

# CAFE RECOMMENDER

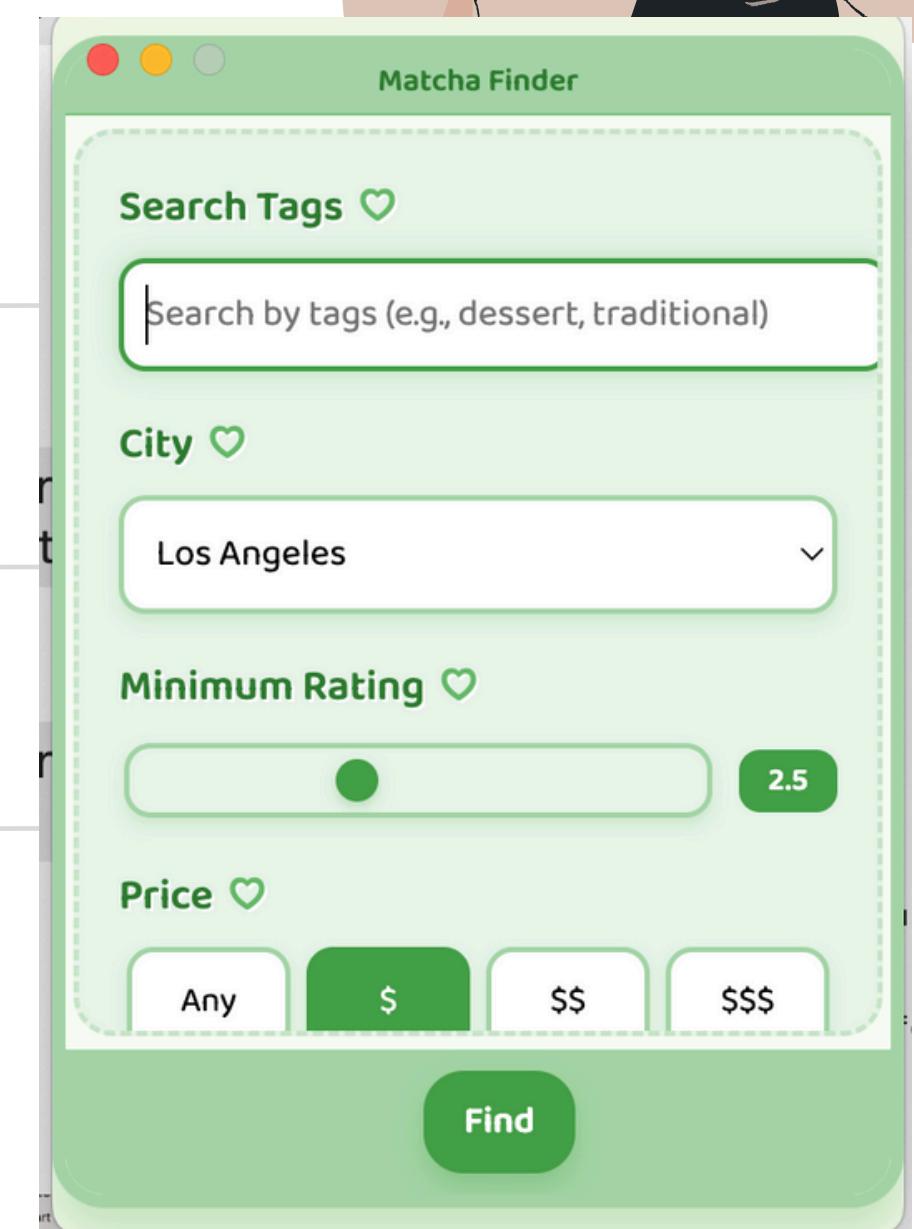
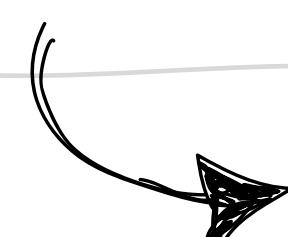
KIMBERLY CUI

# PURPOSE

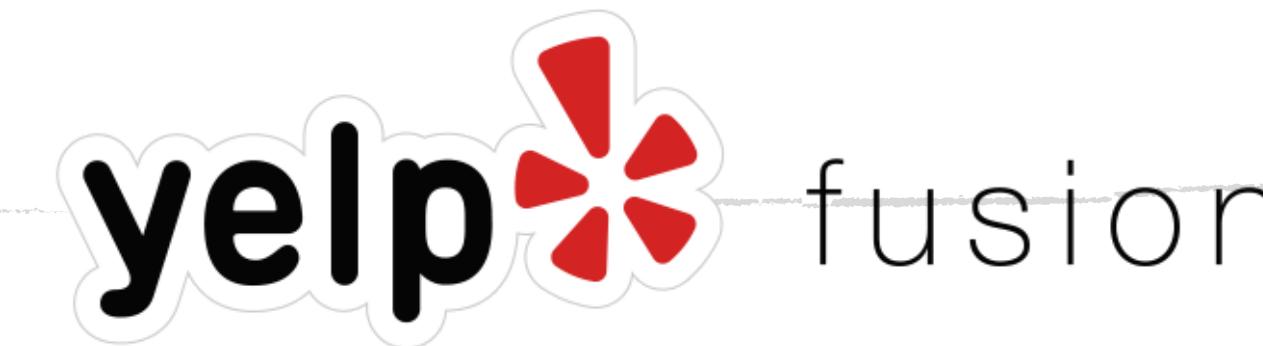
NEED TO FIND A NEW CAFE?  
OVERWHELMED BY TOO MANY CHOICES?  
STRICT CRITERIA?

NO FEAR!

- Find your VIBE with this cafe recommender!
- matcha-focused spots only 
- Price, rating, and city filters
- Reviews-based vibe categorization
-  cute iPod-style UI



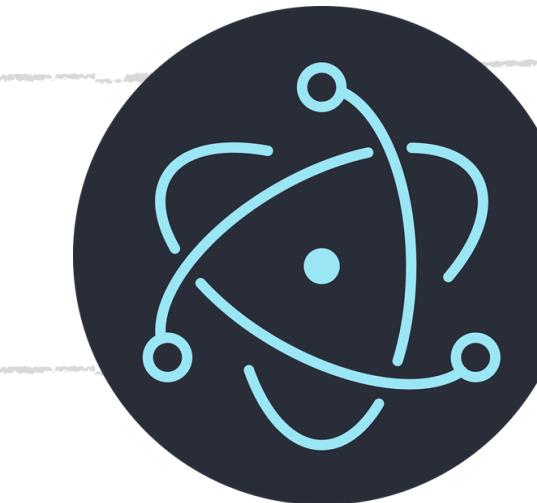
## WHAT I USED



fusion

**YAKE**

*(yet another  
keyword  
extractor)*



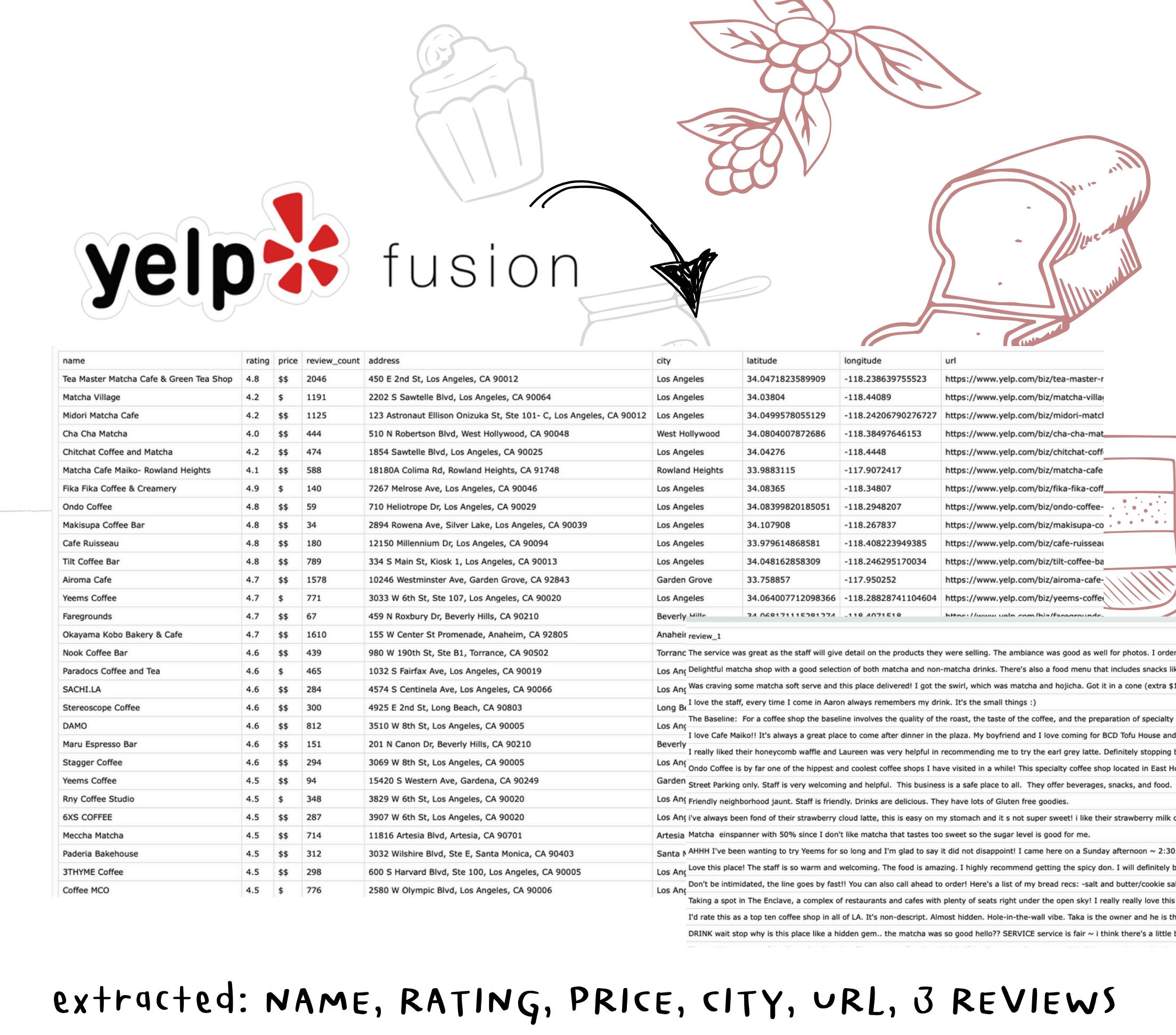
# DATASET

```

script.py 2 ✘ matcha_analysis.py 9+ ✘ review_analysis.py 2 •

Users > kimberlycui > Desktop > script.py > ...
1 1 import requests
2 2 import pandas as pd
3 3 import time
4 4
5 5 SEARCH_URL = "https://api.yelp.com/v3/businesses/search"
6 6 REVIEW_URL = "https://api.yelp.com/v3/businesses/{}/reviews"
7 7
8 8 # Define a list of locations (you can expand this)
9 9 locations = ["Los Angeles, CA", "San Francisco, CA", "New York, NY", "Seattle, WA"]
10 10
11 11 # Function to fetch data
12 12 def get_yelp_data(location, term="matcha cafe", limit=50):
13 13     businesses = []
14 14     for offset in range(0, 250-limit, limit): # Yelp allows up to 1000 results
15 15         print(offset)
16 16         params = {
17 17             "term": term,
18 18             "location": location,
19 19             "limit": limit,
20 20             "offset": offset,
21 21             "categories": "cafes,coffee",
22 22             "sort_by": "rating",
23 23             "price": "1,2,3,4",
24 24         }
25 25         response = requests.get(SEARCH_URL, headers=HEADERS, params=params)
26 26
27 27         if response.status_code == 200:
28 28             data = response.json()
29 29             businesses.extend(data.get("businesses", []))
30 30         else:
31 31             print(f"Error {response.status_code}: {response.json()}")
32 32             break # Stop if there's an error
33 33
34 34
35 35

```



# YAKE (YET ANOTHER KEYWORD EXTRACTOR)

FIRST COMBINE ALL 3 REVIEWS!

CONFIGURE THE YAKE KEYWORD EXTRACTOR

DEFINE A KEYWORD EXTRACTION FUNCTION

APPLY KEYWORD EXTRACTION TO DATASET

- `lan="en"`: Language of the text (English).
- `n=2`: Extract keywords that consist of two-word phrases (bigrams).
- `dedupLim=0.5`: controls how aggressively YAKE avoids returning duplicated keywords.
- `top=5`: Extracts the five most important keyword phrases per combined review.

- function takes the text of combined reviews and applies YAKE
- returns a list of the top 5 keyword phrases extracted from that text

- Applies function (`extract_yake_keywords`) to each combined review.
- Stores the resulting list of keywords in a new column (`tags`)

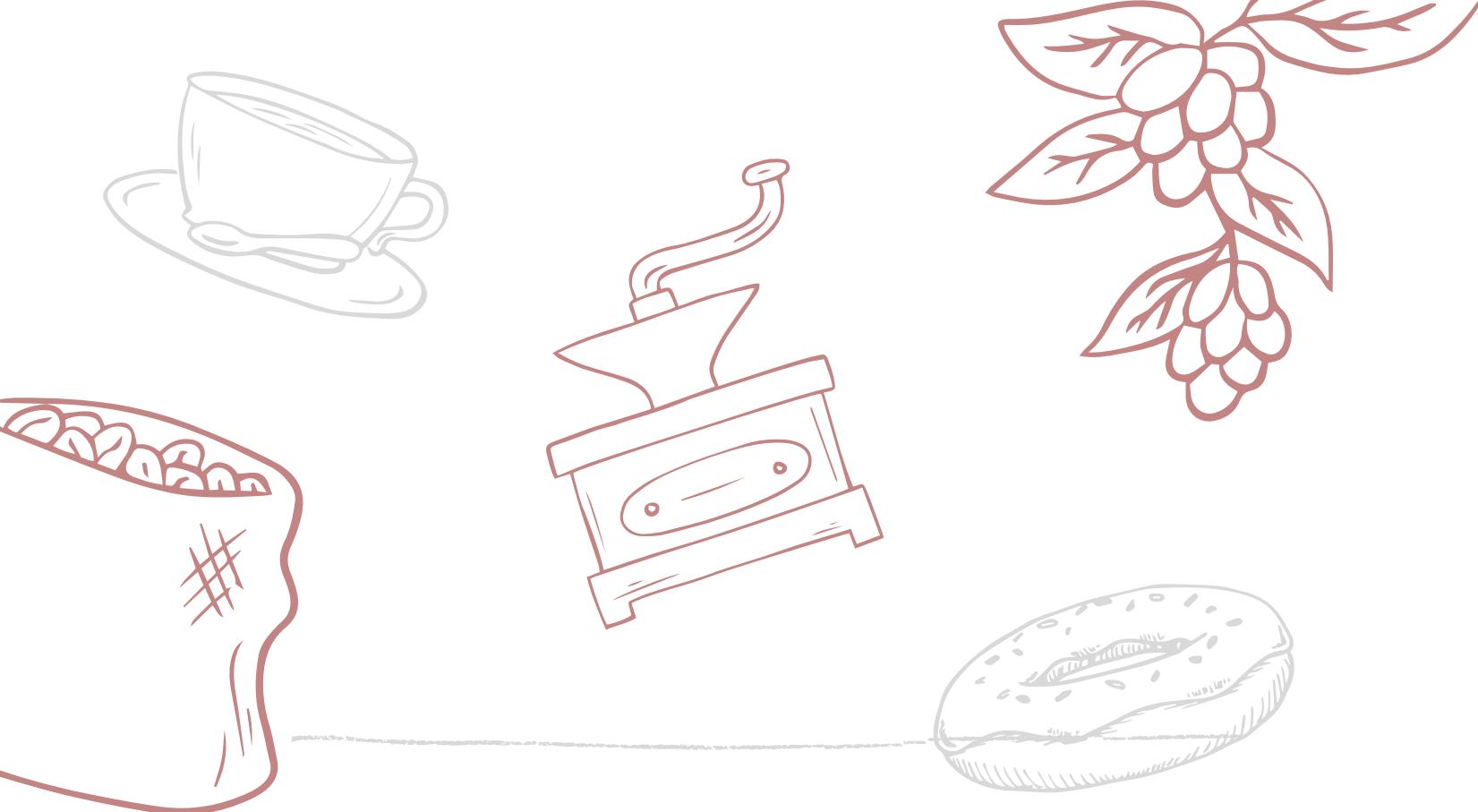
```
# YAKE keyword extractor configuration
kw_extractor = yake.KeywordExtractor(
    lan="en",
    n=2, # Bigrams (2-word phrases), adjust if needed
    dedupLim=0.5,
    top=5, # Top 5 keywords per document
    features=None
)

# Extract keywords using YAKE
def extract_yake_keywords(text):
    keywords = kw_extractor.extract_keywords(text)
    return [kw[0] for kw in keywords]

# Apply keyword extraction
df["tags"] = df["combined_reviews"].apply(extract_yake_key

# Save the updated dataframe
output_file = "matcha_cafes_with_tags.csv"
df.to_csv(output_file, index=False)
```





HOW DOES  
YAKE EXTRACT  
KEYWORDS?

## 1. TERM FREQUENCY

## 2. TERM POSITION

- A. PHRASES OCCURRING EARLIER IN THE TEXT CAN HAVE HIGHER RELEVANCE.

## 3. TERM RELATEDNESS TO CONTEXT

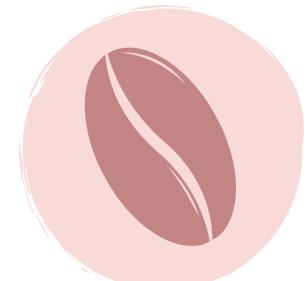
## 4. CASE (UPPERCASE, TITLE-CASE)

## 5. STOP WORDS AND COMMON WORDS

- A. COMMON, LOW-VALUE WORDS (E.G., "AND," "THE," "WITH") ARE TYPICALLY IGNORED

## 6. PHRASE UNIQUENESS

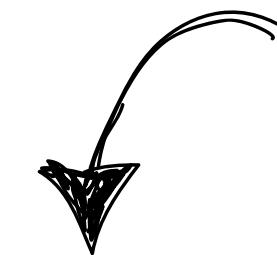
- 7. YAKE PREFERENCES PHRASES THAT ARE LESS GENERIC, GIVING PRIORITY TO TERMS UNIQUELY CHARACTERIZING A DOCUMENT OR REVIEW.



# KEYWORDS EXAMPLE



UPSIDE DOWN



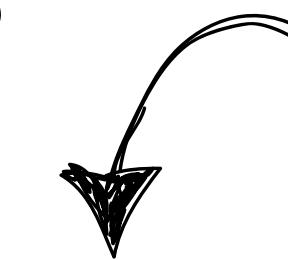
donation based  
friendly people and  
atmosphere  
great events to meet people  
great selection to drinks to  
enjoy

Just had an amazing first  
experience here! I attended  
one of their coffee tastings.  
Due to the recent tragedies  
here in LA they did it a little  
differently,...

If you're looking for a coffee  
experience that transcends  
the ordinary, look no further  
than Cold Brew. This  
delightful beverage is the  
epitome of smooth,...

[**'donation based', 'based friendly', 'friendly people', 'atmosphere great', 'great events'**]

ANKO



Pretty good! Got the mango  
bingsoo and blueberry latte.  
The bingsoo was smaller  
than usual but the ice was a  
lot more creamy and  
flavorful. The latte was...

Anko still has the best taiyaki  
in Koreatown (I always have  
to stop by and order 2 red  
bean and 2 custard)! The  
outside is always crispy,  
while the inside...

Cozy and quaint cafe, super  
sweet and clean aesthetic  
with chill music ambiance.  
Got the green grape  
refresher, which I am a  
sucker for anything grape...

[**'Pretty good', 'good', 'mango bingsoo', 'blueberry latte', 'latte'**]



# PERCENTAGE MATCH BREAKDOWN

## TAG MATCHING: 50%

- still uses the same exact/partial matching system
- most important factor after location filtering

## RATING MATCHING: 35%

- higher minimum ratings filter out lower-rated cafes
- Partial credit for cafes that almost meet your minimum

## PRICE MATCHING: 15%

- Exact price matches get full points
- Similar prices get partial credit

A screenshot of a mobile application interface. On the left, there's a large image of a white mug filled with coffee. To the right, the search results for "Acurrúcame Café" are displayed. The first result is "Acurrúcame Café" located in Los Angeles with a rating of 4.9 and a price point of \$. Below the name are three green tags: "friendly space", "space", and "calm". To the right of the tags is a green button containing the text "50%".

**Acurrúcame Café**  
Los Angeles | Rating: 4.9 | \$  
friendly space space calm ...

50%

A screenshot of a mobile application interface. On the left, there's a large image of a white mug filled with coffee. To the right, the search results for "Matcha Village" are displayed. The first result is "Matcha Village" located in Los Angeles with a rating of 4.2 and a price point of \$. Below the name are three green tags: "good selection", "matcha", and "non-matcha drinks". To the right of the tags is a green button containing the text "35%".

**Matcha Village**  
Los Angeles | Rating: 4.2 | \$  
good selection matcha non-matcha drinks ...

35%

A screenshot of a mobile application interface. On the left, there's a large image of a white mug filled with coffee. To the right, the search results for "Fika Fika Coffee & Creamery" are displayed. The first result is "Fika Fika Coffee & Creamery" located in Los Angeles with a rating of 4.9 and a price point of \$. Below the name are three green tags: "grey latte", "honeycomb waffle", and "earl grey". To the right of the tags is a green button containing the text "15%".

**Fika Fika Coffee & Creamery**  
Los Angeles | Rating: 4.9 | \$  
grey latte honeycomb waffle earl grey ...

15%



