Chatbot Project

Madison Montgomery, Darren Lin, Donying Zhou, Tyler Dougherty, Xuyang Bai

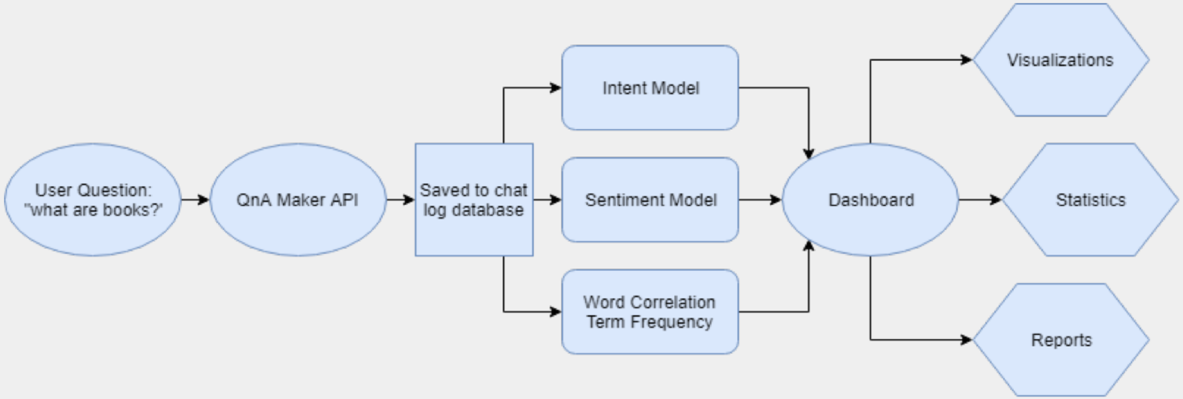
M.S. Business Analytics @ Rensselaer Polytechnic Institute

The Problem

- Learning curve for some users
  - Confusion → Negative reviews
- Need to obtain instant feedback
  - Mitigate customer dissatisfaction and increase usability
- No way to determine user intent
  - Is the feedback plausible?

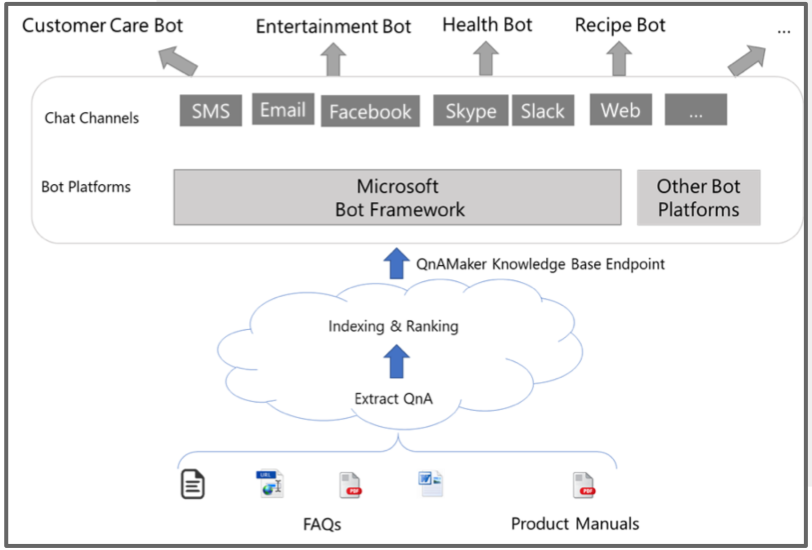
Task

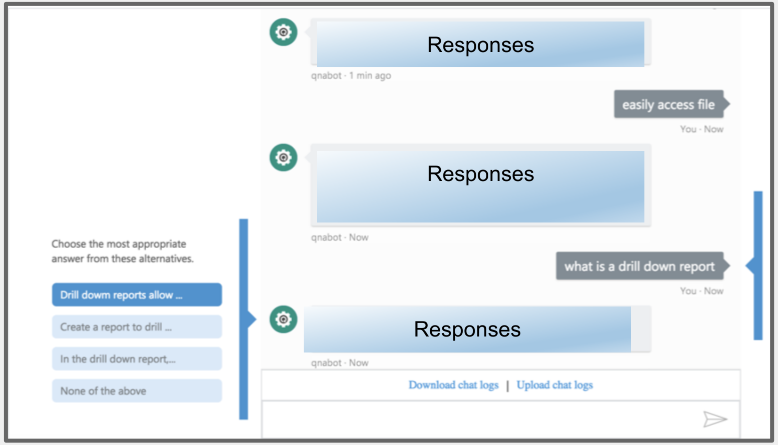
Develop an integrated chat bot application responsive to user questions, partnered with high level analyses of text data integrating various text mining models and visualizations



QnA Maker

- Formulated dictionary of possible questions and answers
- Used Microsoft QnA Maker to create initial chat bot
- Tested by asking each question 4-5 different ways
- Continuously retrained to form correlations
- Exported chat log for analysis

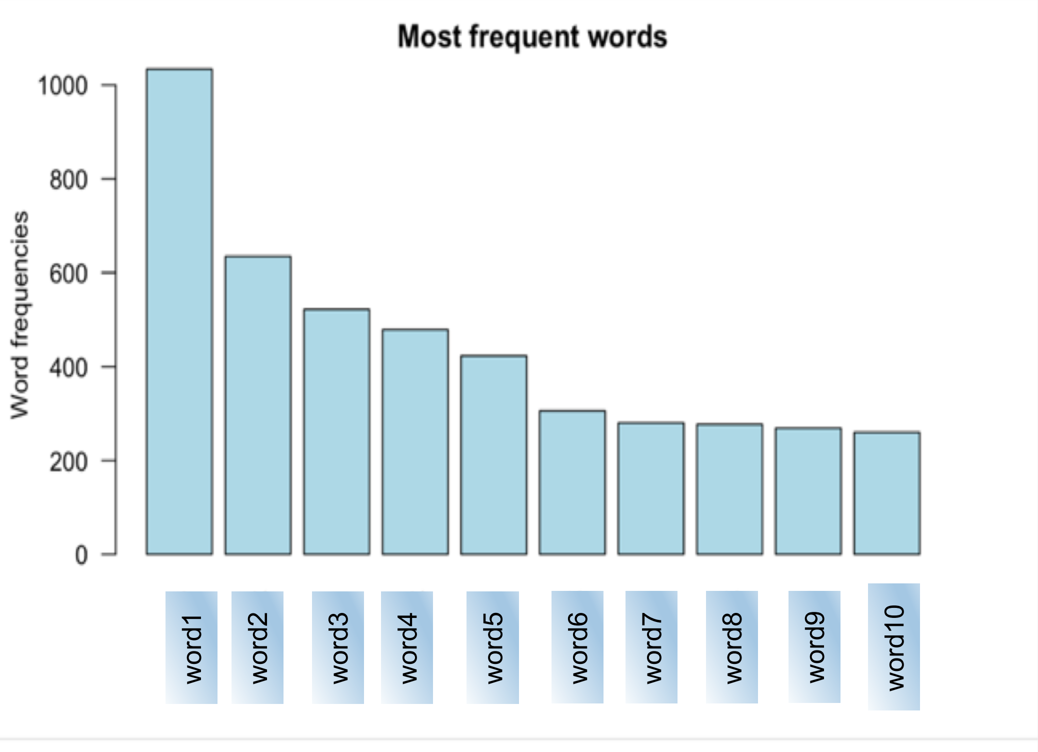




Term Frequency- Tools Introduction

- Tools: Text mining in R
- Purposes:
  - higher frequency->more important
- Eliminate whitespace; lowercase; stopwords; stemming
- Create Term-Document Matrix

Term Frequency- Current Situation

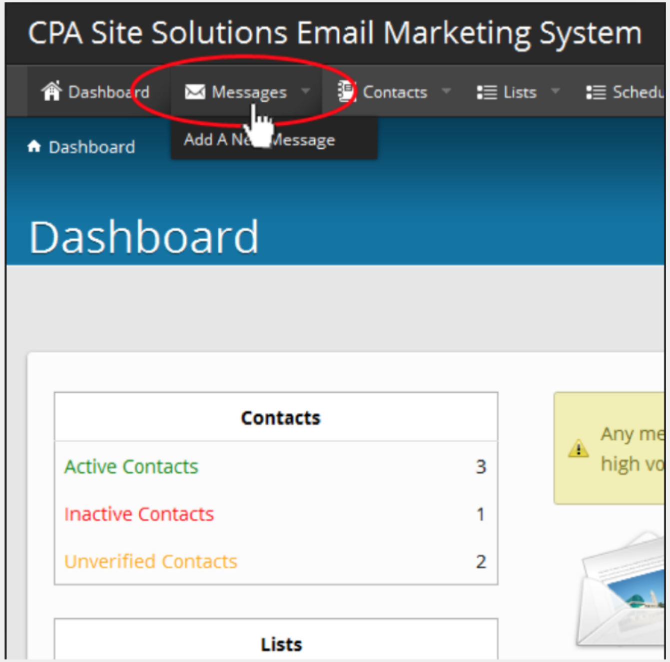


Term Frequency- Usage

Usage: Understanding Customer Behaviors and Need:

Historical TF trend of the same group of user -> e.g. seasonality

Tailored customer services plan



Term Frequency- Business insights

Problem: Customers have question on expanding the options and further analyze their financial statement

Suggestion: Clear direction on dimension settings

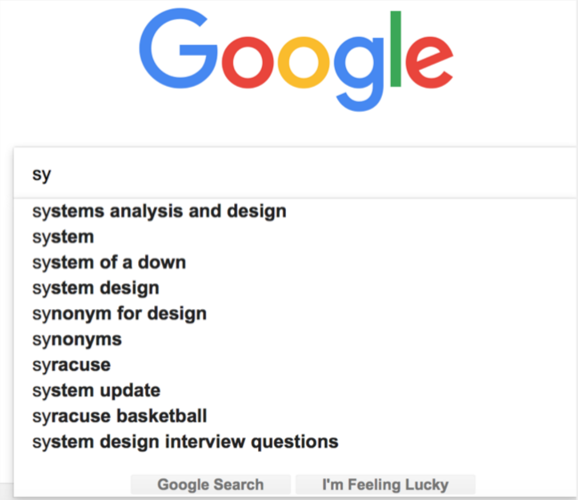
Solutions

Keep Current System Design:

comment bar

meta key to each dimension

give options to choose



Word Correlation - Concepts

- Correlated word pairs: Pairs of words that are likely to appear in same questions
- Correlation coefficient: Measure of correlation between two words
  - Positive correlation coefficient: The two words are likely to appear in same questions
  - Negative correlation coefficient: The two words are likely to appear in different questions
- Stop words: Words that do not contain much useful information, such as “a”, “the”, “of”, etc.
- Stop words are excluded from word correlation analysis

Word Correlation - Calculation

- Tool used: R packages
- Data cleansing before analysis:
  - Define and remove stop words
  - Singularize each noun to avoid duplication
- Calculating correlation coefficients of word X and word Y:
  - N11: Number of questions that contain both word X and word Y
  - N10: Number of questions that contain word X but do not contain word Y
  - N01: Number of questions that contain word Y but do not contain word X
  - N00: Number of questions that contain neither word X nor word Y

$$Cor(X,Y) = \frac{\mathrm{N11} * \mathrm{N00} - \mathrm{N10} * \mathrm{N01}}{\sqrt{(N11+N10)*(N11+N01)*(N11+N00)*(N10+N00)*(N01+N00)}}$$

Word Correlation - Output and visualization

- Output:

  A list of word pairs with their corresponding correlation coefficients OR a matrix of correlation coefficients

- Visualization:

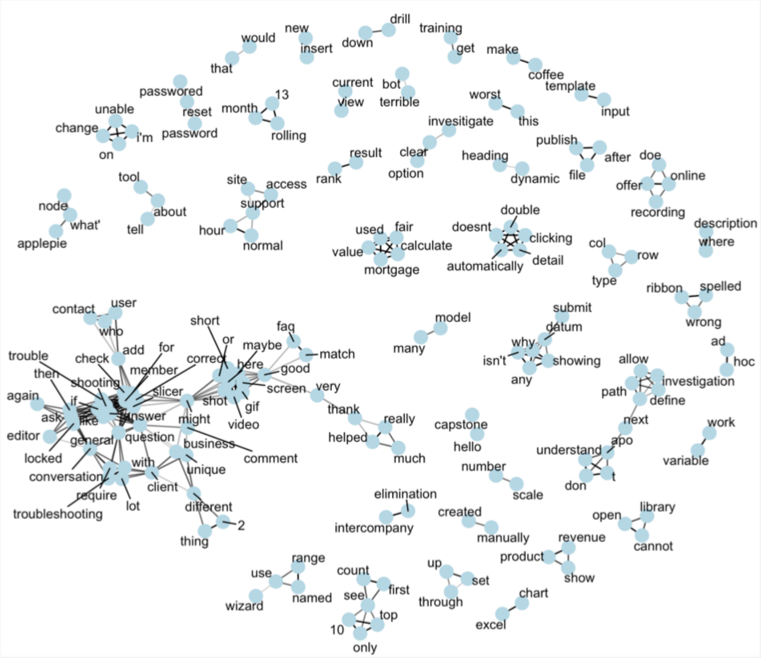
  Visualizing highly correlated words with link-node graph

- Tools used:

  R plotting packages or Power BI

Word Correlation - Visualization and interpretation

- Interpreting visualization: In the plot, only highly correlated word pairs (cor>0.5) are shown.
- Associations among words are formed through correlations, helping recommendation



NLP

- Field of computer science related to artificial intelligence
- Extracts meaningful data from conversations between a computer and a user

Human: “Wake me up at nine a.m.”

Computer: {
user_intent: “set_alarm”,
time: 9,
response: “OK, alarm set for 9 a.m.”
}

- Incorporate NLP into the bot to extract user intent

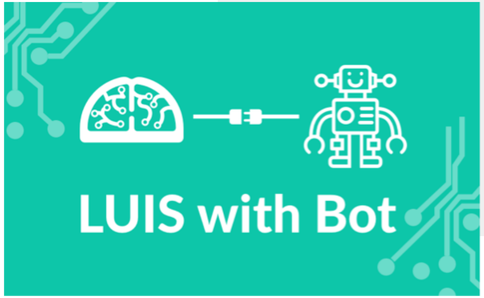
Human: “Make me a copy of this report.”

Computer: {
user_intent: “duplicate_file”,
data: “C:/Data/Reports/2018/Jan/Fancy_Report.xlsx”,
response: “OK, I have copied the file.”
}

LUIS

- Machine learning-based service that parses natural language
  - Output: structured object that contains interpreted intent and arguments
- General misalignment between goal of project and what Luis provides
  - Project: Q&A
    - "What is..." -> "This is the definition of..."
    - "How do you..." -> "Open the menu, then..."
  - Luis: Querying some action
    - "Print out 2 copies" ->

```
{ “intent”: “print_document”, “copies”: 2 }
```



Classifying Questions by Intent

- The problem: Too many ways to ask a particular question → Unable to quantify overall issues; messy visualizations
- The solution: Create a classifier that can accurately read in a question and output its intent

Question → Intent

"How can I create a report?"
"how to write report"
"how create report"
→ Create Report

Bag-Of-Words Model

- Machine learning algorithms prefer well defined fixed-length inputs and outputs
  - BoW model extracts features from text to be used in algorithms
- The model involves two things:
  - A vocabulary of known words
  - A measure of the frequency of the occurrence of each word
- “Bag” refers to the fact that the order of words is not significant
  - The model only cares about whether a word is in the document, not where

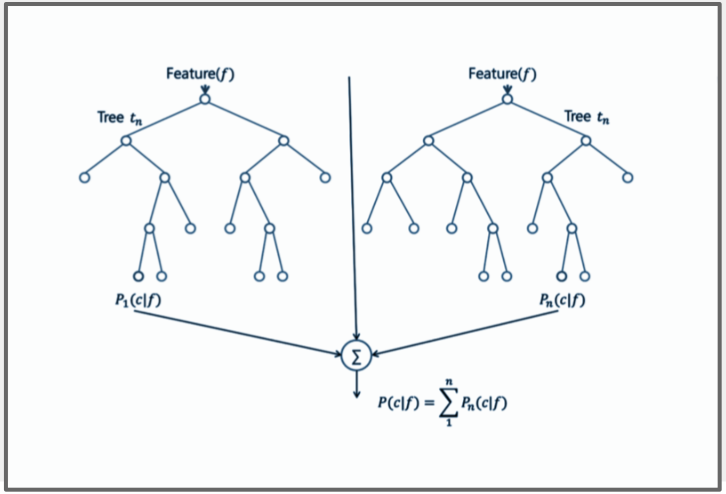
“do you offer online recordings”
“do you offer online training”

→

do	you	offer	online	recordings	training
1	1	1	1	0	1

Random Forest Classifier

- A random forest was used to classify and make predictions on the new questions
- Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction
  - “Random” refers to the tree picking the best feature to split on amongst a random subset of features → Better results than classic decision trees



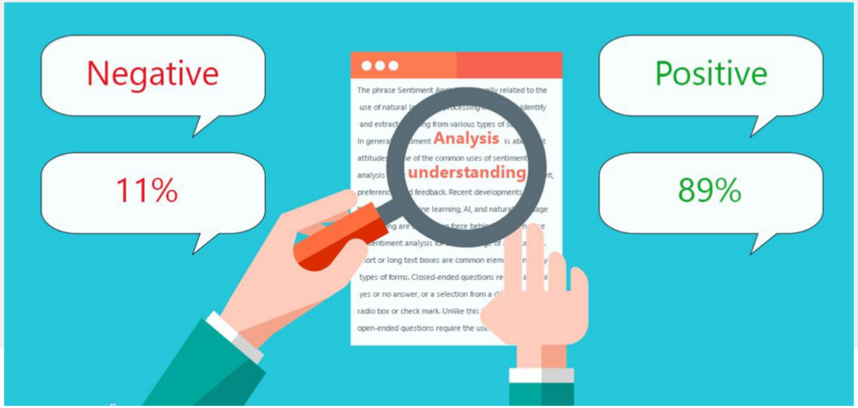
- Overfitting can be avoided by increasing the number of trees in the model
- Features are extracted from BoW and used in random forest predictions
- Trained on 300 questions with labelled intents
- Tested on chat log of 10,000+ questions

Sentiment Analysis

- Code that identifies/categorizes opinions expressed in text
- Useful for determining attitude and emotion
- Looked at:
  - Google Natural Language API
    - Magnitude + score
  - Amazon Comprehend API
    - Positive, Negative, Neutral, Mixed
  - Microsoft Azure Text Analytics API
    - Score
  - Python NLTK
    - Requires existing data

Sentiment - Azure Text Analytics

- Easy and simple to work with
  - 1 value is easy to understand and put in a visualization for the dashboard
- Integration
  - Q&A Maker
  - PowerBI
  - The company already uses Microsoft for everything else



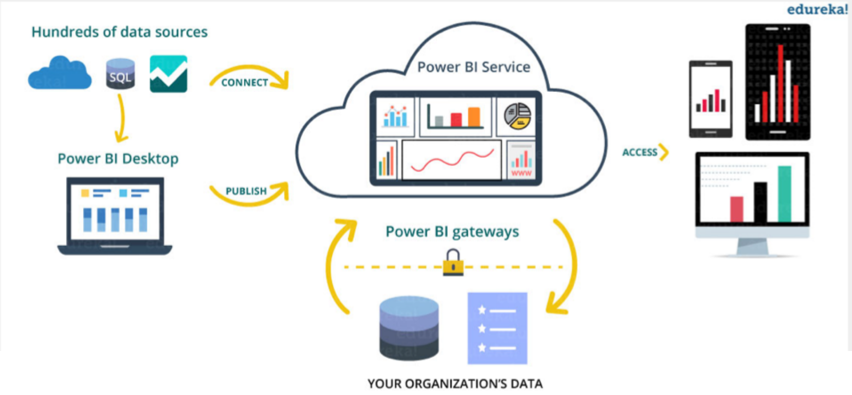
Business Intelligence Dashboards

- Most practical way to leverage BI
- Gather, consolidate, and analyze business driving data
- Sleek, real-time visibility
- Identifies areas of improvement
- Easy to use → everyone can create reports and make decisions
- Customizable

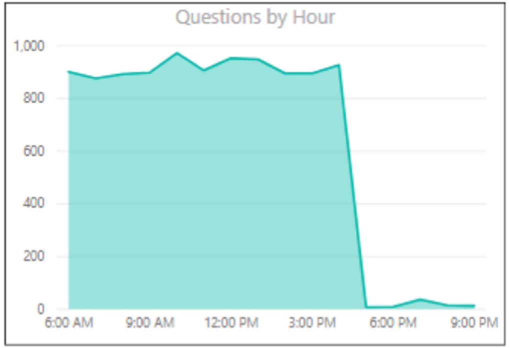
“SMBs live and die by the data in their systems”

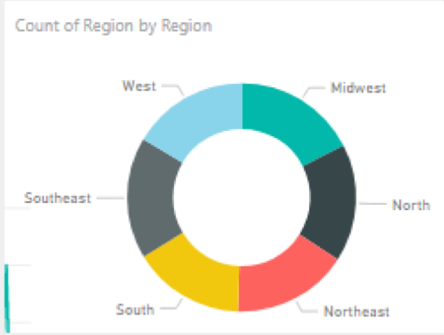
Power BI Features

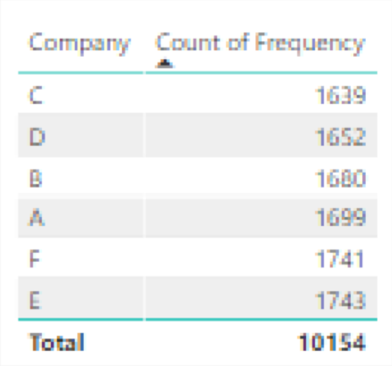
- Microsoft cloud-based analytics service
- Self service BI capabilities and visualizations
- Based on Excel add-ins
    - Power Query, Power Pivot, Power Viewer
- Data exploration
    - Interactive reports with natural language capabilities
- On premise and cloud data
    - Pre-built services for 54 common programs (GitHub, SQL, etc.)



Visualizations







Company	Count of Frequency
C	1639
D	1652
B	1680
A	1699
F	1741
E	1743
Total	**10154**

Data Changes

- Frequency → not useful
- Binned score, time, sentiment
- New groups for time
- Changed region
- Red flag
- Have we helped

Next Steps

- Deploy the bot!
  - Gathering real data will be the key to the bot's success
- Possibly explore different frameworks for the bot
  - Other APIs
  - Larger, faster databases that can handle high amounts of message requests
- Streamlining
  - Linking all components together (i.e. the bot, chat log, dashboard, etc.)
  - Real-time data; dynamic dashboard