B.C. and colleagues [4] where they reported reasonable performance on identifying Twitter texts. The approach involves using feature such as question marks, 5W1H and adding words such as “which”, “why”, “would” and “can” (**Figure 3**). 12,000 labelled inputs were created to train the classifier for the learning-based method. The data points consist of statements labelled from Yelp’s reviews and also manually constructed statements and questions. The model was trained using Naive Bayes and Random Forest, with Random Forest performing better than Naive Bayes.

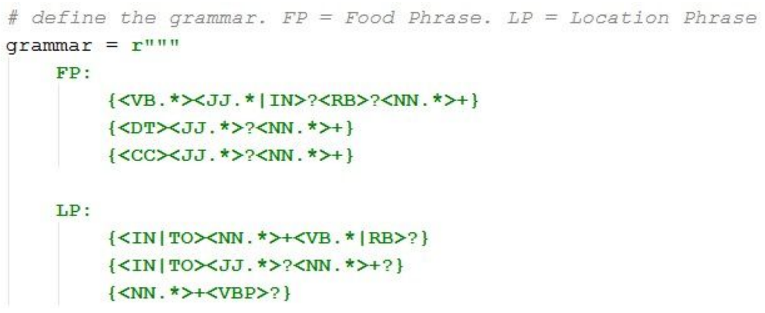


```
where_features = ['where is', 'where are', 'where can', 'where was', 'where you', 'where to', 'where']
who_features = ['who is', 'who was', 'who are', 'who were', 'who to', 'who did', 'who do', 'who']
what_features = ['what is', 'what was', 'what are', 'what were', 'what to', 'what did', 'what do', 'what']
when_features = ['when is', 'when was', 'when are', 'when were', 'when to', 'when did', 'when do', 'when']
why_features = ['why is', 'why was', 'why are', 'why were', 'why did', 'why do', 'why']
which_features = ['which is', 'which was', 'which are', 'which were', 'which did', 'which do', 'which']
how_features = ['how is', 'how was', 'how are', 'how were', 'how did', 'how do', 'how']
would_features = ['why would', 'would i', 'would you']
can_features = ['can you', 'could you', 'could i']
qm = ['\\?']
```

Figure 3: Identified Features for Statement/Question Classification

Information Extraction

Using the parsed dictionary, the State Machine extracts food, cuisine, and location (i.e. states) by identifying specific grammar patterns in common question statement (**Figure 4**), and subsequently applying a combination of rules based and dictionary to identify the named entities.

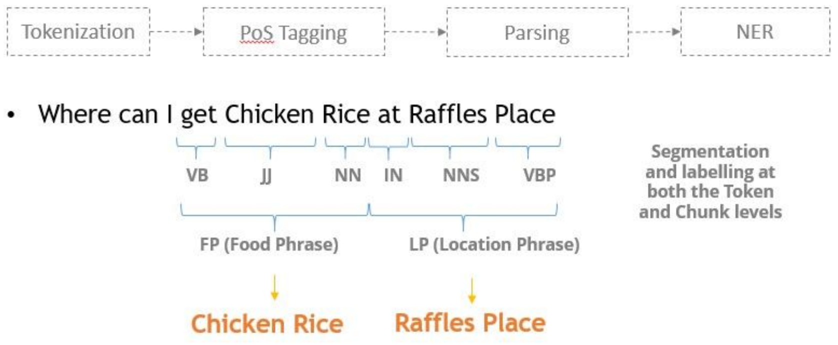


```
# define the grammar. FP = Food Phrase. LP = Location Phrase
grammar = r"""
    FP:
        {<VB.*><JJ.*|IN>?<RB>?<NN.*>+}
        {<DT><JJ.*>?<NN.*>+}
        {<CC><JJ.*>?<NN.*>+}

    LP:
        {<IN|TO><NN.*>+<VB.*|RB>?}
        {<IN|TO><JJ.*>?<NN.*>+?}
        {<NN.*>+<VBP>?}
```

Figure 4: Food Phrase & Location Phrase Grammars

An example showing how the parsed dictionary is used to extract the relevant named entities is illustrated in **Figure 5**.



Tokenization
PoS Tagging
Parsing
NER
• Where can I get Chicken Rice at Raffles Place
VB
JJ
NN
IN
NNS
VBP
Segmentation and labelling at both the Token and Chunk levels
FP (Food Phrase)
LP (Location Phrase)
Chicken Rice
Raffles Place

Figure 5: Illustration of Information Extraction by State Machine

Document Retrieval

In order to perform this step, we had to utilize SQL queries based on the user's input to retrieve the required set(s) of information from the database. There are 2 steps to the document retrieval process. The first step is to retrieve the most relevant food or cuisine based on the user's input. For example, if the user wanted "Coffee", Jiakbot will return a business related to "Coffee" such as "Coffee Bean and Tea Leaves".

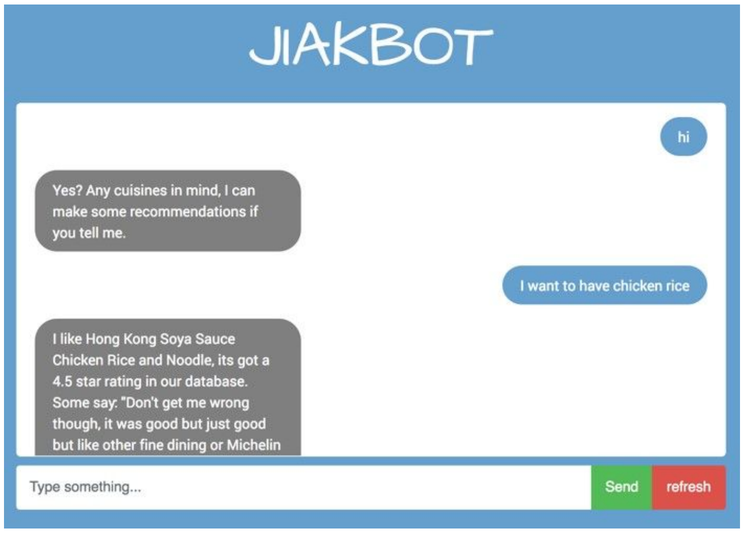
The second step, would be to retrieve a relevant statement telling the user more about the business and include it as part of the response. This is retrieved from statements tokenized from the reviews. In order to decide which statement to return Cosine Similarity was used to compare statements within randomly selected 10000 statements. 10000 was chosen due to performance considerations of retrieving and comparing too many statements.

Through our empirical testing, it was found that, if the Cosine Similarity is less than 0.6, the returned statement does not make adequate sense for it to be used to generate a response. Hence, we set the Cosine Similarity limit to 0.6, whereby if the result is less than 0.6, a generic response is returned to the user. Based on the tests conducted by our team, returning the next most relevant document in terms of Cosine Similarity did not always retrieve something relevant.

Response Generation

The retrieval-based model [5], which makes use of a repository of predefined responses and heuristics, is used to construct a response. In JiakBot's case, the heuristic is a set of rules matching the state, type of user input (statement, question, rhetoric) and whether a document is successfully retrieved from the database. Each statement input is further categorised based on its nature - 'greeting', 'intent', 'about user', 'about bot', 'yes', 'no'. Based on each category in which each statement is classified as, a set of predefined responses have been created and the response output is randomly generated from the set to give a logical response.

As an illustration, if a valid location is detected (i.e. Dhoby Ghaut, City Hall, Raffles Place, Bras Basah) and a relevant food review is found, a default format of response with a food review (Response 1) from the database is generated. However, if no relevant food review is found, another default format of response (Response 2) is produced. Similarly, if food/cuisine is detected and relevant food review is found, Response 1 is generated (**Figure 6**). If food/cuisine is detected but no relevant food review is found, Response 2 is displayed.



hi

Yes? Any cuisines in mind, I can make some recommendations if you tell me.

I want to have chicken rice

I like Hong Kong Soya Sauce Chicken Rice and Noodle, its got a 4.5 star rating in our database. Some say: "Don't get me wrong though, it was good but just good but like other fine dining or Michelin

Figure 6: Example of Response Output

Putting Things Together

The components and the bot was then hosted locally using the Django Framework to build the front-end. Given that there are multiple components developed by multiple team members. Github was used as repository for source control. The repository is available at: www.github.com/junquant/taproject

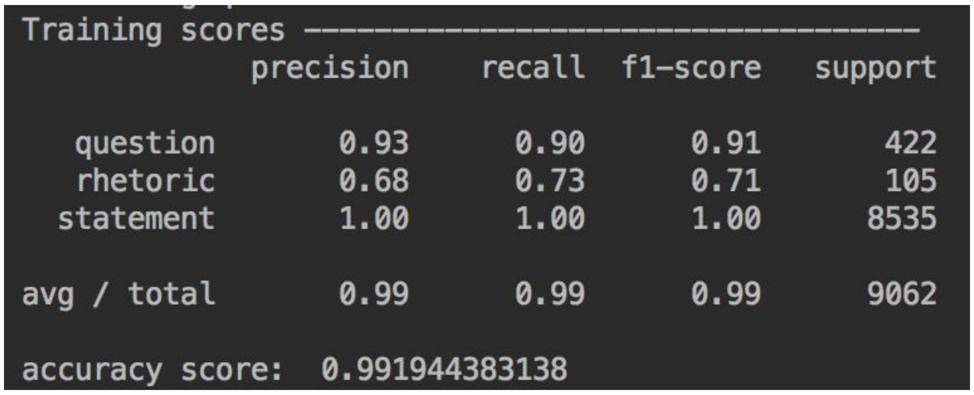
RESULTS AND ANALYSES

Statement/Question Classifier

The data set used to train the classifier consists of 12083 statements, questions and rhetorical questions. Hold-out validation was used to validate the model and the data into training and test set. The size of each data set is as follows.

Size of data set: 12083
Size of training data set: 9062
Size of test data set: 3021

The training and test results are presented below, the scores are rounded to 2 decimal places.

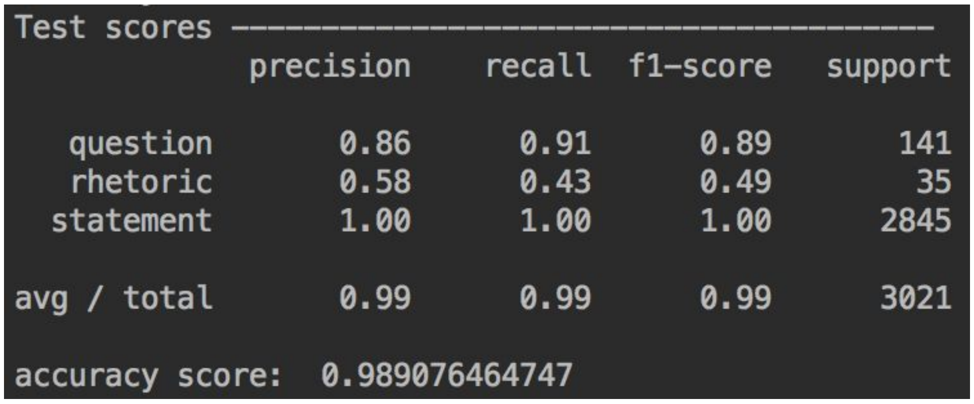


```
Training scores ------------------------------------
              precision    recall  f1-score   support

    question       0.93      0.90      0.91       422
    rhetoric       0.68      0.73      0.71       105
   statement       1.00      1.00      1.00      8535

avg / total        0.99      0.99      0.99      9062

accuracy score:  0.991944383138
```



```
Test scores ---------------------------------------
              precision    recall  f1-score   support

    question       0.86      0.91      0.89       141
    rhetoric       0.58      0.43      0.49        35
   statement       1.00      1.00      1.00      2845

avg / total        0.99      0.99      0.99      3021

accuracy score:  0.989076464747
```

We observe that the classifier performs well in classifying statements and not so well in determining whether the user’s input is a rhetorical question. The errors in classifying whether an input is a question are typically caused by the absence of the “?” feature and the way the question was phrased. Some examples are shown below.

Question: Recommend where to go for nasi biryani - **Classified as** Statement
Question: I have been to that place before please recommend another one - **Classified as** Statement

The low precision and recall scores for rhetorical questions were caused by the rhetorical questions being classified as questions. There were no rhetorical questions being classified as statements. Some of the errors are presented below.

Rhetorical Question: isnt it lonely out there? **Classified as** Question
Rhetorical Question: do you ever wonder about the universe? **Classified as** Question
Rhetorical Question: dont you think so? **Classified as** Question

The errors were primarily caused by the presence of the “?” feature.

Chatbot Performance

Chatbots are difficult to evaluate due to a couple of reasons:

1. There is no specific goal to guide the interaction. Hence, conventional evaluation metrics such as task completion rate are not appropriate.
2. Every conversation is subjective as users respond differently to statements/questions.

Thus, we adopted the crowd-sourcing method [6] to evaluate JiakBot and our focus is on the appropriateness of the response provided. To do so, we created a compilation of human-generated responses and JiakBot responses to a fixed set of statements to create 12 sets of conversations **(Refer to Appendix JiakBot Eval 1 and 2 for more details).**

Due to time constraints, the team only managed to engage six independent evaluators to rate how appropriate they felt the responses were with respect to the inputs. The evaluators are not aware whether the response is from JiakBot or human-generated, so as not to influence their rating. For each response, the evaluator chooses between two labels: “Inappropriate” and “Appropriate”.

JiakBot was fine tuned, based on the results of the first round of evaluation, and JiakBot version 2.0 was launched. Enhancements were made to JiakBot’s vocabulary and logical design by identifying frequently asked questions and keywords. A repeat evaluation was conducted on and significant improvement was noted across all input types, except Rhetorical. (**Figure 7**).

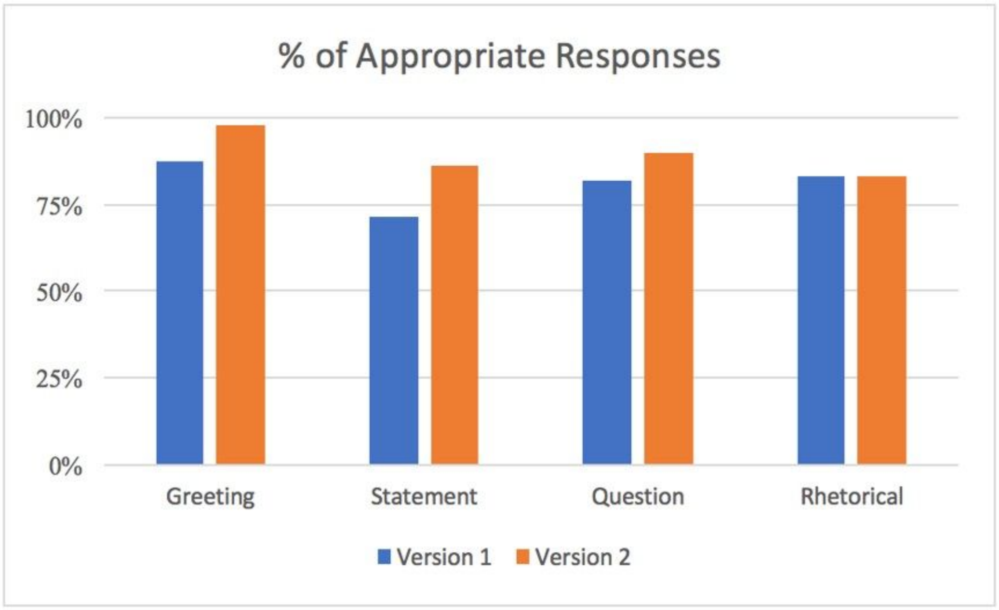


Figure 7: Results of JiakBot version 1 & 2 Evaluation

Additionally we do acknowledge that this would be an iterative process which would require the bot the go through several tests, collection of feedback and improvement. The scope may expand or be more focused as we scale the project accordingly and hence, for JiakBot version 2.0 we believe the current evaluation methods would remain appropriate in the evaluation and improvement of the overall chatbot.

In future, we hope to leverage on other existing evaluation techniques [7] such as utilizing Grice's conversational maxims to measure the linguistic perspective of the conversational interactions as well as apply AI-oriented measures such as the Turing Test.

Topic Modelling

We had initially wanted to do topic modelling to come up with a set of coherent words to describe the corpus in general and each review in specific. This very specific word sets will then be further used in generating refined responses by the chatbot. For example, if using the topic model, we would be able to detect the topics from the user input and based on the topics, extract a relevant response from statements with that topic. However, while conducting the analysis, we realised that while few topics seemed consistent, many seemed incongruous and difficult to interpret.

From our analysis, the reasons for our results were:

1. The vocabularies in the corpus were very distinct due to large usage of words like cuisine name (e.g. ramen, kaya toast, etc.) and localised language (e.g. Singlish) causing a very flat distribution of words. This is compounded by the fact that there are variations in these words.

2. Spelling errors was another major challenge, causing flat distribution of the words, as every incorrect spelling was treated as a new word, thus distorting relative distributions.
3. With the corpus containing only reviews relating to food, it is hard to discover distinct topics. The nuances of the different vocabularies used to describe different foods and cuisines could not be teased out using the standard LDA algorithm.
4. Even though we filtered the stopwords using NLTK corpus, there are still a significant number of words having little or no information i.e. stopwords. Therefore, a more comprehensive stopwords lexicon was needed and was subsequently applied. This brought some improvement over the initial topics but still not enough to consistently summarize each review.

DISCUSSION AND CONCLUSION

Building a chatbot is an interesting and challenging task which requires putting several NLP techniques together. Numerous research papers and chatbot source codes have been shared online and one should leverage on existing works and findings. However, as each language and country has its unique nuances, developing a localised chatbot requires customisation.

Research on topic extraction showed that “Noun” words and phrases are more indicative of the underlying topic than other parts-of-speech tagging methods. It would be interesting to see if applying topic modelling techniques over this curtailed corpus improves our understanding of the resulting topics. Additionally, even though we used only the standard LDA due to the tight project timeline, there are a range of topic modelling techniques that could be tested and evaluated to find the most appropriate type.

Furthermore, natural language processing and more specifically, chatbot improvement is an iterative process as the bot and developers progressively learn more about user’s behavior and expectations.

FUTURE WORKS

To personalize JiakBot further, a personality could be incorporated to JiakBot such that, it is able to chat and joke in localised context, making it truly “Singaporean”. Should JiakBot become successful, we could monetize JiakBot through collaborations with other businesses such as restaurant promotions, malls and Google Ads. Through the implementation of such features, we would be able to create “stick-iness” amongst our users to constantly consult our bot for the best food places and deals. By incentivising the use of JiakBot, all parties will benefit; restaurants gain exposure, customers are well-informed, JiakBot is able to improve with more data and money earned, as a result, can be channeled back for further development of the bot.

On NLP front, using conversational history and context, JiakBot could be trained to provide more relevant responses and improve on its logical design and develop a more robust corpus. Location features could be added so that the address of the recommended restaurant can be provided to users. We could also expand JiakBot’s geographical coverage to beyond CBD.

REFERENCES

[1] AluxBot - A Chatbot that Encourages the Care for the Environment. (2016). International Journal of Computer Science Issues, 13(6), 120-123.

[2] Yi Fei Wang, & Stephen Petrina. (2013). Using Learning Analytics to Understand the Design of an Intelligent Language Tutor – Chatbot Lucy. International Journal of Advanced Computer Science and Applications, 4(11), 124-131.

[3] Chatbots Raise Questions About the Future of Customer Service. (2016, April 27). PR Newswire, p. PR Newswire, Apr 27, 2016.

[4] *Question Identification On Twitter*. 1st ed. Baichuan Li, Xiance Si, Michael R. Lyu, Irwin King, and Edward Y. Chang, 2017. Print.

[5] "Ultimate Guide To Leveraging NLP & Machine Learning For Your Chatbot". *Chatbot's Life*. N.p., 2017. Web. 21 Apr. 2017.

[6] Zhou Yu, Ziyu Xu, Alan W. Black, Alexander I.. (2016). Chatbot Evaluation and Database Expansion via Crowdsourcing.

[7] Chayan Chakrabarti & George F. Luger. (2013) A Framework for Simulating and Evaluating Artificial Chatter Bot Conversations.