Next Word Prediction-Language Model

Abstract

Language modeling involves predicting the next word in a sequence given the sequence of words already present. A language model is a key element in many natural language processing models such resolving customers inquiries through chat or answering the questions through emails.

Background:

In customer Service business especially in messaging or Chats or email supports, customer representative often struggle to response fast if they have limited knowledge of business area wherein the inquiry is about and need respond fast for better service and improved customer satisfaction.

Customer often choose chat or email option to contact customer representative because they can perform multitasking and expecting resolution in same conversation so that they do not need to explain again, on the other hand customer representative handle multiple chats so sometimes they need to revisit the previous text to understand the customers intents of messaging. The model that would always reads previous text and understand the context and predict the next words as agent start responding, this way they do not need to worry about appropriate language plus context of the chat.

With this model, their average response time which is used to calculate their performance will be improved and they can handle multiple messages together.

Business Problem:

Business Stakeholder wanted to build model that would learn from previously provided chat or email resolution history and suggest the next word when representative start providing resolution to the customers inquiry.

- Build the model that would predict the next word based on previous context.

Data Explanation:

Relational Strategies in Customer Service (RSiCS) Dataset
Human-computer data from three live customer service Intelligent Virtual Agents (IVAs) in the domains of travel and telecommunications were collected, and annotators marked all text that was deemed unnecessary to the determination of user intention.

Data was collected from four sources. The conversation logs of three commercial customer service IVAs and the Airline forums on TripAdvisor.com during August 2016.
Dataset numbering used in files:

- TripAdvisor.com airline forum
- Train travel IVA
- Airline travel IVA
- Telecommunications support IVA

Here to train the model mainly data is used from airline travel IVA because all four datasets are huge and would need better hardware's to train the NN model

Fields:

- Dataset ID: Dataset that the request originated from.
- Group ID: The group of 4 annotators that the selections originated from.
- Request ID: Unique ID of a request to allow joining between different files.
- Threshold: The threshold to merge selections by.
- MergedSelections: If at least annotators marked a character as unnecessary then it will be contained within the selected portion denoted by [and].
- Unselected: All text from MergedSelections not contained by [and].
- Selected: All text from MergedSelections contained by [and].
- Greeting: If a greeting of some kind (Hi, How are you) is present in Selected
- Backstory: If self-exposure language is present in Selected. The user is telling the audience about themselves, their situation, what led them to contact the agent or ask their question.
- Justification: If justification language is present in Selected. The user is giving facts to build credibility that their request or statement is true. Also, can be why they need resolution or a consequence if something is not resolved.
- Rant: If ranting is present in Selected. Excessive complaining or negative narrative.
- Gratitude: If some expression of gratitude to the audience for past or future help is present in Selected.
- Other: If some or all of the highlighted section does not contain any relational language in Selected. Could be additional facts the user gave but annotators determined was unnecessary to determine their intention, or a general question such as Can you help?.
- Express Emotion: If any emotional language not covered by Rant is present in Selected

Methods:

To build this model, CRISP-DM methodology will be used and followed each stage from it to make sure right product is built with minimal issues.
The model would be built using Recurrent neural networks (RNN) using TensorFlow and Keras.

To build this model, RNN's LSTM method is used, which has a hidden state and a memory cell with three gates that are forgotten, read, and input gate. The model will be a sequential model. We are going to create embedding layer and specify the input dimensions and output dimensions, then we will add an LSTM layer to our architecture.

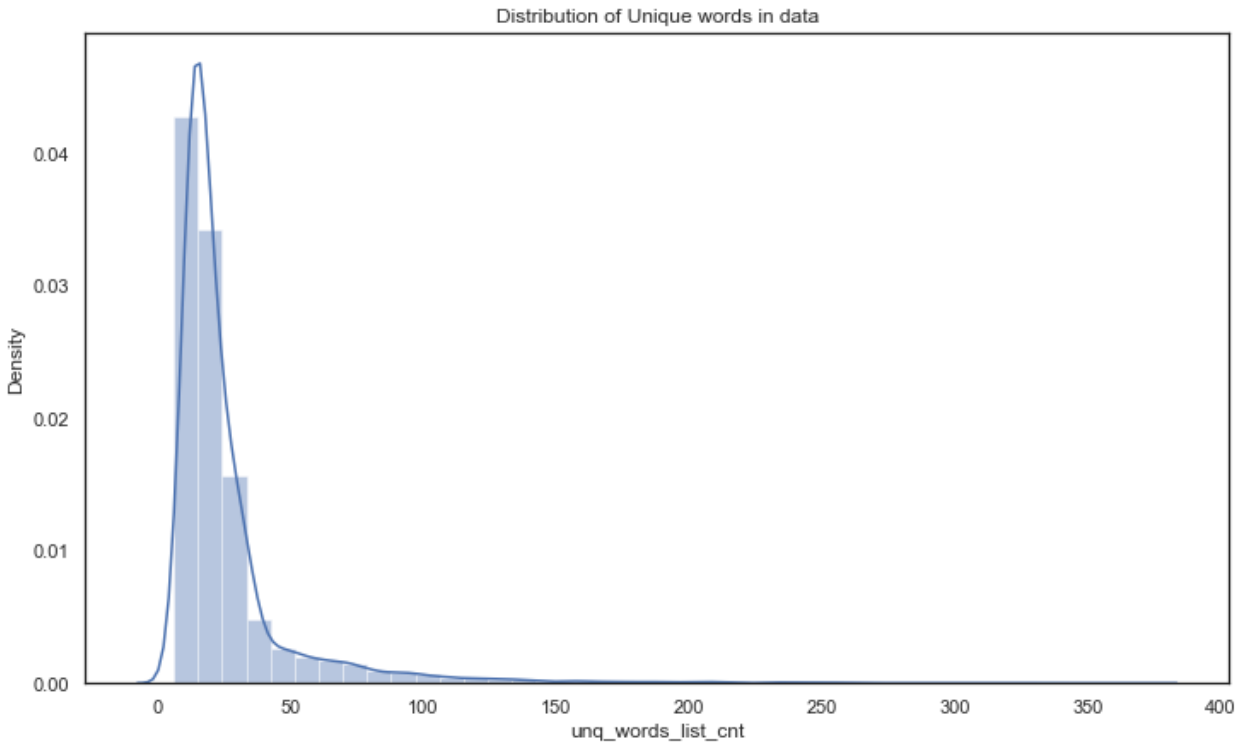
We will give it 128 units and make sure we return the sequences as true. This is to ensure that we can pass it through another LSTM layer. For the next LSTM layer, we will also pass it through another 128 units, but we don't need to specify return sequence as it is false by default. We will pass this through a hidden layer using the dense layer function with softmax set as the activation. Finally, we pass it through an output layer with the specified vocab size and a softmax activation. The softmax activation ensures that we receive a bunch of probabilities for the outputs equal to the vocab size. The entire code for our model structure is as shown below. After we look at the model code, we will also look at the model summary and the model plot.

Analysis:

To build model is to analyze the data and make sure we pre-process it so model can accept it. The first step is the clean the sample responses by combining customers and representatives' responses and then cleaned these texts to make sure no special characters or symbols present in the text data.

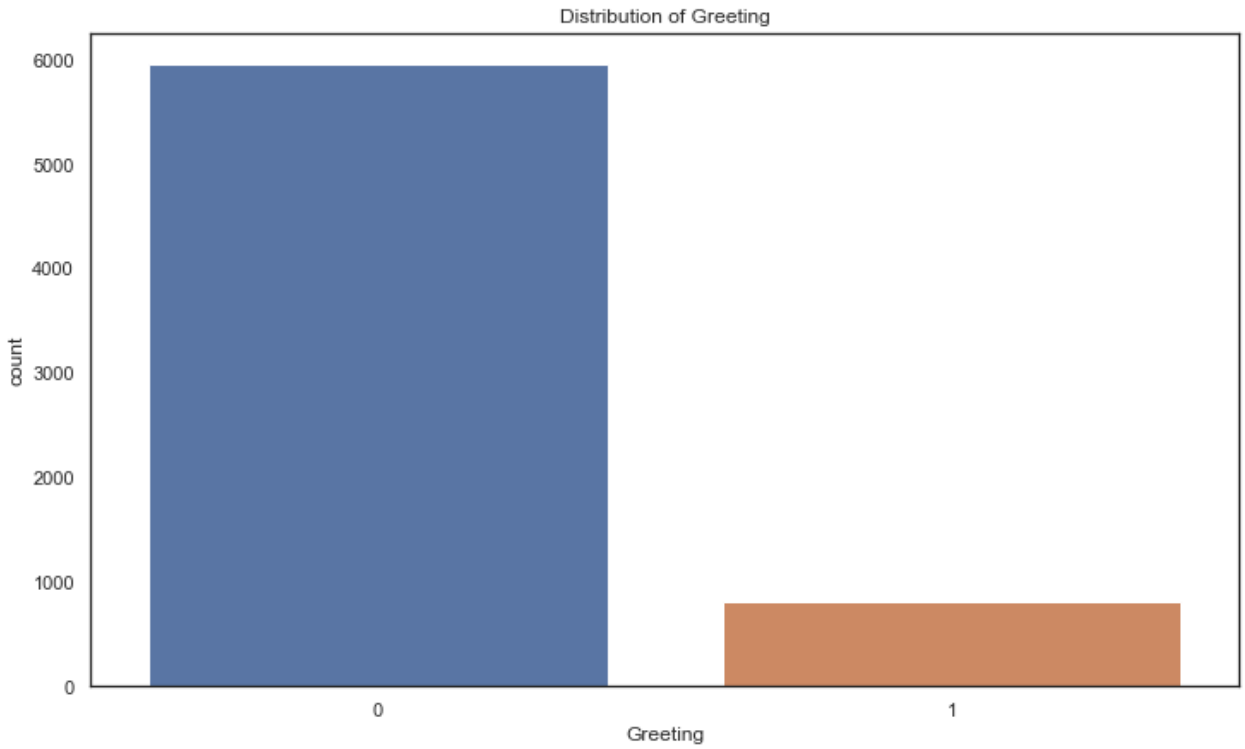
Used NLTK's tokenizer to tokenize the data and counted unique words from each record and plotted the distribution of it to see how each sample request and response have unique words.

The distribution of unique words counts shows that there are majority of the request and responses have 10-30 unique words and few responses have more than 150 words, but the samples are very few for such records in the dataset.

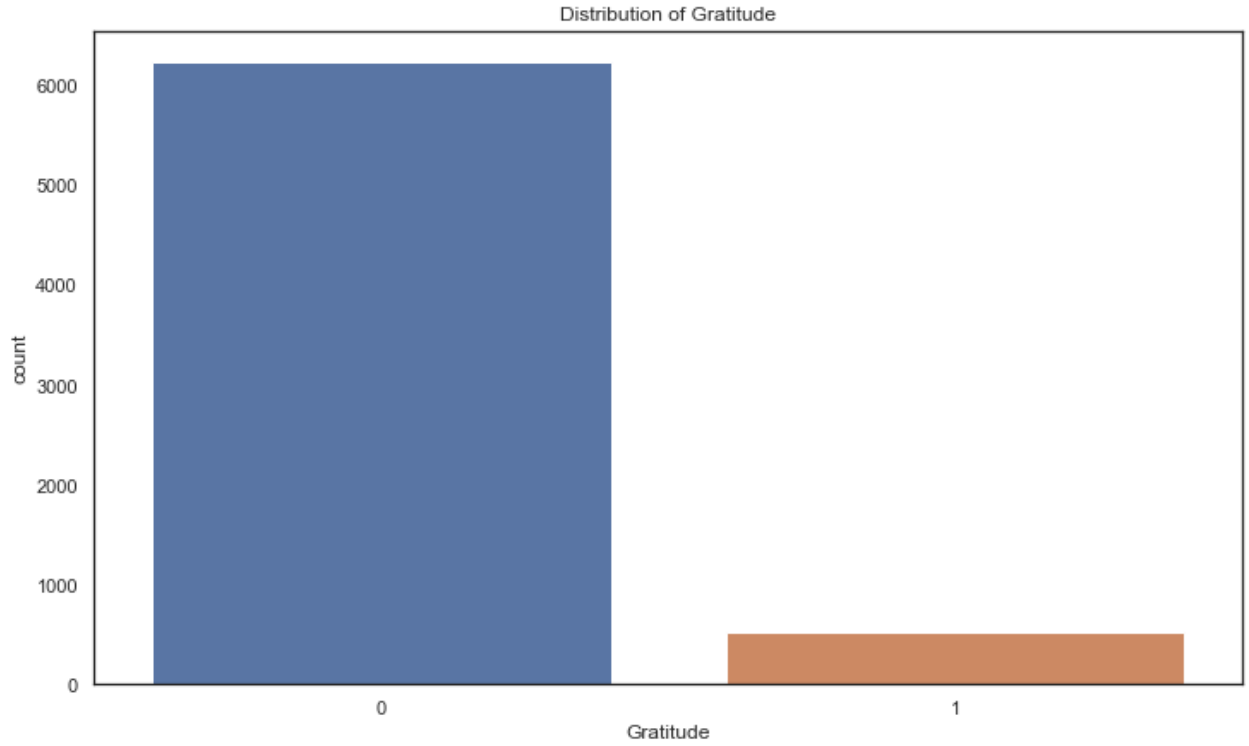


Datasets have few columns which shows that whether Gratitude or Greetings or Ranting Present in the response texts.
Plotted bar chart to see how it is distributed among the datasets. This would help understand how the responses are and there are more intent related words that just Greeting or Gratitude or Rantings.

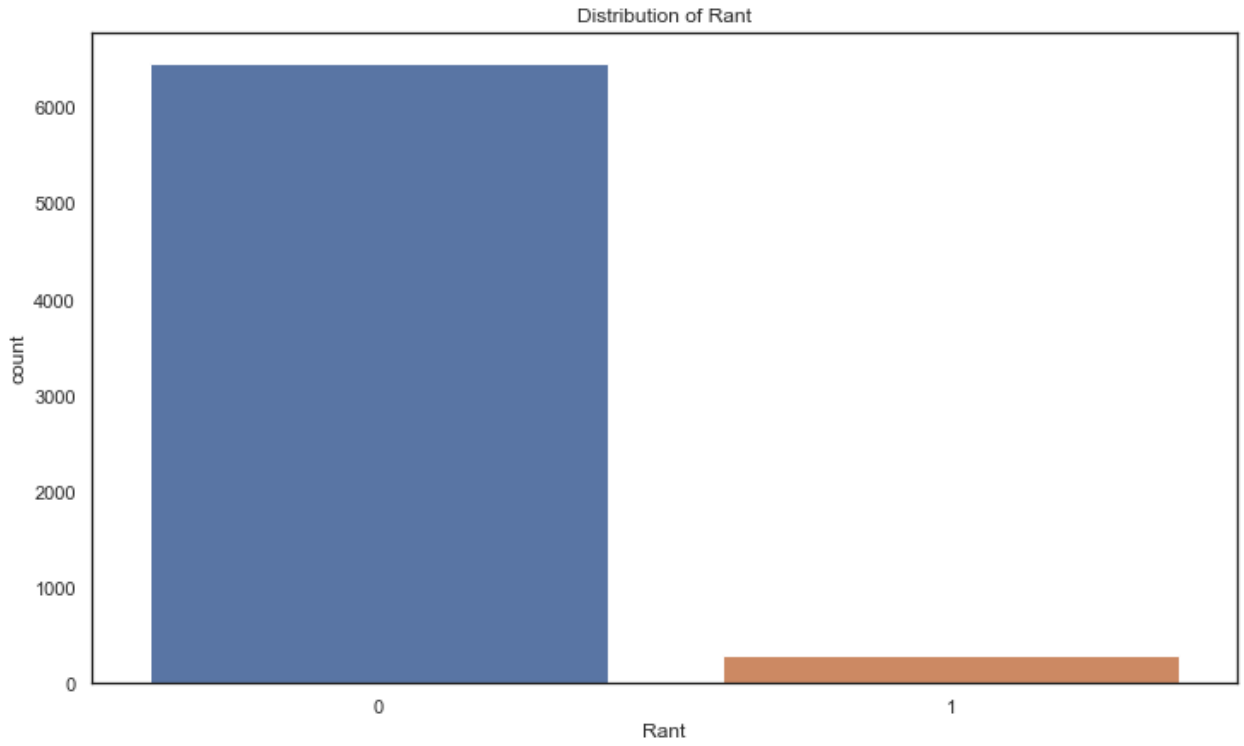
Majority of the responses do not have Greeting present, very few responses have greeting present, based on this bar chart it shows that only 11%-12% responses have greeting present.



For Gratitude, there are less than 5% responses contains Gratitude related words.



For Ranting, there are less than 3% responses have negative or complaining related words.



Conclusion:

The model is predicting the next words based on given input sample text, the generate next word or series of words can be handled through argument as configuration and changed based on model performance.

The next word prediction language model is generating the relevant word and can be used in production to help agent type response fast.

Assumptions:

The assumption made as part of this model building is that the training data is business related so model would learn and understand the context of that business or domain so it would generate relevant next words.
This model would understand English language and help predict next words. The training data used is specific to airline inquiries domain and model would predict next words relevant to that context.

Limitations:

Since the model is trained on specific domain, so it would predict next word in context with the domain and given sample text. Model may predict grammatically incorrect word if such words are present in training data.
This model is training using neural networks (RNN) so it would understand the context of given seed but may not produce meaningful words if very short feed is passed or no feed is passed.

Challenges:

This model is language model and need GPU enabled machines to train the model. The model trained on GPUs would give better result and training time would be faster. Since datasets in acquired in its raw format, there could be spelling errors or non-standard shortcuts being used and to clean this up need better resources and hardware.

Future Uses:

This model can be used for product larger text based on given texts where companies need to write communicational or information contexts. This model can be used for any application wherein people need to type in few words for searching or querying.

Recommendations:

This model needs to be phased out for few agents who can use this and give feedback based on context it generates and then rolled out for bigger population. For better and faster result, new responses need to be used for training the model as it is used on production system so it would learn new context and will be capable to predict such words.

Implementation Plan:

The baselined model would be deployed into production system in phases. The newly generated text and sample responses used to re-train the model, This process would be continuously running so model would give better result. Product system need to be upgraded to support required hardware to re-train the model.

Ethical Assessment:

All the personal identifiable information (PII) present in the datasets have been removed by tokenizing it. All numerical characters contained PII such as account number SSN, phone numbers are masked with '#' and alphabetic characters such as name, address, company name are masked with 'cname' and 'pname'.

This data is acquired in August 2016, made sure that it not biased based on gender or specific geographic area. Model would be trained purely from the historical data and will make sure there would not be any ethical bias.

Appendix:

- https://nextit-public.s3-us-west-2.amazonaws.com/rsics.html
- https://www.tensorflow.org/guide/keras/rnn
- https://dida.do/blog/ethics-in-natural-language-processing