

▼ DoQA Cooking Dev Data Cleaning

```

import json
import pandas as pd
from pathlib import Path
import re
import matplotlib.pyplot as plt
import itertools # mote up later
from nltk import bigrams, ngrams, trigrams
import collections # must move up
import nltk
from nltk.corpus import stopwords

import warnings
warnings.filterwarnings("ignore")

with open("/content/doqa-cooking-dev-v2.1.json", 'r') as file:
    json_data = json.load(file)

data = json_data['data']

titles = []
backgrounds = []
all_paragraphs = []
for entry in data:
    title = entry["title"]
    background = entry["background"]
    paragraphs = [p["context"] for p in entry["paragraphs"]]
    # Store each variable in separate lists
    titles.append(title)
    backgrounds.append(background)
    all_paragraphs.append(paragraphs)

title_df = pd.DataFrame({'Title': titles})
background_df = pd.DataFrame({'Background': backgrounds})
paragraphs_df = pd.DataFrame({'Paragraphs': all_paragraphs})

print(title_df)


```

	Title
0	What to add to the batter of the cake to avoid...
1	Is bacon fat supposed to congeal at room tempe...
2	Is it safe to microwave Pyrex containers immed...
3	How to ensure that eggs get hard boiled on a g...
4	Clarified butter for gumbo roux
..	...
195	Why are mushrooms safe for everyone to touch a...
196	Why does olives with the pit left always taste...
197	What kind of reaction is this?
198	Should a sourdough starter be left open/ajar a...
199	What can I substitute for Cointreau

[200 rows x 1 columns]

Begin Cleaning

```

title_df_list = title_df['Title'].tolist()

title_df_list[:3]

["What to add to the batter of the cake to avoid hardening when the gluten formation can't be avoided?",
 'Is bacon fat supposed to congeal at room temperature?',
 'Is it safe to microwave Pyrex containers immediately after removing them from the freezer and removing the
 plastic lid?']

def clean_string(text): # can be Title, Background, Paragraphs
    """re.sub(pattern, repl, string).
    Returns the string obtained by replacing the leftmost
    non-overlapping occurrences of pattern in string by the
    replacement thus removing any urls
    """
    return " ".join(re.sub("([^\0-9A-Za-z \t])|(\w+:\/\/\S+)", "", str(text)).split())

TextOnlyTitle = [clean_string(Title) for Title in title_df_list]#can be Title, Background, Paragraphs

TextOnlyTitle[:1] # Can be Title, Background, Paragraphs

['What to add to the batter of the cake to avoid hardening when the gluten formation cant be avoided']

ListlowercasewordsTitle = [Title.lower().split() for Title in TextOnlyTitle]

ListlowercasewordsTitle[:1]

[['what',
 'to',
 'add',
 'to',
 'the',
 'batter',
 'of',
 'the',
 'cake',
 'to',
 'avoid',
 'hardening',
 'when',
 'the',
 'gluten',
 'formation',
 'cant',
 'be',
 'avoided']]

data = ListlowercasewordsTitle[:3]
for x in data:
    print(x, end=' ')

['what', 'to', 'add', 'to', 'the', 'batter', 'of', 'the', 'cake', 'to', 'avoid', 'hardening', 'when', 'the', 'gl

TextOnlyTitle = list(itertools.chain(*ListlowercasewordsTitle))

TextOnlyTitle[:2]

['what', 'to']

```

```
len(TextOnlyTitle)
```

```
1679
```

```
UniqueWordsTitle = set(TextOnlyTitle)
```

```
len(UniqueWordsTitle)
```

```
627
```

```
CountTextOnlyTitle = collections.Counter(TextOnlyTitle)
```

```
CountTextOnlyTitle.most_common(10)
```

```
[('to', 62),
 ('a', 55),
 ('the', 50),
 ('how', 49),
 ('is', 41),
 ('i', 38),
 ('for', 29),
 ('in', 29),
 ('what', 27),
 ('of', 25)]
```

```
CleanTitle = pd.DataFrame(CountTextOnlyTitle.most_common(10),
                          columns=['words', 'count'])
```

```
CleanTitle
```

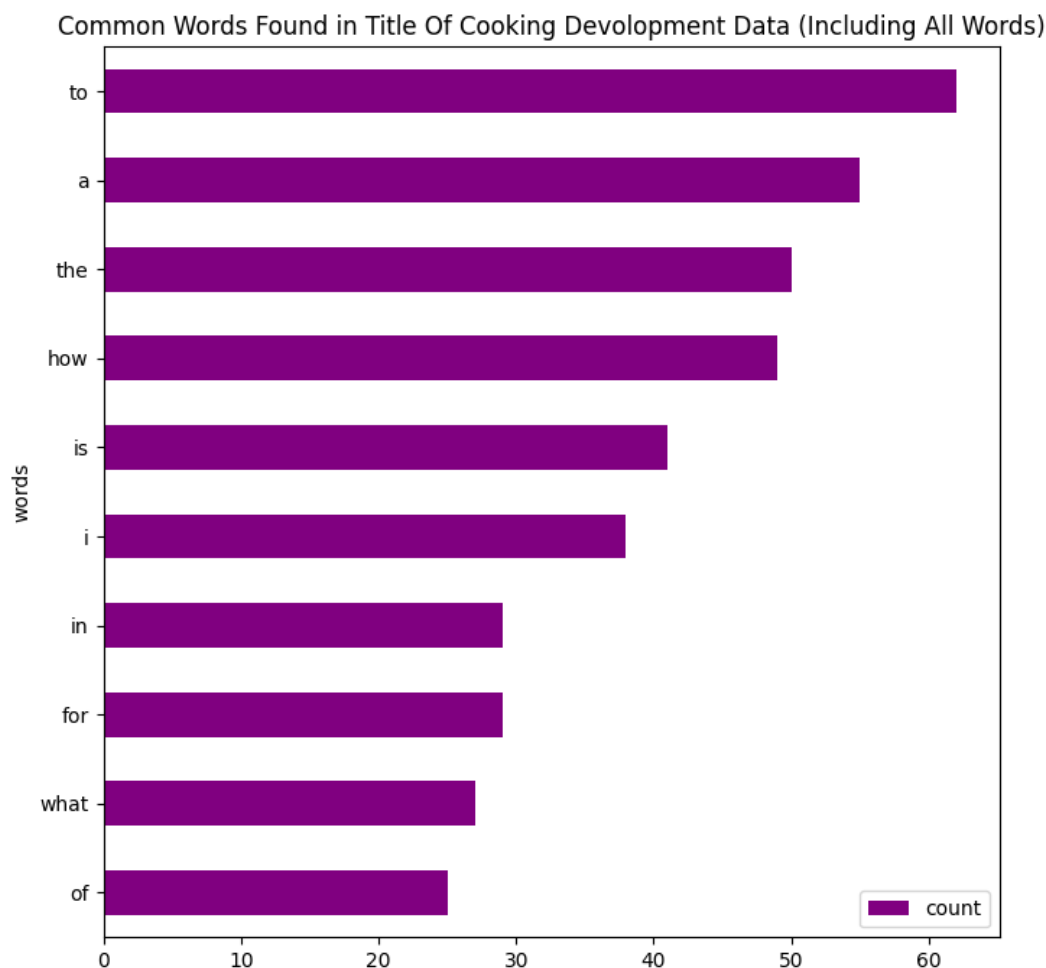
	words	count
0	to	62
1	a	55
2	the	50
3	how	49
4	is	41
5	i	38
6	for	29
7	in	29
8	what	27
9	of	25

```
fig, ax = plt.subplots(figsize=(8, 8))
```

```
# Plot horizontal bar graph
CleanTitle.sort_values(by='count').plot.barh(x='words',
                                              y='count',
                                              ax=ax,
                                              color="purple")
```

```
ax.set_title("Common Words Found in Title Of Cooking Development Data (Including All Words)")
```

```
plt.show()
```



```
background_df_list = background_df['Background'].tolist()
```

```
background_df_list[:3]
```

["So, over mixing batter forms gluten, which in turn hardens the cake. Fine.The problem is that I don't want lumps in the cakes, and the above statement prevents me from fine mixing the batter. So, is there something which I can add to the batter (more milk?) to make the outcome soft despite Gluten?"]

'My grandma told me its a good idea to save the bacon drippings in a sealable container to cook with later. I remember when I used to watch her cook with it, it was always solid. I have started saving the fat from my bacon, only the bottom of the can is the only part that ever congeals. The top always seems sort of semi-liquid. Is that ok? When cooking with it, what part should I use and what is the difference between solid and merely viscous bacon fat?'

'I am attempting to bulk prep some frozen convenience food and have determined that a standard Pyrex glass container would be ideal for freezing (non-liquid) meals (think burritos or tacos), but I want to ensure it is safe to induce a huge temperature swing, or if there is a better way to do this.'

```
def clean_string(text): # can be Title, Background, Paragraphs
```

```
    """re.sub(pattern, repl, string).
```

```
    Returns the string obtained by replacing the leftmost
    non-overlapping occurrences of pattern in string by the
    replacement thus removing any urls
    """
```

```
    return " ".join(re.sub("([^\0-9A-Za-z \t])|(\w+:\/\/\/\S+)", "", str(text)).split())
```

```
TextOnlyBackground = [clean_string(Background) for Background in background_df_list]#can be Title, Background, Parag
```

```
TextOnlyBackground[:1] # Can be Title, Background, Paragraphs
```

```
['So over mixing batter forms gluten which in turn hardens the cake FineThe problem is that I dont want lumps
in the cakes and the above statement prevents me from fine mixing the batter So is there something which I can
add to the batter more milk to make the outcome soft despite Gluten']
```

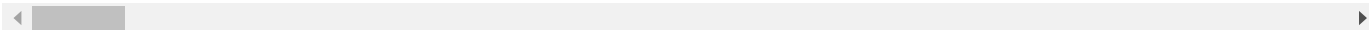
```
ListlowercasewordsBackground = [Background.lower().split() for Background in TextOnlyBackground]
```

```
data = ListlowercasewordsBackground[:3]
```

```
for x in data:
```

```
    print(x, end=' ')
```

```
    ['so', 'over', 'mixing', 'batter', 'forms', 'gluten', 'which', 'in', 'turn', 'hardens', 'the', 'cake', 'finethe'
```



```
TextOnlyBackground = list(itertools.chain(*ListlowercasewordsBackground))
```

```
TextOnlyBackground[:2]
```

```
['so', 'over']
```

```
len(TextOnlyBackground)
```

```
20212
```

```
UniqueWordsBackground = set(TextOnlyBackground)
```

```
len(UniqueWordsBackground)
```

```
3011
```



```
CountTextOnlyBackground = collections.Counter(TextOnlyBackground)
```

```
CountTextOnlyBackground.most_common(25)
```

```
[('the', 1050),
 ('i', 707),
 ('a', 610),
 ('to', 580),
 ('and', 486),
 ('it', 432),
 ('of', 386),
 ('is', 337),
 ('in', 312),
 ('for', 225),
 ('that', 201),
 ('with', 180),
 ('this', 170),
 ('or', 161),
 ('have', 160),
 ('on', 129),
 ('my', 118),
 ('be', 117),
 ('but', 114),
 ('if', 105),
 ('at', 105),
 ('there', 98),
 ('are', 94),
 ('as', 94),
 ('some', 92)]
```

```
CleanBackground = pd.DataFrame(CountTextOnlyBackground.most_common(10),
                                columns=['words', 'count'])
```

CleanBackground

	words	count	
0	the	1050	
1	i	707	
2	a	610	
3	to	580	
4	and	486	
5	it	432	
6	of	386	
7	is	337	
8	in	312	
9	for	225	

```
fig, ax = plt.subplots(figsize=(8, 8))
```

```
# Plot horizontal bar graph
```

```
CleanBackground.sort_values(by='count').plot.barh(x='words',
                                                    y='count',
                                                    ax=ax,
                                                    color="purple")
```

```
ax.set_title("Common Words Found in Background Of Cooking Development Data (Including All Words)")
```

```
plt.show()
```

Common Words Found in Background Of Cooking Development Data (Including All Words)



```
#importing stop word dictionary
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
True
```

```
<=
```

```
#Defining The Stop Words
```

```
stop_words = set(stopwords.words('english')) #there are 179 stop words
```

```
'''
```

```
# View a few words from the set
```

```
list(stop_words)[0:5]
```

```
['do', 'herself', 'some', 'her', 'while']
```

```
'''
```

```
ListlowercasewordsBackground[0] #list each lower case tweet
```

```
['so',
 'over',
 'mixing',
 'batter',
 'forms',
 'gluten',
 'which',
 'in',
 'turn',
 'hardens',
 'the',
 'cake',
 'finethe',
 'problem',
 'is',
 'that',
 'i',
 'dont',
 'want',
 'lumps',
 'in',
 'the',
 'cakes',
 'and',
 'the',
 'above',
 'statement',
 'prevents',
 'me',
 'from',
 'fine',
 'mixing',
 'the',
 'batter',
 'so',
 'is',
 'there',
 'something',
 'which',
```

```
'i',
'can',
'add',
'to',
'the',
'batter',
'more',
'milk',
'to',
'make',
'the',
'outcome',
'soft',
'despite',
'gluten']
```

```
BackgroundWithoutStopwords = [[word for word in TextOnlyBackground if not word in stop_words] #works
for TextOnlyBackground in ListlowercasewordsBackground]
```

```
BackgroundWithoutStopwords[0]
```

```
['mixing',
'batter',
'forms',
'gluten',
'turn',
'hardenes',
'cake',
'finethe',
'problem',
'dont',
'want',
'lumps',
'cakes',
'statement',
'prevents',
'fine',
'mixing',
'batter',
'something',
'add',
'batter',
'milk',
'make',
'outcome',
'soft',
'despite',
'gluten']
```

```
BackgroundWithoutStopword = list(itertools.chain(*BackgroundWithoutStopwords))
```

```
CountBackgroundsWithoutStopwords = collections.Counter(BackgroundWithoutStopword)
```

```
CountBackgroundsWithoutStopwords.most_common(10)
```

```
[('chicken', 81),
('like', 79),
('would', 78),
('im', 76),
('make', 65),
('water', 61),
('cook', 59),
('use', 58),
('add', 49),
('food', 43)]
```



```
BackgroundWithoutStopwords = pd.DataFrame(CountBackgroundsWithoutStopwords.most_common(25),
                                           columns=['words', 'count'])
```

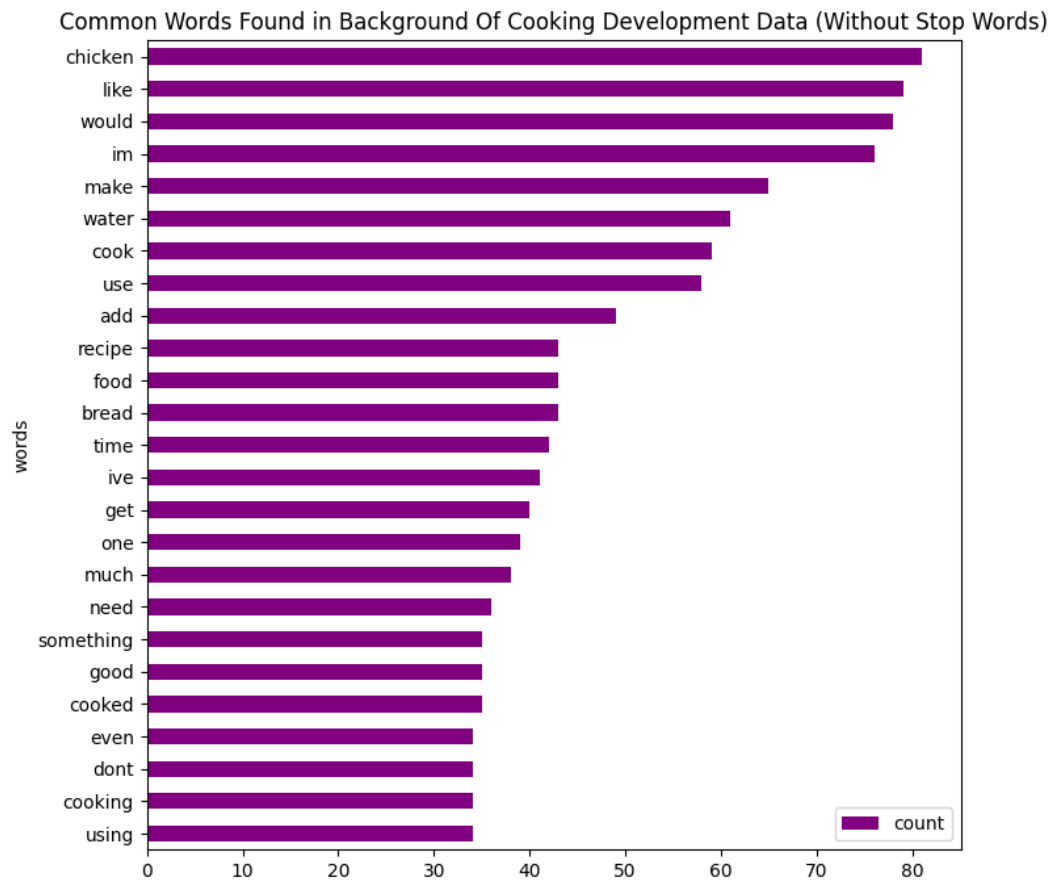
```
fig, ax = plt.subplots(figsize=(8, 8))
```

```
# Plot horizontal bar graph
```

```
BackgroundWithoutStopwords.sort_values(by='count').plot.barh(x='words',
                                                             y='count',
                                                             ax=ax,
                                                             color="purple")
```

```
ax.set_title("Common Words Found in Background Of Cooking Development Data (Without Stop Words)")
```

```
plt.show()
```



```
# Create list of lists containing bigrams in tweets
```

```
Backgroundbigram = [list(bigrams(Background)) for Background in BackgroundWithoutStopwords]
```

```
# View bigrams for the first tweet
```

```
Backgroundbigram[:]
```

```
[(['w', 'o'), ('o', 'r'), ('r', 'd'), ('d', 's')],
 [(['c', 'o'), ('o', 'u'), ('u', 'n'), ('n', 't')]]
```

▼ Paragraph

```
#background_df = pd.DataFrame({'Background': backgrounds})
```

```
paragraphs_df_list = paragraphs_df['Paragraphs'].tolist()
```

```
paragraphs_df_list[:3]
```

```
[['Milk won\'t help you - it\'s mostly water, and gluten develops from flour (more accurately, specific proteins in flour) and water.The way to reduce gluten development is to incorporate more fat into the batter. Lipids are hydrophobic and will prevent further hydration of the glutenin.Using a lower-protein flour will also help. If you\'re not already using cake flour, the reason it\'s called cake flour is because of the lower protein content.That being said, have you actually tried leaving the batter coarse? Just because the batter is lumpy does not mean that the cake will have big lumps. The entire mixture is wet, so unless you leave huge lumps of dry flour in the batter, you won\'t end up with a lumpy cake. There\'s a difference between "don\'t overmix" and "don\'t mix" - you\'re supposed to mix enough to incorporate, just don\'t try homogenize it. CANNOTANSWER'],
```

```
['To answer what I think is the question (you put all of the grease into a container and there\'s a residue at the top), bacon drippings are not 100% fat. There are also solid pieces of bacon in there and other "impurities" from the curing process.When rendering bacon fat, you should line the container with a paper towel first (or cheesecloth if you have it). Pour the bacon drippings onto the paper towel and the fat will drain out the bottom; the solids will be left behind and you can dispose of them. You\'ll be left with (mostly) pure fat.The rendered fat will most definitely congeal; the vessel, once cooled, should contain only a solid, off-white substance. CANNOTANSWER'],
```

```
['It not huge, it\'s just the difference from freezer to room temperature you are worrying aboutE.g. -20Â°C to 20Â°C, is A 40Â°C shift. The shift was going to be 20Â°C to 100+Â°C anyway. There is no physical reasons why this would be anymore stressfulFrom room temperature you are raising it 80Â°C, from frozen you are raising it 120Â°C. Not a problem in the normal temperature range for glassThe freezing temperature of water (most common item in food) has no relation to the freezing temperature of glass etcPyrex and other glasses can be damaged if one part of them is instantly heated or cooled by a 100Â°C or so CANNOTANSWER"]]
```

```
TextOnlyParagraphs = [clean_string(paragraphs) for paragraphs in paragraphs_df_list]#can be Title, Background, Parag
```

```
TextOnlyParagraphs[:1] # Can be Title, Background, Paragraphs
```

```
['Milk wont help you its mostly water and gluten develops from flour more accurately specific proteins in flour and waterThe way to reduce gluten development is to incorporate more fat into the batter Lipids are hydrophobic and will prevent further hydration of the gluteninUsing a lowerprotein flour will also help If youre not already using cake flour the reason its called cake flour is because of the lower protein contentThat being said have you actually tried leaving the batter coarse Just because the batter is lumpy does not mean that the cake will have big lumps The entire mixture is wet so unless you leave huge lumps of dry flour in the batter you wont end up with a lumpy cake Theres a difference between dont overmix and dont mix youre supposed to mix enough to incorporate just dont try homogenize it CANNOTANSWER']
```

```
ListlowercasewordsParagraphs= [Paragraphs.lower().split() for Paragraphs in TextOnlyParagraphs]
```

```
data = ListlowercasewordsParagraphs[:3]
```

```
for x in data:
```

```
    print(x, end= ' ')
```

```
['milk', 'wont', 'help', 'you', 'its', 'mostly', 'water', 'and', 'gluten', 'develops', 'from', 'flour', 'more',
```

```
▶
```

```
TextOnlyParagraphs = list(itertools.chain(*ListlowercasewordsParagraphs))
```

```
TextOnlyParagraphs[:2]
```

```
['milk', 'wont']
```

```
len(TextOnlyParagraphs)
```

```
23862
```

```
UniqueWordsParagraphs = set(TextOnlyParagraphs)
```

```
len(UniqueWordsParagraphs) #15816/101330=15.6% Unique%=UniqueWords/TextOnlyTweet
```

```
3463
```

```
ParagraphsWithoutStopwords = [[word for word in TextOnlyParagraphs if not word in stop_words] #works  
                                for TextOnlyParagraphs in ListlowercasewordsParagraphs]
```

```
ParagraphsWithoutStopwords[0]
```

```
'gluteninusing',  
'lowerprotein',  
'flour',  
'also',  
'help',  
'youre',  
'already',  
'using',  
'cake',  
'flour',  
'reason',  
'called',  
'cake',  
'flour',  
'lower',  
'protein',  
'contentthat',  
'said',  
'actually',  
'tried',  
'leaving',  
'batter',  
'coarse',  
'batter',  
'lumpy',  
'mean',  
'cake',  
'big',
```

```
    'enough',  
    'incorporate',  
    'dont',  
    'try',  
    'homogenize',  
    'cannotanswer']
```

```
ParagraphsWithoutStopword = list(itertools.chain(*ParagraphsWithoutStopwords))
```

```
CountParagraphsWithoutStopwords = collections.Counter(ParagraphsWithoutStopword)
```

```
CountParagraphsWithoutStopwords.most_common(10)
```

```
[('cannotanswer', 200),  
 ('would', 105),  
 ('water', 88),  
 ('use', 74),  
 ('sauce', 71),  
 ('cooking', 70),  
 ('like', 68),  
 ('make', 64),  
 ('get', 60),  
 ('dough', 59)]
```

```
ParagraphsWithoutStopwords = pd.DataFrame(CountParagraphsWithoutStopwords.most_common(25),  
                                           columns=['words', 'count'])
```

```
fig, ax = plt.subplots(figsize=(8, 8))
```

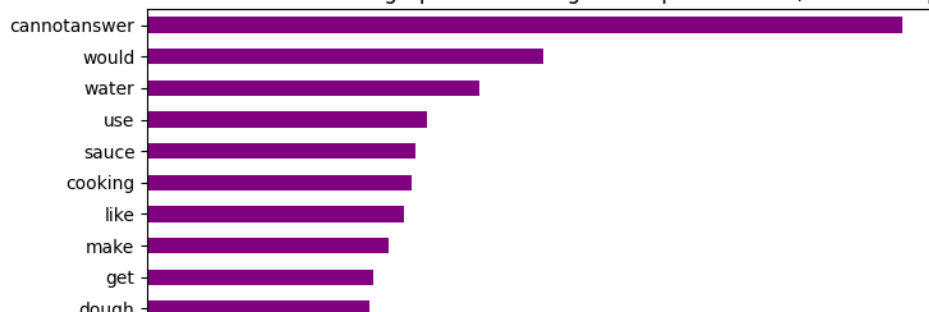
```
# Plot horizontal bar graph
```

```
ParagraphsWithoutStopwords.sort_values(by='count').plot.barh(x='words',  
                    y='count',  
                    ax=ax,  
                    color="purple")
```

```
ax.set_title("Common Words Found in Paragraphs Of Cooking Development Data (Without Stop Words)")
```

```
plt.show()
```

Common Words Found in Paragraphs Of Cooking Development Data (Without Stop Words)



```
# Create list of lists containing bigrams in tweets
```

```
Paragraphsbigram = [list(bigrams(Paragraphs)) for Paragraphs in ParagraphsWithoutStopwords]
```

```
g
```

```
# View bigrams for the first tweet
```

```
Paragraphsbigram[:]
```

```
[('w', 'o'), ('o', 'r'), ('r', 'd'), ('d', 's')],  
[('c', 'o'), ('o', 'u'), ('u', 'n'), ('n', 't')]]
```

