Multilingual Culinary Chatbot: Multilingual Recipe Recommendation Chatbot Based on Ingredients

Mt. SAC CISB 63 Final Project Fall 2023

Vedavit Shetty

The Multilingual Culinary Chatbot redefines the recipe recommendation experience by leveraging the analytical prowess of NLP techniques. It intricately processes user-provided ingredients using spaCy's sophisticated NER for precise identification, complemented by NLTK's tokenization and POS tagging for a deeper linguistic analysis. By employing a TF-IDF vectorization approach alongside cosine similarity, the chatbot adeptly surfaces the most pertinent recipes. Its multilingual capacity, actualized through TextBlob, enriches this culinary voyage, offering recipe translations that cater to a global audience.

Github Publication: https://github.com/vedavitshetty/Multilingual-Culinary-Chatbot (https://github.com/vedavitshetty/Multilingual-Culinary-Chatbot)

In [1]:
```
#ignore warnings:
import warnings
warnings.filterwarnings("ignore")
!pip install textblob
!python -m textblob.download_corpora
```

```
Requirement already satisfied: textblob in /Users/vedavitshetty/anacond
a3/lib/python3.11/site-packages (0.17.1)
Requirement already satisfied: nltk>=3.1 in /Users/vedavitshetty/anacon
da3/lib/python3.11/site-packages (from textblob) (3.8.1)
Requirement already satisfied: click in /Users/vedavitshetty/anaconda3/
lib/python3.11/site-packages (from nltk>=3.1->textblob) (8.0.4)
Requirement already satisfied: joblib in /Users/vedavitshetty/anaconda
3/lib/python3.11/site-packages (from nltk>=3.1->textblob) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in /Users/vedavitshetty/
anaconda3/lib/python3.11/site-packages (from nltk>=3.1->textblob) (202
2.7.9)
Requirement already satisfied: tqdm in /Users/vedavitshetty/anaconda3/l
ib/python3.11/site-packages (from nltk>=3.1->textblob) (4.65.0)
[nltk_data] Downloading package brown to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package brown is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package conll2000 to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package conll2000 is already up-to-date!
[nltk_data] Downloading package movie_reviews to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package movie_reviews is already up-to-date!
Finished.
```

In [2]:
```
#!pip install WordCloud

# Import necessary libraries:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
from nltk.chunk import ne_chunk
import re
from wordcloud import WordCloud
from textblob import TextBlob
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# download stopwords, punkt, and averaged_perceptron_tagger, maxent_ne_c
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package words to
[nltk_data]     /Users/vedavitshetty/nltk_data...
[nltk_data]   Package words is already up-to-date!
```

Out[2]:
```
True
```

In [3]:
```
recipes = pd.read_csv('recipes.csv', index_col=0)
```

In [4]:
```
recipes
```

Out[4]:

	Title	Ingredients	Instructions	Image_Name	Cleaned_Ingredients
0	Miso-Butter Roast Chicken With Acorn Squash Pa...	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...	Pat chicken dry with paper towels, season all ...	miso-butter-roast-chicken-acorn-squash-panzanella	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...
1	Crispy Salt and Pepper Potatoes	['2 large egg whites', '1 pound new potatoes (...	Preheat oven to 400°F and line a rimmed baking...	crispy-salt-and-pepper-potatoes-dan-kluger	['2 large egg whites', '1 pound new potatoes (...
2	Thanksgiving Mac and Cheese	['1 cup evaporated milk', '1 cup whole milk', ...	Place a rack in middle of oven; preheat to 400...	thanksgiving-mac-and-cheese-erick-williams	['1 cup evaporated milk', '1 cup whole milk', ...
3	Italian Sausage and Bread Stuffing	['1 (¾- to 1-pound) round Italian loaf, cut in...	Preheat oven to 350°F with rack in middle. Gen...	italian-sausage-and-bread-stuffing-240559	['1 (¾- to 1-pound) round Italian loaf, cut in...
4	Newton's Law	['1 teaspoon dark brown sugar', '1 teaspoon ho...	Stir together brown sugar and hot water in a c...	newtons-law-apple-bourbon-cocktail	['1 teaspoon dark brown sugar', '1 teaspoon ho...
...	...	...	...	...	...
13496	Brownie Pudding Cake	['1 cup all-purpose flour', '2/3 cup unsweeten...	Preheat the oven to 350°F. Into a bowl sift to...	brownie-pudding-cake-14408	['1 cup all-purpose flour', '2/3 cup unsweeten...
13497	Israeli Couscous with Roasted Butternut Squash...	['1 preserved lemon', '1 1/2 pound butternut s...	Preheat oven to 475°F.\nHalve lemons and scoop...	israeli-couscous-with-roasted-butternut-squash...	['1 preserved lemon', '1 1/2 pound butternut s...
13498	Rice with Soy-Glazed Bonito Flakes and Sesame ...	['Leftover katsuo bushi (dried bonito flakes) ...	If using katsuo bushi flakes from package, moi...	rice-with-soy-glazed-bonito-flakes-and-sesame-...	['Leftover katsuo bushi (dried bonito flakes) ...
13499	Spanakopita	['1 stick (1/2 cup) plus 1 tablespoon unsalted...	Melt 1 tablespoon butter in a 12-inch heavy sk...	spanakopita-107344	['1 stick (1/2 cup) plus 1 tablespoon unsalted...
13500	Mexican Poblano, Spinach, and Black Bean "Lasa...	['12 medium to large fresh poblano chiles (2 1...	Lay 4 chiles on their sides on racks of gas bu...	mexican-poblano-spinach-and-black-bean-lasagne...	['12 medium to large fresh poblano chiles (2 1...

13501 rows × 5 columns

Exploratory Data Analysis (EDA)

Check for missing or null values and handle it

In [5]:
```
recipes.isnull().sum()
```

Out[5]:
```
Title                 5
Ingredients           0
Instructions          8
Image_Name            0
Cleaned_Ingredients   0
dtype: int64
```

In [6]:
```
recipes.dropna(inplace=True)
```

Inspect the Data

In [7]:
```
recipes.describe()
```

Out[7]:

	Title	Ingredients	Instructions	Image_Name	Cleaned_Ingredients
count	13493	13493	13493	13493	13493
unique	13302	13471	13464	13464	13471
top	Potato Latkes	[]	Place ingredients in blender in the order list...	#NAME?	['']
freq	5	6	5	30	6

Drop the rows where the ingredients are not listed

In [8]:
```
# Filter out rows where Cleaned_Ingredients is "[]"
recipes = recipes[recipes['Ingredients'] != '[]']
```

In [9]:
```
recipes.describe()
```

Out[9]:

	Title	Ingredients	Instructions	Image_Name	Cleaned_Ingredients
count	13487	13487	13487	13487	13487
unique	13296	13470	13458	13458	13470
top	French 75	['1 cube or 1/2 teaspoon sugar', '4 dashes Pey...	Place ingredients in blender in the order list...	#NAME?	['1 cube or 1/2 teaspoon sugar', '4 dashes Pey...
freq	5	4	5	30	4

Data looks good, let's continue to inspect the data

In [10]:
```
recipes.head()
```

Out[10]:

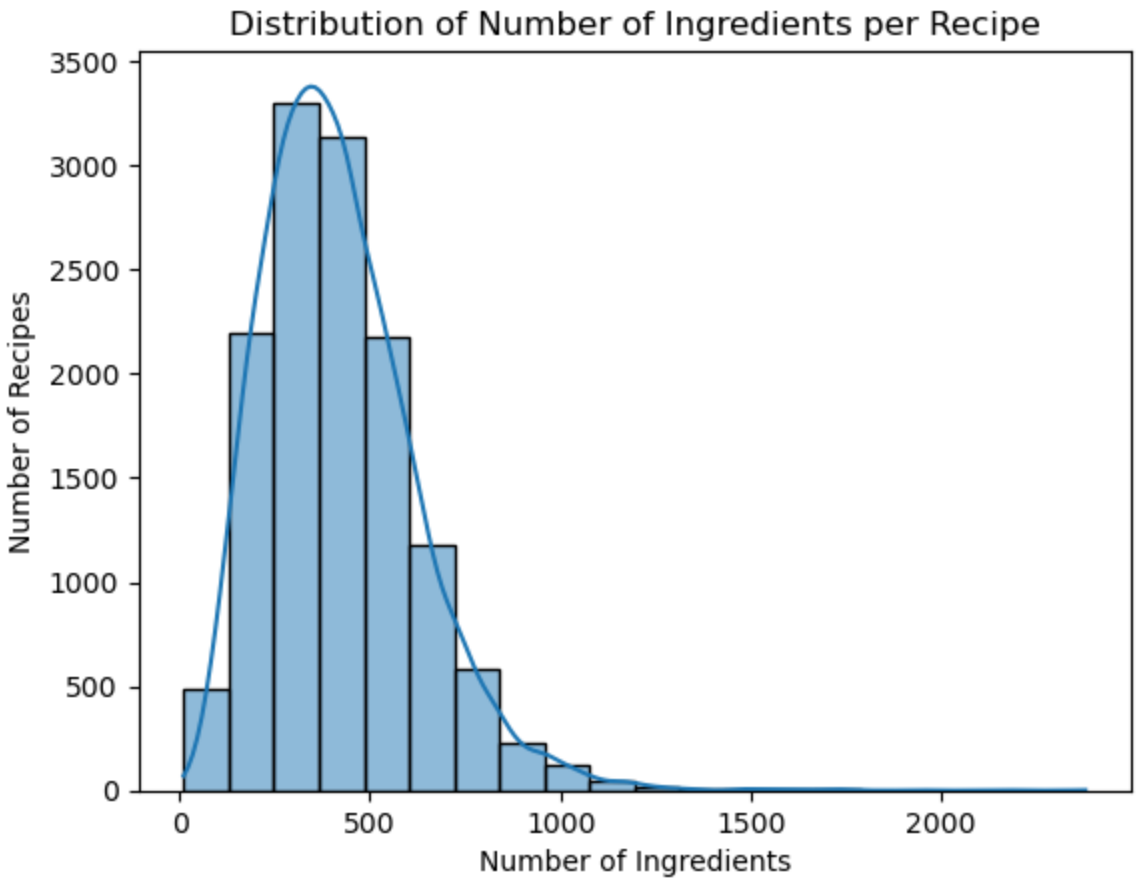
	Title	Ingredients	Instructions	Image_Name	Cleaned_Ingredients
0	Miso-Butter Roast Chicken With Acorn Squash Pa...	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...	Pat chicken dry with paper towels, season all ...	miso-butter-roast-chicken-acorn-squash-panzanella	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...
1	Crispy Salt and Pepper Potatoes	['2 large egg whites', '1 pound new potatoes (...	Preheat oven to 400°F and line a rimmed baking...	crispy-salt-and-pepper-potatoes-dan-kluger	['2 large egg whites', '1 pound new potatoes (...
2	Thanksgiving Mac and Cheese	['1 cup evaporated milk', '1 cup whole milk', ...	Place a rack in middle of oven; preheat to 400...	thanksgiving-mac-and-cheese-erick-williams	['1 cup evaporated milk', '1 cup whole milk', ...
3	Italian Sausage and Bread Stuffing	['1 (¾- to 1-pound) round Italian loaf, cut in...	Preheat oven to 350°F with rack in middle. Gen...	italian-sausage-and-bread-stuffing-240559	['1 (¾- to 1-pound) round Italian loaf, cut in...
4	Newton's Law	['1 teaspoon dark brown sugar', '1 teaspoon ho...	Stir together brown sugar and hot water in a c...	newtons-law-apple-bourbon-cocktail	['1 teaspoon dark brown sugar', '1 teaspoon ho...

In [11]:
```
recipes.info()
```

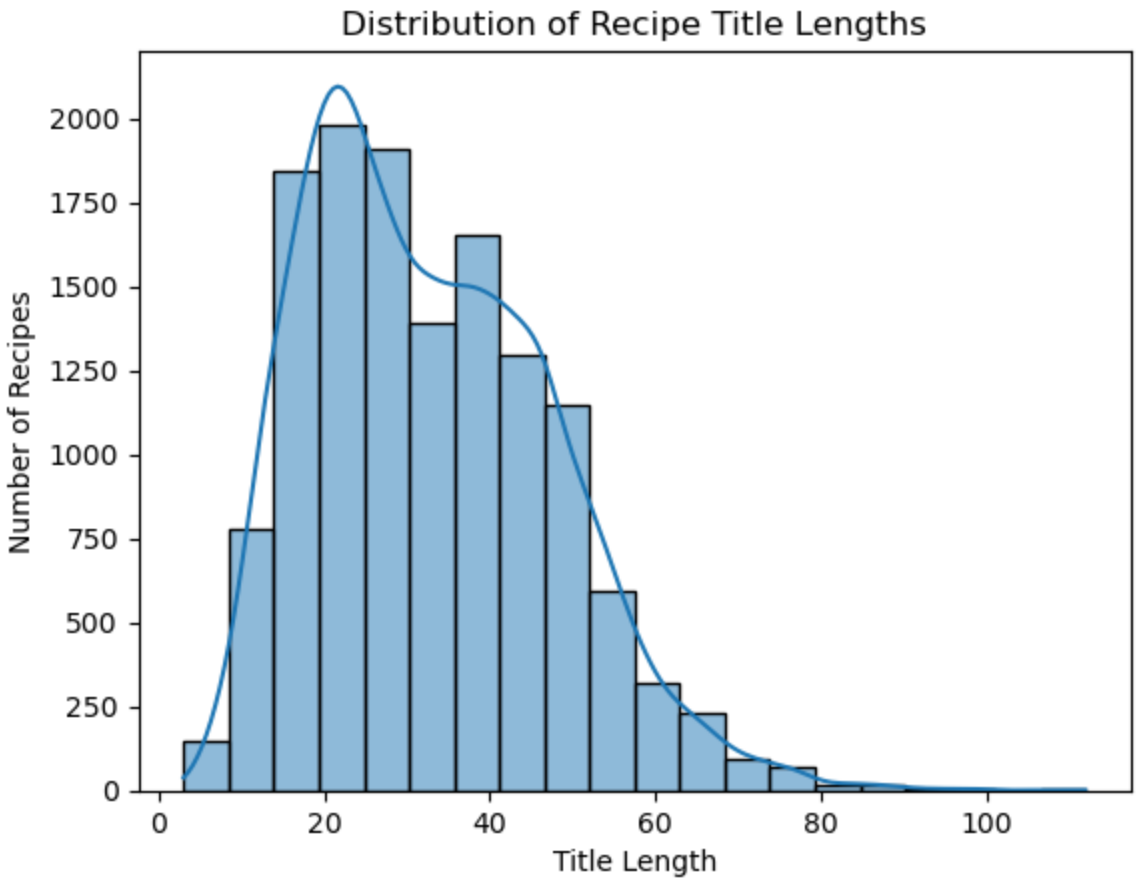
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13487 entries, 0 to 13500
Data columns (total 5 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   Title                13487 non-null  object
 1   Ingredients          13487 non-null  object
 2   Instructions         13487 non-null  object
 3   Image_Name           13487 non-null  object
 4   Cleaned_Ingredients  13487 non-null  object
dtypes: object(5)
memory usage: 632.2+ KB
```

Visualizations

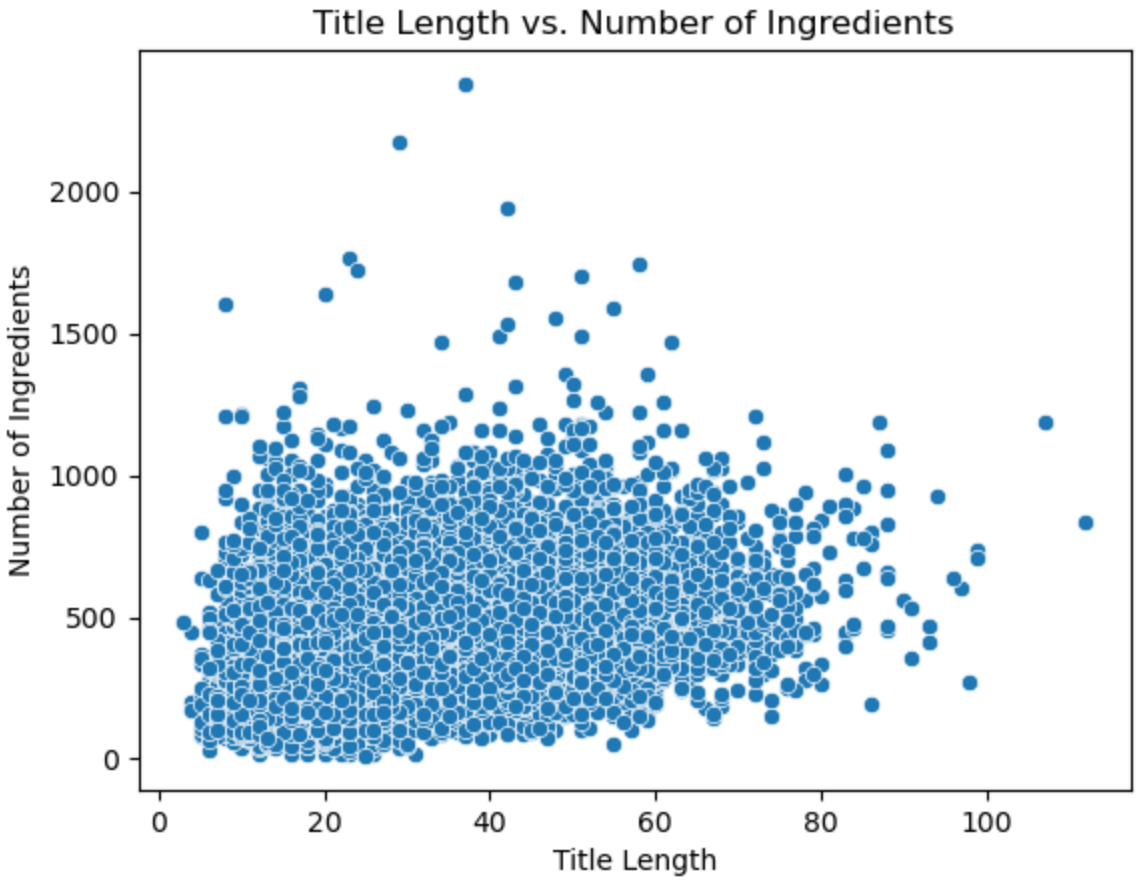
In [12]:
```
# Distribution of Number of Ingredients per Recipe
recipes['num_ingredients'] = recipes['Ingredients'].apply(len)
sns.histplot(recipes['num_ingredients'], bins=20, kde=True)
plt.title('Distribution of Number of Ingredients per Recipe')
plt.xlabel('Number of Ingredients')
plt.ylabel('Number of Recipes')
plt.show()
```



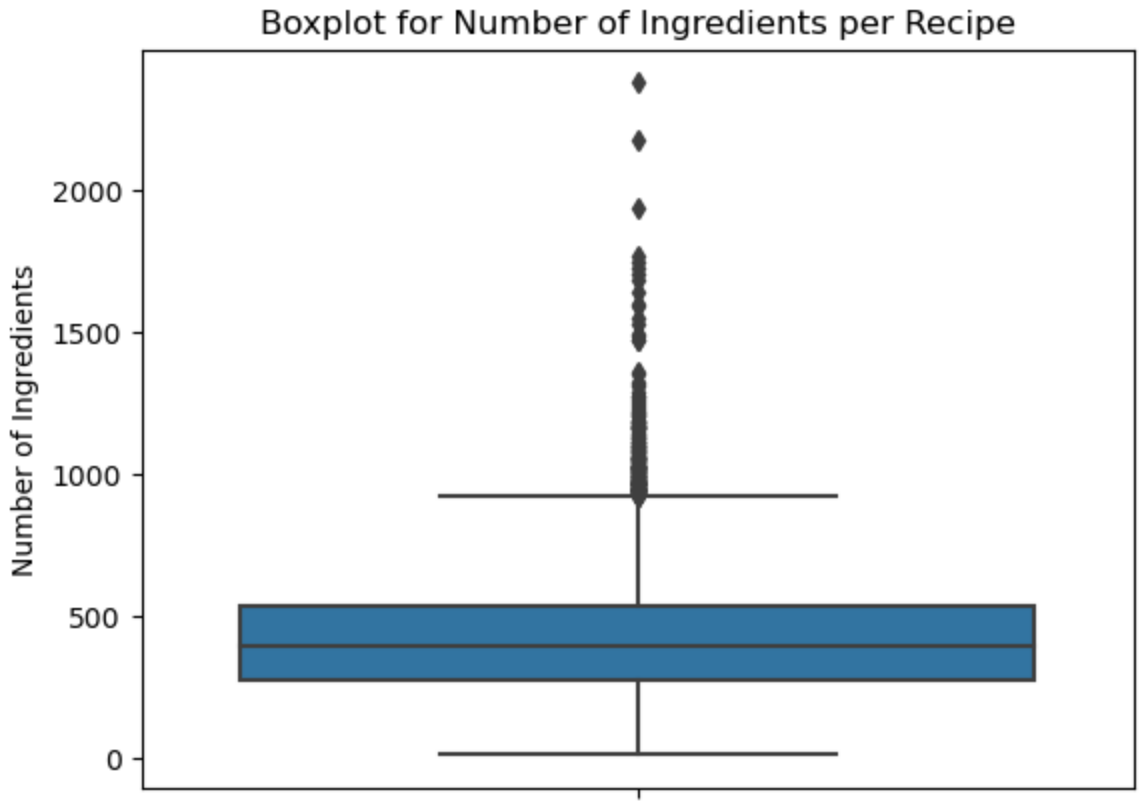
In [13]:
```
# Recipe Title Lengths
recipes['title_length'] = recipes['Title'].apply(len)
sns.histplot(recipes['title_length'], bins=20, kde=True)
plt.title('Distribution of Recipe Title Lengths')
plt.xlabel('Title Length')
plt.ylabel('Number of Recipes')
plt.show()
```



In [14]:
```
sns.scatterplot(x=recipes['title_length'], y=recipes['num_ingredients'])
plt.title('Title Length vs. Number of Ingredients')
plt.xlabel('Title Length')
plt.ylabel('Number of Ingredients')
plt.show()
```



In [15]:
```
sns.boxplot(y=recipes['num_ingredients'])
plt.title('Boxplot for Number of Ingredients per Recipe')
plt.ylabel('Number of Ingredients')
plt.show()
```



Some analysis:

- 75% of recipes have about 600 or less ingredients with the middle 50% containing about 100 to 600 ingredients
- There's not a strong correlation between recipe title and number of ingredients
- The distribution for recipe title lengths appears to be normally distributed around 20 to 40 characters.

Apply NLP Techniques to preprocess the data

Removing Stopwords and Lowercasing

In [16]:
```
stop = set(stopwords.words('english'))

recipes['Cleaned_Ingredients'] = recipes['Cleaned_Ingredients'].apply(la
recipes.head()
```

Out[16]:

	Title	Ingredients	Instructions	Image_Name	Cleaned_Ingredients	num_ingredients	title
0	Miso-Butter Roast Chicken With Acorn Squash Pa...	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...	Pat chicken dry with paper towels, season all ...	miso-butter-roast-chicken-acorn-squash-panzanella	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...	828	
1	Crispy Salt and Pepper Potatoes	['2 large egg whites', '1 pound new potatoes (...	Preheat oven to 400°F and line a rimmed baking...	crispy-salt-and-pepper-potatoes-dan-kluger	['2 large egg whites', '1 pound new potatoes (...	251	
2	Thanksgiving Mac and Cheese	['1 cup evaporated milk', '1 cup whole milk', ...	Place a rack in middle of oven; preheat to 400...	thanksgiving-mac-and-cheese-erick-williams	['1 cup evaporated milk', '1 cup whole milk', ...	289	
3	Italian Sausage and Bread Stuffing	['1 (¾- to 1-pound) round Italian loaf, cut in...	Preheat oven to 350°F with rack in middle. Gen...	italian-sausage-and-bread-stuffing-240559	['1 (¾- 1-pound) round italian loaf, cut 1-inc...	558	
4	Newton's Law	['1 teaspoon dark brown sugar', '1 teaspoon ho...	Stir together brown sugar and hot water in a c...	newtons-law-apple-bourbon-cocktail	['1 teaspoon dark brown sugar', '1 teaspoon ho...	218	

Tokenization

In [17]:
```
recipes['Tokenized_Ingredients'] = recipes['Cleaned_Ingredients'].apply(v
recipes.head()
```

Out[17]:

	Title	Ingredients	Instructions	Image_Name	Cleaned_Ingredients	num_ingredients	title
0	Miso-Butter Roast Chicken With Acorn Squash Pa...	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...	Pat chicken dry with paper towels, season all ...	miso-butter-roast-chicken-acorn-squash-panzanella	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...	828	
1	Crispy Salt and Pepper Potatoes	['2 large egg whites', '1 pound new potatoes (...	Preheat oven to 400°F and line a rimmed baking...	crispy-salt-and-pepper-potatoes-dan-kluger	['2 large egg whites', '1 pound new potatoes (...	251	
2	Thanksgiving Mac and Cheese	['1 cup evaporated milk', '1 cup whole milk', ...	Place a rack in middle of oven; preheat to 400...	thanksgiving-mac-and-cheese-erick-williams	['1 cup evaporated milk', '1 cup whole milk', ...	289	
3	Italian Sausage and Bread Stuffing	['1 (¾- to 1-pound) round Italian loaf, cut in...	Preheat oven to 350°F with rack in middle. Gen...	italian-sausage-and-bread-stuffing-240559	['1 (¾- 1-pound) round italian loaf, cut 1-inc...	558	
4	Newton's Law	['1 teaspoon dark brown sugar', '1 teaspoon ho...	Stir together brown sugar and hot water in a c...	newtons-law-apple-bourbon-cocktail	['1 teaspoon dark brown sugar', '1 teaspoon ho...	218	

POS (Part-of-Speech) Tagging

In [18]:
```
recipes['POS_Tagged_Ingredients'] = recipes['Tokenized_Ingredients'].app
recipes.head()
```

Out[18]:

	Title	Ingredients	Instructions	Image_Name	Cleaned_Ingredients	num_ingredients	title
0	Miso-Butter Roast Chicken With Acorn Squash Pa...	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...	Pat chicken dry with paper towels, season all ...	miso-butter-roast-chicken-acorn-squash-panzanella	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...	828	
1	Crispy Salt and Pepper Potatoes	['2 large egg whites', '1 pound new potatoes (...	Preheat oven to 400°F and line a rimmed baking...	crispy-salt-and-pepper-potatoes-dan-kluger	['2 large egg whites', '1 pound new potatoes (...	251	
2	Thanksgiving Mac and Cheese	['1 cup evaporated milk', '1 cup whole milk', ...	Place a rack in middle of oven; preheat to 400...	thanksgiving-mac-and-cheese-erick-williams	['1 cup evaporated milk', '1 cup whole milk', ...	289	
3	Italian Sausage and Bread Stuffing	['1 (¾- to 1-pound) round Italian loaf, cut in...	Preheat oven to 350°F with rack in middle. Gen...	italian-sausage-and-bread-stuffing-240559	['1 (¾- 1-pound) round italian loaf, cut 1-inc...	558	
4	Newton's Law	['1 teaspoon dark brown sugar', '1 teaspoon ho...	Stir together brown sugar and hot water in a c...	newtons-law-apple-bourbon-cocktail	['1 teaspoon dark brown sugar', '1 teaspoon ho...	218	

Word Cloud

In [19]:
```
from wordcloud import WordCloud

text = ' '.join(recipes['Cleaned_Ingredients'])
wordcloud = WordCloud().generate(text)

plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

Frequency Distribution

In [20]:
```
all_words = ' '.join(recipes['Cleaned_Ingredients']).split()
fdist = FreqDist(all_words)
fdist.most_common(10)
```

Out[20]:
```
[("'1", 37955),
 ('cup', 28116),
 ("'2", 21701),
 ('teaspoon', 15987),
 ('tablespoons', 15543),
 ("'1/2", 13440),
 ('fresh', 10109),
 ('cups', 10075),
 ("'1/4", 9645),
 ('1/2', 9173)]
```

NER (Named Entity Recognition)

In [21]:
```
# NER (Named Entity Recognition) Function
# This function identifies notable entities within the given text.
# Specifically, it searches for Geopolitical entities (GPE), Personal nar
def named_entity_recognition(text):
    tree = ne_chunk(nltk.pos_tag(word_tokenize(text)))
    named_entities = []
    for subtree in tree.subtrees():
        if subtree.label() in ['GPE', 'PERSON', 'ORGANIZATION']:
            entity = " ".join([word for word, tag in subtree.leaves()])
            named_entities.append(entity)
    return named_entities

# Get the NER for the first 100 recipes
Named_Entities = recipes['Instructions'].head(100).apply(named_entity_re
Named_Entities
```

Out[21]:
```
0     [Pat, Cut, Sprinkle, Set, Place, Pat, Place, T...
1                                                    []
2                                  [Cook, Sprinkle, Bake]
3                                   [Brown, Whisk, Cooks]
4                                         [Shake, Strain]
                              ...
95    [Purée, Season, Mash, Cook, Toss, Sprinkle, Se...
96    [Cook, Set, Scrape, Pulse, Chill, Scrape, Spri...
97                                           [Carefully]
98                [Pat, Cook, Turn, Whisk, Dollop, Bake]
99                           [Whisk, Arrange, Cook, Tear]
Name: Instructions, Length: 100, dtype: object
```

Translation

In [22]:
```
# Translation Function
# This function translates the given text into the specified target langu
# If the translation fails for any reason, it returns the original text.
def translate_text(text, target_lang='es'):
    try:
        blob = TextBlob(text)
        return blob.translate(from_lang='en', to=target_lang).string
    except:
        return text

# Applying the translation function to the first 100 titles in the recipe
spanish_recipe_title = recipes['Title'].head(100).apply(lambda x: transla
spanish_recipe_title
```

Out[22]:
```
0     Pollo asado miso-mordido con squash bellota pa...
1                  Patatas crujientes de sal y pimienta
2                      Acción de Gracias Mac and Cheese
3                   Salchicha italiana y relleno de pan
4                                       Ley de Newton
                            ...
95          Tazones de maíz y garbanzos con miso tahini
96                    Tarta de crema de fruta de piedra
97    Plátanos Hornados con crema y queso (plátano h...
98              Tamale pastel con tomate y maíz frescos
99    Panzanella en el camino carbonizado con vinagr...
Name: Title, Length: 100, dtype: object
```

In [23]:
```
# Applying the translation function to the first 100 instructions in the
spanish_recipe_instructions = recipes['Instructions'].head(100).apply(lam
spanish_recipe_instructions
```

Out[23]:
```
0     Pase el pollo seco con toallas de papel, sazon...
1     Precaliente el horno a 400 ° F y línea una ban...
2     Coloque una rejilla en el medio del horno; Pre...
3     Precaliente el horno a 350 ° F con rejilla en ...
4     Revuelva el azúcar morena y el agua caliente e...
                            ...
95    Purée jalapeños, jengibre, ajo, cilantro, jugo...
96    Precaliente un horno a 375 ° F. Cocine la mant...
97    Precaliente el horno a 400 ° F.\nColoque los p...
98    Precaliente el horno a 450 ° F. Calienta una s...
99    Batir ajo, salmuera, mostaza y sal en un tazón...
Name: Instructions, Length: 100, dtype: object
```

Subjectivity

In [24]:
```
# Subjectivity Analysis Function
# This function returns the subjectivity score of the given text.
# The score ranges from 0 (objective) to 1 (subjective).
def get_subjectivity(text):
    return TextBlob(text).sentiment.subjectivity

recipes['Instructions_Subjectivity'] = recipes['Instructions'].apply(get_
recipes.head()
```

Out[24]:

	Title	Ingredients	Instructions	Image_Name	Cleaned_Ingredients	num_ingredients	title_lengt
0	Miso-Butter Roast Chicken With Acorn Squash Pa...	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...	Pat chicken dry with paper towels, season all ...	miso-butter-roast-chicken-acorn-squash-panzanella	['1 (3½–4-lb.) whole chicken', '2¾ tsp. kosher...	828	5
1	Crispy Salt and Pepper Potatoes	['2 large egg whites', '1 pound new potatoes (...	Preheat oven to 400°F and line a rimmed baking...	crispy-salt-and-pepper-potatoes-dan-kluger	['2 large egg whites', '1 pound new potatoes (...	251	3
2	Thanksgiving Mac and Cheese	['1 cup evaporated milk', '1 cup whole milk', ...	Place a rack in middle of oven; preheat to 400...	thanksgiving-mac-and-cheese-erick-williams	['1 cup evaporated milk', '1 cup whole milk', ...	289	2
			Preheat				

TF-IDF Weighting & Matrix Creation

In [25]:
```
# Applying Term Frequency-Inverse Document Frequency (TF-IDF) technique
vectorizer = TfidfVectorizer(tokenizer=lambda x: x, lowercase=False, pre
matrix = vectorizer.fit_transform(recipes['Tokenized_Ingredients'])
```

Cosine Similarity

In [26]:
```
# Function to compute cosine similarity between user's input ingredients
def compute_cosine_similarity(user_tokens):
    user_vector = vectorizer.transform([user_tokens])
    cosine_scores = cosine_similarity(user_vector, matrix)
    recipes['cosine_score'] = cosine_scores[0]
    return recipes
```

Extract Ingredients with Spacy

In [27]:
```
#!pip install spacy
#!python -m spacy download en_core_web_sm
import spacy
nlp = spacy.load("en_core_web_sm")

def parse_ingredients_with_spacy(user_input):
    doc = nlp(user_input)
    ingredients = [ent.text for ent in doc.ents if ent.label_ == "FOOD"]
    return ingredients
```

Recipe Recommendation Function

In [28]:
```
# Function to recommend the top recipes based on the ingredients provide
def recommend_recipes(user_ingredients):
    # Feedback message
    print("Let me find some recipes for you...")

    # Convert user ingredients to tokens
    user_tokens = [ingredient.lower().strip() for ingredient in user_ing

    # Compute cosine similarity scores for recipes
    compute_cosine_similarity(user_tokens)

    # Calculate count of exact ingredient matches
    recipes['match_count'] = recipes['Tokenized_Ingredients'].apply(lamb

    # Return top 5 recipes with highest match count and then highest cos
    top_recipes = recipes.sort_values(by=['match_count', 'cosine_score']
    return top_recipes[['Title', 'Ingredients', 'Instructions']]
```

In [34]:
```
# Function to display the recommended recipes to the user in a structure
def display_recommendations(user_input):
    # Get the recommended recipes
    recommended = recommend_recipes(user_input)

    if recommended.empty:
        print("I couldn't find any recipes with those ingredients. Try d
        return

    print(f"Here are some recipes I found for you:")

    # Display each recipe
    for index, row in recommended.iterrows():
        print("\n---\n")
        print(f"Recipe {index + 1}: {row['Title']}")
        print(f"Ingredients: {row['Ingredients']}")
        print(f"Instructions: {row['Instructions']}")

    print("\n---\n")
```

Test the Refined Recommendation System

In [36]:
```
while True:
    user_input = input("\nEnter the ingredients you have (comma-separate
    if user_input.lower() == 'exit':
        print("Goodbye! Happy cooking!")
        break
    display_recommendations(user_input)
    # Ask if the user wants to continue or not
    continue_choice = input("Would you like to search for more recipes (
    if continue_choice.lower() not in ['yes', 'y']:
        print("Thank you for using the Culinary Matcher. Have a great da
        break
```

```
g broth and pomegranate molasses with the tomato paste. Wrap one onion
layer around about 2 to 3 tablespoons filling and arrange seam-side dow
n in a 2-quart shallow baking dish. Repeat with remaining onion layers
and filling.
Pour liquid over stuffed onions, cover with foil, and roast until tende
r and some of the liquid is absorbed, 2 hours (you can roast for up to
3 additional hours for softer, more savory onions). Remove from oven an
d uncover. Preheat broiler and broil until golden brown, 1 to 2 minute
s. Let cool for 10 minutes. To serve, drizzle with oil, and garnish wit
h pomegranate seeds, parsley, and cilantro.

---

Recipe 6993: Anchovy Butter
Ingredients: ['1/2 cup softened unsalted butter', '4 minced garlic clov
es', '8 anchovies packed in oil, drained and minced', '1/2 teaspoon hot
paprika', '1/2 teaspoon fresh lemon juice', 'Kosher salt']
Instructions: In a medium bowl, combine 1/2 cup softened unsalted butte
r, 4 minced garlic cloves, 8 anchovies packed in oil, drained and mince
d, 1/2 teaspoon hot paprika, 1/2 teaspoon fresh lemon juice, and kosher
salt to taste. Mix with a fork until smooth and spread on steak. Or tra
```

In [41]:
```
# Function to display the recommended recipes translated into the user's
def display_recommendations_in_different_languages(user_input, language=
    # Get the recommended recipes
    recommended = recommend_recipes(user_input)

    if recommended.empty:
        print("I couldn't find any recipes with those ingredients. Try d
        return

    print(f"Here are some recipes I found for you in {language}:")

    # Display each recipe
    for index, row in recommended.iterrows():
        print("\n---\n")
        print(f"Recipe {index + 1}: {translate_text(row['Title'], langua
        print(f"Ingredients: {translate_text(row['Ingredients'], languag
        print(f"Instructions: {translate_text(row['Instructions'], langu

    print("\n---\n")
```

In [42]:
```
# Dictionary to map language names to codes
language_codes = {
    'english': 'en',
    'spanish': 'es',
    'german': 'de',
    'french': 'fr',
    'italian': 'it',
    'portuguese': 'pt',
    'dutch': 'nl',
    'russian': 'ru',
    'japanese': 'ja',
    'chinese': 'zh',
    # Add more languages and their codes as necessary
}

def get_language_code(language_name):
    # Convert to lowercase and strip any extra whitespace
    language_name = language_name.lower().strip()
    return language_codes.get(language_name, 'en')  # Default to English
```

In [44]:
```
while True:
    # Get user input for ingredients
    user_ingredients = input("\nEnter the ingredients you have (comma-sep
    if user_ingredients.lower() == 'exit':
        print("Goodbye! Happy cooking!")
        break

    # Get user input for language preference
    language_name = input("Which language would you like the recipes in?
    language_code = get_language_code(language_name)

    # Call the function to display recommendations in the chosen language
    display_recommendations_in_different_languages(user_ingredients, lang

    # Ask if the user wants to continue or not
    continue_choice = input("Would you like to search for more recipes (Y
    if continue_choice.lower() not in ['yes', 'y']:
        print("Thank you for using the Culinary Matcher. Have a great day
        break
```

```
olden brown on all sides, turning frequently, about 30 minutes. Transfe
r to plates and serve.

---

Recipe 10963: Roast Chicken Legs with Lemon and Thyme
Ingredients: ['4 whole chicken legs or 4 chicken thighs and 4 drumstick
s (2 1/2 to 3 pounds total)', '3 tablespoons extra-virgin olive oil',
'5 (3- to 4-inch) sprigs fresh thyme', '2 garlic cloves, smashed', '3/4
teaspoon salt', '1/2 teaspoon black pepper', '4 (1/4-inch-thick) lemon
slices']
Instructions: Put oven rack in upper third of oven and preheat oven to
500°F.
Toss chicken with oil, thyme sprigs, garlic, salt, and pepper in a larg
e bowl, then transfer to a large (17- by 12-inch) shallow heavy baking
pan (1 inch deep).
Bake chicken 10 minutes, then add lemon slices to pan. Continue to bake
until chicken is golden and cooked through, 15 to 20 minutes more. Serv
e chicken with lemon slices.
```

Summary

Key phases of the project's development include:

- Exploratory Data Analysis (EDA): Initiated with rigorous data cleansing, followed by visualization to distill the dataset's core attributes.
- Natural Language Processing (NLP): The application of spaCy for precise NER and NLTK for foundational NLP tasks such as tokenization and POS tagging has created a robust analytical framework. The creation of a Word Cloud has visually captured the essence of prevalent ingredients.
- Visualization: Advanced through the creation of histograms and scatter plots, these visual aids render the data's complex patterns more accessible.
- Translation: Leveraged TextBlob for its initial multilingual support, the system transcends linguistic barriers, making culinary discovery universally accessible.

- Recommendation System: Anchored by TF-IDF vectors and cosine similarity, it meticulously aligns user preferences with the culinary database.
- Interactive Testing: The chatbot's user interface facilitates a seamless interaction, soliciting ingredient inputs and dispensing recipes in a multitude of languages, with an emphasis on English, Spanish, and Japanese.

Conclusion

The Multilingual Culinary Chatbot embodies a seamless fusion of culinary insight and technological innovation, empowering users to transform available ingredients into extraordinary meals. It accentuates recipe relevance through its multilingual capabilities, broadening its appeal to a diverse international user base. Poised for future growth, the chatbot's modular design is ready to accommodate expanded language options, personalized culinary recommendations, and potential e-commerce linkages for a complete from-pantry-to-plate experience.

In [ ]: