AI Dining Suggestion App

By Bao Pham

Advisor: Dr. Chris Pollett
Committee Member: Dr. Mike Wu
Committee Member: Dr. Kevin Montgomery

Agenda

1. Introduction
2. Preliminaries
3. Implementation
4. Experiments
5. Conclusion

Introduction

Introduction

- Trying to decide what to eat can sometimes be challenging and time-consuming for people.
- Google Maps and Yelp are two main applications of the web that provide lists of restaurants.
- One’s taste at a certain time does not always depend on the reviews of a restaurant
- Google Maps and Yelp apps still offer the traditional way of typing keywords, and the apps show the list of potential matches.

Introduction

Issues:

- Google Maps and Yelp do not really learn about a user’s taste.
- Lack of simplicity with regard to functionality.

Proposed Solution

- Use AI models to learn the user's dining pattern over time to help make restaurant suggestions.
- Create friendly UI app with simplicity to change the way of how data is presented to user.

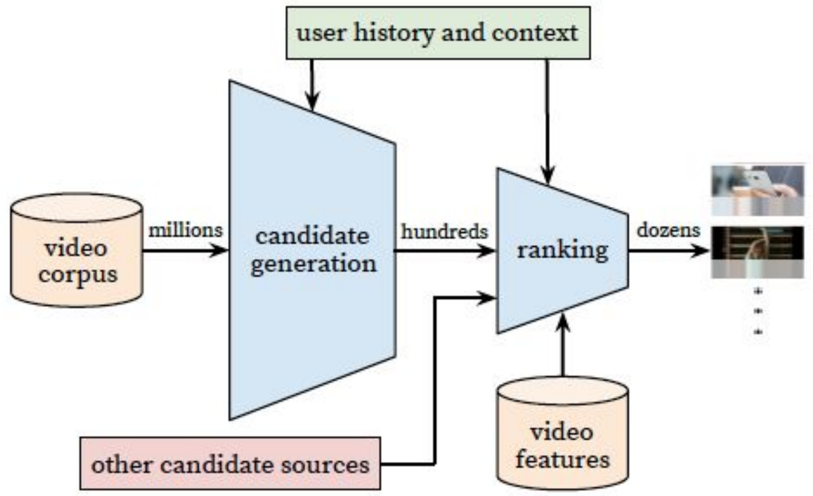
Goal: to reduce time when it comes to food decision making process.

Related Work

- Several papers discuss improvements to Yelp’s recommendation systems using neural networks and machine learning algorithms.
- These models are static and would only provide the same output if given the same input of restaurants because they are trained on Yelp data.
- Not much work found in terms of how the restaurant data is presented to the user.
- **They indicated that machine learning methods can be useful for recommendation systems.**

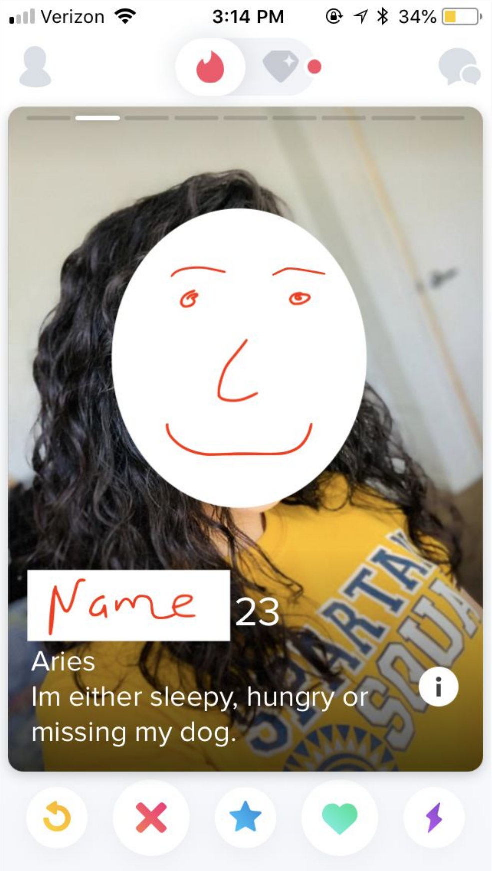
Related Work

- Youtube's recommendation system paper (2016):
  - It has two layers
  - Candidate Generation filters the options that are irrelevant for an average user
  - Ranking ranks the options that are filtered by the the first layer for a specific user



Related Work

- UI was designed following Tinder's format:
  - Tinder is a dating app which has gained lots of traction and has become a success due to its simplicity
  - Options are presented as a deck of cards

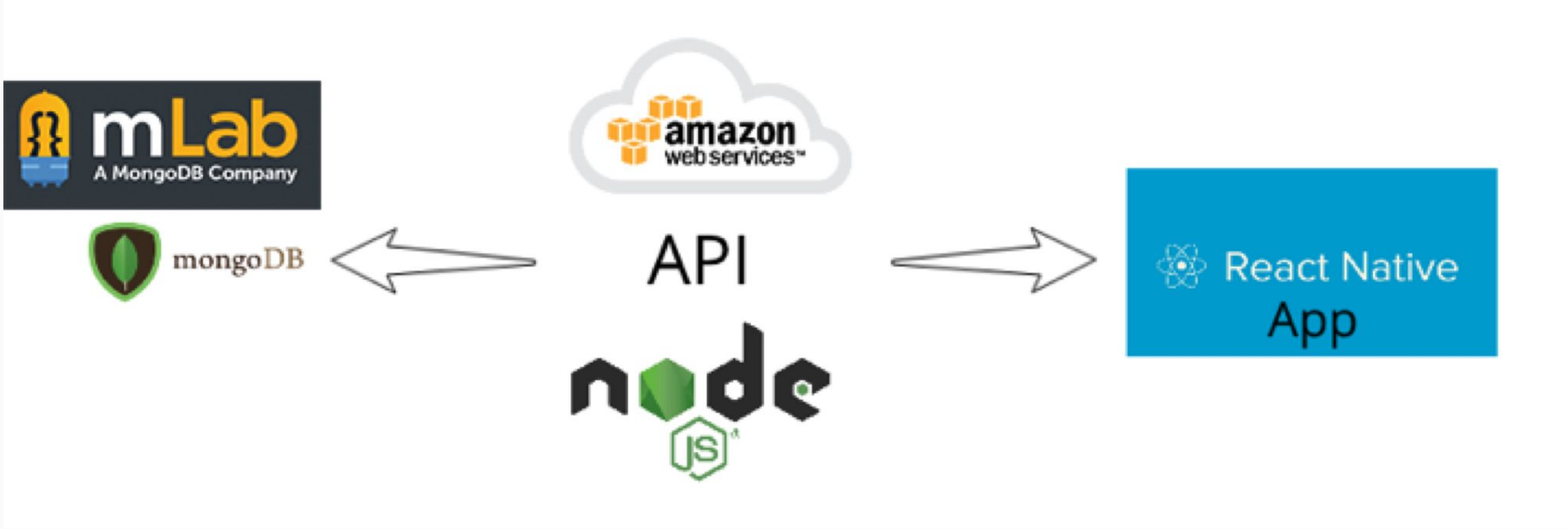


Preliminaries

Technology Stack

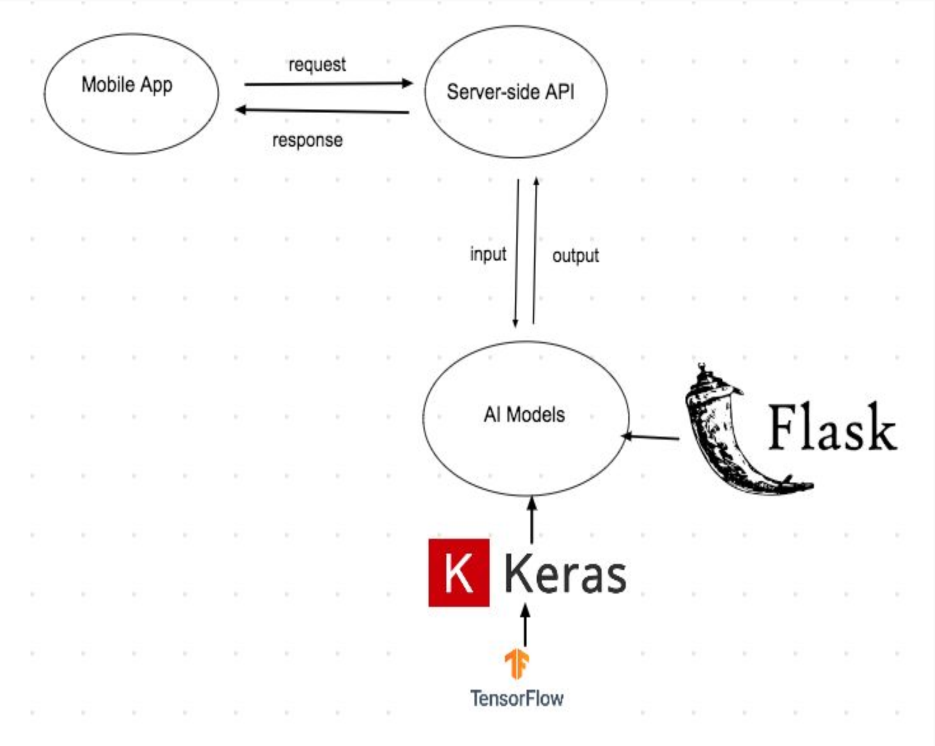
- React Native
  - Javascript
  - Multi-platform (iOS & Android)
  - Strong community support with lots of plugins
- MongoDB
  - NoSQL, flexible schema
  - JSON objects
  - mLab
- NodeJS
  - Javascript
  - RESTful
  - Amazon Web Services

Technology Stack

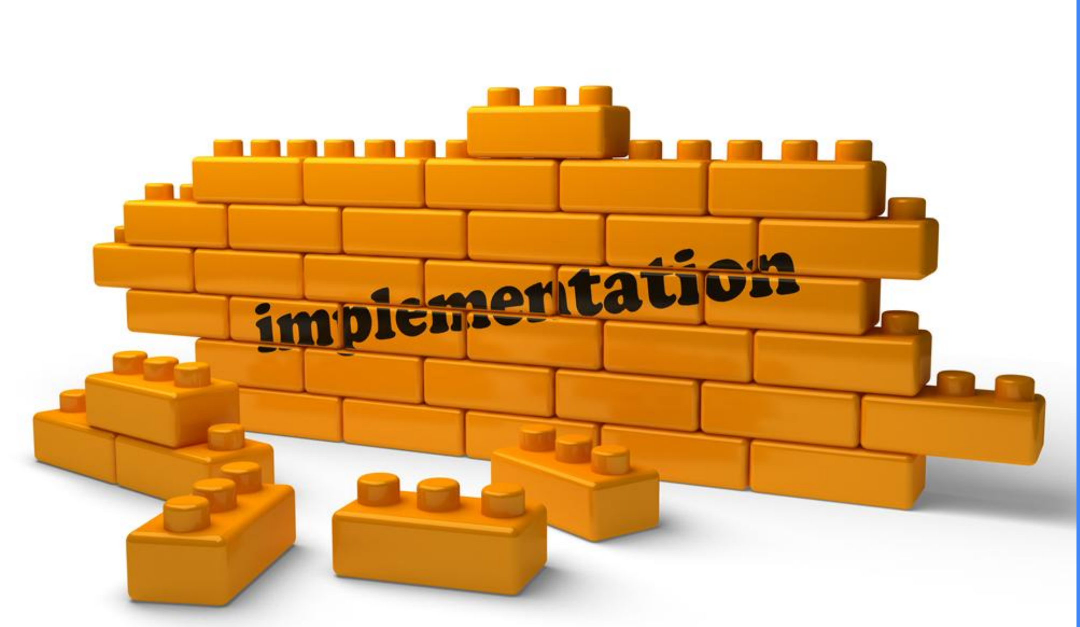


AI Technology

- Tensorflow
  - Common neural network framework
- Keras
  - Neural network library
  - Different implementations of commonly used NNs
- Flask
  - Web framework for Python



Implementation

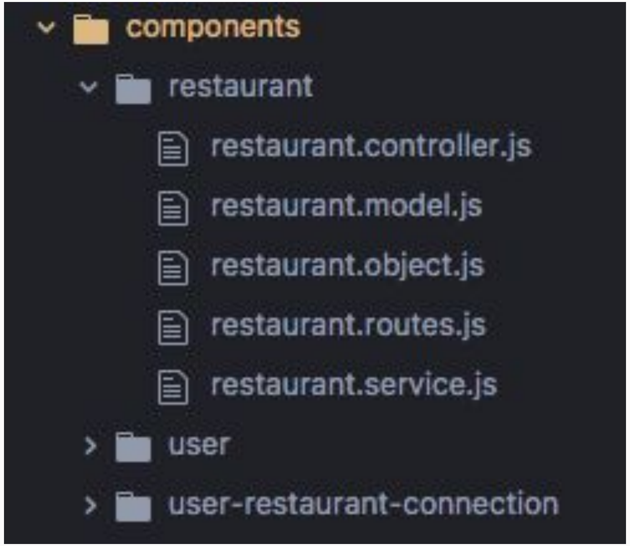


App Design

- Swiping left or right to show like or dislike
- When finalizing the decision, the app should show name and address of the restaurant

Implementation

- RESTful API with each component structure as follows:
    - Router
    - Controller
    - Service
    - Object
    - Model



- Data collection with Google and Yelp APIs

Implementation


```
{
    "geometry": {
        "location": {
            "lat": 37.344459,
            "lng": -121.838002
        },
        "viewport": {↔}
    },
    "icon": "https://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png",
    "id": "022beae10dff8f3de57a303f0e0605479c758cef",
    "name": "La Mejor Taqueria",
    "opening_hours": {
        "open_now": true
    },
    "photos": [↔],
    "place_id": "ChIJG798FjfNj4AR7Th0RBwZWU8",
    "plus_code": {
        "compound_code": "85V6+QQ San Jose, California",
        "global_code": "849W85V6+QQ"
    },
    "price_level": 2,
    "rating": 4.3,
    "reference": "ChIJG798FjfNj4AR7Th0RBwZWU8",
    "scope": "GOOGLE",
    "types": [
        "restaurant",
        "point_of_interest",
        "food",
        "establishment"
    ],
    "vicinity": "2003 Story Rd # 975, San Jose"
},
```

Implementation


```
{
    "id": "DFz_ahwurm2iI-diUEOiLw",
    "alias": "mexican-hot-dogs-san-jose",
    "name": "Mexican Hot Dogs",
    "image_url": "https://s3-media4.fl.yelpcdn.com/bphoto/HdjIieAZdScDLtRs-xIsqg/o.jpg",
    "is_closed": false,
    "url": "https://www.yelp.com/biz/mexican-hot-dogs-san-jose?adjust_creative=WCXgJyb8BoQN0PJeH47AMQ&utm_campaign=yelp_api_v3&utm_medium
        =api_v3_business_search&utm_source=WCXgJyb8BoQN0PJeH47AMQ",
    "review_count": 36,
    "categories": [↔],
    "rating": 5,
    "coordinates": {
        "latitude": 37.351923,
        "longitude": -121.824805
    },
    "transactions": [],
    "price": "$",
    "location": {
        "address1": "2770 Story Rd",
        "address2": "",
        "address3": "",
        "city": "San Jose",
        "zip_code": "95127",
        "country": "US",
        "state": "CA",
        "display_address": [
            "2770 Story Rd",
            "San Jose, CA 95127"
        ]
    },
    "phone": "",
    "display_phone": "",
    "distance": 606.3315765431433
},
```

App Design

- Challenges:
  - Yelp query limit
  - Only 3 images/restaurant from Yelp
  - Duplicate restaurants from both Yelp and Google
  - Large amount of queries made to Yelp and Google APIs -> surpass the free tiers
- Solutions:
  - Switch all of the queries to Google except initial query
  - Map restaurants using location to avoid duplicates
  - Create a cache system to avoid repetitive queries

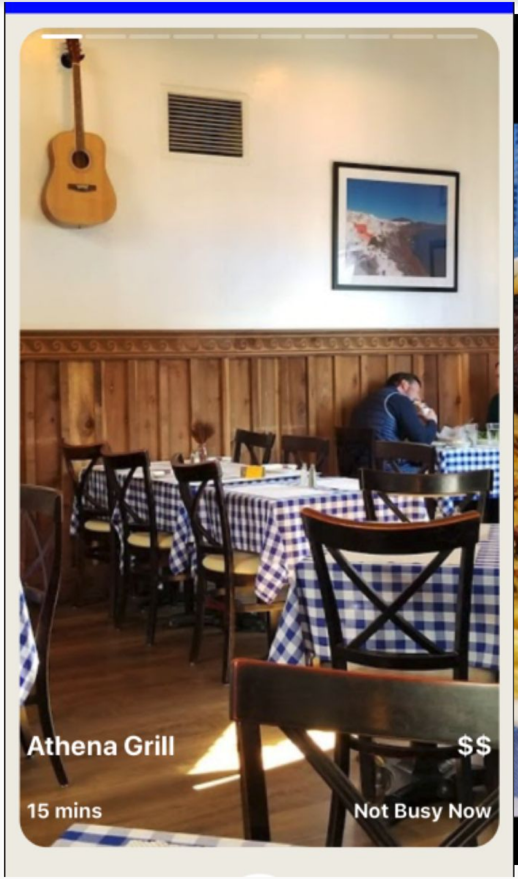
External Services for API

- Food decisions rely on some other factors, not just the restaurants themselves.
- Three extra types of information:
  - Temperature
  - Travel time
  - Busy hours


```
"coord": {
    "lon": -121.83,
    "lat": 37.35
},
"weather": [
    {
        "id": 502,
        "main": "Rain",
        "description": "heavy intensity rain",
        "icon": "10n"
    },
    {
        "id": 701,
        "main": "Mist",
        "description": "mist",
        "icon": "50n"
    }
],
"base": "stations",
"main": {
    "temp": 287.67,
    "pressure": 1002,
    "humidity": 89,
    "temp_min": 287.05,
    "temp_max": 288.75
},
```

Front-end Implementation

- Authentication (Unique ID for each user)
- React Native Deck Swiper
- React Native Image Viewer



Filtering and Processing Data for AI Training

- Filtering data:
  - Yelp's challenge dataset
  - 12 Gigabytes (business, user, review, photo data)
  - Not all businesses are restaurants
  - Filtered by categories, attributes
- Processing data:
  - Tensorflow can't train string typed data
  - Converted data to integer and float type using a mapping system
- 1,418,418 reviews with ratings, 32,516 restaurants, 191,904 users and 69 restaurant categories for ~3 Gigabytes of data

Filtering and Processing Data for AI Training

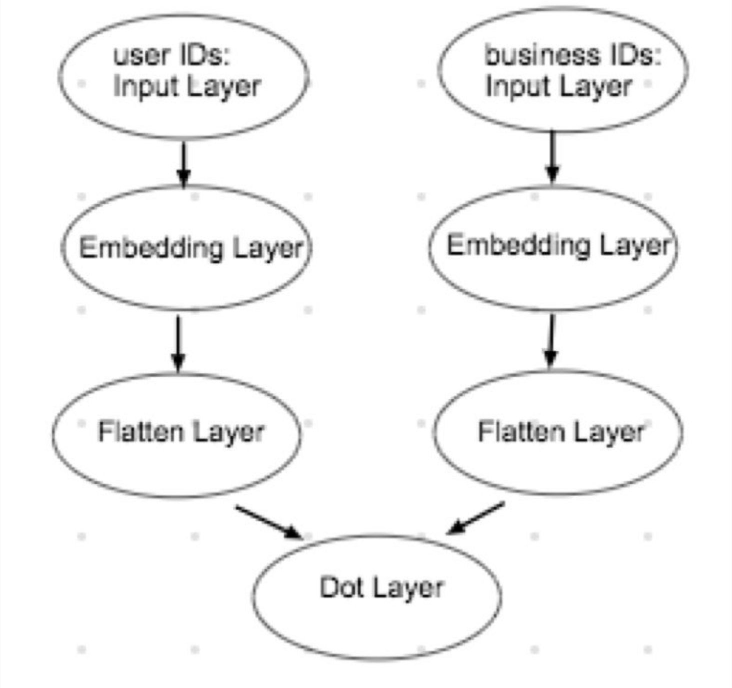

```
{
   "business_id":"Apn5Q_b6Nz61Tq4XzPdf9A",
   "name":"Minhas Micro Brewery",
   "neighborhood":"",
   "address":"1314 44 Avenue NE",
   "city":"Calgary",
   "state":"AB",
   "postal_code":"T2E 6L6",
   "latitude":51.0918130155,
   "longitude":-114.031674872,
   "stars":4.0,
   "review_count":24,
   "is_open":1,
   "attributes":{
      "BikeParking":"False",
      "BusinessAcceptsCreditCards":"True",
      "BusinessParking":"{'garage': False, 'street': True, 'validated': False, 'lot': False, 'valet': False}",
      "GoodForKids":"True",
      "HasTV":"True",
      "NoiseLevel":"average",
      "OutdoorSeating":"False",
      "RestaurantsAttire":"casual",
      "RestaurantsDelivery":"False",
      "RestaurantsGoodForGroups":"True",
      "RestaurantsPriceRange2":"2",
      "RestaurantsReservations":"True",
      "RestaurantsTakeOut":"True"
   },
   "categories":"Tours, Breweries, Pizza, Restaurants, Food, Hotels & Travel",
   "hours":{
      "Monday":"8:30-17:0",
      "Tuesday":"11:0-21:0",
      "Wednesday":"11:0-21:0",
      "Thursday":"11:0-21:0",
      "Friday":"11:0-21:0",
      "Saturday":"11:0-21:0"
   }
},
```

Filtering and Processing Data for AI Training

```
user_id business_id category rating
0 0 6 5
0 1 0 1
0 2 2 2
0 3 6 5
0 4 2 5
0 5 6 5
0 6 0 4
1 15 0 5
1 16 5 2
1 17 0 5
1 18 14 3
1 19 0 2
1 20 0 5
1 21 9 5
```

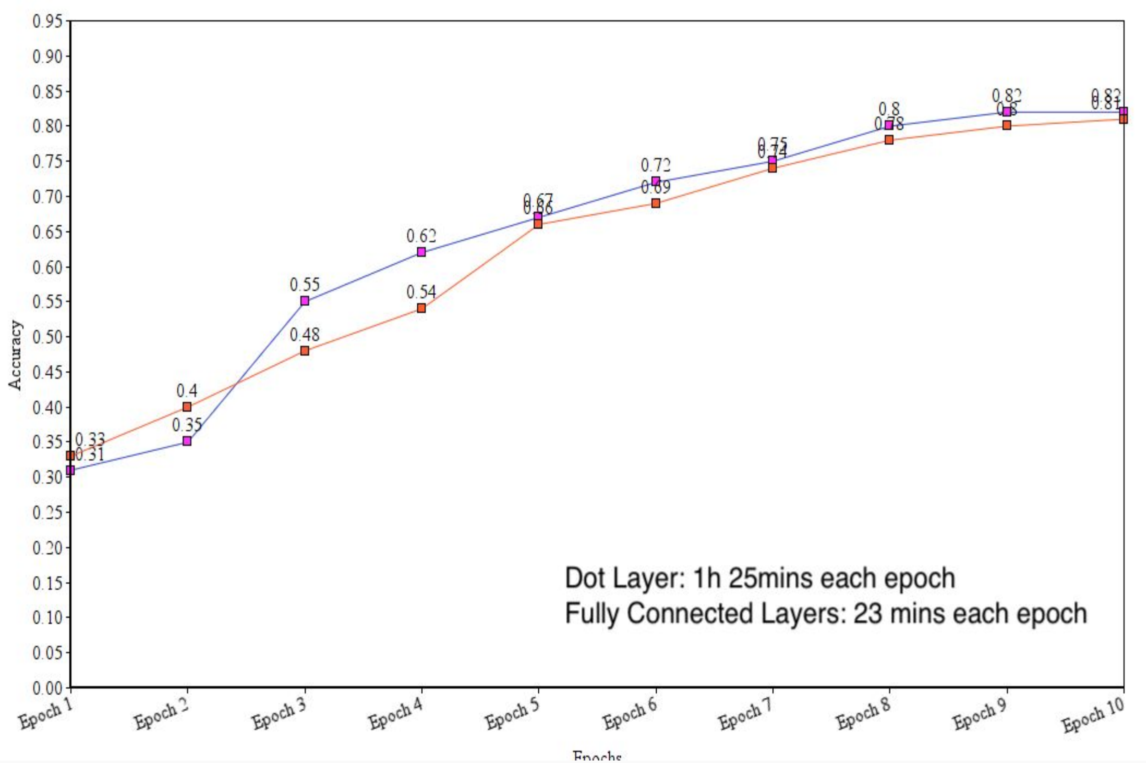
Training Static AI Model

- Matrix Factorization as a baseline model
- Using business, user IDs and ratings



Training Static AI Model

- Improvements on baseline model
- Replaced Dot Layer by three fully connected layers: 128, 64, 1 with two Dropout Layers with dropout rate 0.2
- Dataset was sparse
- Some restaurants have one rating
- Some users only rates once
- Optimal architecture at this point given training time trade-off



Training Reinforcement Learning Model

How to prepare data when each action recorded from the app is an object? (LSTM)

- Unfolded each object into a string of field
- Unlike video games when feedback is learned immediately from each action -> training stack is 20 actions
- Used Label Encoding and One Hot Encoding

Training Reinforcement Learning Model

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

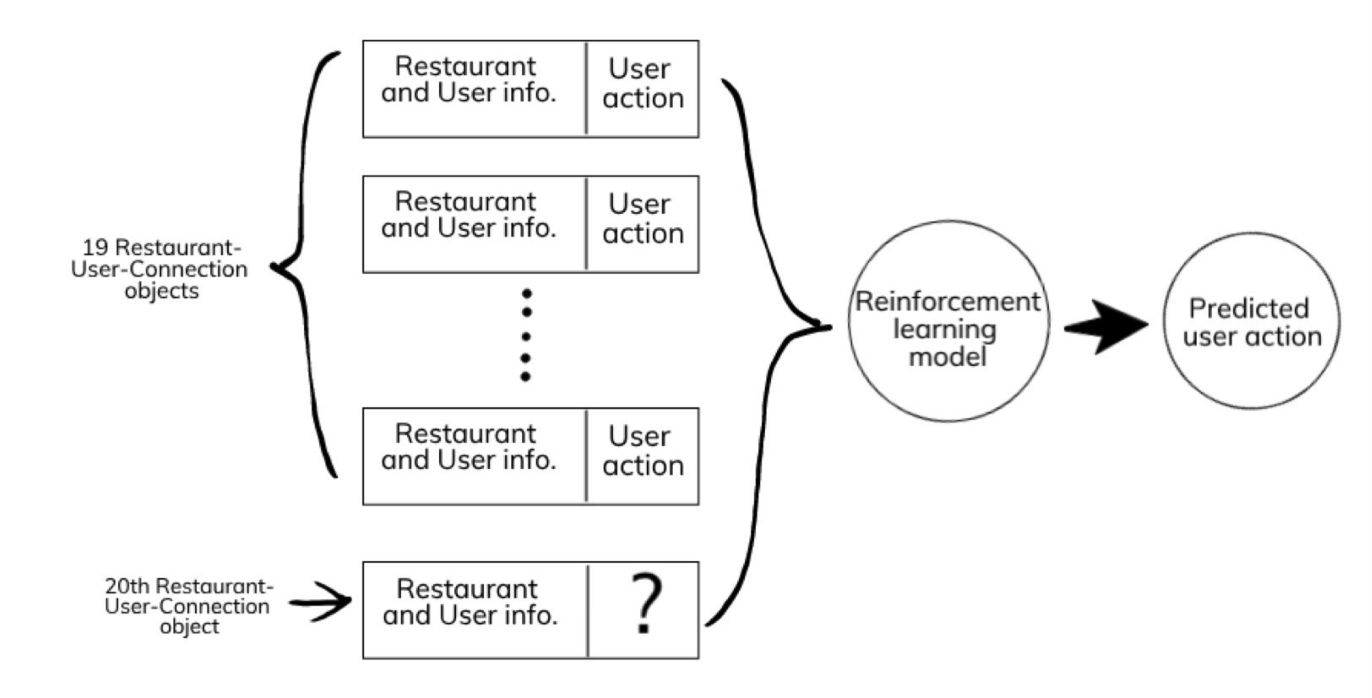
Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

- A day value = 0 -> 6 for Mon -> Sun
- 69 restaurant categories = 0 -> 68

Training Reinforcement Learning Model

- One action string was converted into:

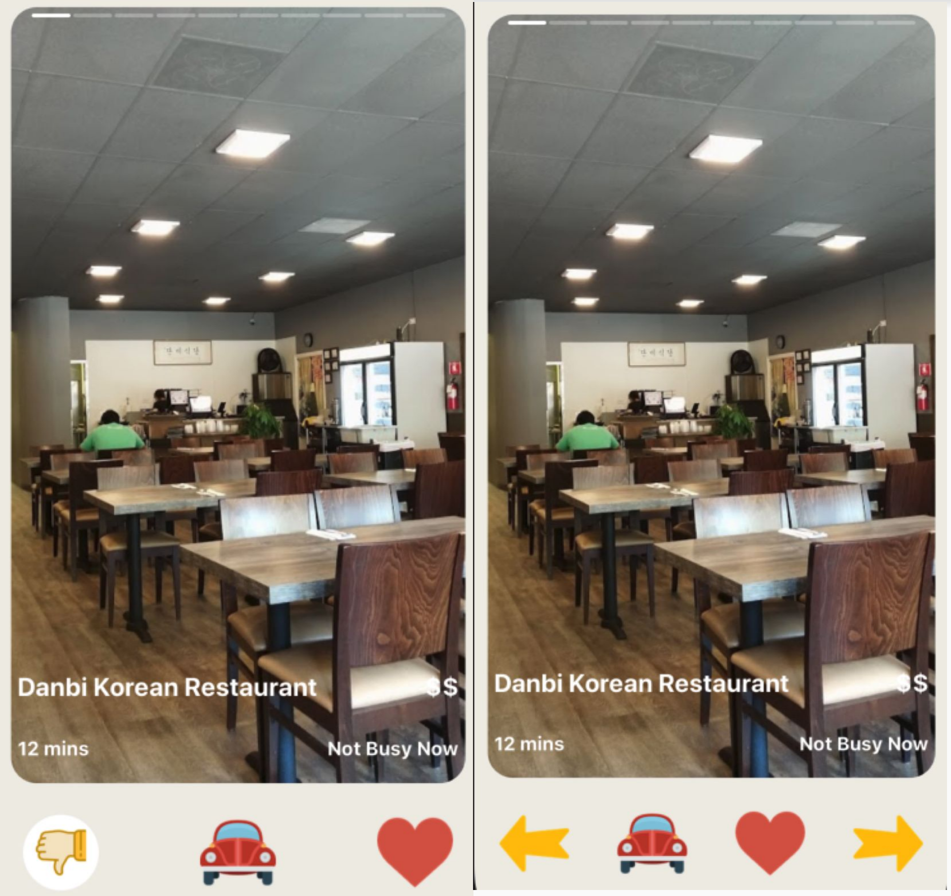
  [0, 4, 11, 16, 20, 51, 1, 0, 0, 0]



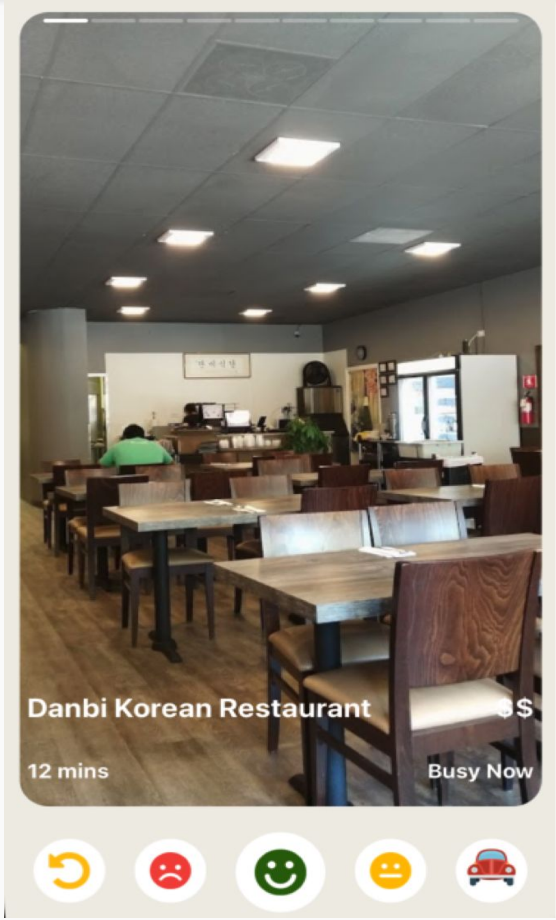
Experiments

App Experiment

- Five testers is sufficient as a study's result from Nielsen Norman Group
- The formula only holds for comparable users who will be using the app in fairly similar ways
- A/B testing
- Great influence from Tinder
- "Dislike" and "Like" were confusing



App Experiment



AI Experiment

- Used an average user from Yelp's dataset
- First layer filtered the low rated restaurants
- At first, the testers did not notice the difference from the reinforcement learning model (accuracy only 70%)
- As more data came in, the accuracy gradually went up
- Model learned some restaurants that users had disliked to push down to the stack

AI Experiment

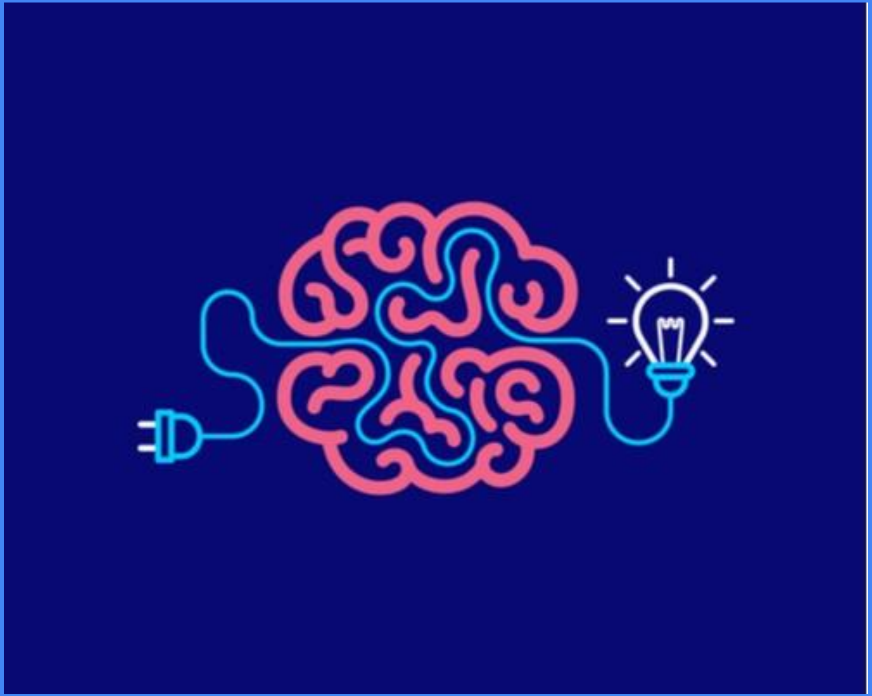
```
_________________________________________________________________
ayer (type)                 Output Shape              Param #
=================================================================
stm_1 (LSTM)                (None, 20)                1760
_________________________________________________________________
lense_1 (Dense)             (None, 4)                 84
=================================================================
'otal params: 1,844
'rainable params: 1,844
lon-trainable params: 0
_________________________________________________________________
lone
:poch 1/5
:019-04-26 15:03:01.026312: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your
 CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
'MA
00/300 [==============================] - 3s 10ms/step - loss: 0.8267 - acc: 0.3083
:poch 2/5
00/300 [==============================] - 2s 8ms/step - loss: 0.6900 - acc: 0.4533
:poch 3/5
00/300 [==============================] - 2s 8ms/step - loss: 0.6356 - acc: 0.6742
:poch 4/5
00/300 [==============================] - 2s 8ms/step - loss: 0.5989 - acc: 0.7017
:poch 5/5
00/300 [==============================] - 2s 8ms/step - loss: 0.5750 - acc: 0.7017
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 20)                1760
_________________________________________________________________
dense_1 (Dense)              (None, 20)                420
_________________________________________________________________
dense_2 (Dense)              (None, 15)                315
_________________________________________________________________
dense_3 (Dense)              (None, 10)                160
_________________________________________________________________
dense_4 (Dense)              (None, 4)                 44
=================================================================
Total params: 2,699
Trainable params: 2,699
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/5
2019-04-27 14:56:05.425934: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU
 supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
600/600 [==============================] - 5s 9ms/step - loss: 0.7159 - acc: 0.4567
Epoch 2/5
600/600 [==============================] - 5s 8ms/step - loss: 0.6325 - acc: 0.5512
Epoch 3/5
600/600 [==============================] - 5s 8ms/step - loss: 0.5808 - acc: 0.6988
Epoch 4/5
600/600 [==============================] - 5s 8ms/step - loss: 0.5491 - acc: 0.7532
Epoch 5/5
600/600 [==============================] - 5s 8ms/step - loss: 0.5287 - acc: 0.8034
```

AI Experiment


```
address: "408 Barber Ln, Milpitas"
▶ busy_hours: (7) [{…}, {…}, {…}, {…}, {…}, {…}, {…}]
dislikes: null
▶ distance: {distance: {…}, duration: {…}, durationInTraffic: {…}}
favorited: null
id: "8KayUocIVMfgfEuqP1Q-rw"
in_db: true
likes: null
▶ location: {lat: 37.4218596148519, lng: -121.916425418683}
name: "Pepper Lunch USA"
open_now: true
▶ photos: (10) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
place_id: "ChIJQfdhqCbJj4ARE9zXHEhMh-Q"
predicted_action: 2
predicted_rating: 3.4051642417907715
price: 2
rating: 3.5
▶ temperature: {temp: 292.59, pressure: 1015, humidity: 72, temp_min: 289.15, temp_max: 295.93}
```

Conclusion



Conclusion

- The simplicity of the app reduced the time taken to search for restaurants.
  - Typing keywords and searching take ~10 minutes
  - After loading a deck of restaurants, only take few seconds to swipe (3-5 minutes)
- With more data and time, reinforcement learning can really enhance the recommendation systems
  - Accuracy increased from 70% -> 80% from 500 records -> 1800 records

Future Work

- Other AI architectures can be experimented
    - More layers, other type of neural networks (DFF, RNN,etc.)
    - Other types of data
- Group functionality can be derived from individual’s preferences
    - When forming suggestions for a group, different lists of suggestions for individuals can be combined.

References

[1] Sawant, Sumedh, and Gina Pai. *Yelp Food Recommendation System - Machine Learning*. Stanford University, 2013, cs229.stanford.edu/proj2013/SawantPai-YelpFoodRecommendationSystem.pdf.

[2] Seo, Sungyong, et al. *Representation Learning of Users and Items for Review* ...University of Southern California, Visa Inc., 2017, pdfs.semanticscholar.org/4946/89f4522619b887e515aea2b205490b0eb5cd.pdf.

[3] Covington, Paul, et al. "Deep Neural Networks for YouTube Recommendations – Google AI." *Google AI*, Google Inc., 2016, ai.google/research/pubs/pub45530.

[4] Hasselt, Hado van, et al. "Deep Reinforcement Learning with Double Q-Learning - ArXiv." *Google DeepMind*, Google Inc., 2016, arxiv.org/pdf/1509.06461.pdf.

[5] "Google APIs." *Google Developers*, Google, developers.google.com/products/.

[6] "Yelp API." *Yelp Developers*, www.yelp.com/developers/.

[7] "Supplemental Nutrition Assistance Program (SNAP)." *Food and Nutrition Service*, Committee on Examination of the Adequacy of Food Resources and SNAP Allotments; Food and Nutrition Board; Committee on National Statistics; Institute of Medicine; National Research Council; Caswell JA, Yaktine AL, Editors. Supplemental Nutrition Assistance Program: Examining the Evidence to Define Benefit Adequacy. Washington (DC): National Academies Press (US); 2013, www.fns.usda.gov/snap/supplemental-nutrition-assistance-program-snap.

[8] OpenWeatherMap.org. "Current Weather and Forecast." *Open Weather Map*, openweathermap.org/.

[9] "Yelp Dataset Challenge." *Yelp Dataset*, www.yelp.com/dataset/challenge.

[10] How Many Test Users in a Usability Study? (n.d.). Retrieved from https://www.nngroup.com/articles/how-many-test-users/

Acknowledgement

- Advisor: Professor Chris Pollett
- Committee members:
  - Dr. Mike Wu
  - Dr. Kevin Montgomery
- The great testers

Demo

"A demo is worth a thousand pictures"

Thank You!