Trends Extraction:

Here is quick walkthrough of the demo and this is mainly the flow:

1) Import necessary libraries

2) Create functions to extract articles from newsAPI.

3) Preprocess the articles to get trends from the extracted articles.

4) Extract trends from twitter API based on location.

5) Append trends from twitter to newsAPI.

6) Personalization for user.

- Importing necessary python libraries:

```
import pprint
import requests
import pandas as pd
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.tokenize import RegexpTokenizer
nltk.download('wordnet')
from nltk.stem.wordnet import WordNetLemmatizer
import matplotlib.pyplot as plt
from twitter import *
import sys
sys.path.append(".")
import config
import warnings
warnings.filterwarnings('ignore')
```

- These are secret keys to access the newsAPI. Alternate keys can be used if no of free hits diminishes. However, note that this is for development/research purposes only. Also newsAPI provides the following domain news and country codes are two letter codes to the API as a parameter.

```
secret='b88ebbaa632c4961984bc611b1b6d916'
secret_alt_1 = 'b7d9851f7a01459fb6e160952e93ebdf'
secret_alt_2='454c7b6933c5427cbf17ef85e4cf1ca7'
secret_alt_3='38c913a811df4a95ad68eae9ca127418'
#topics=['business', 'entertainment', 'general', 'health', 'science', 'sports', 'technology']
country=['us','ca','au'] #few of the country codes newapi provides.
```

- Parameters to be provided to the API:

```
parameters = {

    'pageSize': 30,  # maximum is 100
    'apiKey': secret # your own API key
}
```

- Code snippet to print list of websites and category of news they cater:

```
name=[]
category=[]

url_sources='https://newsapi.org/v2/sources?'
response = requests.get(url_sources, params=parameters)
response_json = response.json()
df_name=pd.DataFrame()
df_category=pd.DataFrame()

for i in response_json['sources']:
    df_name=df_name.append([i['name']])
    df_category=df_category.append([i['category']])
    df_name.reset_index(drop=True,inplace=True)
df_name=df_name.rename(columns={0:'name'})
name.append(df_name['name'].to_list())
df_category.reset_index(drop=True,inplace=True)
df_category=df_category.rename(columns={0:'category'})
category.append(df_category['category'].to_list())

name = [item for sublist in name for item in sublist]
category = [item for sublist in category for item in sublist]
category_dict = {name[i]: category[i] for i in range(len(name))}

url_df = pd.DataFrame(list(zip(name, category)),columns =['name', 'category'])

print("List of websites and the category of news the website provides:")
```

OUTPUT:

```
List of websites and the category of news the website provides:
```

	name	category
0	ABC News	general
1	ABC News (AU)	general
2	Aftenposten	general
3	Al Jazeera English	general
4	ANSA.it	general
5	Argaam	business
6	Ars Technica	technology
7	Ary News	general
8	Associated Press	general
9	Australian Financial Review	business
10	Axios	general

- Function to extract top headlines from newsAPI based on country codes(ex: US)

```
#extracts articles based on country codes
def extract(country_code):
    title=[]
    url=[]
    category=[]
    name=[]

    url_country='https://newsapi.org/v2/top-headlines?country='+country_code+'&apiKey='+secret
    df_title=pd.DataFrame()
    df_url=pd.DataFrame()
    df_category=pd.DataFrame()
    df_name=pd.DataFrame()

    response = requests.get(url_country, params=parameters)

# Convert the response to JSON format and pretty print it
    response_json = response.json()
    for i in response_json['articles']:
        df_title=df_title.append([i['title']])
        df_url=df_url.append([i['url']])
        df_name=df_name.append([i['source']['name']])

        #df_category=df_category.append(pd.Series(topic),ignore_index=True)
    df_title.reset_index(drop=True,inplace=True)
    df_title=df_title.rename(columns={0:'Title'})
    title.append(df_title['Title'].to_list())
    df_url.reset_index(drop=True,inplace=True)
    df_url=df_url.rename(columns={0:'url'})
    url.append(df_url['url'].to_list())

    df_name=df_name.rename(columns={0:'name'})
    name.append(df_name['name'].to_list())
```

```
 name = [item for sublist in name for item in sublist]
# category = [item for sublist in category for item in sublist]
# df_category.reset_index(drop=True,inplace=True)
 #df_category=df_category.rename(columns={0:'Title'})
 #category.append(df_category['Title'].to_list())
 title = [item for sublist in title for item in sublist]
 url = [item for sublist in url for item in sublist]
 #category = [item for sublist in category for item in sublist]


 for item in name:
     if item in category_dict.keys():
         category.append(category_dict[item])
     else:
         category.append('general')
 df = pd.DataFrame(list(zip(title, url,name,category)),columns =['title', 'url','name','category'])
 df['country']=country_code


 return df
```

- Extracting top 12 articles from US, Australia, Canada and India for reference.

```
us_df=extract('us')
print("Top 12 US articles:")
us_df.head(12)
```

OUTPUT:

```
Top 12 US articles:
CPU times: user 89.9 ms, sys: 4.39 ms, total: 94.3 ms
Wall time: 716 ms
```

	title	url	name	category	country
0	'Extremely serious' train derailment in Aberde...	https://www.cnn.com/2020/08/12/uk/aberdeen-tra...	CNN	general	us
1	Sumner Redstone, billionaire media tycoon, dea...	https://www.cnbc.com/2020/08/12/sumner-redston...	CNBC	general	us
2	Microsoft's two-screen Surface Duo isn't an iP...	https://www.cnet.com/features/microsofts-two-s...	CNET	general	us
3	New Zealand's largest city goes back into lock...	https://www.nbcnews.com/news/world/new-zealand...	NBC News	general	us
4	Coronavirus may spread much farther than 6 fee...	https://www.cbsnews.com/news/coronavirus-sprea...	CBS News	general	us
5	How Kamala Harris' Indian relatives helped sha...	https://www.cnn.com/2020/08/12/asia/kamala-har...	CNN	general	us
6	Ohio State's Justin Fields gives three-letter ...	https://www.foxnews.com/sports/ohio-state-just...	Fox News	general	us
7	Tesla's stock jumps after stock split news, Ba...	https://www.marketwatch.com/story/teslas-stock...	MarketWatch	general	us
8	Cold case murder, sexual assault of 17-year-ol...	https://www.usatoday.com/story/news/nation/202...	USA Today	general	us
9	Mauritius oil spill: Rush to pump out oil befo...	https://www.bbc.com/news/world-africa-53750151	BBC News	general	us
10	Ancient 'Terror Crocodiles' Preyed On Dinosaur...	https://www.npr.org/2020/08/12/901520688/teeth...	NPR	general	us
11	China to Bring Up WeChat, TikTok in Next U.S. ...	https://finance.yahoo.com/news/china-bring-wec...	Yahoo Entertainment	general	us

```
aus_df=extract('au')
print("Top 12 Australia articles:")
aus_df.head(12)
```

OUTPUT:

```
Top 12 Australia articles:
```

	title	url	name	category	country
0	All square: Suns and Bombers can't be split - AFL	https://www.afl.com.au/news/484608/all-square-...	Afl.com.au	general	au
1	'Extremely serious incident': Passenger train ...	https://www.smh.com.au/world/europe/train-dera...	The Sydney Morning Herald	general	au
2	The Bachelor: James Weir recaps episode 1 | An...	https://www.news.com.au/entertainment/tv/reali...	News.com.au	general	au
3	Microsoft Surface Duo is coming Sept. 10 for $...	https://www.cnet.com/news/microsoft-says-surfa...	CNET	general	au
4	Westpac survey: consumer confidence plunges - ...	https://www.news.com.au/national/breaking-news...	News.com.au	general	au
5	'I'd have someone speak to the players every d...	https://wwos.nine.com.au/nrl/brad-fittlers-sol...	Nine	general	au
6	Georgallis one of six staffers culled as part ...	https://www.smh.com.au/sport/nrl/georgallis-on...	The Sydney Morning Herald	general	au
7	First Look At The Newest King Of Fighters CG A...	https://www.kotaku.com.au/2020/08/first-look-a...	Kotaku Australia	general	au
8	Coronavirus: does the common cold protect you ...	https://theconversation.com/coronavirus-does-t...	The Conversation Africa	general	au
9	Policeman investigated over leak of footage, s...	https://www.theage.com.au/national/victoria/po...	The Age	general	au
10	Timeline points to Australian Defence Force of...	https://www.theage.com.au/national/victoria/ti...	The Age	general	au
11	Xbox Series X to launch in November, but Halo ...	https://amp.theguardian.comgames/2020/aug/12/x...	Theguardian.comgames	general	au

```
can_df=extract('ca')
print("Top 12 Canada articles:")
can_df.head(12)
```

OUTPUT:

```
Top 12 Canada articles:
```

	title	url	name	category	country
0	British fossil hunters find bones of new dinos...	https://www.thechronicleherald.ca/news/world/b...	TheChronicleHerald.ca	general	ca
1	Microsoft Surface Duo's design is winning me o...	https://www.cnet.com/news/microsoft-surface-du...	CNET	general	ca
2	The Xbox Series X could launch on November 6th...	https://www.theverge.com/2020/8/12/21364657/mi...	The Verge	technology	ca
3	Man charged in brutal fatal attack on Alberta ...	https://www.cp24.com/news/man-charged-in-bruta...	CP24 Toronto's Breaking News	general	ca
4	Interim NDP leader, 23, confesses to 'a little...	https://www.cbc.ca/news/canada/new-brunswick/n...	CBC News	general	ca
5	School outbreaks of COVID-19 will happen. Doct...	https://www.cbc.ca/news/health/covid-19-school...	CBC News	general	ca
6	A Broken Cable Has Smashed a Huge Hole in The ...	https://www.sciencealert.com/a-broken-cable-sm...	ScienceAlert	general	ca
7	Five fun things about the Perseid meteor showe...	https://lfpress.com/news/local-news/five-fun-t...	The London Free Press	general	ca
8	Groups vow to fight sexist, racist coverage of...	https://globalnews.ca/news/7267990/kamala-harr...	Global News	general	ca
9	Xiaomi's affordable Redmi K30 Ultra flagship i...	https://www.techradar.com/news/xiaomis-afforda...	TechRadar	technology	ca
10	Interior Health COVID-19 update - The Nelson D...	http://thenelsondaily.com/news/interior-health...	The Nelson Daily	general	ca
11	What do the Rangers do with Lafreniere until t...	https://news.google.com/__i/rss/rd/articles/CB...	Google News	general	ca

```
ind_df=extract('in')
print("Top 12 India articles:")
ind_df.head(12)
```

OUTPUT:

```
Top 12 India articles:
```

	title	url	name	category	country
0	Explained: Study zeroes in on most effective f...	https://indianexpress.com/article/explained/co...	The Indian Express	general	in
1	'Sanjay Dutt is a strong fighter,' says Subhas...	https://timesofindia.indiatimes.com/entertainm...	The Times of India	general	in
2	Kareena Kapoor, Saif Ali Khan confirm they are...	https://www.hindustantimes.com/bollywood/karee...	Hindustan Times	general	in
3	"Immature": Sharad Pawar On Grand-Nephew's Sus...	https://www.ndtv.com/india-news/immature-shara...	NDTV News	general	in
4	Realme to launch two new budget smartphones C1...	https://www.livemint.com/technology/tech-news/...	Livemint	general	in
5	MIUI 12 Update for Mi 10, Select Redmi Note Ph...	https://gadgets.ndtv.com/mobiles/news/miui-12-...	NDTV News	general	in
6	Edu-tech company Unacademy picks bid papers, s...	https://www.hindustantimes.com/cricket/edu-tec...	Hindustan Times	general	in
7	Sushant Singh Rajput case: Sushant's family la...	https://indianexpress.com/article/entertainmen...	The Indian Express	general	in
8	Samsung details Note20 Ultra's VRR display and...	https://www.gsmarena.com/samsung_talks_about_n...	GSMArena.com	general	in
9	Rajasthan peace deal done, 4 big reasons why C...	https://www.hindustantimes.com/india-news/raja...	Hindustan Times	general	in
10	Bengaluru violence: Muslim youths form human c...	https://timesofindia.indiatimes.com/city/benga...	The Times of India	general	in
11	NASA satellite finds 66 new exoplanets, 2,100 ...	https://www.tribuneindia.com/news/science-tech...	The Tribune India	general	in

- List of states in India. This list is taken so as to provide state wise news trends:

```
states_in_india=[
'Assam',
'Andhra Pradesh',
'Arunachal Pradesh',
'Bihar',
'Chhattisgarh',
'Goa',
'Gujarat',
'Haryana',
'Himachal Pradesh',
'Jammu and Kashmir',
'Jharkhand',
'Karnataka',
'Kerala',
'Madhya Pradesh',
'Maharashtra',
'Manipur',
'Meghalaya',
'Mizoram',
'Nagaland',
'Odisha',
'Punjab',
'Rajasthan',
'Sikkim',
'Tamil Nadu',
'Telangana',
'Tripura',
'Uttar Pradesh',
'Uttarakhand',
'West Bengal',
'Andaman and Nicobar Islands',
'Chandigarh',
'Dadar and Nagar Haveli',
'Daman and Diu',
'Delhi',
'Lakshadweep',
'Puducherry'
]
```

- Function to extract state wise popular news articles:

```
def extract_local():
    article=[]
    st=[]
    for state in states_in_india:
        parameters = {

        'pageSize': 30,  # maximum is 100
        'apiKey': secret,
        'sortBy': 'popularity',
        # 'q': 'Bangalore AND Bengaluru'
        #'q': 'Manipur'
        #'q':'Karnataka'
        'qInTitle': state

        }


        local_article=pd.DataFrame()
        article_state=pd.DataFrame()
        url='https://newsapi.org/v2/everything?'
        response = requests.get(url, params=parameters)
        response_json = response.json()
        for item in response_json['articles']:
            local_article=local_article.append([item['title']])
            article_state=article_state.append([state])
        if(local_article.empty==False):
            local_article.reset_index(drop=True,inplace=True)
            local_article=local_article.rename(columns={0:'title'})

            article.append(local_article['title'].to_list())
            article_state.reset_index(drop=True,inplace=True)
            article_state=article_state.rename(columns={0:'state'})
            st.append(article_state['state'].to_list())
```

```
article = [item for sublist in article for item in sublist]
st = [item for sublist in st for item in sublist]

states_df = pd.DataFrame((zip(article,st)),columns =['title', 'state/UT'])
return states_df
```

- Preview of state wise articles:

```
states_df=extract_local()
print("Articles of states:")
states_df.head()
```

- Now, we have collected news articles of country specific, state specific. This is processed using text extraction methods using python to extract particular important keywords as a hashtag. The code for the same is as below:

```
def extract_phrase(input_df):
    import pke
    import string
    from nltk.corpus import stopwords

 # 1. create a TopicRank extractor.
    extractor = pke.unsupervised.TopicRank()
    total=""
    for i in range(0,len(input_df)):
        total=total+"".join(input_df['title'][i])+" "
 # 2. load the content of the document.
    extractor.load_document(input=total)

 # 3. select the longest sequences of nouns and adjectives, that do
 #    not contain punctuation marks or stopwords as candidates.
    pos = {'NOUN', 'PROPN', 'ADJ'}
    stoplist = list(string.punctuation)
    stoplist += ['-lrb-', '-rrb-', '-lcb-', '-rcb-', '-lsb-', '-rsb-']
    stoplist += stopwords.words('english')
    stoplist  += ["using", "show", "result", "large", "also", "iv", "one", "two", "new", "previously", "shown"]
    extractor.candidate_selection(pos=pos, stoplist=stoplist)

 # 4. build topics by grouping candidates with HAC (average linkage,
 #    threshold of 1/4 of shared stems). Weight the topics using random
 #    walk, and select the first occuring candidate from each topic.
    extractor.candidate_weighting(threshold=0.74, method='average')

 # 5. get the 10-highest scored candidates as keyphrases
    keyphrases = extractor.get_n_best(n=50)
    lll=[]
    for i in range(len(keyphrases)):
        app=keyphrases[i][0]
        app='#'+app
        lll.append(app)
    df=pd.DataFrame(lll,columns=['trends'])
    return df
```

- Considering Rajasthan as a specific case for localized news:

```
rajasthan_df=states_df[states_df['state/UT']=='Rajasthan']
rajasthan_df.head()
rajasthan_df.reset_index(drop=True,inplace=True)
```

- Extracting hashtags from news articles using the text preprocessor function:

```
import warnings
warnings.filterwarnings('ignore')
raj_df=extract_phrase(rajasthan_df)
ind_df=extract_phrase(states_df)

can_df=extract_phrase(can_df)
aus_df=extract_phrase(aus_df)
us_df=extract_phrase(us_df)
```

- Now, we also gather trends from twitter and append it to the final list. Note that you need to have developer access to twitter API. Follow this link to get started: https://developer.twitter.com/en/docs/getting-started

- Code snippet to extract trends from twitter. This code snippet shall redirect to your twitter developer account to provide the OTP.

```
from twitter import *
app_name = 'hastag extracter'
consumer_key = 'JKv3MqdQYRrq98m2uR9bFq4ac'
consumer_secret = 'aE6NLvxSon8FSs52BJYAxpzIvuPwTO6s1v6K3X6nF1go1x9tp9'
access_key, access_secret = oauth_dance(app_name, consumer_key, consumer_secret)


twitter = Twitter(auth = OAuth(access_key,
                  access_secret,
                  consumer_key,
                  consumer_secret))

world=1
india=23424848
#uk=23424975
us=23424977
#russia=23424936
australia=23424748
canada=23424775



#bangalore=2295420
results_world = twitter.trends.place(_id = world)
results_can = twitter.trends.place(_id = canada)
results_aus = twitter.trends.place(_id = australia)
results_us = twitter.trends.place(_id = us)
results_in = twitter.trends.place(_id = india)

twitter_trends_world=[]
twitter_trends_can=[]
twitter_trends_aus=[]
twitter_trends_us=[]
twitter_trends_ind=[]
```

```
for location in results_world:
    for trend in location["trends"]:
        twitter_trends_world.append('#'+trend['name'])

for location in results_can:
    for trend in location["trends"]:
        twitter_trends_can.append(trend['name'])

for location in results_aus:
    for trend in location["trends"]:
        twitter_trends_aus.append(trend['name'])

for location in results_us:
    for trend in location["trends"]:
        twitter_trends_us.append(trend['name'])

for location in results_in:
    for trend in location["trends"]:
        twitter_trends_ind.append(trend['name'])
```

```
Hi there! We're gonna get you all set up to use hastag extracter.
Opening: https://api.twitter.com/oauth/authorize?oauth_token=ZbF47QAAAAABF86-AAABc-LUgg4


In the web browser window that opens please choose to Allow
access. Copy the PIN number that appears on the next page and paste or
type it here:

Please enter the PIN: 6128250
```

- Code snippet to append twitter trends to newsAPI extracted trends to generate the final list.

```
world_df=pd.DataFrame(twitter_trends_world,columns=['trends'])


twitter_trends_can=pd.DataFrame(twitter_trends_can,columns=['trends'])
#can_df.head()
#can_df=can_df.drop(columns=['Occurence'])
can_df.append(twitter_trends_can)
can_df.reset_index(drop=True)

twitter_trends_aus=pd.DataFrame(twitter_trends_aus,columns=['trends'])
#aus_df.head()
#aus_df=aus_df.drop(columns=['Occurence'])
aus_df.append(twitter_trends_aus)
aus_df.reset_index(drop=True)


twitter_trends_us=pd.DataFrame(twitter_trends_us,columns=['trends'])
#us_df.head()
#us_df=us_df.drop(columns=['Occurence'])
us_df.append(twitter_trends_us)
us_df.reset_index(drop=True)


twitter_trends_ind=pd.DataFrame(twitter_trends_ind,columns=['trends'])
twitter_trends_ind.head()
#ind_df.head()
#ind_df=ind_df.drop(columns=['Occurence'])
ind_df.append(twitter_trends_ind)
ind_df.reset_index(drop=True)
print("twitter trends appended")
```

- Finally a simulation of how user chooses at what level he chooses to see the trends. Here, it is considered the user belongs to Rajasthan:

  1) World level (shows top 20 trends):

```
print('if the user chooses world trends:')
world_df.head(20)
```

```
if the user chooses world trends:
```

	trends
0	#Neymar
1	##雪花ラミィ初配信
2	##galatakulesi
3	#聡ちゃん
4	##StopSoros
5	##ProphetForAll
6	##エビが出てくるかもしれないボタン
7	#流星群
8	#Adulto Ney
9	#SeniSeviyorum Hatice
10	#bambam
11	#Stonehaven
12	#Richarlison
13	#流れ星
14	##あなたの恋愛スタイル診断
15	##精神年齢診断
16	##昭和の死語クイズ
17	##東大王
18	##YouthDay
19	##FelizMiercoles

2) National level (shows top 20 trends):

```
print('if the user chooses india only trends:')
ind_df.head(20)
```

```
if the user chooses india only trends:
```

	trends
0	#cases
1	#andhra pradesh
2	#bihar
3	#jammu
4	#deaths
5	#floods
6	#test results
7	#longding karnataka
8	#days
9	#updates
10	#kashmir
11	#assam
12	#lockdown
13	#maharashtra lawyer
14	#goa govt
15	#july
16	#telangana
17	#west bengal
18	#andaman
19	#kerala

3) Local level (Shows top 20 trends):

```
print('if the user chooses rajasthan only trends:')
raj_df.head(20)
```

```
if the user chooses rajasthan only trends:
```

	trends
0	#rajasthan lessons
1	#sc proceedings gehlot
2	#top developments rajasthan crisis
3	#cong mlas rajasthan
4	#sachin pilot
5	#rajasthan assembly justice
6	#congress truce
7	#bjp
8	#pakistan hindu migrant family
9	#jodhpur power games
10	#horse trading
11	#dead
12	#rates
13	#guv delay cm advice
14	#state hike
15	#locust swarms
16	#south africa
17	#members
18	#house
19	#rajasthan cm gehlot rajasthan cabinet