Data Science and Applications

Chatbot for Recipe Recommendation

Final Project: Report

Michael Khalil
mkhal120@uOttawa.ca

Gehad Abo Kamar
gabok033@uOttawa.ca

Khadija Taha
khesh072@uOttawa.ca

Youssef Metwally
ymetw027@uOttawa.ca

Click here to experience the Recipes' Agent

Check the github repo here

Check the colab notebook here

-Submitted To-

Professor
Arya Rahgozar

Teaching Assistant
Migao Wu

Methodology:

Having a good process ensures a comprehensive and repeatable method for conducting analysis. Project work can occur in several phases at once.

The iterative nature of the project lifecycle is intended to more closely portray a real project, in which aspects of the project move forward and may return to earlier stages as new information is uncovered and team members learn more about various stages of the project.

1) Problem Formulation:

Cooking is a hobby for some and a big problem for others. However, you can always use a helping hand for cooking. We must take advantage of living in the age of artificial intelligence and data science along with advanced technological solutions created for us to facilitate some of our daily tasks. With this help, we won't need to fall into an endless long line due to a single ingredient that is not available at home.

Many people nowadays suffer from many health conditions such as diabetes and obesity. So, in order to overcome this, the food we eat is the most important thing to think about. Also, we want to eat delicious food that does not conflict with what some think of eating healthy food.

The aim of this project is to help people who cook to identify a healthy and tasty recipe for cooking in the kitchen using the ingredients available at home and based on past users' ratings. Then, we provide them a step by step guide to help them cook.

2) Data Source:

Data scraped from "*allrecipes.com*" website using "*apify.com*" that scrapes recipes from Allrecipes with ingredients and directions. This easy-to-use actor gets an Allrecipes dataset in different formats according to our needs. We chose the json format.

3) Data Preparation:

We extracted the data into a data frame using the “recipe_scrapers” Python library. Here is a look at our data.

	title	total_time	prep	cook	yields	ingredients	instructions	calories	nutrients	rating
0	Breakfast Cups	30	10 mins	20 mins	18 item(s)	cooking spray, 18 refrigerated biscuits (unbak...	Preheat oven to 400 degrees F (200 degrees C)....	190	{'calories': '190.4 calories', 'carbohydrate Co...	4.268966
1	Easy Shakshuka	45	10 mins	35 mins	4 serving(s)	1 tablespoon olive oil, 2 cloves garlic, mince...	Heat the vegetable oil in a deep skillet over ...	294	{'calories': '293.5 calories', 'carbohydrate Co...	4.282609
2	Lemon-Ricotta Cornmeal Waffles	30	10 mins	20 mins	4 serving(s)	1 cup all-purpose flour, ½ cup cornmeal, ¼ cup...	Preheat a waffle iron according to manufacture...	406	{'calories': '406.1 calories', 'carbohydrate Co...	4.800000

	make_it_again	image	meal type	url
0	82	https://imagesvc.meredithcorp.io/v3/mm/image?u...	breakfast	https://www.allrecipes.com/recipe/233747/break...
1	2	https://imagesvc.meredithcorp.io/v3/mm/image?u...	breakfast	https://www.allrecipes.com/recipe/190276/easy-...
2	27	https://imagesvc.meredithcorp.io/v3/mm/image?u...	breakfast	https://www.allrecipes.com/recipe/274974/lemon...

The dataset contains different types of recipes such as desserts, breakfast, dinner, etc. In total, 999+ recipes are available in our dataset. Each recipe consists of the below features:

- Title

- Total_time
- Prep
- Cook
- Yields
- Ingredients
- Instructions
- Calories
- Nutrients
- Rating
- make_it_again ratio
- Image URL
- Meal Type
- Receipe URL

Before building the recommendation engine, we need to clean the data. We should clear the texts from the characters and words that we do not need. While we were scraping the data, some special characters came with that such as "[" among the words. We started to clean those characters. Also, we removed the stop words. We applied the following clearing operations for ingredients attribute that can be summarized as follows:

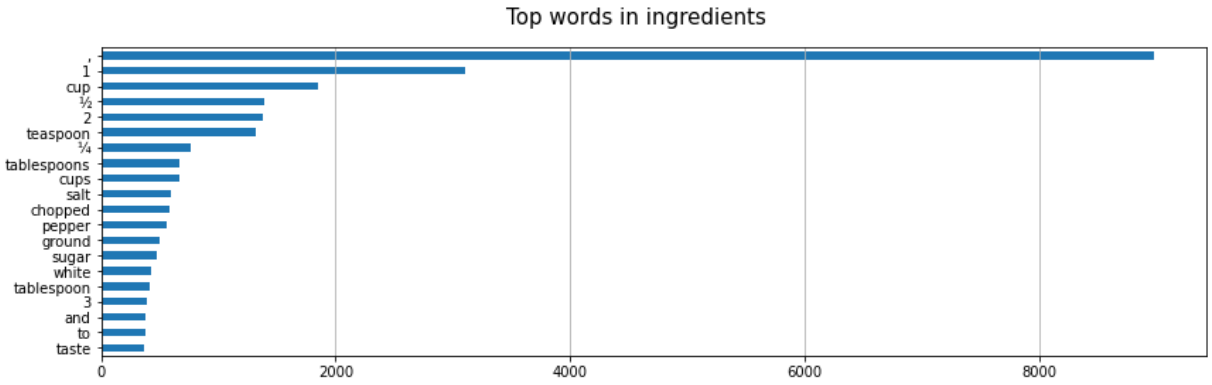
- Remove brackets and strings inside.
- Remove fractions such as '½' using regular expressions.
- Remove any number.
- Replace some words with different spelling such as:
    - 'ricottum', 'ricotta'
    - 'olife', 'olive'

    - 'pitum', 'pita'
    - 'flmy', 'flour'
- Remove extra stop words found in the data:
    - ['teaspoons', 'tablespoons', 'chopped', 'ground', 'minced', 'cups', 'extract', 'shredded', 'and', 'or', 'all-purpose', 'all purpose', 'allpurpose', 'cloves', 'to taste', 'slices', 'sliced', 'shredded', 'of', 'sizes', 'sized', 'large', 'small', 'big', ...
- Separate 'salt pepper' into two different ingredients 'salt' and 'pepper'.

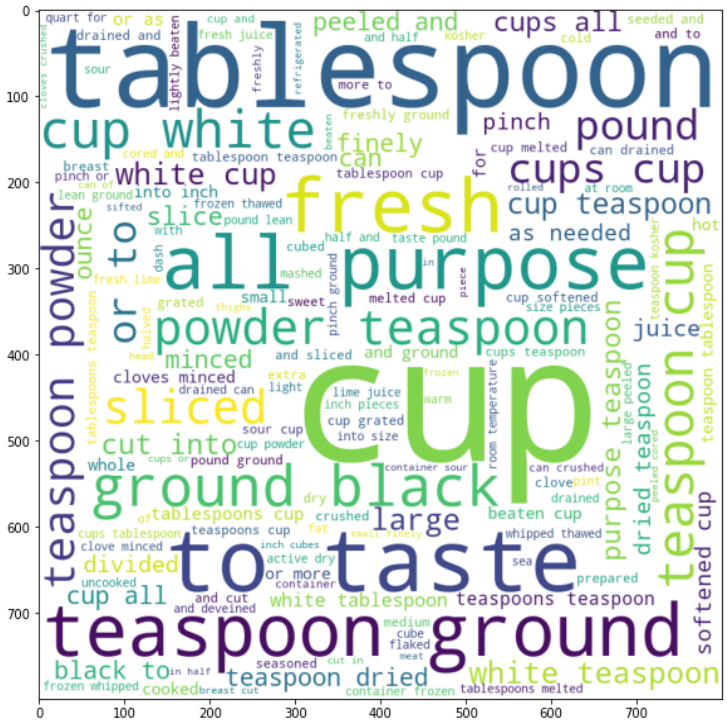
Here is a look at before and after doing the steps

Before

Top words



Example of words that we want to remove (stop words)



After

Before

	ingredients
0	[cooking spray, 18 refrigerated biscuits (unbaked), 8 ounces breakfast sausage, 7 large eggs, ½ cup milk, salt and ground black pepper to taste, 1 cup mild shredded Cheddar cheese]
1	[1 tablespoon olive oil, 2 cloves garlic, minc...
2	[1 cup all-purpose flour, ½ cup cornmeal, ¼ cu...

After

	ingredients filtered
0	[cooking spray, egg, milk, cheddar cheese, sal... cooking spray,egg,milk,cheddar cheese,salt,pepper
1	[olive oil, garlic, onion, bell pepper, tomato...
2	[flour, cornmeal, sugar, baking, baking soda, ...

Let’s have a look at some brief information about the most frequent ingredients in our dataset.

salt: 560

sugar: 398

pepper: 330

egg: 302

butter: 277

flour: 257

garlic: 228

onion: 203

milk: 199

vanilla: 142

olive oil: 141

water: 141

baking: 100

vegetable oil: 92

cinnamon: 86

mayonnaise: 79

bell pepper: 76

cayenne pepper: 69

baking soda: 67

cheddar cheese: 65

It appears that the most frequent ingredient is salt! It isn't unexpected.

The previous work done was for the ingredients attribute only. For the instructions attribute, we did the same cleaning operations done for ingredients as well as the following:

- Remove extra stop words found in the data:
    - ['.', ',', 'and', 'the', 'a', 'in', 'to','until','with','minutes','of', 'into', 'degrees','bowl','over','for', 'oven','mixture','heat', ';']
- Apply stemming algorithm: PorterStemmer.

4) Data Transformation:

For ingredients, we collected all the ingredients individually to form a vocabulary. For the vocabulary we applied the following:

- TFIDF
- CountVectorizer
- Word2Vec

The aim is to find similarities between individual ingredients

For recipes, we applied the following transformations for the list of ingredients for each recipe:

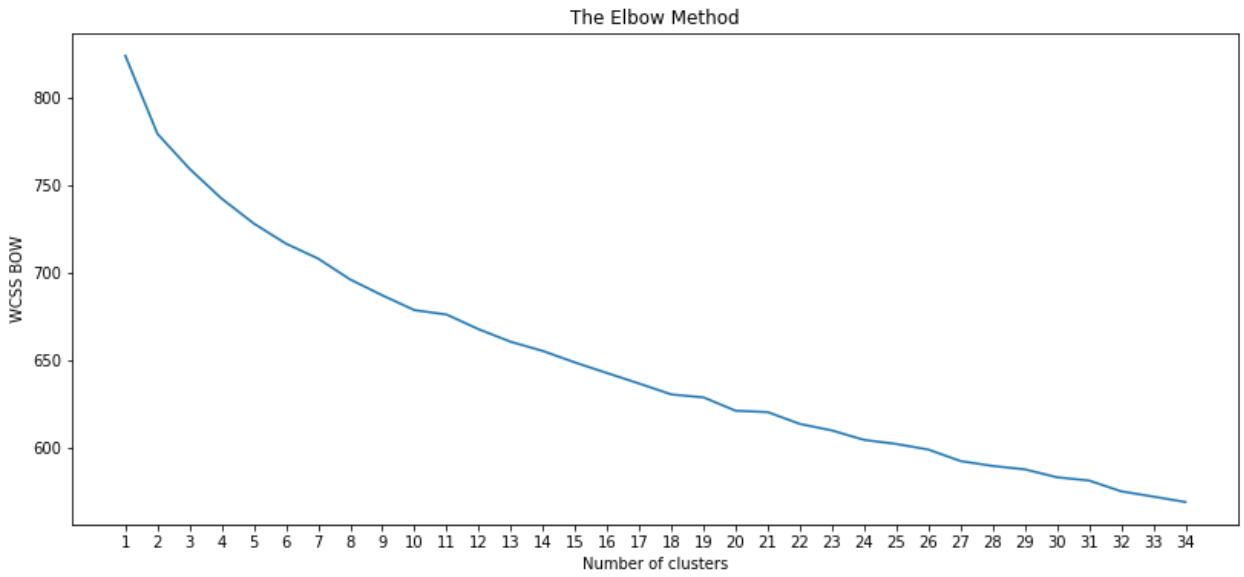
- TFIDF
- CountVectorizer

The aim is to find similarities between recipes through ingredients. Or classify recipes to Meal types based on the ingredients of the recipe

Also, we transformed the instructions data using TFIDF, and Doc2Vec transformation method.

5) Clustering:

We built a classification model using K-Means classification algorithm to classify the ingredients. This would help the user in case he doesn't have a specific ingredient, we can recommend him an ingredient that is similar to the missing ingredient. First, we applied the algorithm on the data transformed by TF-IDF. We chose the number of clusters to be 10.

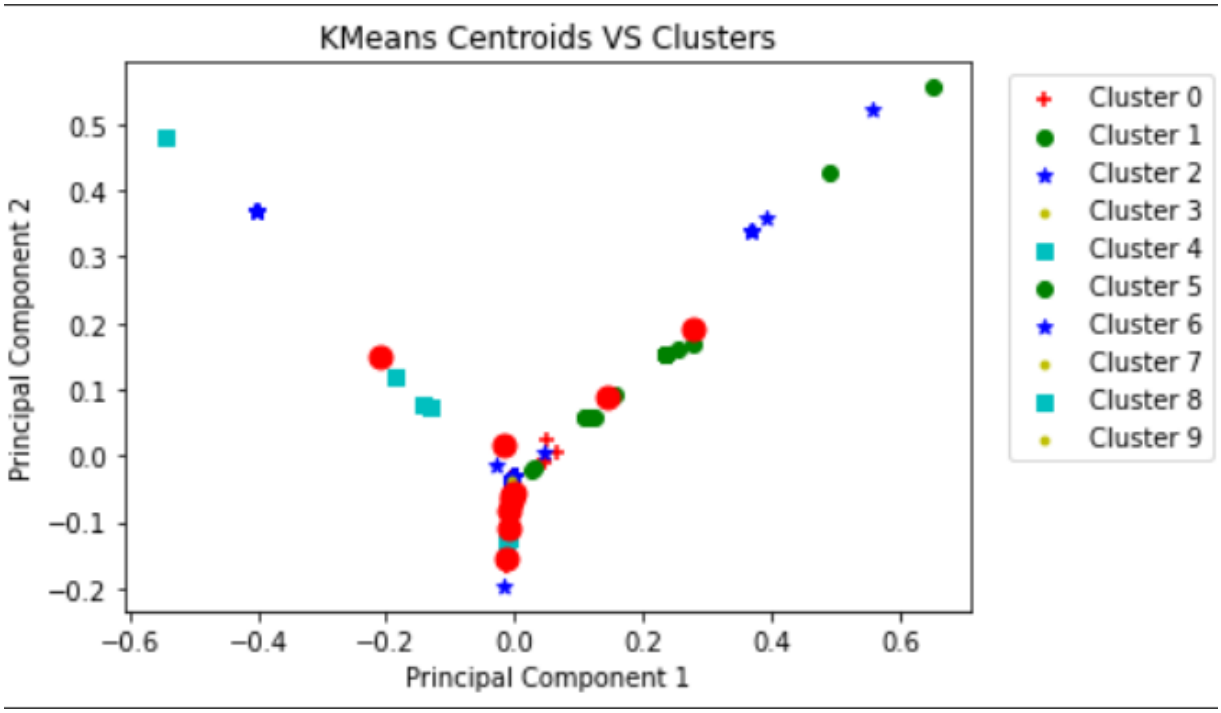


For example, the ingredients in the first resulted cluster (cluster 0) are:

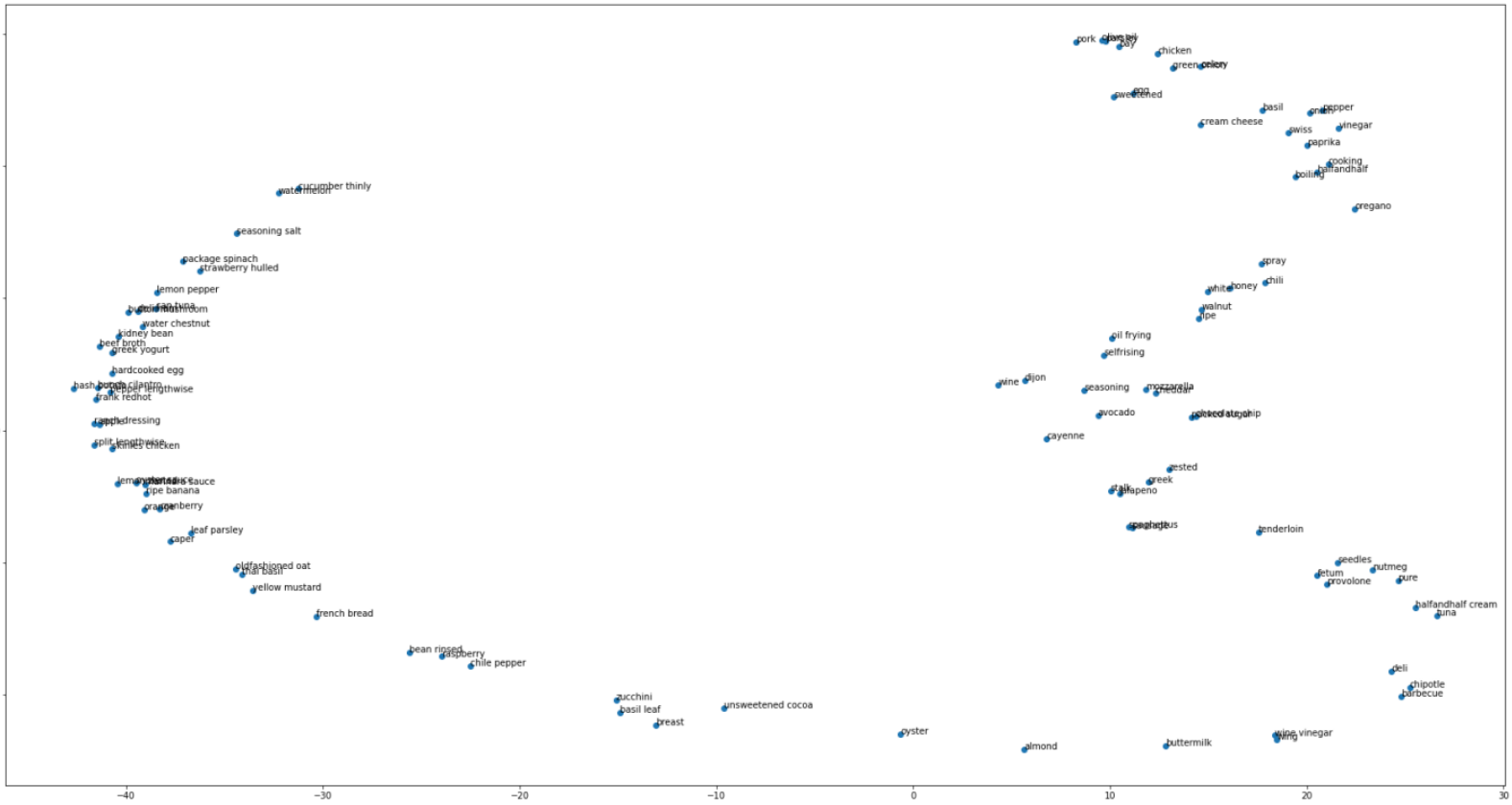
data_index	cluster	
2	milk	0
5	pepper	0
9	bell pepper	0
11	paprika	0
12	jalapeno pepper	0
28	freshly pepper	0
29	cayenne pepper	0
35	cream	0
42	water	0

64	worcestershire sauce	0
70	cream tartar	0
76	pepper flake	0
82	fish sauce	0
98	halfandhalf cream	0
103	boiling water	0
113	smoked paprika	0
126	quart water	0
143	soy sauce	0
147	coconut	0
149	heavy cream	0
189	whipping cream	0
194	spaghettus sauce	0
195	water chestnut	0
218	oyster sauce	0
221	pepper sauce	0
222	chipotle pepper	0
224	barbecue sauce	0
228	barbeque sauce	0
229	chili sauce	0
235	chile pepper	0
241	marinara sauce	0
242	pepper lengthwise	0
248	coconut milk	0
250	sweetened coconut	0
255	sweetened milk	0

We applied Principal component analysis (PCA) as a dimensionality reduction algorithm in order to be able to visualize the clusters in 2D.



For better visualization, we applied the T-SNE dimensionality reduction with Word2Vec algorithm along with the most common vocabulary used in the ingredients data to visualize it.

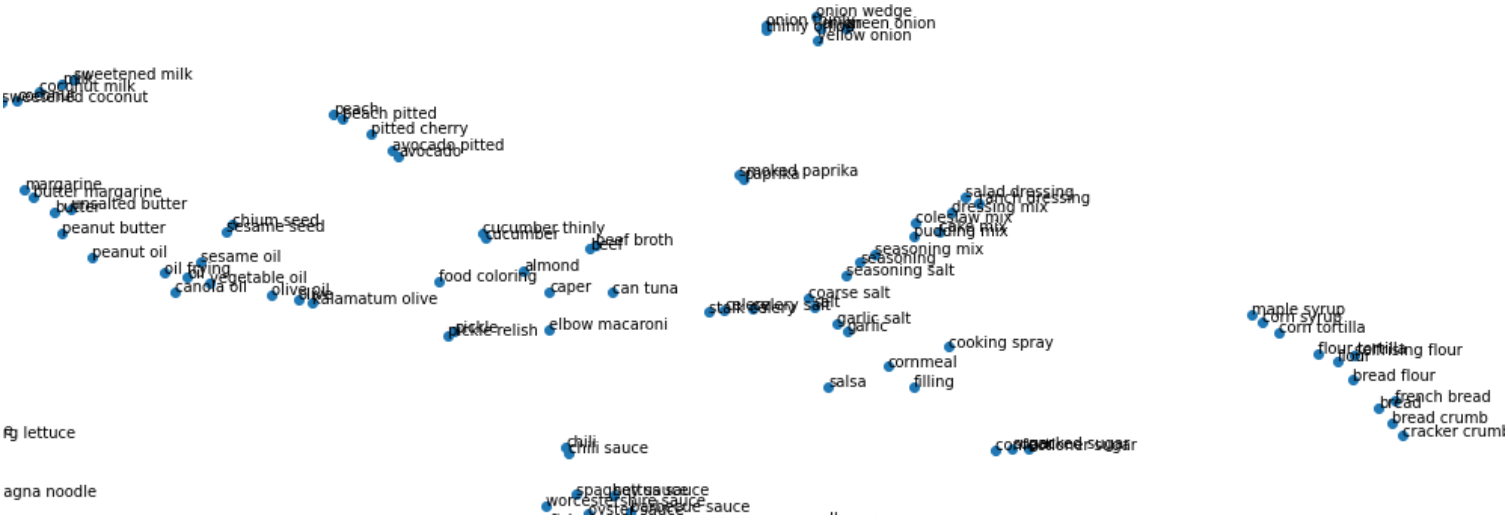


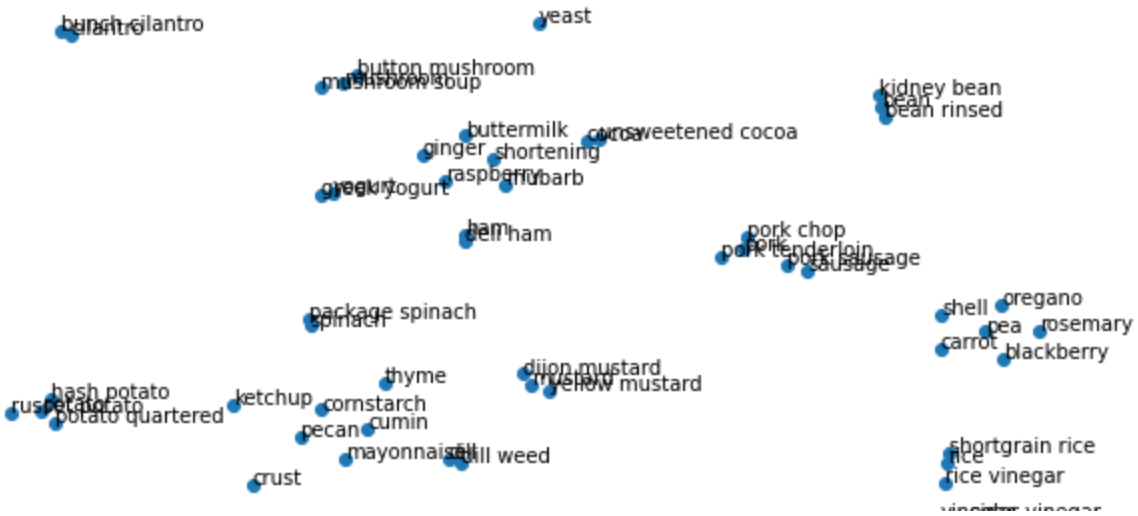
Some CloseUps

We noticed that the ingredients that could be paired together are close to each other. For example, notice that *chicken* is close to *honey* and *thyme*. It could help in suggesting pairings

Another T-SNE graph but with CountVectorizer

Some ClosUps

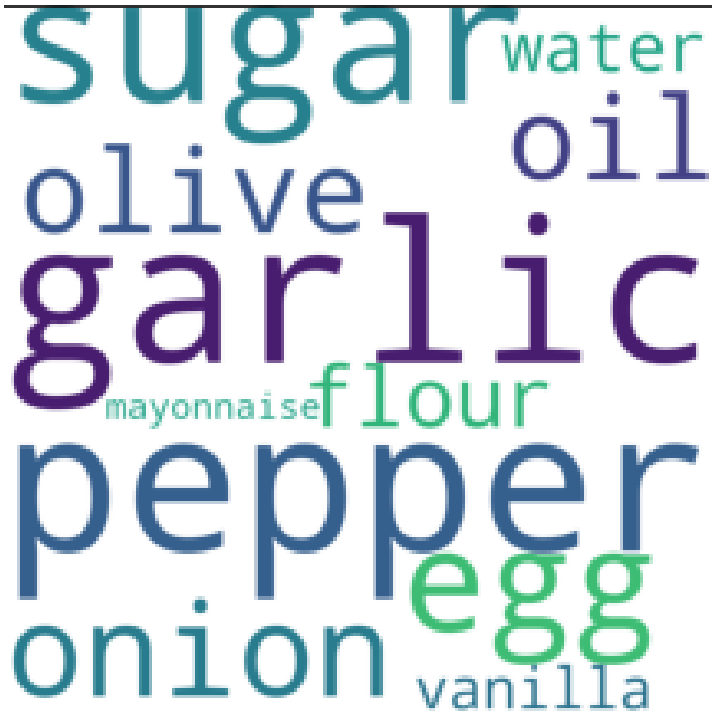




Here it looks like ingredients that are similar to others are close. Notice that different kinds of *potatoes* are clustered together. Same with *mustard, chickens, onions, or chilies.* This could help in suggesting replacements.

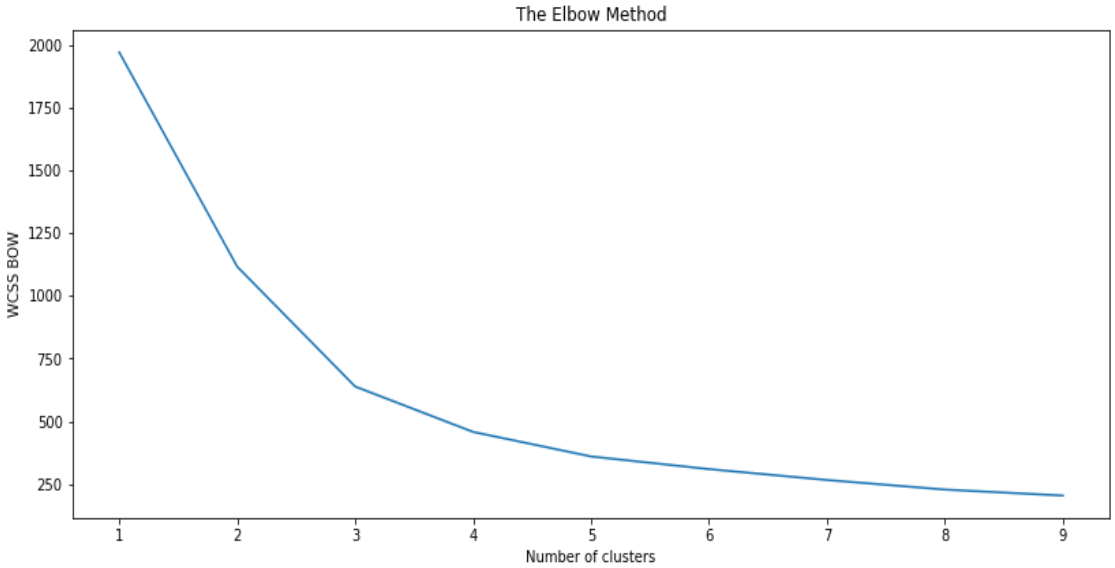
We also could use the help of Word2Vec to find the most similar ingredients to an ingredient in the vocabulary.

For example, to find the most similar ingredient to salt, the result will be:



- [('cumin', 0.9996896982192993),
- ('olive oil', 0.9996840357780457),
- ('water', 0.9996800422668457),
- ('worcestershire sauce', 0.9996656179428101),
- ('mayonnaise', 0.9996472597122192),
- ('honey', 0.9996466040611267),
- ('cayenne pepper', 0.9996353387832642),
- ('oregano', 0.9996302127838135),
- ('oil frying', 0.9996263384819031),
- ('beef', 0.9996222853660583)]

Second, we will apply the K-Means on the **recipes** that are transformed by the CountVectorizer method and PCA transformation.



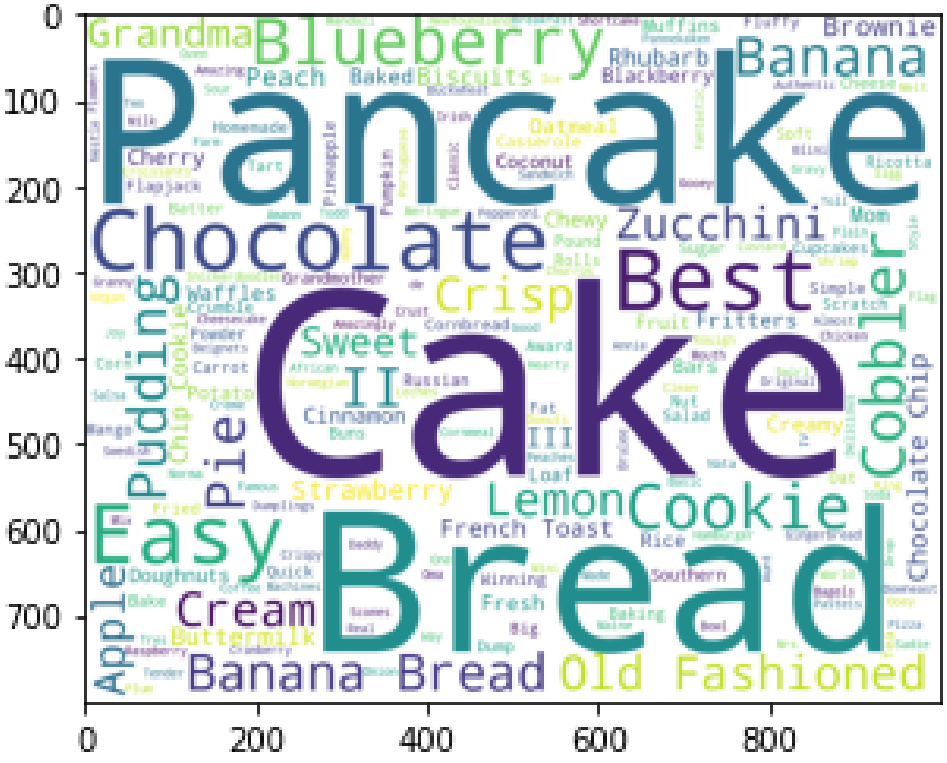
Let us take a look at the data in the resulting cluster 1.

- Mean num of ingredients: 20
  - sugar: 236
  - flour: 168
  - salt: 158
  - egg: 140
  - butter: 125
  - vanilla: 98
  - milk: 94
  - baking: 87
  - baking soda: 67
  - cinnamon: 56
  - buttermilk: 13

Recipes

2	Lemon-Ricotta Cornmeal Waffles
9	Mom's Apple Fritters
10	Good Old Fashioned Pancakes
11	Farm Fresh Zucchini Cranberry Nut Muffins
12	Waffles I
...	...
916	Award Winning Soft Chocolate Chip Cookies
917	Air Fryer Strawberry Crisp
918	Smith Island Cake
919	Coconut Cake with Crushed Pineapple
920	Grandma's Old Fashioned Tea Cakes

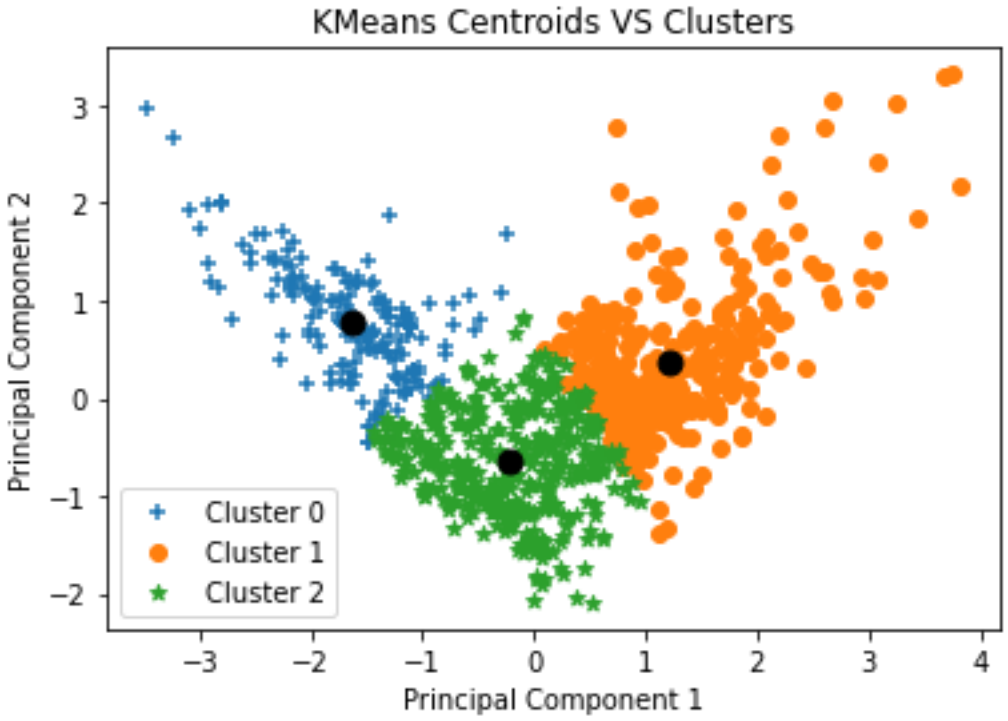
Common Words in recipes



Here is a look at common words in recipes' names in clusters 2 and 3 respectively

We can see that the first cluster is about cakes and baking. But the distinction about the other two clusters are not as clear. They are big on chicken salads we guess.

We visualized the result using PCA.



6) Recommendation Engine:

We used "pairwise_distances_argmin_min" Python metric to measure the distance among ingredients of different recipes to recommend a recipe based on the ingredients that the user provides.

For example, if the user likes **pasta and tomato**, the system recommends:

“Emily’s Excellent Taco Casserole”

'cheddar cheese,lettuce,tomato,salsa,cream'

OR

“Creamy Pasta Bake with Cherry Tomatoes and Basil”

'olive oil,onion,garlic,tomato sauce,tomato paste,whipping cream,parmesan cheese,sugar,tomato,mozzarella cheese,salt,pepper'

Here is a look at the code output with all the details:

```
["Emily's Excellent Taco Casserole" 35 '10 mins' '25 mins' '4 serving(s)'
 list(['6 cups corn tortilla chips', '2 cups vegetarian chili with beans', '1
cup shredded Cheddar cheese', '2 cups shredded lettuce', '2 roma (plum)
tomatoes, chopped', '½ cup salsa', '¼ cup sour cream'])
 '6 cups corn tortilla chips, 2 cups vegetarian chili with beans, 1 cup
shredded Cheddar cheese, 2 cups shredded lettuce, 2 roma (plum) tomatoes,
chopped, ½ cup salsa, ¼ cup sour cream'
 'Preheat the oven to 350 degrees F (175 degrees C).\nPlace chips in the bottom
of a 9 inch square baking dish. Pour chili straight from the can over the
chips. Sprinkle shredded cheese over the top.\nBake for 20 to 25 minutes in the
preheated oven, until chili is bubbling and cheese is melted. Top with lettuce,
tomato, sour cream and salsa in the pan, or after serving (if people are
picky).'
 '477'
 {'calories': '476.9 calories', 'carbohydrateContent': '48.8 g',
'cholesterolContent': '42.5 mg', 'fatContent': '24 g', 'fiberContent': '7.9 g',
'proteinContent': '18.9 g', 'saturatedFatContent': '10.3 g', 'sodiumContent':
'972.4 mg', 'sugarContent': '5.4 g'}
 4.2957746479 31

'https://imagesvc.meredithcorp.io/v3/mm/image?url=https%3A%2F%2Fimages.media-al
lrecipes.com%2Fuserphotos%2F1411575.jpg'
 'dinner'
 'https://www.allrecipes.com/recipe/11679/homemade-mac-and-cheese/'
 list(['cheddar cheese', 'lettuce', 'tomato', 'salsa', 'cream', 'cheddar',
'cheese', 'cheddar', 'cheese', 'cheddar', 'cheese'])
 'cheddar cheese,lettuce,tomato,salsa,cream' 2]
```

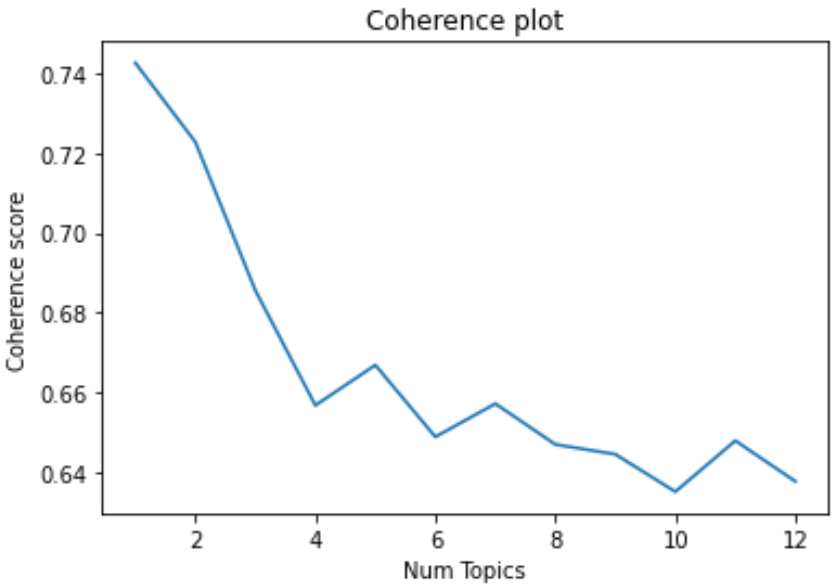
```
['Creamy Pasta Bake with Cherry Tomatoes and Basil' 51 '20 mins' '31 mins'
 '6 serving(s)'
 list(['1 (16 ounce) package penne pasta', '1 tablespoon olive oil', '1 onion,
finely chopped', '3 cloves garlic, minced', '3 (6 ounce) cans tomato sauce', '2
tablespoons tomato paste', '¾ cup heavy whipping cream', '½ cup grated Parmesan
cheese', 'salt and freshly ground black pepper', '1 pinch white sugar', '1
pound cherry tomatoes, halved', '1 ¼ cups shredded mozzarella cheese', '1 small
bunch fresh basil, finely chopped'])
 '1 (16 ounce) package penne pasta, 1 tablespoon olive oil, 1 onion, finely
chopped, 3 cloves garlic, minced, 3 (6 ounce) cans tomato sauce, 2 tablespoons
tomato paste, ¾ cup heavy whipping cream, ½ cup grated Parmesan cheese, salt
and freshly ground black pepper, 1 pinch white sugar, 1 pound cherry tomatoes,
halved, 1\u2009¼ cups shredded mozzarella cheese, 1 small bunch fresh basil,
finely chopped'
 'Bring a large pot of lightly salted water to a boil. Add penne and cook,
stirring occasionally, until tender yet firm to the bite, about 11 minutes.
Drain, reserving 1 cup of cooking water.\nHeat olive oil in a large skillet
over medium heat and cook onion until soft and translucent while penne is
cooking, about 5 minutes. Add garlic and cook an additional 30 seconds. Stir in
tomato sauce and tomato paste and cook until slightly reduced, about 5 minutes.
```

Add cream and Parmesan cheese and season with salt, pepper, and sugar.\nPreheat oven to 400 degrees F (200 degrees C). Grease a baking dish.\nStir some pasta cooking water into the sauce and add cooked penne. Remove from heat and stir in cherry tomatoes, 1/2 the mozzarella cheese, and basil. Pour penne mixture into the prepared baking dish and cover with remaining mozzarella cheese.\nBake in the preheated oven until cheese is melted, about 20 minutes.'
 '532'
 {'calories': '531.6 calories', 'carbohydrateContent': '67.8 g', 'cholesterolContent': '61.7 mg', 'fatContent': '21.2 g', 'fiberContent': '5.3 g', 'proteinContent': '21.4 g', 'saturatedFatContent': '11.1 g', 'sodiumContent': '779.8 mg', 'sugarContent': '7.8 g'}
 4.7666666667 46

'https://imagesvc.meredithcorp.io/v3/mm/image?url=https%3A%2F%2Fimages.media-allrecipes.com%2Fuserphotos%2F4573350.jpg'
 'dinner' 'https://www.allrecipes.com/recipe/244351/indian-tacos/'
 list(['olive oil', 'onion', 'garlic', 'tomato sauce', 'tomato paste', 'whipping cream', 'parmesan cheese', 'sugar', 'tomato', 'mozzarella cheese', 'salt', 'pepper', 'olive', 'oil', 'tomato', 'sauce', 'tomato', 'paste', 'whipping', 'cream', 'parmesan', 'cheese', 'mozzarella', 'cheese', 'olive', 'oil', 'tomato', 'sauce', 'tomato', 'paste', 'whipping', 'cream', 'parmesan', 'cheese', 'mozzarella', 'cheese', 'olive', 'oil', 'tomato', 'sauce', 'tomato', 'paste', 'whipping', 'cream', 'parmesan', 'cheese', 'mozzarella', 'cheese'])
 'olive oil,onion,garlic,tomato sauce,tomato paste,whipping cream,parmesan cheese,sugar,tomato,mozzarella cheese,salt,pepper'
 1]

To evaluate the clustering performance, we calculated the Silhouette Coefficient and it's found to be (0.17660) which is relatively low.

Also, we plotted the Coherence plot:



7) Classification:

In addition to the previous work, we built a classification model to help the user by telling him the type of the meal he/she wants to know.

We splitted the data into 80% training data and 20% testing data. We used, SGD, SVM, Decision Tree, K-Nearest Neighbor as classification algorithms for the experimentation. Then, we built a Pipeline to make the code more efficient, organized, and reusable.

To evaluate the model performance, we applied cross-validation and got the champion model to be SVM.

```
Accuracy: 67.04%
-------------------------------------
Accuracy: 62.01%
-------------------------------------
Accuracy: 50.62%
-------------------------------------
Accuracy: 67.31%
-------------------------------------
```

Then, we tuned its hyper-parameters and got the following results.

```
Training Accuracy: 0.9131614654002713
Mean cross-validation Accuracy: 0.6893372824879674

Best Parameters
----------------
clf__C: 1
clf__gamma: 0.001
clf__kernel: 'linear'
tran1__ngram_range: (1, 1)
tran2__use_idf: True

Training Report
-------
              precision    recall  f1-score   support

   breakfast       0.89      0.92      0.90       152
     dessert       0.95      0.97      0.96       152
      dinner       0.89      0.95      0.92       151
       lunch       0.93      0.88      0.91       130
       snack       0.91      0.83      0.87       152

    accuracy                           0.91       737
```

```
   macro avg       0.91      0.91      0.91       737
weighted avg       0.91      0.91      0.91       737
Testing Report
-------


              precision    recall  f1-score   support

   breakfast       0.65      0.65      0.65        34
     dessert       0.79      0.88      0.84        43
      dinner       0.58      0.81      0.68        32
       lunch       0.76      0.57      0.65        46
       snack       0.50      0.40      0.44        30

    accuracy                           0.67       185
   macro avg       0.66      0.66      0.65       185
weighted avg       0.67      0.67      0.66       185
```

We provided below a sample input and the predicted output to test our classification model.

Input:

```
['egg,milk',
 'chicken',
 'oil,chicken,salt,pepper,cheese',
 'bacon,toast,american cheese',
 'bacon,lettuce,tomato',                    # BLT
 'sandwich',
 'chip',
 'butter,sugar',
 'dough,pepproni,tomato sauce,mozzarella'] # pizza
```

Output:

```
['breakfast', 'dinner', 'dinner', 'breakfast', 'breakfast', 'snack', 'snack',
'dessert', 'dinner']
```

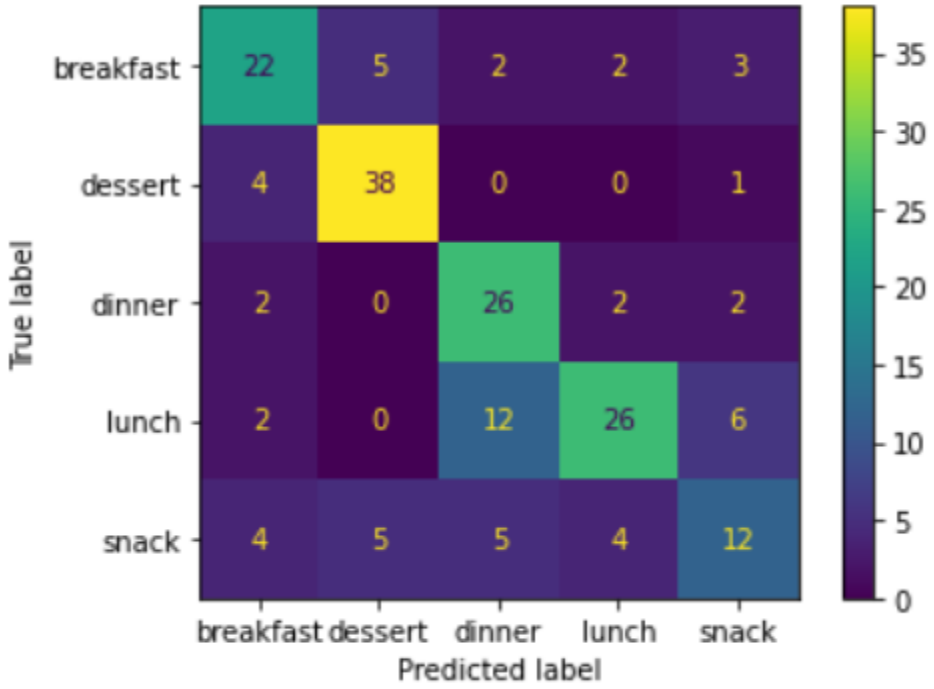
Which are really great results.

8) Chat Bot Evaluation:

- Clusters supported:
    - This is the number of question and answer pairs you're capable of responding to.
    -
- Questions per cluster:
    - For each cluster or topic, these are the variations of questions our chatbot is able to answer.
- Accuracy rate:
    - Within those known clusters and questions, how often is the bot giving a helpful and correct answer? We recommend aiming for 80% accuracy to be top quality

9) Error Analysis:

To make error analysis we must first know when the model predict wrong so we plot confusion matrix then false positive and false negative



12 texts has been wrongly classified as breakfast

10 texts has been wrongly classified as lunch

19 texts has been wrongly classified as dinner

8 texts has been wrongly classified as snack

12 texts has been wrongly classified as dessert

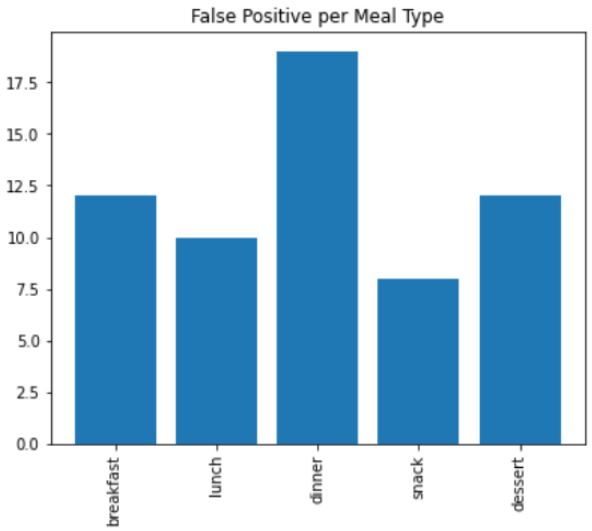
12 of breakfast texts has been wrongly classified

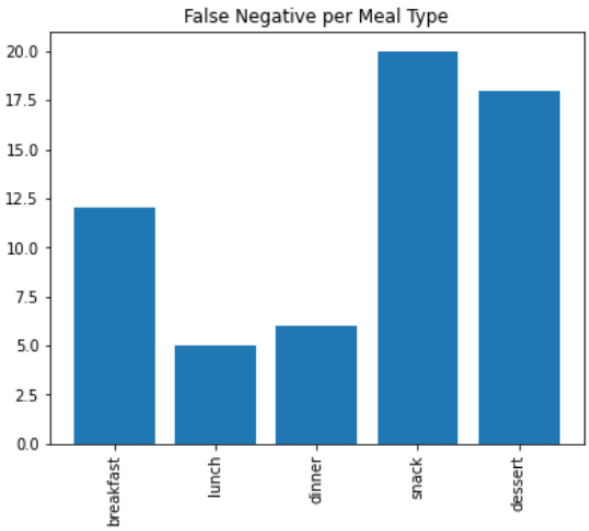
5 of lunch texts has been wrongly classified

6 of dinner texts has been wrongly classified

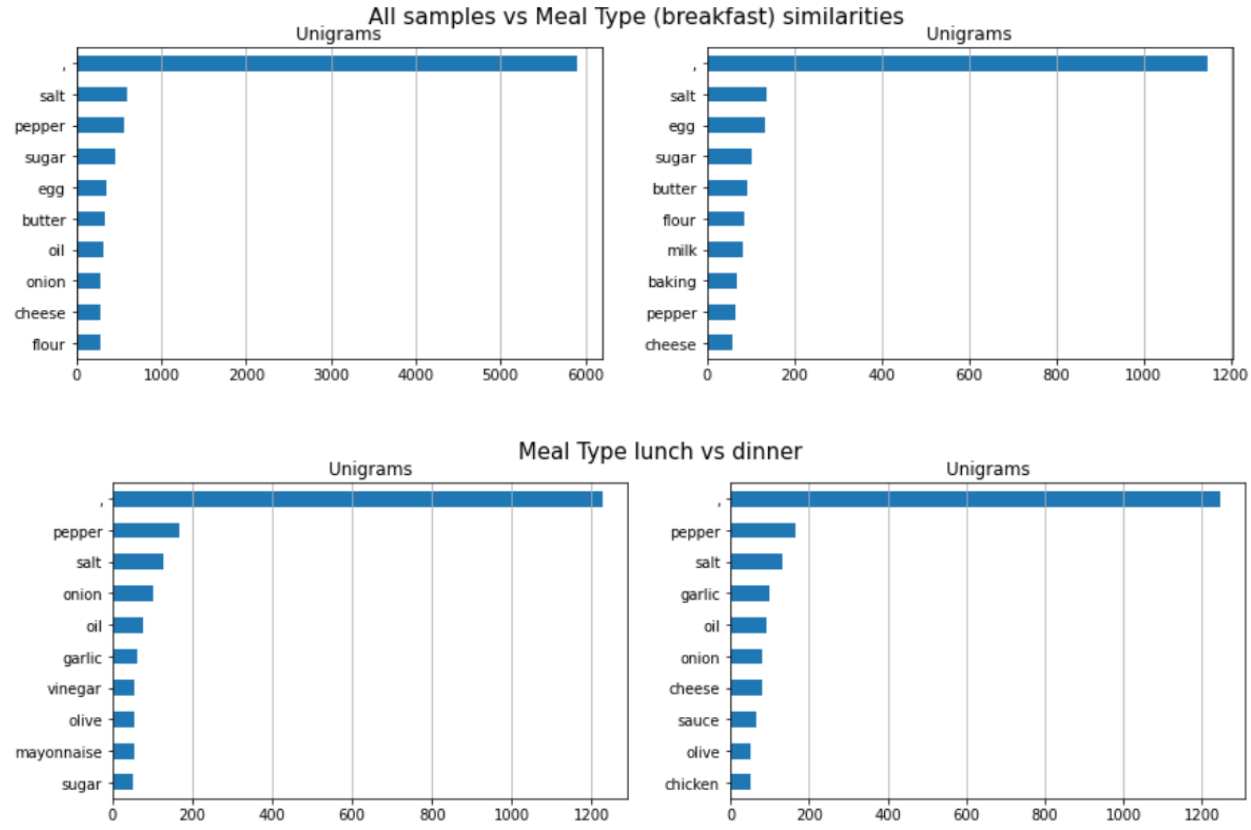
20 of snack texts has been wrongly classified

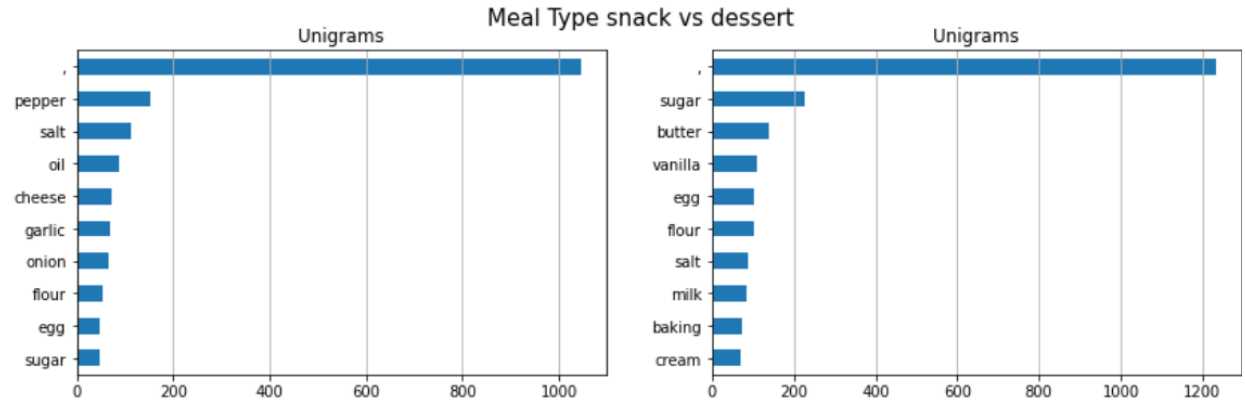
18 of dessert texts has been wrongly classified





Then we plotted the most frequent Ingredients that each meal consists of.





As we can see that there are very similarity from the top Ingredients for each meal like egg and salt. So the model predicted wrong.

Let's see the samples that threw off the model for further analysis.

Easy Homemade Pastrami ['garlic', 'vegetable oil', 'smoked paprika', 'mustard', 'pepper']... classified as dinner should be lunch

Sweet and Savory Slow Cooker Pulled Pork ['beer', 'sugar', 'salt', 'pepper', 'paprika']... classified as dinner should be lunch

Cheesecake-Stuffed Strawberries ['cream cheese', 'confectioner sugar', 'vanilla', 'strawberry hulled']... classified as dessert should be snack

Easy Batter Fruit Cobbler ['butter', 'flour', 'sugar', 'baking', 'salt']... classified as breakfast should be dessert

B and L's Strawberry Smoothie ['strawberry hulled', 'yogurt', 'sugar', 'vanilla']... classified as dessert should be breakfast

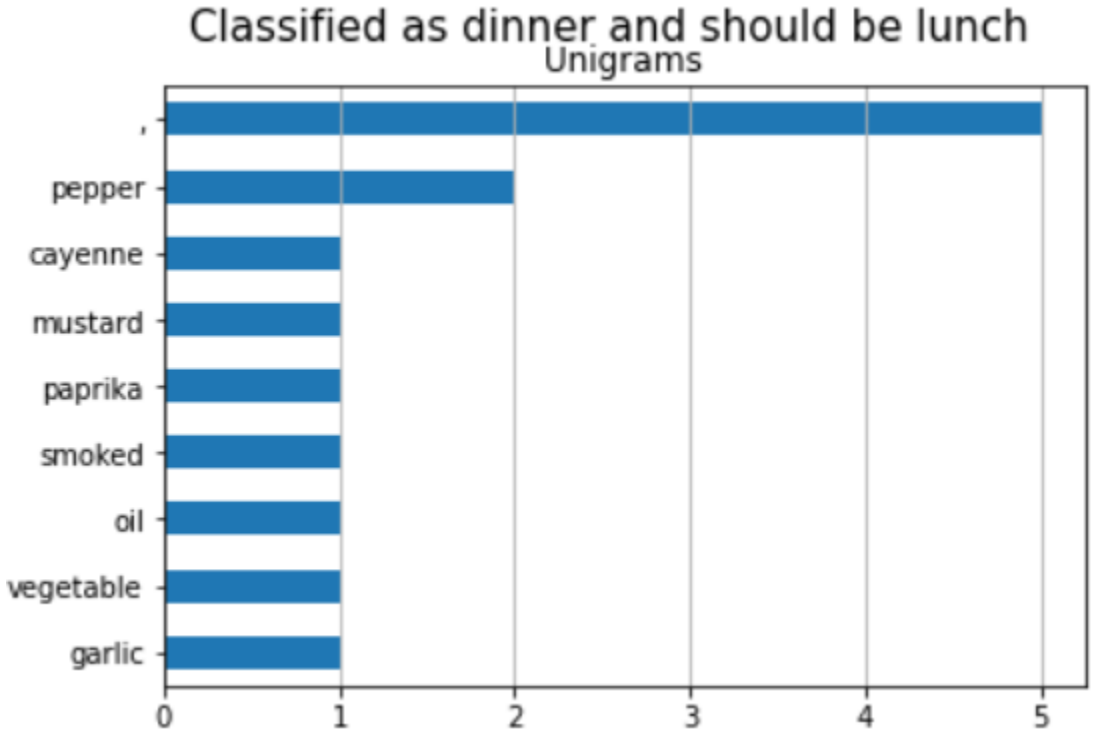
Baby Spinach Omelet ['egg', 'parmesan cheese', 'onion', 'nutmeg', 'salt']... classified as snack should be breakfast

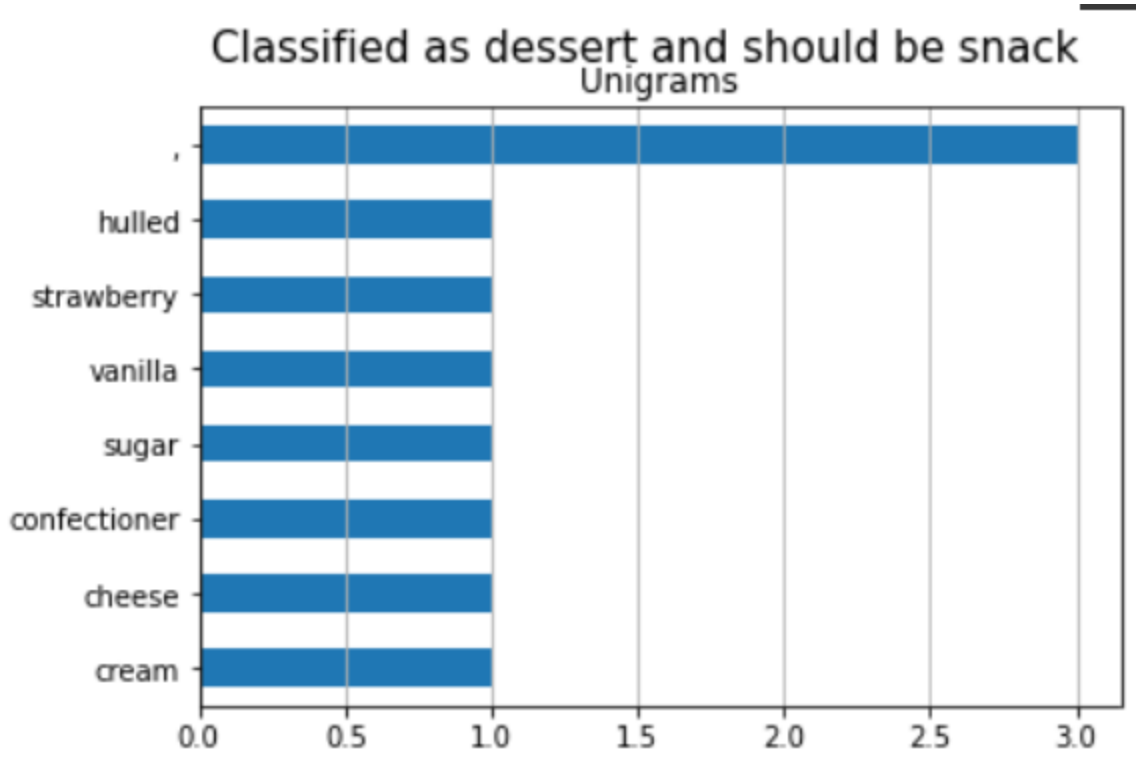
The World's Greatest Crab Recipe ['olive oil', 'butter', 'garlic']... classified as snack should be lunch

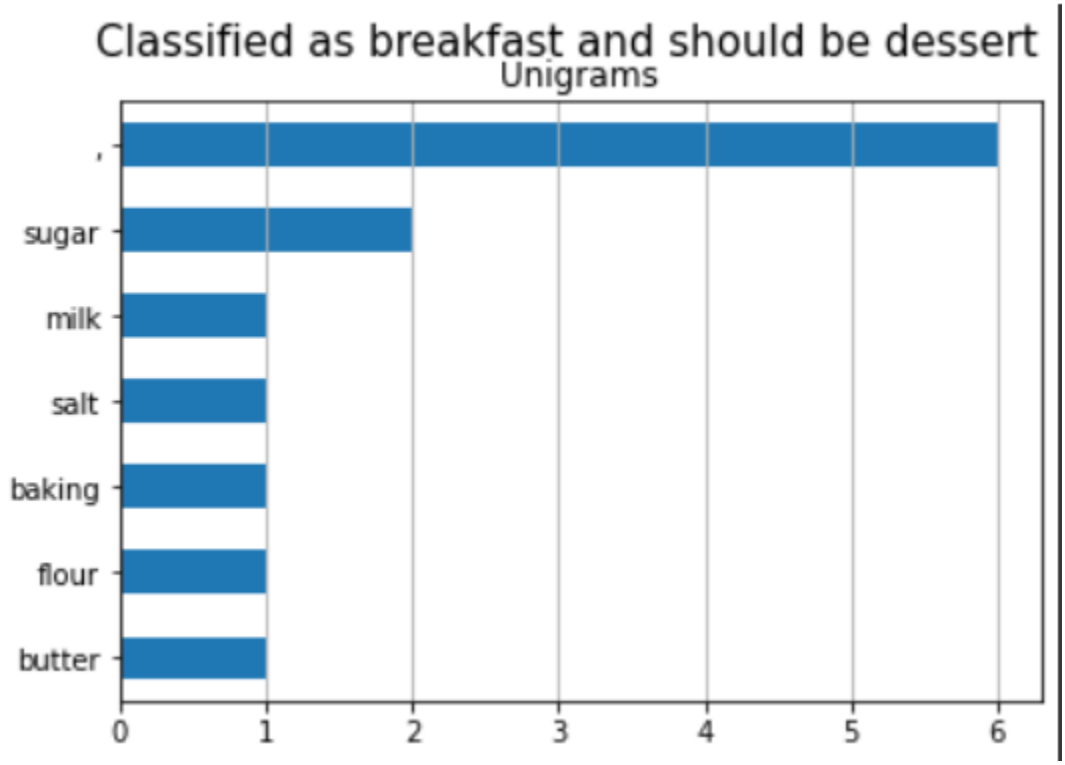
Thai Chicken Spring Rolls ['ginger', 'garlic', 'soy sauce', 'chicken half', 'peanut oil']... classified as dinner should be lunch

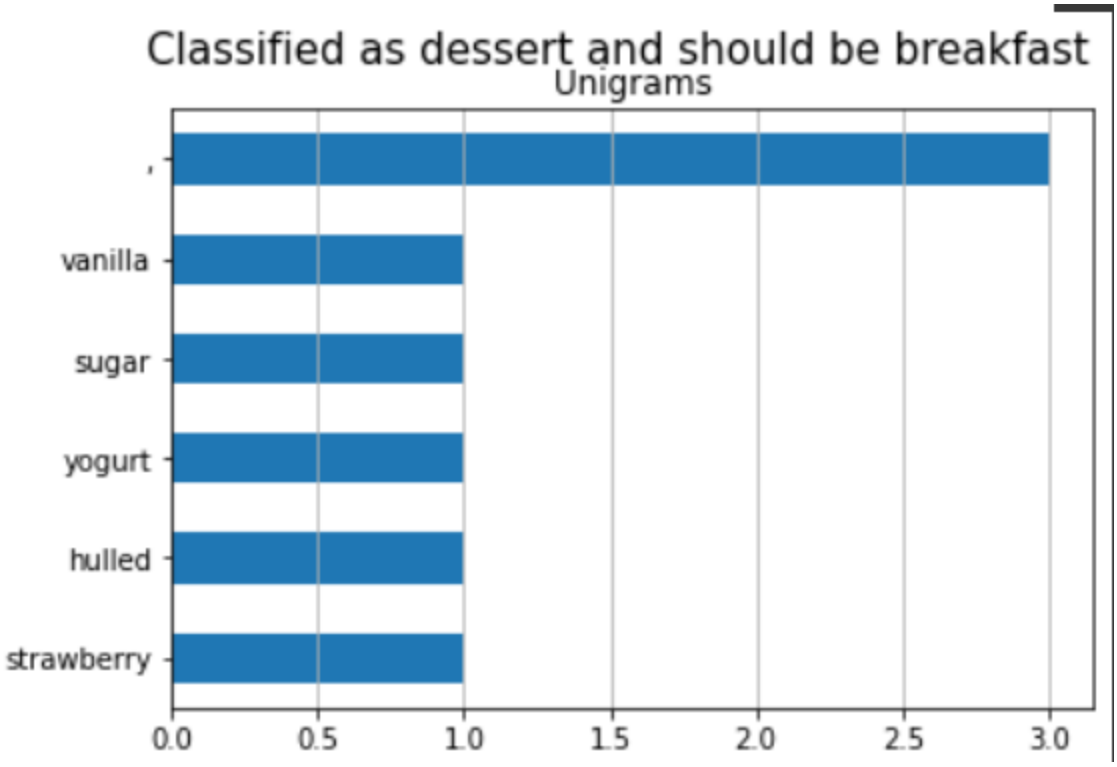
Taco Slaw ['cabbage', 'jalapeno pepper', 'onion', 'carrot', 'cilantro']... classified as snack should be lunch

Ahi Sushi Cups	['water', 'rice vinegar', 'sugar', 'cream cheese', 'salt']...	classified as dinner	should be lunch
Southern Fried Apples	['butter', 'sugar', 'cinnamon']...	classified as dessert	should be breakfast




Classified as dessert and should be snack
Unigrams
,
hulled
strawberry
vanilla
sugar
confectioner
cheese
cream
0.0
0.5
1.0
1.5
2.0
2.5
3.0


Classified as breakfast and should be dessert
Unigrams
,
sugar
milk
salt
baking
flour
butter
0
1
2
3
4
5
6



10) Deployment:

Competency Questions

Competency question example:

1) what needed to cook Waffles I?

2) How to prepare to make a Breakfast Cups?

3) can you recommend meals with garlic or onion?

4) I'd love pasta and milk

We can save the models objects as a binary files using "Pickle" Python library to be loaded for later use in the chat bot building process.

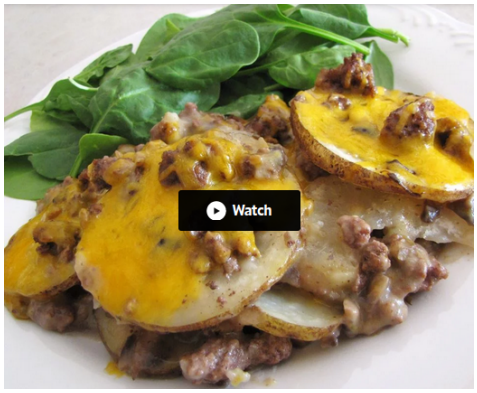
We created a backend with the help of *flask* and *replit.com* to use as webhook

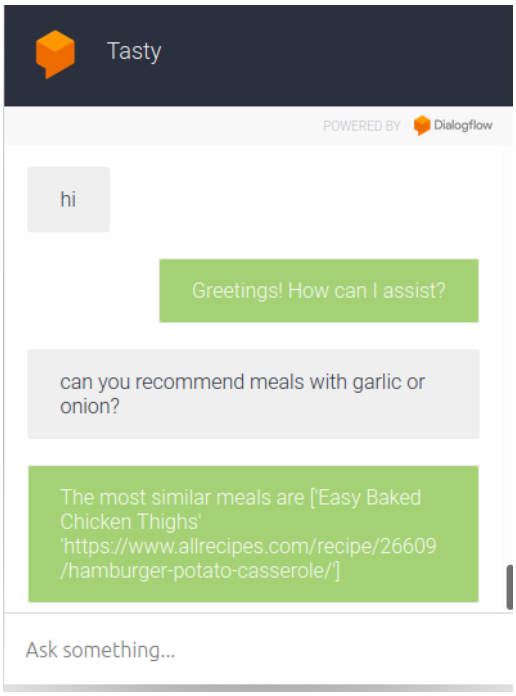
Questions per cluster.

For each cluster or topic, how many variations of questions is your chatbot able to answer?

For each cluster or topic, how many variations of questions is your chatbot able to answer?

Tasty can combine between related items in one cluster.

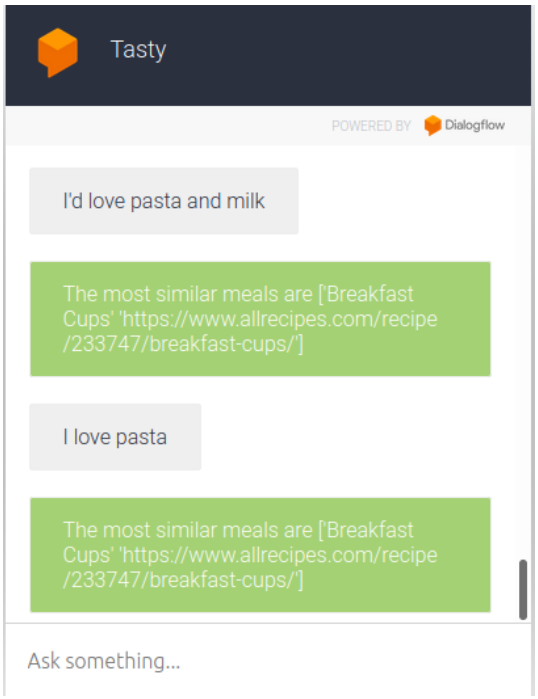




Clusters supported.

This is the number of question and answer pairs you’re capable of responding to.

Tasty chat bot isn't confused in misleading typing, as it's able to split between incorrectly concatenated words.

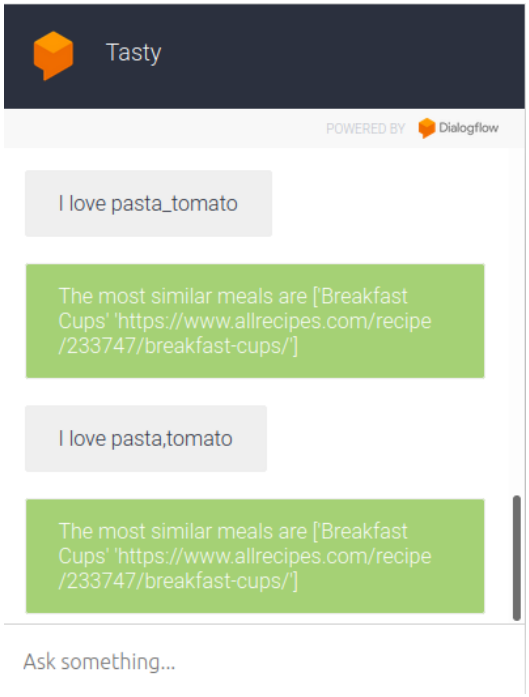


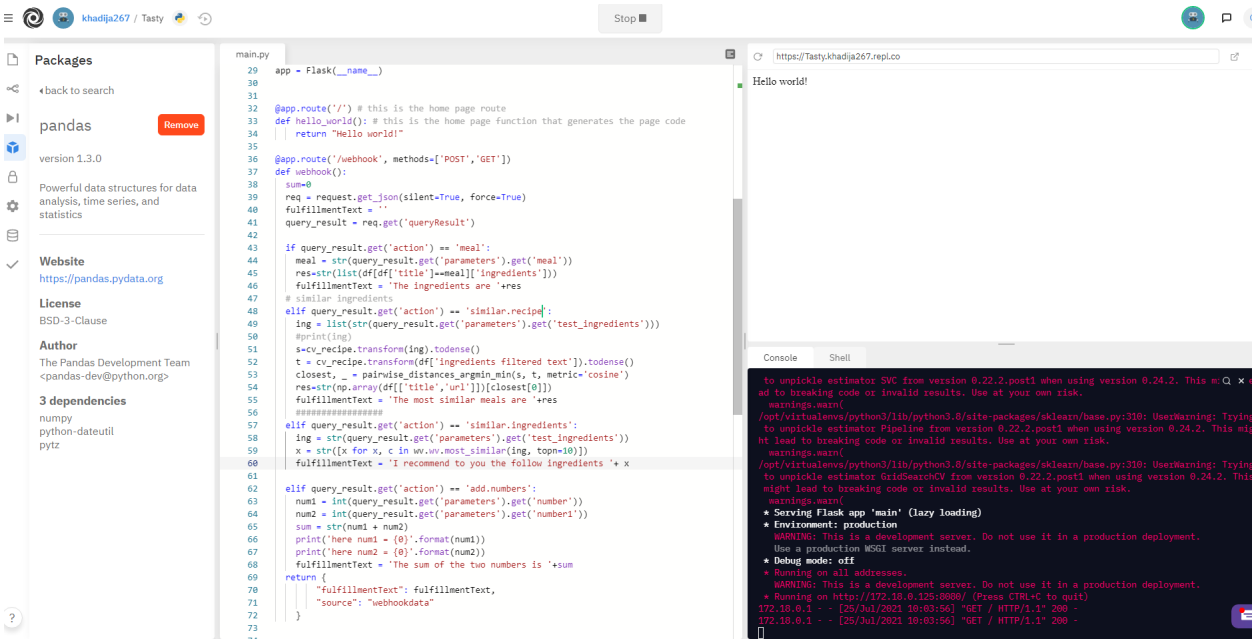
Questions per cluster.

For each cluster or topic, how many variations of questions is your chatbot able to answer?

For each cluster or topic, how many variations of questions is your chatbot able to answer?

Tasty can combine between related items in one cluster.


Tasty
POWERED BY Dialogflow
I love pasta_tomato
The most similar meals are ['Breakfast Cups' 'https://www.allrecipes.com/recipe /233747/breakfast-cups/']
I love pasta,tomato
The most similar meals are ['Breakfast Cups' 'https://www.allrecipes.com/recipe /233747/breakfast-cups/']
Ask something...


khadija267 / Tasty
Stop
Packages
back to search
pandas
Remove
version 1.3.0
Powerful data structures for data analysis, time series, and statistics
Website
https://pandas.pydata.org
License
BSD-3-Clause
Author
The Pandas Development Team <pandas-dev@python.org>
3 dependencies
numpy
python-dateutil
pytz
main.py
app = Flask(__name__)
@app.route('/') # this is the home page route
def hello_world(): # this is the home page function that generates the page code
return "Hello world!"
@app.route('/webhook', methods=['POST','GET'])
def webhook():
sum=0
req = request.get_json(silent=True, force=True)
fulfillmentText = ''
query_result = req.get('queryResult')
if query_result.get('action') == 'meal':
meal = str(query_result.get('parameters').get('meal'))
res=str(list(df[df['title']==meal]['ingredients']))
fulfillmentText = 'The ingredients are '+res
# similar ingredients
elif query_result.get('action') == 'similar.recipe':
ing = list(str(query_result.get('parameters').get('test_ingredients')))
#print(ing)
s=cv_recipe.transform(ing).todense()
t = cv_recipe.transform(df['ingredients filtered text']).todense()
closest, _ = pairwise_distances_argmin_min(s, t, metric='cosine')
res=str(np.array(df[['title','url']])[closest[0]])
fulfillmentText = 'The most similar meals are '+res
################
elif query_result.get('action') == 'similar.ingredients':
ing = str(query_result.get('parameters').get('test_ingredients'))
x = str([x for x, c in wv.most_similar(ing, topn=10)])
fulfillmentText = 'I recommend to you the follow ingredients '+ x
elif query_result.get('action') == 'add.numbers':
num1 = int(query_result.get('parameters').get('number'))
num2 = int(query_result.get('parameters').get('number1'))
sum = str(num1 + num2)
print('here num1 = {0}'.format(num1))
print('here num2 = {0}'.format(num2))
fulfillmentText = 'The sum of the two numbers is '+sum
return {
"fulfillmentText": fulfillmentText,
"source": "webhookdata"
}
https://Tasty.khadija267.repl.co
Hello world!
Console
Shell
to unpickle estimator SVC from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or invalid results. Use at your own risk.
warnings.warn(
/opt/virtualenvs/python3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarning: Trying to unpickle estimator Pipeline from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or invalid results. Use at your own risk.
warnings.warn(
/opt/virtualenvs/python3/lib/python3.8/site-packages/sklearn/base.py:310: UserWarning: Trying to unpickle estimator GridSearchCV from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or invalid results. Use at your own risk.
warnings.warn(
* Serving Flask app 'main' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.18.0.125:8080/ (Press CTRL+C to quit)
172.18.0.1 - - [25/Jul/2021 10:03:56] "GET / HTTP/1.1" 200 -
172.18.0.1 - - [25/Jul/2021 10:03:56] "GET / HTTP/1.1" 200 -

11) References and Resources:

- https://medium.com/swlh/building-a-content-based-food-recommendation-engine-df2ac7d08129
- https://towardsdatascience.com/building-a-food-recommendation-system-90788f78691a
- https://www.marketingaiinstitute.com/blog/4-conversational-ai-metrics-how-to-measure-ai-chatbot-performance
- https://heartbeat.fritz.ai/recommender-systems-with-python-part-i-content-based-filtering-5df4940bd831
- https://towardsdatascience.com/how-to-build-a-restaurant-recommendation-system-using-latent-factor-collaborative-filtering-ffe08dd57dca
- https://www.algolia.com/blog/ai/the-anatomy-of-high-performance-recommender-systems-part-1/
- https://www.datacamp.com/community/tutorials/recommender-systems-python
- https://apify.com/dtrungtin/allrecipes-scraper
- https://github.com/tropicalmentat/ingredients_extraction
- https://replit.com/