Name : Haritha Pulletikurti
Instructor Name : Professor Joel Sokol
ISYE 6501 Project
11/17/2020

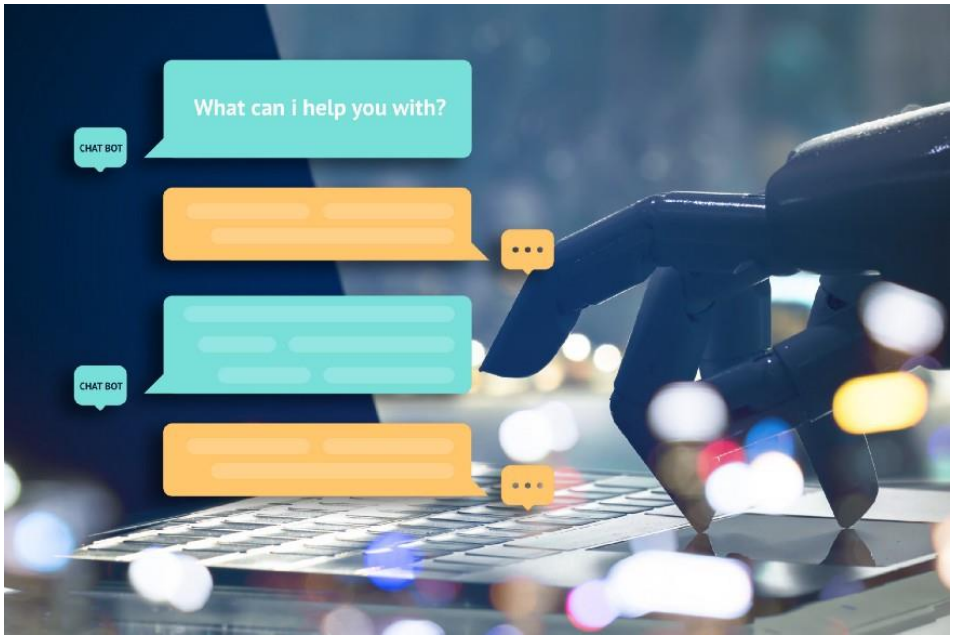
Title: AI/Watson Chatbot – Levi for Roman's Pizza

1. Introduction :

Business challenge for Roman's Pizza was that the customer support was struggling to keep pace with the constant stream of questions being posted on the social media platforms. To handle this situation, the Roman's Pizza company introduced a lateral conversational interface, AI/Watson Chatbot – Levi, that came across as natural language as opposed to a button-driven approach to handle common requests such as those focused on store hours or weekly specials. Levi is designed to be very human-like i.e. it is colloquial in terms of language. Humorous in terms of personality. Intuitive, meaning it has the ability to understand the tone and sentiment of the customer and respond in a natural manner.

1.1 Results:

i. More than 90% incoming queries are answered automatically without human intervention.
ii. Responds immediately to customers, while previous replies took 6 – 8 hours on average. And if there is an issue outside the business hours that Levi cannot handle, it will log the complaint and notify the team to deal with it first thing in the morning. It addresses customer's concern up front and gives an expectation on when things can be resolved.
iii. Frees up staff to focus on more important tasks and establish a healthy work- life balance.



2. Design of Levi:

To Bring Levi to life, there is a custom-built conversational interface, powered by IBM technology under an IBM Embedded Solutions Agreement(ESA). It has four main components –

i. **IBM Watson Assistant** serves as the heart of the solution.
ii. **IBM Watson Tone Analyzer** monitors chat instances identifying when the conversations need to be escalated to the human staff. It tracks the tone and when it hits a threshold – swear words, aggression – It will send an SMS message to the support team that they have someone who needs help.
iii. **IBM API Connect** to integrate with the Facebook Messenger app.
iv. **IBM Cloud** where the entire Levi Infrastructure is hosted.

For more details please see https://www.ibm.com/case-studies/romans-pizza-api-watson-cloud

3. Data Collection and Preparation

3.1 Data: The data we require is in the form of the following:

i. **Entities:** These are objects or things the Chat Bot is taking action on and needs to recognize. – For example, Toppings like onion, pineapple, tomato, mushrooms etc.
ii. **Intent:** Things the user will say: " I am hungry", "I want to order pizza", "Pizza for pick up", etc.
iii. **Responses to intent:** scenario responses you need to come up with like "What type of pizza do you want"?
iv. **Other Data:** Store timings, Employee Availability, Number of Employees, Day of week, Number of pending orders, Time for making each pizza, Available Menu, Number of Customer orders served per day, Cost of each menu item etc.

3.2 Data Collection:

Data can be collected from the previous customer call recordings and can be scripted.

3.3 How often it might need to be refreshed and the model re-run:

During the initial days of releasing the Levi Bot, the daily data can be collected and fed into for training the models every week. After the Levi Bot, starts performing well, the data can be refed once every month and the model can be retrained.

3.4 Data Preparation Phase:

i. **Data Cleanup**: We need to clean the raw textual data before feeding into the model.
   a) Remove the punctuation and special characters, repetitive information, duplicate information.
   b) Tokenize the sentences into words.
   c) Lowercase all the words and use lemmatization

" Lemmatization in linguistics is the process of grouping the different inflected forms of a word so they can be analyzed as a single item."
Example: lemmetizer.lemmeatize("Pineapples") ➔ pineapple
The above function is from Python.

d) After cleaning the data and got lists of words of sentences, convert the words into indexes so they can be used as input. In Python there is a "Tokenizer Class of Keras". Then we can pad them to make equal length so can be used in the model.
e) Do the same steps for Output words as well.

ii. **NLP:** Also, we can use Neural Networks and Deep Learning Techniques such as Natural Language Processing (NLP) for script cleaning, to extract important verbs and also there are tools for Parts of Speech.

iii. **Binarize the textual data:** We can binarize the chosen words . We can use each selected word as a feature.

#	topping	hungry	pineapple	pizza	order	card	information	buy
1	1	0	1	1	0	0	0	0
2	0	0	0	1	1	0	0	0

Binarized data shows
" topping , pineapple , pizza " in record 1,
" pizza, order" for record 2 etc.

4. Training and Test Sets:

Once the Data is ready for the model, split the dataset into training and test set.
Usually we can use 80% for training and 20% for testing.

5. Building the Watson Assistant

i. K means clustering using key words:

We can use K means clustering to cluster the different data based on different intents and topics. We can use "SKlearn.cluster in KMeans library"

Example code:

```
TOTAL_CLUSTERS = 50
# Number of clusters
kmeans = KMeans(n_clusters=TOTAL_CLUSTERS)
# Fitting the input data
kmeans = kmeans.fit(row_bools)
# Getting the cluster labels
labels = kmeans.predict(row_bools)
# add cluster back to data frame
row_bools['cluster'] = labels
row_bools['cluster'].value_counts().head()
```

Sample output:

```
10    4400
15    3000
22    2653
2     2518
19    2492
Name: cluster, dtype: int64
```

ii. **N- grams per cluster:** This is the final phase of the process before the manual analysis and extraction process begins. We can use the NLTK library to pull all the n-grams we need.

N-grams are sequences of varying sizes found in the text.

Example:" The topping is onion" has the following 3 n-grams of 2 dimensions (bigrams)

- The topping
- topping is
- is onion

The reason we are using this approach is to find how many times certain sequential word patterns are used in different clustered transactions/dialogs. This is to translate similar dialogs into chatbot scenarios. We can run n-grams of dimension 2,3,4,6. The larger, the more revelatory to find complex and repeated patterns.

iii. **Chatbot Levi :** Once the customer has begun the dialog, based on the intent, cluster is selected and from above list, a script can be crafted based on the scenario. More data that is fed, the more the model is trained and the more relevant answer we get.

6 Building the IBM Watson Tone Analyzer:

i. Data Collection and Preparation :

The data collection, preparation and the model building is all common as the Watson Assistant, but we need to extend the model to make sure the customers mood can be captured by the Analyzer, so it can escalate the query to for human intervention.

ii. Attaching Weights to possible mood intent words :

We can attach weights to the trained input data giving lower weights to "Happy words" and higher weights to "seemingly angry or frustrated words and average weight for "if the mood is on the okay side".

iii. Using K means clustering :

The data is clustered using K Means based on topics – "Toppings", "Billing" , "Timings" etc. Same as in Building Watson Assistant.

iv. Using Logistic Regression to predict the Tone of the Customer:

a. Then based on the weights we can use logistic regression on each cluster to determine the probability of the customer typed sentence to be on the happier side or the possibly angrier side or okay side.
b. If the prediction is Angry phrase – then give a soft reply and immediately send SMS to the customer service for immediate human intervention.

7 Estimating the timing for "delivery of pizza or human intervention" or any payment queries:

i. Data:

Data might be Store timings, Employee Availability, Current Wait time, Pizza making time, Cost etc.

ii. Using Linear Regression to estimate the timing for the timing queries :

The Tone Analyzer and Watson Assistant both can use the Linear Regression to estimate the time to deliver the pizza to the customer in case the query is about the time.

If the query is asked after hours of store operation, the Assistant can estimate the time that the requested task can be accomplished or the earliest time the employee can answer the query.

iii. Using Linear Regression for Payment Queries:

The Tone Analyzer and Watson Assistant both can use the Linear Regression to estimate the cost predictions of the pizza order to the customer in case the query is about the payment.

Summary

1. Building the Watson Assistant for Levi Bot:

Given	Textual data collected from earlier call data of customers Prepare the textual data words from sentences, Store timings, Employee Availability, Number of Employees, Day of week, Number of pending orders, Time for making each pizza, Available Menu, Number of Customer orders served per day, Cost of each item etc.
Use	Prepare the data using NLP , KMeans clustering to cluster data based on intents and topics Logistic regression on each cluster
To	Generate a reasonable answer to the query

2. Building the Watson Tone Analyzer for Levi Bot:

Given	Textual data collected from earlier call data of customers - build the words from sentences, assign weights for the words which indicate the moods – Happier words have lower weights, Sad or Angry tone words can have higher weights

Use	Prepare the data using NLP , KMeans clustering to prepare data based on intents and topics Logistic regression on each cluster
To	To estimate a possible tone of the customer based on a threshold of weight. If the prediction is angrier or sad then send SMS for immediate human intervention. Generate a reasonable answer to the query .

3. Estimating the time for timing/ payment questions:

Given	Textual data collected from earlier call data of customers - build the words from sentences, Store timings, Employee Availability, Number of Employees, Day of week, Number of pending orders, Time for making each pizza, Available Menu, Number of Customer orders served per day, Cost of each menu item etc.
Use	Prepare the data using NLP (Natural Language Processing) KMeans clustering to prepare data based on intents and topics, Linear regression on each cluster
To	Estimate the possible time of delivery/Payment for the queried service and provide a reasonable answer to the customer.

Conclusion

To build the Levi Chat Bot,

- We can collect the data from the prior customer call recordings to the customer support (usually the calls are all recorded in most cases) and also can be fetched from the generic store information database. Once the Levi Chat bot is live, we can collect the monthly data to refed and retrain the bot to make it more intelligent.
- We can use, Unsupervised models like Natural Language Processing Algorithms for textual data processing and KMeans Algorithm for clustering the data based on different topics or intents.
- We can use Supervised models like Linear Regression and Logistic Regression to build the Watson Assistant and Tone Analyzer to predict or estimate the input and output responses and to take the correct action based on the scenario.
- We can use tools like api.ai to build and integrate them to social media sites where the chatbots can come into use
- We can use Cloud to host the whole infrastructure of the application.

In my research for this Project, I came across lot of methods of building the chatbot.
I only presented one reasonable way of approaching this problem using the knowledge that I gained from the class during this semester.

Works Cited

https://amunategui.github.io/Extracting-Chatbot-Scenarios-From-Customer-Service-Scripts/index.html.

https://www.ibm.com/downloads/cas/YLNZM1AR

https://medium.com/@jinrajjain/automatic-utterances-clustering-for-chatbots-5d5f23afc082

https://towardsdatascience.com/a-brief-introduction-to-intent-classification-96fda6b1f557

https://www.guru99.com/supervised-vs-unsupervised-learning.html

https://www.ibm.com/case-studies/romans-pizza-api-watson-cloud