Apple vs Google: A Comparative Sentiment Analysis in Tweets Related to Apple and Google Products

Business understanding

In today's digital age, social media platforms like Twitter provide a massive amount of real-time insights into consumer opinions, especially regarding products, brand and services. For tech giants like Apple and Google, monitoring sentiment around their products can be a crucial business strategy. Understanding how consumers feel about their offerings—whether positively, negatively, or neutrally—helps inform marketing strategies, product development, and customer support.

Project overview

Business problem

Apple and Google are two of the most influential tech companies globally, with millions of users and customers who actively discuss their products on platforms like Twitter. However, manually analyzing sentiment in these discussions would be both time-consuming and impractical given the sheer volume of tweets. Therefore, automating sentiment analysis using an NLP model will allow both companies to process and understand consumer opinions quickly and at scale. This could give them a competitive edge by allowing for a timely response to customer feedback, identifying emerging issues, and improving customer satisfaction and brand loyalty

Project objectives

Main Objective

- The primary objective is to develop an NLP-based machine learning model that can accurately classify the sentiment of tweets related to Apple and Google products as positive, negative, or neutral.

Specific Objectives

- Preprocess tweet data effectively (removing noise, handling stopwords, tokenization) to improve model accuracy.

- Build and evaluate multiple machine learning and deep learning models (Logistic Regression, Random Forest, Neural Network, LSTM) to classify tweet sentiments.
- Identify the best performing models and give recommendations.

Justification

Understanding sentiment at scale is critical for organizations in highly competitive industries like tech. By automating the analysis of millions of tweets related to Apple and Google products, companies can quickly gauge customer satisfaction, identify emerging trends, and proactively address negative sentiments. Sentiment analysis also allows these companies to measure the impact of new product releases, marketing campaigns, and public relations efforts.

Research questions

- What preprocessing techniques significantly improve model accuracy on tweet text?
- Which machine learning or deep learning model performs best in classifying tweet sentiments?
- How do different sentiment classes (positive, negative, neutral/no emotion) perform across models?

Data understanding

Data Collection

The dataset for this project comes from CrowdFlower and contains over 9,000 tweets labeled with sentiment (positive, negative, or neutral). These labeled examples provide a training set to build and evaluate the sentiment classification model. The data is enriched with various features such as tweet text, tweet ID, user ID, and the sentiment label, which is crucial for training the model.

To further improve the model, external data such as newer tweets about Apple and Google products or additional labeled sentiment datasets might be useful for retraining or fine-tuning.

Load libraries

In [ ]:
```
!pip install emoji
import pandas as pd
import re
import json
import emoji
import string
import nltk
import matplotlib.pyplot as plt
import tensorflow as tf

from nltk.corpus import stopwords
from wordcloud import WordCloud
from nltk import FreqDist
from nltk.stem.wordnet import WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from nltk.tokenize import word_tokenize
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from sklearn.metrics import classification_report
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.metrics import accuracy_score, confusion_matrix,classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics

nltk.download('wordnet')
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('punkt_tab')

import warnings
warnings.filterwarnings("ignore")
```

```
Requirement already satisfied: emoji in /usr/local/lib/python3.11/dist-packages (2.1
4.1)
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
```

Load the data

In [ ]:
```
df = pd.read_csv("tweet_product_company.csv", encoding="latin-1")
df.head()
```

Out[ ]:

	tweet_text	emotion_in_tweet_is_directed_at	is_there_an_emotion_directed_at_a_brand_or_
0	.@wesley83 I have a 3G iPhone. After 3 hrs twe...	iPhone	Negative
1	@jessedee Know about @fludapp ? Awesome iPad/i...	iPad or iPhone App	Positive
2	@swonderlin Can not wait for #iPad 2 also. The...	iPad	Positive
3	@sxsw I hope this year's festival isn't as cra...	iPad or iPhone App	Negative
4	@sxtxstate great stuff on Fri #SXSW: Marissa M...	Google	Positive

In [ ]:
```
df.tail()
```

Out[ ]:

	tweet_text	emotion_in_tweet_is_directed_at	is_there_an_emotion_directed_at
9088	Ipad everywhere. #SXSW {link}	iPad	
9089	Wave, buzz... RT @mention We interrupt your re...	NaN	No emotion to
9090	Google's Zeiger, a physician never reported po...	NaN	No emotion to
9091	Some Verizon iPhone customers complained their...	NaN	No emotion to
9092	Ï¡Ïàü_ÊÎÒ£Áââ_£â_ÛâRT @...	NaN	No emotion to

In [ ]:
```
df.sample(5)
```

Out[ ]:

	tweet_text	emotion_in_tweet_is_directed_at	is_there_an_emotion_directed_at_a_
7655	#sxsw=appreciation for Twitter. My account has...	NaN	No emotion towar
5217	RT @mention #Apple to Open Pop-Up Shop at #SXS...	NaN	No emotion towar
7432	{link} Report: Apple to Open Pop-Up Store at #...	NaN	No emotion towar
8246	Excitement in the social network space with Go...	Other Google product or service	
2626	Slides are available on the @mention site for ...	NaN	No emotion towar

Data cleaning

Correct formats

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9093 entries, 0 to 9092
Data columns (total 3 columns):
 #   Column                                              Non-Null Count  Dtype 
---  ------                                              --------------  ----- 
 0   tweet_text                                          9092 non-null   object
 1   emotion_in_tweet_is_directed_at                     3291 non-null   object
 2   is_there_an_emotion_directed_at_a_brand_or_product  9093 non-null   object
dtypes: object(3)
memory usage: 213.2+ KB
```

- **tweet_text**: The data type `object` is appropriate as it contains text data.
- **emotion_in_tweet_is_directed_at**: The `object` type is suitable since it likely contains categorical or string values indicating emotion direction.
- **is_there_an_emotion_directed_at_a_brand_or_product**: The `object` type is correct as it likely holds categorical values (e.g., "Yes" or "No").

All columns have the correct data type (`object`) for their respective content.

Missing Values

In [ ]:
```
df.isna().sum()
```

Out[ ]:

	0
tweet_text	1
emotion_in_tweet_is_directed_at	5802
is_there_an_emotion_directed_at_a_brand_or_product	0

dtype: int64

- The dataset has a total of 9093 rows.
- The **tweet_text** is the primary feature and is almost complete.
- The other two features have missing values and might require further preprocessing before use in sentiment analysis or classification tasks.

Removed blank row with the missing value in **tweet_text** since there is no tweet to draw sentiment from.

In [ ]:
```
df.dropna(subset=["tweet_text"],inplace=True)
df.isna().sum()
```

Out[ ]:

	0
tweet_text	0
emotion_in_tweet_is_directed_at	5801
is_there_an_emotion_directed_at_a_brand_or_product	0

dtype: int64

Replaced missing values on column **emotion_in_tweet_is_directed_at** with `'Unknown'` : Since we have over 60% of missing values in the ' emotion is directed at' column, we will assume the sentiments weren't directed toward any specific brand and fill the missing data with 'Unknown'.

In [ ]:
```
product_names = df.emotion_in_tweet_is_directed_at.unique()
df.emotion_in_tweet_is_directed_at.fillna("Unknown",inplace=True)
df.isna().sum()
```

Out[ ]:

	0
tweet_text	0
emotion_in_tweet_is_directed_at	0
is_there_an_emotion_directed_at_a_brand_or_product	0

dtype: int64

Keeping the original text

In [ ]:
```
df["original_tweet"]= df.tweet_text
```

Feature Engineering

Removing unwanted Text

In [ ]:
```
# cleaning unwanted characters
def remove_unwanted_text(text):
    if isinstance(text, str):
        return re.sub(r'[^\x00-\x7F]+', '', text)
    return text


df['tweet_text'] = df['tweet_text'].apply(remove_unwanted_text)

df.tweet_text
```

Out[ ]:

	tweet_text
0	.@wesley83 I have a 3G iPhone. After 3 hrs twe...
1	@jessedee Know about @fludapp ? Awesome iPad/i...
2	@swonderlin Can not wait for #iPad 2 also. The...
3	@sxsw I hope this year's festival isn't as cra...
4	@sxtxstate great stuff on Fri #SXSW: Marissa M...
...	...
9088	Ipad everywhere. #SXSW {link}
9089	Wave, buzz... RT @mention We interrupt your re...
9090	Google's Zeiger, a physician never reported po...
9091	Some Verizon iPhone customers complained their...
9092	___RT @mention Google Tests Check-in Offers At...

9092 rows × 1 columns

dtype: object

Lowercasing

Convert all `tweet_text` to lowercase to maintain consistency

In [ ]:
```
df.tweet_text = df.tweet_text.str.lower()
```

We did Lowercasing to make texts uniform

Converting Transcript to strings

In [ ]:
```
def transcription_to_strings(df):
    # matches basic sad face :(, :-(
    sad_face = re.compile(r'[:;8]?[\'\-]?\([/\\]')

    # matches crying face :'(, :'-(
    crying_face = re.compile(r'[:;8]?[\'\-]?\'\(')

    # matches complex sad faces with slashes, tears, etc.
    complex_sad_face = re.compile(r'[:;8]?[\'\-]?\([/\\]\)?')

    # edge case: faces with tears or creative symbols like T_T, TT_TT
    crying_edge_case = re.compile(r'(T_T|tt_t|TT_TT)')

    # apply pattern replacement across the DataFrame's text column
    df['tweet_text'] = df['tweet_text'].apply(lambda x:
        crying_edge_case.sub('crying face',
```

```
        complex_sad_face.sub('complex sad face',
        crying_face.sub('crying face',
        sad_face.sub('sad face', x)))))

transcription_to_strings(df)
```

- Translated emoticons into phrases like 'sad face', 'crying face' to preserve emotional context for sentimental analysis

Dealing with emojis

In [ ]:
```
for index in range(df.tweet_text.shape[0]):
    df.tweet_text.iloc[index] = emoji.demojize(df.tweet_text.iloc[index])
```

- Changed emojis into text formart since emojis carry alot of meaning and helps models learn them.

Replacing abbreviation with its full form

In [ ]:
```
# Loading abbreviation dictionary from abbr.txt
with open('abbr.txt', 'r') as file:
    abbr_dict = json.load(file)

# expand abbreviations in a tweet
def expand_abbr_in_tweet(text):
    return " ".join([abbr_dict.get(word, word) for word in text.split()])

# apply the function to each tweet in the DataFrame
df['tweet_text'] = df['tweet_text'].apply(expand_abbr_in_tweet)
```

- Replaced abreviation with its full form so as they won't be removed as stopwords

Removing Links

First check if there is any links

In [ ]:
```
def text_processor(process, pattern):
    def extract_pattern(text):
        if isinstance(text, str):
            return re.findall(pattern, text)
        return []

    # apply pattern extraction
    links_list = df['tweet_text'].apply(extract_pattern)

    # flatten the list of lists
    all_links = [link for sublist in links_list for link in sublist]

    # check if anything was found
```

```
    if all_links:
        print(f"{process} found in the dataset: {len(all_links)} matches")
        print(all_links[:5])  # optional: show a few samples
    else:
        print(f"No {process} found.")
```

In [ ]:
```
def remove_pattern(pattern):
    df['tweet_text'] = df['tweet_text'].apply(lambda x: re.sub(pattern, '', x))
```

In [ ]:
```
to_remove_or_check = "Links"
pattern = r'http\S+|www\S+'
text_processor(to_remove_or_check,pattern)
```

```
Links found in the dataset: 48 matches
['http://ht.ly/49n4m', 'http://bit.ly/ieavob', 'http://bit.ly/gvlrin', 'http://j.mp/grn7pk)', 'http://bit.ly/axzwxb']
```

Removing links found

In [ ]:
```
remove_pattern(pattern)
```

Checking if any links are remaining

In [ ]:
```
text_processor(to_remove_or_check,pattern)
```

```
No Links found.
```

Removing Usernames

Checking for twitter usernames

In [ ]:
```
to_remove_or_check = "Usernames"
pattern = r'@\S+'
text_processor(to_remove_or_check,pattern)
```

```
Usernames found in the dataset: 7192 matches
['@wesley83', '@jessedee', '@fludapp', '@swonderlin', '@sxsw']
```

Removing Usernames

In [ ]:
```
remove_pattern(pattern)
```

Checking if any usernames are remaining

In [ ]:
```
text_processor(to_remove_or_check,pattern)
```

```
No Usernames found.
```

Removing Hashtags

Let us first check if there is any hashtags

In [ ]:
```
to_remove_or_check = "Hashtags"
pattern = r'#\S+'
text_processor(to_remove_or_check,pattern)
```

```
Hashtags found in the dataset: 15853 matches
['#rise_austin,', '#sxsw.', '#sxsw', '#ipad', '#sxsw.']
```

Removing hashtags

In [ ]:
```
remove_pattern(pattern)
```

In [ ]:
```
text_processor(to_remove_or_check,pattern)
```

```
No Hashtags found.
```

Removing Punctuation

In [ ]:
```
# removeD punctuation
def remove_punctuation(text):
    return text.translate(str.maketrans('', '', string.punctuation))

# applying the function to the tweet_text column
df['tweet_text'] = df['tweet_text'].apply(remove_punctuation)
```

Removing Stopwords

In [ ]:
```
product_names = list(product_names)
```

In [ ]:
```
#adding product names when there combined to accomodate such instances
product_names_joined = [str(name1).lower() + str(name2).lower() for name1 in produc
                        for name2 in product_names if not pd.isna(name1) or not pd.
product_names.extend(product_names_joined)
# converting all product names to lowercase
product_names= [str(name).lower() for name in product_names]
```

In [ ]:
```
# getting stopwords
stopwords_list = stopwords.words('english')

#removing punctuation from stopwords since punctuation have been removed
stopwords_list= [word.translate(str.maketrans('', '', string.punctuation)) for word

# extending the list with lowercase product names
stopwords_list.extend(product_names)
```

In [ ]:
```
for index in range(df.tweet_text.shape[0]):
    tweet_text = df.tweet_text.iloc[index].split()
    #removing stopwords from text
    tweet_no_stopword = " ".join([word for word in tweet_text if word not in stopwo
    df.tweet_text.iloc[index] = tweet_no_stopword
```

Removing Numbers

In [ ]:
```
for index in range(df.tweet_text.shape[0]):
    tweet = df.tweet_text.iloc[index].split()
```

```
    df.tweet_text.iloc[index] = " ".join([re.sub(r'\b\d+\b', '', text) for text in
```

In [ ]:
```
# Comparing cleaned and original data
df[["tweet_text","original_tweet"]].head(10)
```

Out[ ]:

	tweet_text	original_tweet
0	3g hrs tweeting dead need upgrade plugin stat...	.@wesley83 I have a 3G iPhone. After 3 hrs twe...
1	know awesome application likely appreciate des...	@jessedee Know about @fludapp ? Awesome iPad/i...
2	wait also sale	@swonderlin Can not wait for #iPad 2 also. The...
3	hope years festival crashy years app	@sxsw I hope this year's festival isn't as cra...
4	great stuff fri marissa mayer tim oreilly tech...	@sxtxstate great stuff on Fri #SXSW: Marissa M...
5	new applications communication showcased confe...	@teachntech00 New iPad Apps For #SpeechTherapy...
7	starting around corner hop skip jump good time...	#SXSW is just starting, #CTIA is around the co...
8	beautifully smart simple idea retweet wrote ap...	Beautifully smart and simple idea RT @madebyma...
9	counting days plus strong canadian dollar mean...	Counting down the days to #sxsw plus strong Ca...
10	excited meet show sprint galaxy still running	Excited to meet the @samsungmobileus at #sxsw ...

- This displays side by side comparison of raw and cleaned tweets to verify the preprocessing pipeline

In [ ]:
```
df.head()
```

Out[ ]:

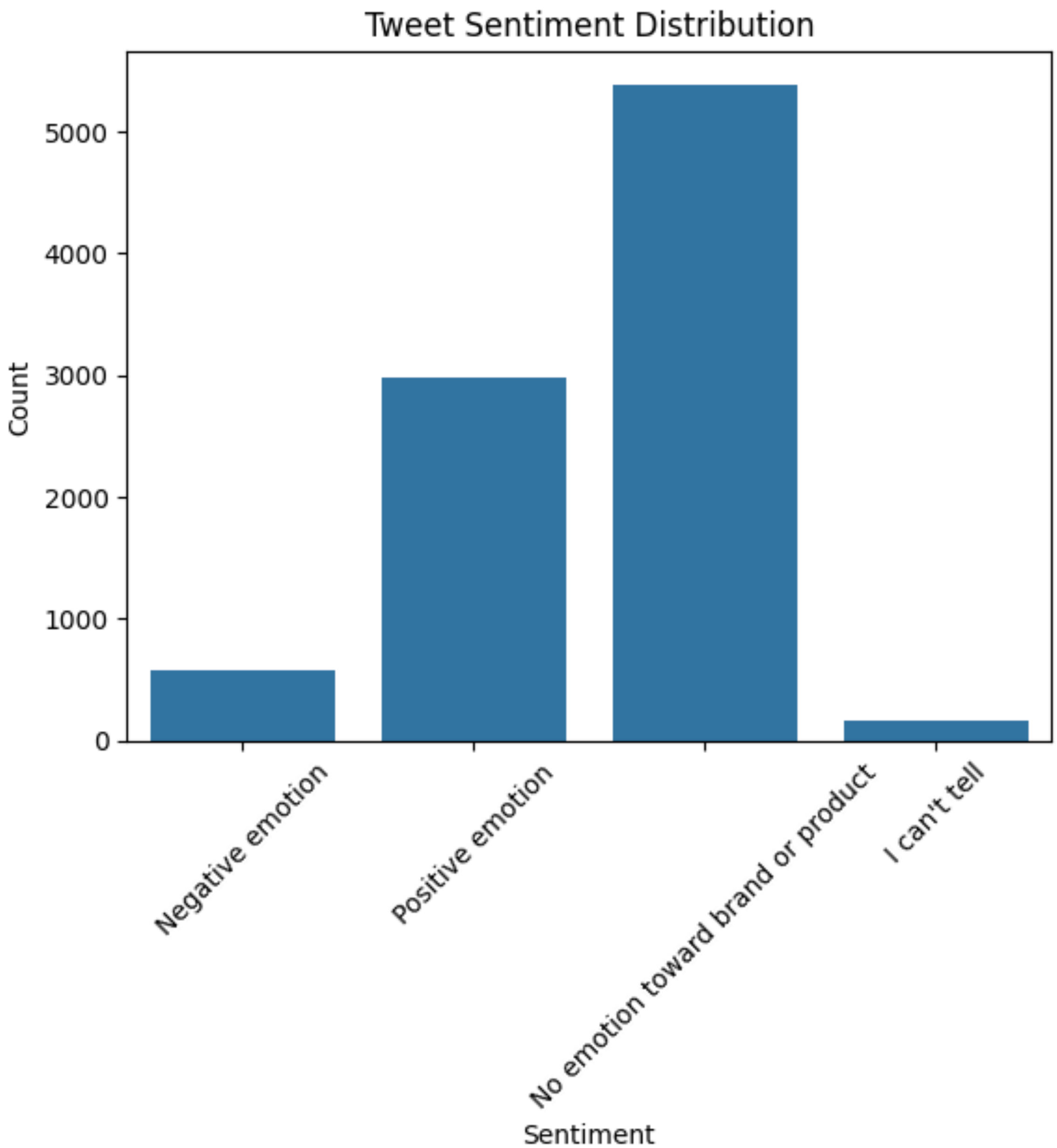
	tweet_text	emotion_in_tweet_is_directed_at	is_there_an_emotion_directed_at_a_brand_or_pr
0	3g hrs tweeting dead need upgrade plugin stat...	iPhone	Negative er
1	know awesome application likely appreciate des...	iPad or iPhone App	Positive er
2	wait also sale	iPad	Positive er
3	hope years festival crashy years app	iPad or iPhone App	Negative er
4	great stuff fri marissa mayer tim oreilly tech...	Google	Positive er

- The function is to confirm the above changes were executed successfully.

which shows that links, stopwords, punctuation, hashtags, usernames were dropped as is in tweet_text column

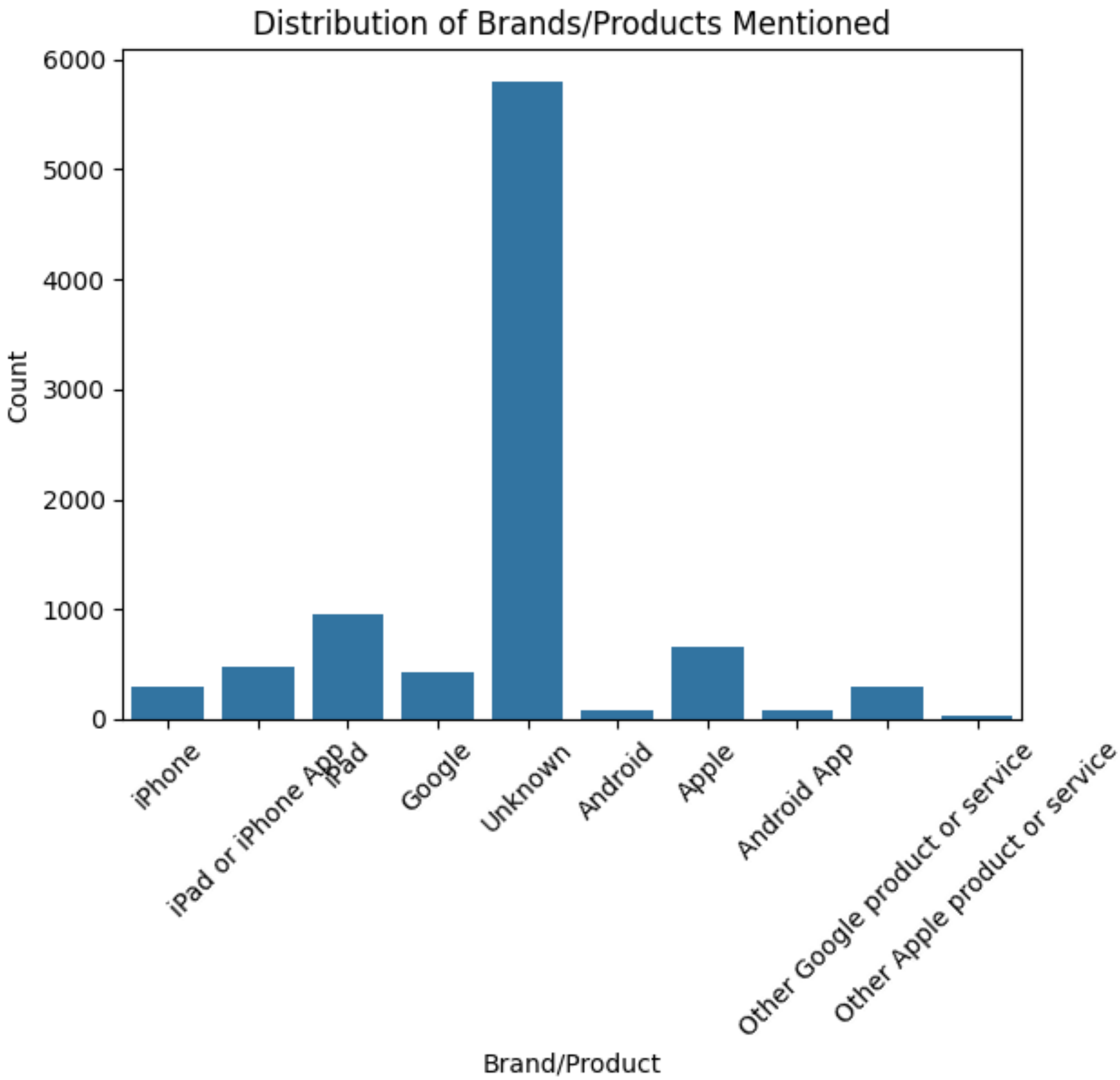
In [ ]:

```
# sentiment distribution plot
import seaborn as sns
sns.countplot(data=df, x='is_there_an_emotion_directed_at_a_brand_or_product')
plt.title('Tweet Sentiment Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



- 'No emotion towards brand or product' had the highest count which shows that most sentiments/reaction given was generalized/neutral

In [ ]:
```
# distribution of brands/products mentioned countplot
sns.countplot(data=df, x='emotion_in_tweet_is_directed_at')
plt.title('Distribution of Brands/Products Mentioned')
plt.xlabel('Brand/Product')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



- The above plot shows the frequency of each product/brand mentions.

Most of the data had missing values on the brand/product column hence filling the blanks with unknown- in this case unknown showing that the sentiments did not mention any product/brand name.

Tokenization

In [ ]:
```
for index in range(df.tweet_text.shape[0]):
    df.tweet_text.iloc[index] = word_tokenize(df.tweet_text.iloc[index])
```

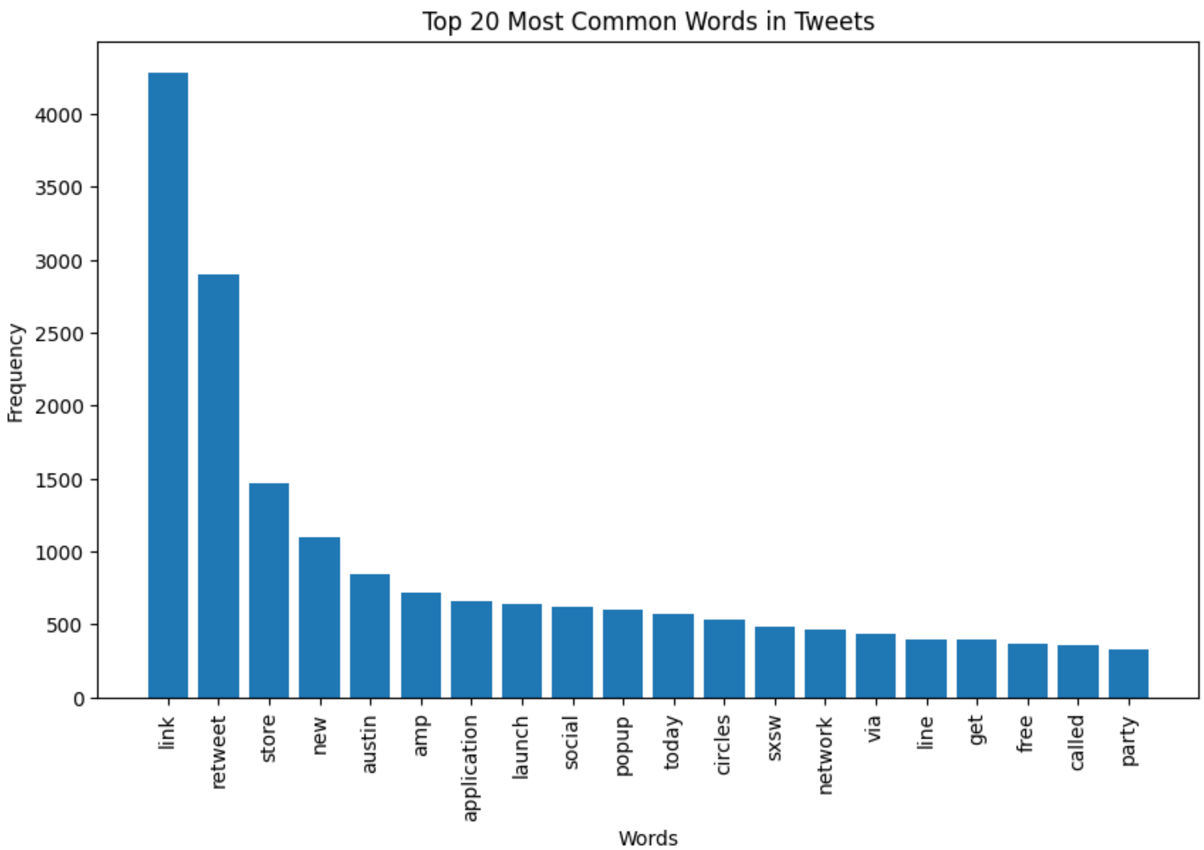
- The text was stripped down the tweet text into smaller units(tokens). Therefore allowing vectorization of the cleaned text which is a numerical representation of the tokens to capture the semantic meanings and relationship between words.

In [ ]:
```
tokens = []
for index in range(df.tweet_text.shape[0]):
    text=df.tweet_text.iloc[index]
    for word in text:tokens.append(word)
# creating frequency distribution from the tokens
freqdist = FreqDist(tokens)

# getting 200 most common words
most_common = freqdist.most_common(200)

print(most_common)
```

```
[('link', 4284), ('retweet', 2903), ('store', 1465), ('new', 1093), ('austin', 842),
('amp', 722), ('application', 655), ('launch', 643), ('social', 623), ('popup', 59
7), ('today', 576), ('circles', 529), ('sxsw', 487), ('network', 462), ('via', 435),
('line', 401), ('get', 393), ('free', 364), ('called', 361), ('party', 332), ('majo
r', 302), ('mobile', 300), ('like', 290), ('time', 272), ('one', 272), ('temporary',
264), ('opening', 256), ('people', 255), ('possibly', 244), ('great', 223), ('downto
wn', 222), ('see', 220), ('going', 218), ('day', 216), ('check', 215), ('maps', 21
4), ('go', 212), ('open', 210), ('need', 203), ('mayer', 203), ('marissa', 192), ('g
ot', 183), ('know', 182), ('googles', 182), ('come', 174), ('applications', 168),
('win', 168), ('first', 166), ('good', 165), ('us', 162), ('pop', 160), ('ipad2', 15
9), ('next', 148), ('want', 146), ('love', 145), ('cool', 143), ('panel', 142), ('sh
op', 142), ('best', 140), ('design', 138), ('app', 135), ('game', 135), ('make', 13
5), ('thanks', 135), ('news', 134), ('think', 133), ('big', 130), ('set', 128), ('se
arch', 128), ('use', 128), ('awesome', 126), ('would', 126), ('around', 125), ('las
t', 124), ('music', 123), ('users', 122), ('talk', 121), ('show', 119), ('anyone', 1
18), ('video', 118), ('using', 115), ('right', 114), ('says', 114), ('download', 11
3), ('rumor', 110), ('really', 109), ('guy', 109), ('even', 109), ('launching', 10
8), ('session', 108), ('still', 107), ('coming', 105), ('year', 104), ('location', 1
01), ('apples', 100), ('congress', 100), ('booth', 100), ('hey', 99), ('ipads', 99),
('buy', 97), ('team', 96), ('case', 96), ('future', 96), ('6th', 96), ('way', 96),
('heard', 95), ('week', 95), ('products', 95), ('cant', 89), ('tonight', 89), ('digi
tal', 89), ('twitter', 87), ('find', 86), ('everyone', 85), ('may', 84), ('blackberr
y', 84), ('fun', 84), ('phone', 84), ('thing', 83), ('back', 83), ('look', 82), ('lo
oking', 81), ('getting', 81), ('could', 81), ('nice', 80), ('also', 79), ('many', 7
9), ('2s', 79), ('away', 79), ('ever', 78), ('web', 78), ('wins', 78), ('facebook',
77), ('wait', 76), ('temp', 76), ('designing', 75), ('tv', 75), ('yes', 74), ('lon
g', 73), ('already', 73), ('giving', 71), ('quotgoogle', 71), ('bing', 71), ('includ
es', 69), ('uberguide', 69), ('interesting', 69), ('live', 69), ('fast', 68), ('muc
h', 68), ('oh', 67), ('interactive', 67), ('ready', 67), ('others', 67), ('looks', 6
6), ('take', 66), ('available', 66), ('every', 66), ('please', 66), ('sure', 65),
('tomorrow', 65), ('work', 65), ('night', 65), ('made', 65), ('smart', 63), ('sell',
63), ('friends', 63), ('product', 63), ('someone', 62), ('rt', 62), ('platform', 6
1), ('tweet', 61), ('gets', 61), ('cc', 61), ('action', 61), ('itunes', 60), ('detai
ls', 60), ('street', 60), ('battery', 60), ('keep', 60), ('else', 59), ('two', 59),
('white', 59), ('yet', 59), ('better', 59), ('years', 58), ('theres', 58), ('relie
f', 58), ('wow', 57), ('tech', 56), ('meet', 56), ('join', 56), ('thats', 56), ('tec
hnology', 56), ('saw', 56), ('updates', 55), ('post', 55), ('seen', 55), ('japan', 5
5), ('hotpot', 55), ('na', 54)]
```

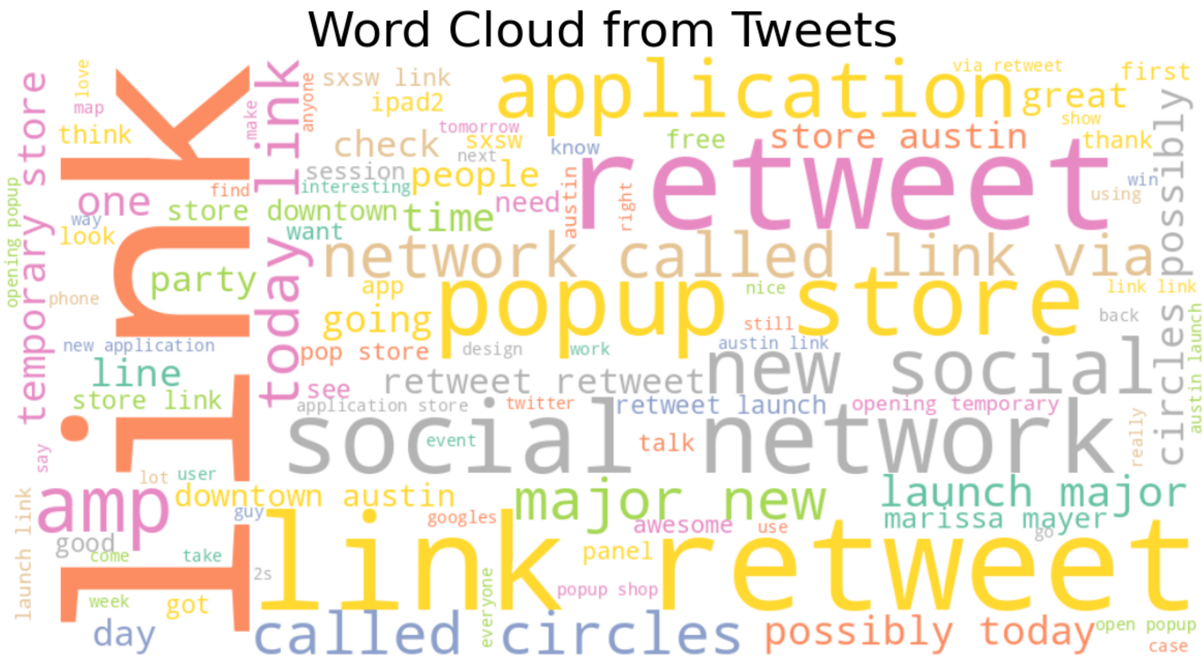
In [ ]:
```
#  top 20 most common words for visualization barplot
top_20 = freqdist.most_common(20)

# splitting the words and their frequencies into separate lists
words, frequencies = zip(*top_20)
```

```
plt.figure(figsize=(10, 6))
plt.bar(words, frequencies)
plt.xticks(rotation=90)
plt.title('Top 20 Most Common Words in Tweets')
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.show()
```



- The bar plot gives a quick idea of the common word in the tweets.

In [ ]:
```
#merging all English text into one string for the word cloud
tweet_data = ' '.join(' '.join(t) for t in df['tweet_text'])
# generating the word cloud with text data
wordcloud = WordCloud(width=1000, height=500,
                      background_color='white',
                      colormap='Set2',
                      max_words=100).generate(tweet_data)

# plotting
plt.figure(figsize=(15, 10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud from Tweets', fontsize=34);
```

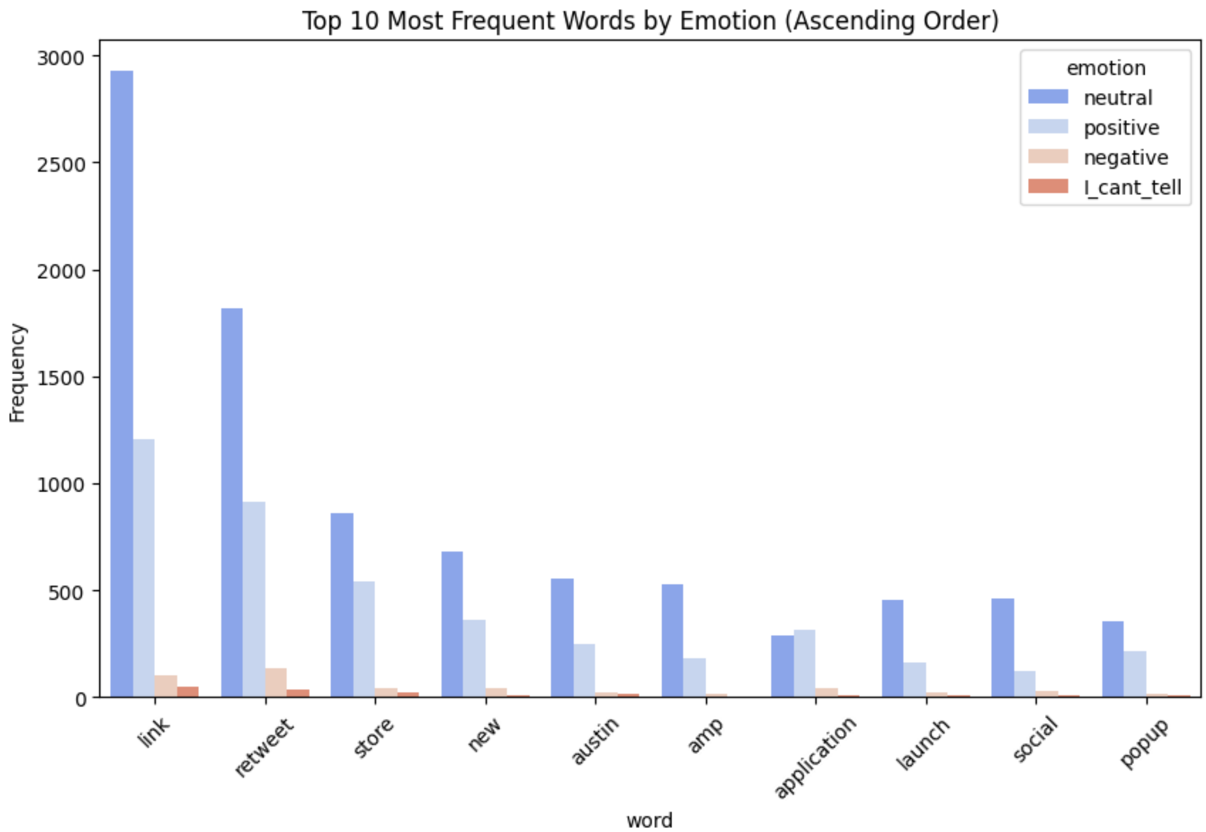


- The word cloud was used to create a visual summary of most used words.

the big fonts show the frequent used words in the tweets eg. link, link retweet, application.

In [ ]:
```
word_emotion_counts = {}

# iterating through each tweet
for index in range(df.shape[0]):
    text = df.tweet_text.iloc[index]
    emotion = df.is_there_an_emotion_directed_at_a_brand_or_product.iloc[index]

    for word in text:
        if word not in word_emotion_counts:
            word_emotion_counts[word] = [0, 0, 0, 0]  # [neutral, positive, negativ

        if emotion == 'No emotion toward brand or product':
            word_emotion_counts[word][0] += 1
        elif emotion == 'Positive emotion':
            word_emotion_counts[word][1] += 1
        elif emotion == 'Negative emotion':
            word_emotion_counts[word][2] += 1
        else :
            word_emotion_counts[word][3] += 1

word_emotion_counts
emotion_data = []
for word, counts in word_emotion_counts.items():
    emotion_data.append({'word': word, 'neutral': counts[0], 'positive': counts[1],

emotion_df = pd.DataFrame(emotion_data)

# getting the top 10 words by frequency (neutral + positive + negative counts)
top_words = emotion_df.set_index('word').sum(axis=1).nlargest(10).index
top_words_df = emotion_df[emotion_df['word'].isin(top_words)]
```

```
# reshape the dataFrame for plotting
top_words_melted = top_words_df.melt(id_vars='word', value_vars=['neutral', 'positi
                                     var_name='emotion', value_name='count')

# bar plot with hue for emotion
plt.figure(figsize=(10, 6))
sns.barplot(x='word', y='count', hue='emotion', data=top_words_melted, palette='coo
            order=top_words)
plt.title('Top 10 Most Frequent Words by Emotion (Ascending Order)')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.show()
```



- The word-sentiment mapping above was used to associate each word with how often it appears in 'positive', 'negative', 'no emotion' and 'I can't tell'

Lemmatization

In [ ]:
```
lemmatizer = WordNetLemmatizer()

for index in range(df.tweet_text.shape[0]):
    text = df.tweet_text.iloc[index]
    tweet_text =[lemmatizer.lemmatize(word) for word in text]
    df.tweet_text.iloc[index] = tweet_text
```

- Lemmatization was used to normalize words to their base forms, reduce noise and improve interpretability.

In [ ]:
```
df.head()
```

Out[ ]:

	tweet_text	emotion_in_tweet_is_directed_at	is_there_an_emotion_directed_at_a_brand_or_p
0	[3g, hr, tweeting, dead, need, upgrade, plugin...	iPhone	Negative e
1	[know, awesome, application, likely, appreciat...	iPad or iPhone App	Positive e
2	[wait, also, sale]	iPad	Positive e
3	[hope, year, festival, crashy, year, app]	iPad or iPhone App	Negative e
4	[great, stuff, fri, marissa, mayer, tim, oreil...	Google	Positive e

Vectorization/TF-IDF

In [ ]:
```
# Prepared the text
texts = [" ".join(text) for text in df.tweet_text]

# CountVectorizer
count_vect = CountVectorizer()
count_matrix = count_vect.fit_transform(texts)
count_df = pd.DataFrame(count_matrix.toarray(), columns=count_vect.get_feature_name
print("CountVectorizer:\n")
print(count_df)

# TfidfVectorizer
tfidf_vect = TfidfVectorizer()
tfidf_matrix = tfidf_vect.fit_transform(texts)
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=tfidf_vect.get_feature_name
```

```
CountVectorizer:

      0310apple  100  103011p  1045am3  10am  10k  10mins  10pm  10x  10x2  \
0             0    0        0        0     0    0       0     0    0     0
1             0    0        0        0     0    0       0     0    0     0
2             0    0        0        0     0    0       0     0    0     0
3             0    0        0        0     0    0       0     0    0     0
4             0    0        0        0     0    0       0     0    0     0
...         ...  ...      ...      ...   ...  ...     ...   ...  ...   ...
9087          0    0        0        0     0    0       0     0    0     0
9088          0    0        0        0     0    0       0     0    0     0
9089          0    0        0        0     0    0       0     0    0     0
9090          0    0        0        0     0    0       0     0    0     0
9091          0    0        0        0     0    0       0     0    0     0

      ...  zite  zlf  zms  zombie  zomg  zone  zoom  zuckerberg  zynga  zzzs
0     ...     0    0    0       0     0     0     0           0      0     0
1     ...     0    0    0       0     0     0     0           0      0     0
2     ...     0    0    0       0     0     0     0           0      0     0
3     ...     0    0    0       0     0     0     0           0      0     0
4     ...     0    0    0       0     0     0     0           0      0     0
...   ...   ...  ...  ...     ...   ...   ...   ...         ...    ...   ...
9087  ...     0    0    0       0     0     0     0           0      0     0
9088  ...     0    0    0       0     0     0     0           0      0     0
9089  ...     0    0    0       0     0     0     0           0      0     0
9090  ...     0    0    0       0     0     0     0           0      0     0
9091  ...     0    0    0       0     0     0     0           0      0     0

[9092 rows x 8553 columns]
```

- Converted tweet text into numerical features using countVectorizer and TF-IDF.

This emphasizes emotionally charged/ brand-specific terms that are rare but meaningful

In [ ]:
```
top_n = 5

print("\nTF-IDF Vectorizer (Top Words per Document):\n")
for i, row in tfidf_df.iterrows():
    doc_tfidf = row[row > 0].sort_values(ascending=False).head(top_n).round(3)
    print(f"\nDocument {i + 1}:")
    print(doc_tfidf.reset_index().rename(columns={'index': 'Word', i: 'TF-IDF'}))
```

Streaming output truncated to the last 5000 lines.

```
Document 8449:
         Word  TF-IDF
0  resurgence   0.458
1         fam   0.385
2    enjoying   0.374
3     showing   0.337
4      center   0.331

Document 8450:
        Word  TF-IDF
0  reassured   0.348
1    terrace   0.348
2    rooftop   0.323
3       soft   0.323
4    hacking   0.308

Document 8451:
        Word  TF-IDF
0        n26   0.449
1     terror   0.449
2      level   0.377
3  40075959p   0.347
4        red   0.327

Document 8452:
         Word  TF-IDF
0     torrent   0.446
1      expect   0.437
2  unofficial   0.437
3        hear   0.377
4         way   0.320

Document 8453:
       Word  TF-IDF
0     tough   0.482
1    urinal   0.472
2   holding   0.406
3        ur   0.396
4  tweeting   0.389

Document 8454:
          Word  TF-IDF
0         week   0.451
1  comfortable   0.370
2         wise   0.358
3       typing   0.330
4        sixth   0.304

Document 8455:
      Word  TF-IDF
0    index   0.343
1      dow   0.329
2     mood   0.329
3  predict   0.318
```

```
4   theory   0.318

Document 8456:
        Word  TF-IDF
0      young   0.586
1    chasing   0.293
2       shoe   0.293
3       foot   0.273
4  newspaper   0.269

Document 8457:
       Word  TF-IDF
0   tricked   0.456
1  birthday   0.396
2    either   0.389
3     guess   0.336
4       mom   0.324

Document 8458:
          Word  TF-IDF
0         slow   0.440
1  downloading   0.416
2      torrent   0.411
3       review   0.368
4     probably   0.351

Document 8459:
      Word  TF-IDF
0      net   0.460
1     shit   0.372
2  another   0.300
3       oh   0.290
4      way   0.268

Document 8460:
     Word  TF-IDF
0  peeked   0.496
1    flap   0.474
2   loose   0.448
3   built   0.348
4    look   0.270

Document 8461:
      Word  TF-IDF
0  barroom   0.355
1    brawl   0.355
2    saber   0.355
3     fist   0.355
4    light   0.296

Document 8462:
            Word  TF-IDF
0  hivethinkquot   0.436
1            mob   0.422
2         quotif   0.395
3          flash   0.355
```

```
4          there   0.294

Document 8463:
    Word  TF-IDF
0  smart   0.708
1     oh   0.706

Document 8464:
      Word  TF-IDF
0  flowing   0.492
1    water   0.399
2     near   0.372
3    still   0.284
4   anyone   0.279

Document 8465:
         Word  TF-IDF
0  journalist   0.508
1        pass   0.421
2      afford   0.410
3       spent   0.407
4       money   0.353

Document 8466:
        Word  TF-IDF
0     bother   0.593
1    setting   0.448
2      store   0.385
3     coming   0.372
4  temporary   0.309

Document 8467:
        Word  TF-IDF
0  smarmcake   0.617
1    writeup   0.617
2      thing   0.347
3        got   0.322
4       link   0.117

Document 8468:
        Word  TF-IDF
0  something   0.582
1      dunno   0.411
2       prob   0.388
3        bus   0.361
4  happening   0.356

Document 8469:
         Word  TF-IDF
0  simplicity   0.476
1          jr   0.461
2  developing   0.444
3         lot   0.334
4     product   0.298

Document 8470:
```

```
     Word  TF-IDF
0    went   0.566
1   stock   0.562
2  really   0.440
3   think   0.412

Document 8471:
    Word  TF-IDF
0  truck   0.855
1   need   0.519

Document 8472:
    Word  TF-IDF
0    pop   0.872
1  store   0.489

Document 8473:
         Word  TF-IDF
0   available   0.598
1  blackberry   0.578
2          go   0.474
3     retweet   0.227
4        link   0.180

Document 8474:
    Word  TF-IDF
0   week   0.567
1   next   0.529
2   time   0.456
3  party   0.437

Document 8475:
          Word  TF-IDF
0  boomersquot   0.430
1     quotyour   0.411
2   interested   0.386
3          mom   0.351
4      thought   0.338

Document 8476:
            Word  TF-IDF
0        changed   0.579
1      scheduler   0.354
2         prefer   0.293
3  communication   0.262
4           food   0.243

Document 8477:
     Word  TF-IDF
0     itc   0.383
1   three   0.321
2     met   0.315
3    girl   0.301
4  change   0.297

Document 8478:
```

```
      Word  TF-IDF
0      run   0.650
1  battery   0.544
2     hour   0.531

Document 8479:
     Word  TF-IDF
0    math   0.606
1    wave   0.525
2    buzz   0.443
3     day   0.306
4  circle   0.258

Document 8480:
         Word  TF-IDF
0       canal   0.355
1    flooding   0.355
2  deathstarr   0.339
3         adi   0.339
4       ringo   0.329

Document 8481:
          Word  TF-IDF
0   enthusiast   0.478
1  6thcongress   0.457
2       lining   0.422
3          5pm   0.363
4      outside   0.322

Document 8482:
         Word  TF-IDF
0  hipstapaks   0.747
1      giving   0.463
2        away   0.456
3        link   0.141

Document 8483:
      Word  TF-IDF
0      dst   0.401
1  noticed   0.401
2   sunday   0.336
3     late   0.298
4  morning   0.283

Document 8484:
       Word  TF-IDF
0     615ab   0.334
1  saturday   0.317
2   feeling   0.314
3       sat   0.311
4   welcome   0.308

Document 8485:
       Word  TF-IDF
0     guard   0.520
1  enjoying   0.495
```

```
2  security   0.469
3     ipad2   0.350
4    austin   0.236

Document 8486:
      Word  TF-IDF
0  arrived   0.553
1     nice   0.427
2     shop   0.385
3      pop   0.377
4     time   0.331

Document 8487:
         Word  TF-IDF
0  apparently   0.656
1       place   0.561
2         get   0.386
3      austin   0.325

Document 8488:
         Word  TF-IDF
0    pornthis   0.483
1      latina   0.471
2  impression   0.433
3        find   0.312
4       first   0.277

Document 8489:
         Word  TF-IDF
0    pornthis   0.483
1      latina   0.471
2  impression   0.433
3        find   0.312
4       first   0.277

Document 8490:
      Word  TF-IDF
0  waiting   0.664
1    prize   0.394
2     stop   0.329
3  someone   0.319
4    still   0.294

Document 8491:
       Word  TF-IDF
0  tweeting   0.909
1       new   0.417

Document 8492:
    Word  TF-IDF
0    2am   0.420
1  lobby   0.380
2  hotel   0.330
3   yeah   0.328
4    who   0.320
```

```
Document 8493:
           Word  TF-IDF
0       crucial   0.537
1  experimenting   0.523
2         brand   0.411
3           say   0.308
4       marissa   0.298

Document 8494:
       Word  TF-IDF
0     doubt   0.318
1     speak   0.306
2    useful   0.295
3       min   0.291
4  tweeting   0.286

Document 8495:
      Word  TF-IDF
0  disrupt   0.510
1  putting   0.437
2   tablet   0.377
3     wait   0.338
4     cant   0.329

Document 8496:
         Word  TF-IDF
0  convenient   0.525
1       crazy   0.408
2   brilliant   0.400
3     setting   0.396
4        sell   0.355

Document 8497:
        Word  TF-IDF
0        fix   0.517
1    setting   0.436
2   attendee   0.414
3      ipad2   0.333
4  temporary   0.301

Document 8498:
      Word  TF-IDF
0  society   0.400
1  texting   0.394
2      war   0.360
3     heat   0.360
4    group   0.317

Document 8499:
          Word  TF-IDF
0  grouptexting   0.450
1       society   0.414
2           war   0.372
3          heat   0.372
4          fast   0.303
```

```
Document 8500:
      Word  TF-IDF
0     eric   0.613
1  sitting   0.506
2    ready   0.425
3  getting   0.412
4     link   0.128

Document 8501:
       Word  TF-IDF
0    riding   0.658
1      thru   0.581
2  tweeting   0.479

Document 8502:
        Word  TF-IDF
0       attn   0.649
1   downtown   0.350
2    opening   0.340
3  temporary   0.338
4     launch   0.268

Document 8503:
       Word  TF-IDF
0  unveiled   0.494
1      dear   0.436
2      soon   0.393
3      tell   0.373
4    please   0.353

Document 8504:
         Word  TF-IDF
0       answr    0.41
1        porn    0.41
2        askd    0.41
3      latina    0.37
4  impression    0.34

Document 8505:
        Word  TF-IDF
0   downtown   0.472
1    opening   0.458
2  temporary   0.456
3     launch   0.361
4     austin   0.341

Document 8506:
        Word  TF-IDF
0   banality   0.570
1     bloody   0.545
2  smackdown   0.532
3        via   0.282
4       link   0.124

Document 8507:
        Word  TF-IDF
```

```
0  techstars   0.664
1    advance   0.600
2       help   0.428
3       link   0.126

Document 8508:
         Word  TF-IDF
0      btchin   0.510
1       throw   0.494
2  spazmatics   0.481
3       shout   0.455
4       party   0.240

Document 8509:
       Word  TF-IDF
0     train   0.457
1  opposite   0.457
2   linking   0.448
3   article   0.389
4      easy   0.386

Document 8510:
       Word  TF-IDF
0  pointing   0.470
1     broke   0.443
2       hmm   0.407
3   article   0.377
4     story   0.340

Document 8511:
          Word  TF-IDF
0          15k   0.622
1    developed   0.606
2          wow   0.431
3  application   0.246

Document 8512:
        Word  TF-IDF
0        jet   0.455
1      freak   0.420
2  hilarious   0.412
3   learning   0.365
4      stuff   0.328

Document 8513:
          Word  TF-IDF
0          11m   0.435
1  bitlyea1zgd   0.416
2       valley   0.384
3        raise   0.365
4       player   0.339

Document 8514:
         Word  TF-IDF
0   apartment   0.435
1  streetview   0.407
```

```
2       click   0.385
3        easy   0.358
4      austin   0.339

Document 8515:
       Word  TF-IDF
0       god   0.476
1        oh   0.410
2     rumor   0.362
3  downtown   0.325
4   opening   0.315

Document 8516:
      Word  TF-IDF
0      use   0.484
1  marissa   0.446
2    mayer   0.436
3      map   0.427
4   mobile   0.410

Document 8517:
          Word  TF-IDF
0    sponsored   0.523
1    uberguide   0.496
2     includes   0.496
3  application   0.291
4          new   0.266

Document 8518:
          Word  TF-IDF
0   perception   0.409
1  transparent   0.409
2      brother   0.335
3      control   0.303
4       trying   0.275

Document 8519:
              Word  TF-IDF
0  freecreditscore   0.341
1              liz   0.316
2            phair   0.316
3        crossroad   0.316
4           closer   0.308

Document 8520:
          Word  TF-IDF
0  identifying   0.435
1        apply   0.403
2         bird   0.384
3        quick   0.352
4          mom   0.308

Document 8521:
          Word  TF-IDF
0         cont   0.471
1           cc   0.442
```

```
2     includes   0.433
3    uberguide   0.433
4  application   0.255

Document 8522:
         Word  TF-IDF
0  accessible   0.529
1        site   0.403
2      making   0.399
3        tell   0.399
4       ready   0.374

Document 8523:
        Word  TF-IDF
0         8a   0.348
1  answering   0.322
2      guest   0.316
3    charles   0.298
4       chen   0.298

Document 8524:
         Word  TF-IDF
0        edge   0.357
1      protip   0.346
2   perfectly   0.337
3  connection   0.324
4      signal   0.302

Document 8525:
               Word  TF-IDF
0           cheapen   0.326
1       productquot   0.326
2           quotbut   0.326
3     quotmultiple   0.326
4  monetizationquot   0.326

Document 8526:
           Word  TF-IDF
0       duckett   0.374
1  instrumental   0.374
2        common   0.330
3      creative   0.314
4          name   0.285

Document 8527:
         Word  TF-IDF
0    latitude   0.482
1    checkins   0.448
2      reward   0.415
3  foursquare   0.400
4      follow   0.398

Document 8528:
           Word  TF-IDF
0  breakthrough   0.561
1      latitude   0.450
```

```
2          push   0.441
3           big   0.312
4          year   0.303

Document 8529:
         Word  TF-IDF
0  activation   0.509
1      market   0.381
2       month   0.377
3        huge   0.370
4       share   0.361

Document 8530:
     Word  TF-IDF
0    lego   0.473
1  battle   0.461
2   robot   0.455
3  hosted   0.451
4   start   0.378

Document 8531:
      Word  TF-IDF
0    haiti   0.429
1   sister   0.398
2  mission   0.379
3   winner   0.356
4     trip   0.327

Document 8532:
        Word  TF-IDF
0  executing   0.554
1     loaded   0.530
2   congrats   0.401
3      guide   0.363
4      event   0.329

Document 8533:
           Word  TF-IDF
0  pengairborne   0.488
1          fave   0.435
2         level   0.429
3      congrats   0.369
4       getting   0.310

Document 8534:
       Word  TF-IDF
0   raffled   0.510
1     inbox   0.481
2   winning   0.419
3  congrats   0.386
4    detail   0.334

Document 8535:
        Word  TF-IDF
0         aw   0.501
1  boyfriend   0.478
```

```
2    winning   0.425
3   congrats   0.391
4       case   0.307

Document 8536:
         Word  TF-IDF
0    redcross   0.421
1      afford   0.368
2    consider   0.360
3      attend   0.351
4  earthquake   0.349

Document 8537:
     Word  TF-IDF
0      en   0.497
1     bet   0.443
2  flight   0.424
3   route   0.402
4   thats   0.369

Document 8538:
          Word  TF-IDF
0  accompanied   0.383
1      haircut   0.383
2   instrument   0.361
3       fedora   0.354
4           en   0.347

Document 8539:
    Word  TF-IDF
0   jose   0.351
1    bff   0.336
2  charm   0.325
3  dream   0.305
4    san   0.295

Document 8540:
      Word  TF-IDF
0  plagued   0.619
1   stream   0.448
2  twitter   0.371
3  product   0.336
4    first   0.329

Document 8541:
       Word  TF-IDF
0   florian   0.405
1     bernd   0.405
2   guguchu   0.405
3  terminal   0.375
4    ticket   0.340

Document 8542:
      Word  TF-IDF
0  stretch   0.595
1    block   0.394
```

```
2  getting   0.378
3      buy   0.360
4      one   0.290

Document 8543:
      Word  TF-IDF
0  compete   0.573
1   bigger   0.499
2    start   0.444
3      guy   0.357
4     like   0.312

Document 8544:
            Word  TF-IDF
0  douchebaggery   0.439
1       patience   0.439
2           lack   0.407
3        popping   0.369
4          thats   0.283

Document 8545:
          Word  TF-IDF
0         last   0.505
1          day   0.435
2         free   0.408
3          get   0.384
4  application   0.329

Document 8546:
       Word  TF-IDF
0  advisory   0.374
1   officer   0.374
2  thrilled   0.353
3     board   0.346
4     chief   0.346

Document 8547:
       Word  TF-IDF
0    excuse   0.476
1   walking   0.391
2  attendee   0.365
3      temp   0.336
4       6th   0.323

Document 8548:
      Word  TF-IDF
0    diary   0.404
1  memento   0.404
2  momento   0.386
3   create   0.309
4       io   0.281

Document 8549:
       Word  TF-IDF
0  commuter   0.431
1      juan   0.412
```

```
2       gift   0.349
3    contest   0.349
4    winning   0.339

Document 8550:
        Word  TF-IDF
0      kenny   0.455
1   spanking   0.455
2     enjoys   0.440
3   congrats   0.344
4      happy   0.319

Document 8551:
        Word  TF-IDF
0      kenny   0.455
1   spanking   0.455
2     enjoys   0.440
3   congrats   0.344
4      happy   0.319

Document 8552:
           Word  TF-IDF
0   independent   0.406
1    innovative   0.366
2            ta   0.309
3          film   0.304
4         later   0.302

Document 8553:
              Word  TF-IDF
0      quotgroupon   0.487
1   socialtypequot   0.483
2           living   0.448
3           reward   0.431
4              via   0.268

Document 8554:
              Word  TF-IDF
0      quotgroupon   0.506
1   socialtypequot   0.501
2           living   0.465
3           reward   0.448
4           launch   0.248

Document 8555:
           Word  TF-IDF
0            jl   0.575
1   interesting   0.375
2      possibly   0.294
3        called   0.269
4           via   0.258

Document 8556:
              Word  TF-IDF
0      quotgroupon   0.506
1   socialtypequot   0.501
```

```
2          living   0.465
3          reward   0.448
4          launch   0.248

Document 8557:
       Word  TF-IDF
0       cal   0.582
1  starting   0.423
2   excited   0.407
3    really   0.350
4      make   0.320

Document 8558:
       Word  TF-IDF
0  bathroom   0.409
1   serious   0.409
2      prop   0.399
3    urinal   0.390
4      keep   0.279

Document 8559:
        Word  TF-IDF
0     11quot   0.396
1       n00b   0.396
2  carefully   0.379
3   studying   0.367
4   quotbest   0.343

Document 8560:
        Word  TF-IDF
0        use   0.537
1       open   0.470
2  temporary   0.463
3       sxsw   0.401
4      store   0.288

Document 8561:
          Word  TF-IDF
0  sweepstakes   0.474
1     purchase   0.369
2        early   0.349
3         soon   0.333
4     giveaway   0.323

Document 8562:
        Word  TF-IDF
0       dell   0.400
1  otherwise   0.382
2         pc   0.362
3        pro   0.351
4    macbook   0.328

Document 8563:
       Word  TF-IDF
0    worked   0.608
1      sold   0.468
```

```
2  attendee   0.466
3      sell   0.440

Document 8564:
       Word  TF-IDF
0      warp   0.503
1       5th   0.436
2        st   0.383
3       6th   0.296
4  congress   0.294

Document 8565:
        Word  TF-IDF
0  speakeasy   0.494
1       food   0.428
2      drink   0.421
3     coming   0.362
4        got   0.325

Document 8566:
      Word  TF-IDF
0       va   0.410
1  hottest   0.370
2  discovr   0.340
3  kicking   0.336
4   client   0.329

Document 8567:
       Word  TF-IDF
0  planning   0.473
1    player   0.439
2     crowd   0.378
3     might   0.365
4      wait   0.346

Document 8568:
          Word  TF-IDF
0       cierto   0.468
1          ser   0.453
2  rwwtof6bcet   0.432
3     possibly   0.239
4        major   0.229

Document 8569:
              Word  TF-IDF
0  foreshadowing   0.524
1    quotscarquot   0.524
2    autocorrects   0.475
3    quotsxswquot   0.475

Document 8570:
       Word  TF-IDF
0      spin   0.668
1  released   0.292
2     issue   0.288
3  magazine   0.288
```

```
4      play   0.269

Document 8571:
         Word  TF-IDF
0     chalked   0.489
1  quottattoo   0.489
2    pavement   0.489
3    ipadquot   0.401
4        mean   0.349

Document 8572:
        Word  TF-IDF
0      chief   0.357
1      wiley   0.357
2   engineer   0.350
3  presenter   0.350
4     native   0.324

Document 8573:
          Word  TF-IDF
0         ipod   0.517
1         cont   0.449
2     includes   0.413
3    uberguide   0.413
4  application   0.243

Document 8574:
            Word  TF-IDF
0        toolkit   0.362
1            rei   0.362
2  laptopcharger   0.362
3           bike   0.327
4            h2o   0.327

Document 8575:
        Word  TF-IDF
0  schooling   0.477
1       exec   0.477
2       cnet   0.386
3      story   0.331
4  marketing   0.311

Document 8576:
        Word  TF-IDF
0  makeshift   0.638
1    totally   0.615
2   downtown   0.398
3      store   0.239

Document 8577:
      Word  TF-IDF
0    omaha   0.431
1  rematch   0.431
2     crap   0.389
3  digging   0.374
4  totally   0.333
```

```
Document 8578:
       Word  TF-IDF
0    seemed   0.381
1  offering   0.345
2      beat   0.335
3       sad   0.335
4    number   0.330

Document 8579:
           Word  TF-IDF
0  quottouching   0.629
1      magazine   0.484
2         story   0.455
3     designing   0.404

Document 8580:
          Word  TF-IDF
0  anywaysquot   0.532
1      quotnot   0.461
2        force   0.441
3       hotpot   0.345
4      service   0.334

Document 8581:
       Word  TF-IDF
0  activity   0.554
1        tx   0.465
2    hotpot   0.414
3      week   0.373
4       map   0.315

Document 8582:
      Word  TF-IDF
0   rating   0.378
1  instant   0.370
2     yelp   0.336
3      etc   0.321
4  sharing   0.307

Document 8583:
            Word  TF-IDF
0   netflixstyle   0.429
1    personality   0.410
2  functionality   0.387
3         brings   0.372
4           yelp   0.343

Document 8584:
       Word  TF-IDF
0      outa   0.379
1     maps2   0.379
2  innovate   0.335
3     whole   0.293
4      heat   0.290
```

```
Document 8585:
     Word  TF-IDF
0    used   0.617
1  hotpot   0.593
2  anyone   0.518

Document 8586:
          Word  TF-IDF
0  progression   0.450
1      logical   0.450
2         rate   0.378
3        guess   0.331
4     business   0.296

Document 8587:
         Word  TF-IDF
0       recos   0.343
1          um   0.332
2         eat   0.323
3     covered   0.311
4  restaurant   0.311

Document 8588:
             Word  TF-IDF
0     integration   0.391
1      impressive   0.366
2            yelp   0.362
3    personalized   0.358
4  recommendation   0.339

Document 8589:
       Word  TF-IDF
0  straight   0.429
1       mag   0.412
2   fashion   0.406
3      flip   0.406
4   welcome   0.374

Document 8590:
          Word  TF-IDF
0      transit   0.663
1  citygoround   0.358
2         gtfs   0.358
3       canada   0.316
4        group   0.242

Document 8591:
               Word  TF-IDF
0          response   0.558
1  uidesignguidecom   0.345
2            regard   0.345
3         reference   0.319
4           exactly   0.304

Document 8592:
         Word  TF-IDF
```

```
0  intriguing   0.462
1     chatter   0.408
2         kek   0.401
3       sound   0.335
4         lot   0.281

Document 8593:
          Word  TF-IDF
0  introducing   0.561
1       status   0.508
2      checkin   0.418
3       reward   0.405
4       mobile   0.278

Document 8594:
      Word  TF-IDF
0  control   0.644
1    crowd   0.584
2      pop   0.466
3     link   0.164

Document 8595:
        Word  TF-IDF
0    distrub   0.610
1    dancing   0.489
2  launching   0.352
3    product   0.331
4     circle   0.249

Document 8596:
       Word  TF-IDF
0    affair   0.633
1     grand   0.572
2  industry   0.437
3     party   0.285

Document 8597:
      Word  TF-IDF
0     fond   0.570
1     lost   0.417
2  instead   0.407
3   trying   0.401
4      got   0.311

Document 8598:
       Word  TF-IDF
0  addition   0.519
1       cut   0.470
2     print   0.444
3     place   0.349
4     thing   0.315

Document 8599:
     Word  TF-IDF
0  3block   0.400
1    rent   0.400
```

```
2  radius   0.400
3   avoid   0.336
4  retail   0.302

Document 8600:
        Word  TF-IDF
0  temporary   0.570
1        get   0.505
2       sxsw   0.494
3      store   0.355
4       link   0.223

Document 8601:
        Word  TF-IDF
0        lil   0.640
1    jealous   0.530
2        set   0.371
3  temporary   0.334
4      store   0.208

Document 8602:
    Word  TF-IDF
0  wipad   0.719
1   shot   0.550
2  first   0.400
3   link   0.142

Document 8603:
             Word  TF-IDF
0  nonprofitsquot   0.680
1          quotwe   0.617
2            want   0.397

Document 8604:
        Word  TF-IDF
0    explore   0.632
1  attending   0.466
2       want   0.348
3      check   0.333
4        via   0.283

Document 8605:
        Word  TF-IDF
0        bug   0.363
1  recreated   0.363
2   included   0.346
3     pacman   0.329
4    perfect   0.329

Document 8606:
     Word  TF-IDF
0   sweet   0.554
1   whats   0.509
2  better   0.489
3    time   0.360
4     new   0.255
```

```
Document 8607:
                Word  TF-IDF
0  improvementsquot   0.356
1           economy   0.356
2         continual   0.356
3           benefit   0.341
4             scale   0.321

Document 8608:
         Word  TF-IDF
0     subject   0.462
1     pattern   0.409
2  engagement   0.401
3  interested   0.360
4        hear   0.317

Document 8609:
                Word  TF-IDF
0  nooooooooooooooo   0.692
1               die   0.574
2           battery   0.438

Document 8610:
       Word  TF-IDF
0    borrow   0.385
1   quickly   0.380
2  portable   0.380
3    behind   0.322
4   charger   0.292

Document 8611:
          Word  TF-IDF
0  maintenance   0.620
1         fine   0.538
2          art   0.415
3      battery   0.392

Document 8612:
      Word  TF-IDF
0    amble   0.530
1  vuitton   0.530
2    louis   0.530
3      app   0.307
4   called   0.248

Document 8613:
       Word  TF-IDF
0     death   0.498
1    policy   0.498
2  actually   0.431
3  official   0.420
4       one   0.276

Document 8614:
       Word  TF-IDF
```

```
0    inthat   0.442
1  driskill   0.423
2     stuck   0.390
3    across   0.334
4     block   0.280

Document 8615:
        Word  TF-IDF
0  stumbling   0.469
1    contact   0.432
2       stay   0.402
3       lost   0.384
4      group   0.372

Document 8616:
     Word  TF-IDF
0   1amp2   0.608
1  stupid   0.483
2    find   0.379
3    game   0.349
4     via   0.273

Document 8617:
      Word  TF-IDF
0    price   0.500
1   stupid   0.465
2      mac   0.440
3  special   0.440
4     find   0.365

Document 8618:
        Word  TF-IDF
0  developer   0.789
1       need   0.614

Document 8619:
     Word  TF-IDF
0  choose   0.523
1   buyer   0.433
2  tablet   0.358
3  future   0.307
4   apple   0.305

Document 8620:
        Word  TF-IDF
0   playlist   0.526
1   featured   0.442
2     artist   0.435
3  available   0.403
4       free   0.292

Document 8621:
       Word  TF-IDF
0        tv   0.470
1      dvrs   0.380
2  stopping   0.352
```

```
3      xbox   0.343
4     train   0.343

Document 8622:
     Word  TF-IDF
0     one   0.359
1  tweeps   0.341
2   freak   0.341
3  kindle   0.324
4   paper   0.316

Document 8623:
      Word  TF-IDF
0  tweeted   0.436
1    quoti   0.329
2  weekend   0.326
3    money   0.321
4     gave   0.319

Document 8624:
           Word  TF-IDF
0           12b   0.419
1  livetweeting   0.419
2            jr   0.380
3         drive   0.342
4          mile   0.342

Document 8625:
          Word  TF-IDF
0   retweeting   0.560
1  spreadsheet   0.508
2        added   0.480
3       chance   0.428
4         link   0.117

Document 8626:
     Word  TF-IDF
0    want   0.363
1  awhile   0.344
2    twit   0.304
3  anyway   0.304
4  always   0.254

Document 8627:
             Word  TF-IDF
0  recommendation   0.547
1         thought   0.497
2         twitter   0.437
3            need   0.362
4     application   0.267

Document 8628:
   Word  TF-IDF
0  covr   0.457
1  undr   0.457
2   lwr   0.457
```

```
3  boom   0.396
4  devs   0.366

Document 8629:
       Word  TF-IDF
0   twitter   0.652
1     honor   0.491
2       wow   0.349
3  everyone   0.328
4      look   0.296

Document 8630:
         Word  TF-IDF
0   authority   0.455
1        main   0.455
2  topicality   0.410
3      factor   0.382
4        site   0.307

Document 8631:
    Word  TF-IDF
0    und   0.490
1    mit   0.490
2   noch   0.469
3  heute   0.469
4    via   0.210

Document 8632:
       Word  TF-IDF
0      gcal   0.730
1  calendar   0.573
2     party   0.344
3      link   0.144

Document 8633:
         Word  TF-IDF
0  capitalism   0.454
1        went   0.336
2      choice   0.322
3    ballroom   0.317
4        full   0.312

Document 8634:
       Word  TF-IDF
0  podcasts   0.879
1     party   0.439
2      link   0.184

Document 8635:
      Word  TF-IDF
0     piss   0.364
1  promise   0.355
2     slip   0.355
3   urinal   0.347
4   either   0.335
```

```
Document 8636:
     Word  TF-IDF
0  thanks   0.648
1    time   0.550
2   party   0.527

Document 8637:
     Word  TF-IDF
0   tyson   0.554
1    mike   0.538
2  coming   0.416
3    game   0.394
4     new   0.239

Document 8638:
           Word  TF-IDF
0  accidentally   0.470
1        return   0.432
2            hi   0.406
3          took   0.383
4        please   0.329

Document 8639:
     Word  TF-IDF
0    line   0.599
1      pm   0.549
2    open   0.339
3   popup   0.275
4  austin   0.249

Document 8640:
      Word  TF-IDF
0      tab   0.517
1   galaxy   0.479
2  samsung   0.456
3     find   0.349
4     know   0.303

Document 8641:
     Word  TF-IDF
0    nice   0.498
1     fun   0.497
2   check   0.414
3   going   0.414
4  people   0.402

Document 8642:
      Word  TF-IDF
0      map   0.500
1   mobile   0.481
2    usage   0.351
3  million   0.341
4      use   0.284

Document 8643:
       Word  TF-IDF
```

```
0  hitchery   0.492
1      beta   0.379
2     early   0.379
3       try   0.355
4       let   0.309

Document 8644:
            Word  TF-IDF
0  visualisation   0.417
1          ghost   0.399
2          logic   0.386
3       movement   0.368
4         pacman   0.350

Document 8645:
        Word  TF-IDF
0    greatly   0.358
1  todaydoes   0.358
2     highly   0.324
3     effort   0.301
4       seem   0.301

Document 8646:
       Word  TF-IDF
0  textbook   0.553
1  proposal   0.553
2    update   0.330
3  download   0.315
4      news   0.309

Document 8647:
        Word  TF-IDF
0        gut   0.329
1    inertia   0.329
2    propped   0.329
3  breakfast   0.315
4      couch   0.291

Document 8648:
           Word  TF-IDF
0      magazine   0.527
1  presentation   0.464
2       digital   0.427
3        future   0.421
4         great   0.360

Document 8649:
      Word  TF-IDF
0  rebecca   0.575
1     suck   0.496
2    black   0.492
3      wow   0.409
4     link   0.120

Document 8650:
        Word  TF-IDF
```

```
0  4chan4eva   0.391
1     ducked   0.391
2   exposing   0.378
3       flew   0.378
4     tattoo   0.316

Document 8651:
        Word  TF-IDF
0      omega   0.398
1  nongoogle   0.368
2      swiss   0.359
3       asks   0.339
4   favorite   0.300

Document 8652:
      Word  TF-IDF
0     plug   0.395
1    floor   0.330
2  sitting   0.318
3    never   0.296
4  thought   0.291

Document 8653:
       Word  TF-IDF
0     staff   0.515
1   brought   0.458
2  employee   0.437
3     texas   0.396
4     store   0.350

Document 8654:
        Word  TF-IDF
0       mass   0.406
1         ui   0.392
2      touch   0.392
3    brought   0.385
4  microsoft   0.347

Document 8655:
          Word  TF-IDF
0          npr   0.633
1   affiliated   0.365
2  firefighter   0.365
3        coder   0.349
4          fan   0.257

Document 8656:
            Word  TF-IDF
0        empathy   0.439
1  twitterstream   0.439
2        concern   0.420
3         mostly   0.374
4       grateful   0.374

Document 8657:
          Word  TF-IDF
```

```
0       useful   0.450
1       engine   0.445
2  information   0.432
3      instead   0.409
4        could   0.365

Document 8658:
           Word  TF-IDF
0  distribution   0.514
1       partner   0.476
2        better   0.329
3       looking   0.312
4         would   0.289

Document 8659:
       Word  TF-IDF
0  acoustic   0.334
1      solo   0.326
2       cst   0.322
3       vip   0.310
4       1pm   0.308

Document 8660:
       Word  TF-IDF
0  headache   0.498
1      full   0.483
2      room   0.464
3    design   0.394
4      talk   0.386

Document 8661:
       Word  TF-IDF
0  dinosaur   0.660
1    pirate   0.345
2      ummm   0.345
3     ninja   0.330
4     shark   0.330

Document 8662:
      Word  TF-IDF
0   unveil   0.596
1  rumored   0.516
2  network   0.290
3   circle   0.280
4    today   0.273

Document 8663:
             Word  TF-IDF
0  everywherequot   0.447
1          quottv   0.447
2        computer   0.387
3             cnn   0.368
4         service   0.324

Document 8664:
             Word  TF-IDF
```

```
0          quottv   0.437
1  everywherequot   0.437
2        computer   0.378
3             cnn   0.359
4         service   0.317

Document 8665:
        Word  TF-IDF
0    traffic   0.497
1  timeevery   0.361
2       2yrs   0.346
3     option   0.299
4      worth   0.275

Document 8666:
        Word  TF-IDF
0  launching   0.585
1    network   0.428
2      today   0.404
3     social   0.399
4        new   0.339

Document 8667:
        Word  TF-IDF
0     cicles   0.679
1  launching   0.391
2     called   0.304
3    network   0.286
4      today   0.270

Document 8668:
              Word  TF-IDF
0           secret   0.533
1  quotcirclesquot   0.483
2        launching   0.398
3           called   0.309
4          network   0.291

Document 8669:
              Word  TF-IDF
0           secret   0.538
1  quotcirclesquot   0.487
2        launching   0.401
3           called   0.312
4          network   0.294

Document 8670:
          Word  TF-IDF
0  circlesquot   0.629
1   quotgoogle   0.470
2    launching   0.438
3      network   0.320
4       social   0.298

Document 8671:
       Word  TF-IDF
```

```
0     sauce   0.445
1       bbq   0.408
2  together   0.386
3     found   0.343
4     texas   0.342

Document 8672:
        Word  TF-IDF
0  recipient   0.574
1      lucky   0.460
2      block   0.364
3     around   0.321
4        one   0.267

Document 8673:
           Word  TF-IDF
0  timeeveryday   0.488
1         worth   0.371
2          save   0.349
3         route   0.343
4       traffic   0.335

Document 8674:
      Word  TF-IDF
0    route   0.357
1  feature   0.349
2  traffic   0.349
3   saving   0.338
4    every   0.319

Document 8675:
         Word  TF-IDF
0        shit   0.683
1  ungrateful   0.422
2          bc   0.335
3        turn   0.301
4        play   0.290

Document 8676:
       Word  TF-IDF
0  funnysad   0.460
1      line   0.398
2    longer   0.361
3      kind   0.358
4   release   0.300

Document 8677:
              Word  TF-IDF
0             edit   0.499
1  preferencesquot   0.479
2           quotad   0.479
3             view   0.367
4            think   0.298

Document 8678:
              Word  TF-IDF
```

```
0    quotchallenge   0.363
1       originally   0.347
2  experimentation   0.347
3       successful   0.314
4           moving   0.301

Document 8679:
        Word  TF-IDF
0   eastquot   0.603
1  quotchina   0.603
2     middle   0.523

Document 8680:
            Word  TF-IDF
0     quotfuture   0.381
1     schemaquot   0.381
2      touchquot   0.365
3  quotdesigning   0.320
4           mind   0.290

Document 8681:
             Word  TF-IDF
0      futurequot   0.389
1  gatekeeperquot   0.389
2        quotfast   0.389
3     quotproduct   0.389
4              ht   0.362

Document 8682:
           Word  TF-IDF
0           bij   0.436
1            nu   0.417
2  quottouching   0.417
3      ipadquot   0.357
4      magazine   0.321

Document 8683:
             Word  TF-IDF
0  quotmophiequot   0.439
1       tradeshow   0.369
2           doubt   0.355
3          picked   0.327
4          charge   0.325

Document 8684:
     Word  TF-IDF
0  picked   0.676
1   white   0.581
2   today   0.361
3   store   0.273

Document 8685:
      Word  TF-IDF
0   messed   0.452
1   borrow   0.416
2      min   0.362
```

```
3    bring   0.333
4  charger   0.315

Document 8686:
        Word  TF-IDF
0  quotnerds   0.416
1   wildquot   0.398
2   prepping   0.386
3        aka   0.368
4       gone   0.334

Document 8687:
    Word  TF-IDF
0  popup   0.746
1  store   0.564
2   link   0.355

Document 8688:
     Word  TF-IDF
0   popup   0.618
1  austin   0.560
2   store   0.467
3    link   0.294

Document 8689:
          Word  TF-IDF
0  constuction   0.487
1         line   0.421
2      keeping   0.409
3     standing   0.354
4        thats   0.314

Document 8690:
     Word  TF-IDF
0   block   0.548
1   still   0.499
2    line   0.375
3  circle   0.352
4   popup   0.344

Document 8691:
    Word  TF-IDF
0  store   0.704
1   sxsw   0.489
2  popup   0.466
3   link   0.221

Document 8692:
      Word  TF-IDF
0   kanyes   0.515
1     vevo   0.515
2  longest   0.432
3     show   0.285
4      pop   0.276

Document 8693:
```

```
       Word  TF-IDF
0  catching   0.521
1     hotel   0.442
2  tweeting   0.409
3      room   0.371
4      also   0.344

Document 8694:
         Word  TF-IDF
0  pickupline   0.749
1     charger   0.506
2        best   0.429

Document 8695:
       Word  TF-IDF
0       bad   0.435
1      sure   0.398
2  tomorrow   0.390
3    update   0.377
4     ipads   0.369

Document 8696:
        Word  TF-IDF
0      super   0.488
1       move   0.459
2    another   0.415
3      smart   0.403
4  temporary   0.307

Document 8697:
      Word  TF-IDF
0  groupme   0.444
1   global   0.411
2      add   0.411
3  feature   0.352
4      say   0.273

Document 8698:
         Word  TF-IDF
0    stumbled   0.463
1    terrific   0.463
2        upon   0.429
3  googlebing   0.355
4        full   0.318

Document 8699:
          Word  TF-IDF
0    regularly   0.411
1    interrupt   0.409
2  programming   0.409
3    scheduled   0.405
4         geek   0.349

Document 8700:
                 Word  TF-IDF
0  regularlyscheduled   0.569
```

```
1         programming   0.412
2           interrupt   0.412
3                geek   0.352
4                news   0.318

Document 8701:
         Word  TF-IDF
0      record   0.763
1  experience   0.623
2        link   0.174

Document 8702:
      Word  TF-IDF
0   rabbit   0.551
1   forgot   0.466
2     home   0.397
3     case   0.326
4  opening   0.279

Document 8703:
     Word  TF-IDF
0   whats   0.565
1    take   0.490
2  really   0.489
3    want   0.448

Document 8704:
      Word  TF-IDF
0  gesture   0.628
1    guide   0.445
2     nice   0.411
3     make   0.357
4     free   0.308

Document 8705:
       Word  TF-IDF
0     mount   0.420
1      ahem   0.389
2  pressure   0.380
3    excuse   0.365
4  musthave   0.349

Document 8706:
         Word  TF-IDF
0          tv   0.635
1     program   0.421
2   connected   0.404
3  experience   0.348
4      future   0.302

Document 8707:
     Word  TF-IDF
0  unless   0.492
1    care   0.469
2    post   0.375
3    sure   0.369
```

```
4     guy   0.316

Document 8708:
      Word  TF-IDF
0      59p   0.502
1  journal   0.502
2  momento   0.481
3     grab   0.358
4     sale   0.335

Document 8709:
          Word  TF-IDF
0  communicate   0.323
1     cerebral   0.323
2        palsy   0.323
3       glenda   0.323
4       watson   0.323

Document 8710:
          Word  TF-IDF
0    quotqampa   0.429
1  rankingquot   0.410
2        salon   0.337
3        house   0.318
4       packed   0.309

Document 8711:
           Word  TF-IDF
0  catfightquot   0.429
1        quotim   0.372
2    googlebing   0.329
3         house   0.318
4        packed   0.309

Document 8712:
        Word  TF-IDF
0  algorithm   0.399
1   patented   0.399
2        age   0.368
3         ie   0.356
4     domain   0.350

Document 8713:
        Word  TF-IDF
0     goquot   0.449
1  quotyoure   0.449
2       mine   0.373
3      never   0.326
4   probably   0.324

Document 8714:
      Word  TF-IDF
0      use   0.491
1  marissa   0.452
2    mayer   0.441
3      map   0.432
```

```
4   mobile   0.416

Document 8715:
          Word  TF-IDF
0     housego   0.475
1        lynn   0.423
2      schema   0.363
3      packed   0.357
4  navigation   0.324

Document 8716:
         Word  TF-IDF
0         tsk   0.666
1   synthetic   0.333
2      people   0.327
3        care   0.267
4       house   0.247

Document 8717:
          Word  TF-IDF
0     example   0.507
1        used   0.472
2  navigation   0.458
3        good   0.371
4         via   0.300

Document 8718:
         Word  TF-IDF
0      snazzy   0.521
1    favorite   0.394
2   something   0.353
3         let   0.313
4         app   0.289

Document 8719:
          Word  TF-IDF
0     console   0.402
1   extensive   0.402
2  javascript   0.363
3         api   0.343
4      chrome   0.314

Document 8720:
          Word  TF-IDF
0   cellphone   0.471
1  discussion   0.386
2  unofficial   0.374
3       brain   0.349
4     control   0.349

Document 8721:
      Word  TF-IDF
0     held   0.584
1    giant   0.511
2  someone   0.414
3     love   0.360
```

```
4     like   0.312

Document 8722:
         Word  TF-IDF
0         att   0.777
1  nationwide   0.287
2        wife   0.245
3      number   0.238
4       tried   0.224

Document 8723:
         Word  TF-IDF
0  adelefiona   0.371
1      solace   0.371
2         cry   0.350
3        sort   0.343
4         mix   0.314

Document 8724:
       Word  TF-IDF
0  imparted   0.376
1     recon   0.376
2     relay   0.376
3      phil   0.376
4   helpful   0.348

Document 8725:
      Word  TF-IDF
0  cruisin   0.451
1   looong   0.451
2  checked   0.385
3    venue   0.370
4      saw   0.292

Document 8726:
        Word  TF-IDF
0      notch   0.398
1  inventory   0.339
2    hanging   0.316
3   customer   0.290
4        top   0.281

Document 8727:
       Word  TF-IDF
0   stellar   0.472
1    smooth   0.472
2  purchase   0.368
3  customer   0.344
4      wish   0.323

Document 8728:
      Word  TF-IDF
0     mktg   0.415
1      yup   0.376
2  history   0.355
3     move   0.312
```

```
4   genius   0.309

Document 8729:
      Word  TF-IDF
0  gowalla   0.427
1  already   0.384
2      pic   0.384
3    heard   0.366
4    apple   0.363

Document 8730:
         Word  TF-IDF
0     hanging   0.521
1      center   0.440
2  convention   0.434
3        find   0.392
4         win   0.323

Document 8731:
         Word  TF-IDF
0  controller   0.426
1       lobby   0.415
2     hanging   0.365
3          hi   0.358
4      hilton   0.321

Document 8732:
      Word  TF-IDF
0    belly   0.432
1     sink   0.413
2   bottom   0.390
3  control   0.320
4      top   0.305

Document 8733:
       Word  TF-IDF
0    rental   0.398
1   plucked   0.398
2      lift   0.368
3      taxi   0.368
4  stranger   0.339

Document 8734:
      Word  TF-IDF
0      cam   0.441
1    piece   0.419
2  missing   0.396
3    catch   0.365
4   stream   0.361

Document 8735:
       Word  TF-IDF
0   geeking   0.457
1   theatre   0.457
2      apis   0.447
3  teaching   0.447
```

```
4   youtube   0.415

Document 8736:
            Word  TF-IDF
0       reliving   0.426
1  googleaclueff   0.426
2       throwing   0.385
3            yep   0.335
4            old   0.297

Document 8737:
        Word  TF-IDF
0  partygood   0.586
1    hanging   0.515
2   industry   0.448
3      crowd   0.436

Document 8738:
         Word  TF-IDF
0  connecting   0.613
1      in1102   0.613
2     hanging   0.486
3        link   0.116

Document 8739:
      Word  TF-IDF
0  passage   0.419
1     rite   0.419
2  geeking   0.408
3   waited   0.366
4  missing   0.359

Document 8740:
      Word  TF-IDF
0    route   0.571
1   driver   0.329
2  finding   0.319
3      per   0.297
4  traffic   0.279

Document 8741:
          Word  TF-IDF
0  routearound   0.682
1  environment   0.316
2    efficient   0.301
3        sweet   0.247
4       pretty   0.237

Document 8742:
         Word  TF-IDF
0          c5   0.540
1  precaution   0.540
2   40075959p   0.417
3       valid   0.356
4        code   0.326
```

```
Document 8743:
        Word  TF-IDF
0  completes   0.433
1  kawasakis   0.433
2     letter   0.415
3       auto   0.402
4       four   0.359

Document 8744:
      Word  TF-IDF
0  outside   0.685
1  waiting   0.662
2    store   0.305

Document 8745:
          Word  TF-IDF
0     benieuwd   0.430
1          ben   0.416
2    regularly   0.327
3  programming   0.325
4    interrupt   0.325

Document 8746:
         Word  TF-IDF
0  mediumquot   0.612
1     message   0.496
2          ux   0.466
3     quotthe   0.402

Document 8747:
     Word  TF-IDF
0    need   0.614
1  austin   0.446
2    sure   0.390
3     buy   0.358
4   ipad2   0.331

Document 8748:
        Word  TF-IDF
0     asking   0.564
1       haha   0.446
2  interview   0.438
3   question   0.428
4        guy   0.329

Document 8749:
       Word  TF-IDF
0    medium   0.412
1  platform   0.396
2       let   0.379
3       hey   0.369
4     think   0.340

Document 8750:
       Word  TF-IDF
0      buzz   0.404
```

```
1  tomorrow   0.358
2     store   0.348
3   twitter   0.347
4       6th   0.340

Document 8751:
        Word  TF-IDF
0  indicates   0.381
1     widget   0.381
2         ui   0.305
3     native   0.305
4     reader   0.294

Document 8752:
       Word  TF-IDF
0       v12   0.442
1      sync   0.377
2  schedule   0.312
3    better   0.282
4      live   0.278

Document 8753:
       Word  TF-IDF
0  valuable   0.577
1    device   0.473
2       far   0.473
3   charger   0.470

Document 8754:
        Word  TF-IDF
0   continue   0.391
1   elevator   0.391
2  submitted   0.379
3     upload   0.369
4       drop   0.324

Document 8755:
         Word  TF-IDF
0     billion   0.466
1       drive   0.413
2        mile   0.413
3  navigation   0.358
4        user   0.297

Document 8756:
         Word  TF-IDF
0         12b   0.465
1          jr   0.422
2       drive   0.380
3        mile   0.380
4  navigation   0.329

Document 8757:
        Word  TF-IDF
0   unstable   0.476
1  tweetdeck   0.449
```

```
2     newest   0.440
3    upgrade   0.395
4         io   0.347

Document 8758:
           Word  TF-IDF
0    phonetxt   0.438
1     prepaid   0.419
2     tmobile   0.373
3  suggestion   0.351
4     working   0.288

Document 8759:
       Word  TF-IDF
0   youtube   0.532
1      tube   0.325
2      sock   0.311
3      fuck   0.282
4  laughing   0.245

Document 8760:
       Word  TF-IDF
0      move   0.472
1    genius   0.467
2      temp   0.401
3  downtown   0.327
4      open   0.321

Document 8761:
          Word  TF-IDF
0  meetingwave   0.450
1   submitting   0.435
2    downloads   0.395
3      version   0.318
4         apps   0.311

Document 8762:
       Word  TF-IDF
0  shipment   0.566
1     daily   0.460
2      wait   0.408
3      week   0.388
4       via   0.285

Document 8763:
       Word  TF-IDF
0  unboxing   0.586
1     front   0.474
2      live   0.450
3     video   0.406
4     store   0.215

Document 8764:
        Word  TF-IDF
0  fullforce   0.366
1     hisher   0.366
```

```
2       hint   0.308
3     strong   0.308
4      email   0.281

Document 8765:
      Word  TF-IDF
0      att   0.441
1   single   0.396
2  arrived   0.384
3  without   0.329
4   person   0.321

Document 8766:
            Word  TF-IDF
0      gapminder   0.416
1  visualization   0.392
2       unveiled   0.384
3         source   0.331
4          based   0.313

Document 8767:
                Word  TF-IDF
0               font   0.647
1  googlecomwebfonts   0.351
2            request   0.331
3          hackathon   0.312
4         discussion   0.300

Document 8768:
      Word  TF-IDF
0  without   0.685
1     even   0.585
2      get   0.435

Document 8769:
       Word  TF-IDF
0    itunes   0.663
1   sampler   0.337
2   already   0.321
3  download   0.296
4     music   0.295

Document 8770:
        Word  TF-IDF
0    passage   0.462
1       rite   0.462
2     waited   0.404
3    missing   0.396
4  important   0.363

Document 8771:
        Word  TF-IDF
0   traveler   0.418
1  passenger   0.418
2     heaven   0.400
3    delayed   0.387
```

```
4      thank   0.285

Document 8772:
            Word  TF-IDF
0  dayearthquake   0.551
1       eventful   0.551
2        tsunami   0.463
3           plus   0.411
4           link   0.104

Document 8773:
       Word  TF-IDF
0     high5   0.455
1    normal   0.388
2    buying   0.335
3     queue   0.323
4  probably   0.314

Document 8774:
     Word  TF-IDF
0  garage   0.550
1  notice   0.550
2      ok   0.429
3    band   0.391
4    sxsw   0.240

Document 8775:
            Word  TF-IDF
0  ipaddesigning   0.583
1         boomer   0.424
2            mom   0.413
3       ballroom   0.407
4        heading   0.380

Document 8776:
          Word  TF-IDF
0      admired   0.366
1      bounced   0.366
2       classy   0.366
3  intelligent   0.354
4   successful   0.332

Document 8777:
        Word  TF-IDF
0    southby   0.418
1  expecting   0.395
2    shortly   0.386
3     record   0.362
4       sale   0.292

Document 8778:
       Word  TF-IDF
0  overload   0.508
1   network   0.503
2     major   0.279
3    called   0.267
```

```
4       via   0.256

Document 8779:
            Word  TF-IDF
0        arrives   0.506
1  overshadowing   0.506
2             pm   0.401
3        totally   0.391
4       tomorrow   0.313

Document 8780:
      Word  TF-IDF
0       xd   0.623
1    funny   0.492
2   dionne   0.322
3  warwick   0.322
4  rebecca   0.303

Document 8781:
       Word  TF-IDF
0  guidance   0.466
1   regular   0.412
2  calendar   0.350
3      plan   0.323
4     track   0.308

Document 8782:
       Word  TF-IDF
0   shortly   0.432
1     close   0.378
2     enter   0.352
3      sign   0.335
4  giveaway   0.334

Document 8783:
      Word  TF-IDF
0       3k   0.415
1  porting   0.415
2  syncing   0.415
3   sunday   0.332
4     song   0.298

Document 8784:
       Word  TF-IDF
0  midnight   0.533
1   learned   0.491
2  downtown   0.336
3      open   0.329
4       via   0.288

Document 8785:
        Word  TF-IDF
0       fare   0.516
1     denies   0.463
2       hope   0.369
3     better   0.365
```

```
4  launching   0.330

Document 8786:
     Word  TF-IDF
0  inside   0.559
1    here   0.495
2    view   0.469
3   popup   0.280
4    link   0.266

Document 8787:
        Word  TF-IDF
0    working   0.576
1      larry   0.437
2       eric   0.395
3  different   0.341
4  interview   0.315

Document 8788:
       Word  TF-IDF
0    cabbie   0.561
1  navigate   0.536
2    native   0.449
3      geek   0.347
4       map   0.276

Document 8789:
        Word  TF-IDF
0     mapper   0.455
1  amazingly   0.443
2      entry   0.413
3       easy   0.374
4     detail   0.307

Document 8790:
     Word  TF-IDF
0    kick   0.453
1   visit   0.416
2    blog   0.384
3    info   0.377
4  giving   0.354

Document 8791:
     Word  TF-IDF
0    kick   0.424
1   visit   0.389
2    page   0.383
3    info   0.353
4  giving   0.331

Document 8792:
           Word  TF-IDF
0        loving   0.538
1  presentation   0.447
2        really   0.397
3       marissa   0.355
```

```
4         mayer   0.347

Document 8793:
      Word  TF-IDF
0  another   0.405
1   giving   0.385
2      let   0.373
3      guy   0.335
4     make   0.328

Document 8794:
        Word  TF-IDF
0      taken   0.498
1      style   0.485
2  instagram   0.479
3     saving   0.389
4      photo   0.353

Document 8795:
          Word  TF-IDF
0      craving   0.500
1  foodspotting   0.489
2      created   0.444
3         mind   0.422
4          got   0.289

Document 8796:
          Word  TF-IDF
0  envisioning   0.377
1     rescuing   0.377
2    broadcast   0.361
3      virtual   0.322
4        robot   0.306

Document 8797:
        Word  TF-IDF
0  wikimedia   0.491
1    visited   0.491
2      yahoo   0.443
3  microsoft   0.348
4    website   0.330

Document 8798:
          Word  TF-IDF
0  professional   0.383
1        talent   0.375
2      talented   0.375
3     community   0.352
4        rockin   0.352

Document 8799:
     Word  TF-IDF
0     jam   0.497
1   avoid   0.482
2  packed   0.413
3    long   0.354
```

```
4   party   0.258

Document 8800:
         Word  TF-IDF
0      protip   0.538
1  austinarea   0.524
2       avoid   0.488
3      friday   0.410
4       store   0.174

Document 8801:
      Word  TF-IDF
0    aunty   0.447
1   voxpop   0.447
2  popular   0.358
3    worth   0.340
4    watch   0.305

Document 8802:
   Word  TF-IDF
0   ep3   0.831
1  many   0.532
2  link   0.164

Document 8803:
        Word  TF-IDF
0    bizness   0.419
1  subscribe   0.419
2    channel   0.379
3         au   0.352
4    podcast   0.336

Document 8804:
     Word  TF-IDF
0  harbor   0.524
1    unlv   0.524
2  cheese   0.455
3   pearl   0.409
4     win   0.258

Document 8805:
      Word  TF-IDF
0  channel   0.729
1   loving   0.629
2      new   0.269

Document 8806:
        Word  TF-IDF
0    1045am3   0.612
1  volunteer   0.553
2     loving   0.477
3       need   0.304

Document 8807:
       Word  TF-IDF
0       alt   0.399
```

```
1   fastest   0.399
2  universe   0.399
3     reach   0.399
4        us   0.311

Document 8808:
           Word  TF-IDF
0          full   0.713
1  presentation   0.673
2          link   0.196

Document 8809:
     Word  TF-IDF
0    room   0.531
1  nobody   0.372
2   raise   0.338
3      us   0.313
4  packed   0.289

Document 8810:
    Word  TF-IDF
0  blind   0.731
1    saw   0.524
2    guy   0.437

Document 8811:
          Word  TF-IDF
0         made   0.635
1      awesome   0.562
2         link   0.380
3  application   0.368

Document 8812:
      Word  TF-IDF
0   summon   0.399
1  hopping   0.376
2    magic   0.356
3    short   0.341
4      car   0.315

Document 8813:
          Word  TF-IDF
0  awesomeness   0.641
1     downtown   0.363
2      opening   0.352
3    temporary   0.350
4       launch   0.277

Document 8814:
       Word  TF-IDF
0    artist   0.608
1        cd   0.449
2   various   0.430
3  featured   0.309
4  download   0.256
```

```
Document 8815:
        Word  TF-IDF
0   bursting   0.388
1  unplugged   0.388
2       seam   0.388
3     family   0.331
4  highlight   0.326

Document 8816:
      Word  TF-IDF
0     firm   0.426
1   scared   0.413
2  netbook   0.394
3    droid   0.334
4      god   0.326

Document 8817:
      Word  TF-IDF
0     ajax   0.494
1  parsing   0.494
2     spec   0.466
3   wanted   0.428
4     know   0.268

Document 8818:
        Word  TF-IDF
0      7quot   0.388
1    quotsxsw   0.388
2         hp   0.371
3  presented   0.318
4     window   0.281

Document 8819:
        Word  TF-IDF
0       goer   0.378
1    fucking   0.371
2  hilarious   0.371
3       dear   0.333
4       hall   0.315

Document 8820:
       Word  TF-IDF
0   cursing   0.375
1      zzzs   0.375
2    losing   0.359
3   bicycle   0.347
4  grateful   0.320

Document 8821:
       Word  TF-IDF
0   husband   0.410
1  chatting   0.400
2     skype   0.391
3       sat   0.352
4  spending   0.352
```

```
Document 8822:
      Word  TF-IDF
0  sitbyus   0.464
1  include   0.385
2     suck   0.362
3     dear   0.362
4     film   0.349

Document 8823:
        Word  TF-IDF
0      along   0.491
1   bringing   0.454
2  attending   0.418
3        wow   0.380
4        big   0.329

Document 8824:
   Word  TF-IDF
0  wake   0.542
1  woah   0.491
2  word   0.434
3  buzz   0.396
4  work   0.347

Document 8825:
       Word  TF-IDF
0  lemonade   0.531
1       wp7   0.474
2     stand   0.433
3       two   0.355
4       day   0.269

Document 8826:
       Word  TF-IDF
0    wonder   0.410
1      sell   0.386
2  tomorrow   0.380
3      many   0.377
4     ipads   0.360

Document 8827:
     Word  TF-IDF
0    sold   0.632
1   buyer   0.393
2    went   0.351
3  report   0.324
4  wonder   0.316

Document 8828:
       Word  TF-IDF
0    hollow   0.460
1    barton   0.447
2   sampler   0.351
3    itunes   0.345
4  includes   0.336
```

```
Document 8829:
   Word  TF-IDF
0  hmma   0.417
1  slew   0.417
2   inc   0.399
3  past   0.317
4  mean   0.297

Document 8830:
          Word  TF-IDF
0         58pm   0.377
1       burnbq   0.377
2  panamerican   0.377
3       burner   0.349
4         park   0.327

Document 8831:
      Word  TF-IDF
0     geez   0.547
1   eating   0.483
2  outside   0.369
3  another   0.357
4   people   0.268

Document 8832:
       Word  TF-IDF
0  playlist   0.677
1     enjoy   0.613
2      free   0.376
3      link   0.156

Document 8833:
      Word  TF-IDF
0    cntxt   0.539
1   locatn   0.539
2   contxt   0.270
3    twtng   0.270
4  replace   0.258

Document 8834:
         Word  TF-IDF
0  allthingsd   0.458
1        dang   0.397
2       false   0.375
3       alarm   0.351
4    probably   0.316

Document 8835:
      Word  TF-IDF
0  sampler   0.628
1   itunes   0.617
2     free   0.439
3     link   0.183

Document 8836:
        Word  TF-IDF
```

```
0  barbarian   0.447
1     agency   0.366
2   software   0.362
3      group   0.302
4    company   0.286

Document 8837:
      Word  TF-IDF
0    teeny   0.377
1   steady   0.377
2  filming   0.341
3     held   0.333
4      cam   0.333

Document 8838:
               Word  TF-IDF
0            market   0.547
1  streamsdownloads   0.382
2            winamp   0.382
3              free   0.347
4            direct   0.313

Document 8839:
            Word  TF-IDF
0          ratio   0.578
1     ipadperson   0.333
2  threequarters   0.333
3          macpc   0.333
4           like   0.315

Document 8840:
       Word  TF-IDF
0    target   0.529
1  audience   0.418
2      sell   0.358
3      shop   0.313
4      know   0.296

Document 8841:
          Word  TF-IDF
0           tg   0.446
1         baby   0.374
2  application   0.341
3       boomer   0.339
4         wish   0.319

Document 8842:
       Word  TF-IDF
0    decide   0.434
1   edition   0.413
2  explorer   0.383
3  festival   0.345
4      band   0.334

Document 8843:
       Word  TF-IDF
```

```
0  cellular   0.376
1  detailed   0.368
2    signal   0.337
3  coverage   0.334
4       wan   0.298

Document 8844:
    Word  TF-IDF
0  30min   0.410
1  lunch   0.325
2  saved   0.312
3    ask   0.299
4    wan   0.294

Document 8845:
      Word  TF-IDF
0  blogger   0.704
1    whats   0.615
2      new   0.309
3     link   0.175

Document 8846:
      Word  TF-IDF
0     next   0.531
1    nexus   0.390
2      gen   0.387
3      nfc   0.374
4  playing   0.331

Document 8847:
       Word  TF-IDF
0  playlist   0.603
1  featured   0.506
2    artist   0.498
3      free   0.335
4      link   0.139

Document 8848:
        Word  TF-IDF
0     pluged   0.369
1  oversized   0.369
2   earphone   0.357
3      loose   0.348
4        law   0.329

Document 8849:
      Word  TF-IDF
0  present   0.381
1    enter   0.357
2     must   0.337
3     head   0.327
4     give   0.319

Document 8850:
        Word  TF-IDF
0      tired   0.474
```

```
1  announces   0.398
2       sale   0.341
3    waiting   0.333
4        buy   0.296

Document 8851:
   Word  TF-IDF
0  beta   0.523
1   wan   0.509
2  sign   0.487
3    na   0.461
4  link   0.134

Document 8852:
         Word  TF-IDF
0  accomplish   0.395
1   rocksauce   0.395
2       shang   0.395
3       stalk   0.378
4    austinjs   0.378

Document 8853:
          Word  TF-IDF
0      android   0.392
1  bitlyea1zgd   0.392
2       valley   0.361
3     lightbox   0.349
4        raise   0.344

Document 8854:
                 Word  TF-IDF
0  quotmayorshipsquot   0.447
1             heating   0.428
2            latitude   0.358
3                 war   0.343
4             checkin   0.319

Document 8855:
        Word  TF-IDF
0  extenders   0.474
1    nothing   0.355
2       sold   0.330
3      money   0.329
4    battery   0.313

Document 8856:
         Word  TF-IDF
0    iphone4s   0.433
1   influence   0.376
2   community   0.359
3  innovation   0.359
4       doubt   0.351

Document 8857:
     Word  TF-IDF
0    wish   0.763
```

```
1    free   0.506
2  austin   0.402

Document 8858:
            Word  TF-IDF
0  proliferation   0.540
1        reunion   0.540
2       chatting   0.487
3        someone   0.338
4           like   0.255

Document 8859:
       Word  TF-IDF
0     taker   0.467
1  mashable   0.389
2      part   0.386
3     spent   0.386
4    better   0.322

Document 8860:
      Word  TF-IDF
0   anyway   0.513
1  curious   0.482
2   others   0.364
3     long   0.358
4      day   0.281

Document 8861:
        Word  TF-IDF
0       talk   0.497
1    360idev   0.453
2  basically   0.393
3      along   0.375
4  different   0.353

Document 8862:
            Word  TF-IDF
0     attendence   0.338
1      filmakers   0.338
2  quotsomething   0.323
3    documentary   0.313
4   venturedquot   0.305

Document 8863:
         Word  TF-IDF
0    material   0.408
1       truly   0.408
2    jonathan   0.382
3  understand   0.365
4          vp   0.338

Document 8864:
    Word  TF-IDF
0  sadly   0.553
1  swarm   0.488
2  crowd   0.421
```

```
3    one   0.292
4    via   0.269

Document 8865:
    Word  TF-IDF
0  sadly   0.501
1    via   0.487
2  swarm   0.442
3  crowd   0.381
4    one   0.264

Document 8866:
     Word  TF-IDF
0   swarm   0.653
1   crowd   0.563
2     via   0.360
3  launch   0.320
4    link   0.158

Document 8867:
     Word  TF-IDF
0   swarm   0.700
1   crowd   0.603
2  launch   0.343
3    link   0.170

Document 8868:
     Word  TF-IDF
0   swarm   0.653
1   crowd   0.563
2     via   0.360
3  launch   0.320
4    link   0.158

Document 8869:
     Word  TF-IDF
0   swarm   0.653
1   crowd   0.563
2     via   0.360
3  launch   0.320
4    link   0.158

Document 8870:
      Word  TF-IDF
0      fwd   0.448
1     dark   0.440
2       dj   0.380
3  looking   0.313
4  tonight   0.308

Document 8871:
     Word  TF-IDF
0     cnn   0.621
1   crowd   0.586
2    sxsw   0.364
3  launch   0.333
```

```
4    link   0.165

Document 8872:
               Word  TF-IDF
0              away   0.568
1           licence   0.445
2  bitlypushsxsw11   0.445
3          actually   0.338
4            giving   0.288

Document 8873:
     Word  TF-IDF
0    line   0.558
1     1pm   0.498
2    less   0.487
3  people   0.316
4   popup   0.256

Document 8874:
       Word  TF-IDF
0  interest   0.589
1   towards   0.511
2   working   0.388
3     ready   0.368
4    thanks   0.327

Document 8875:
      Word  TF-IDF
0  tinfoil   0.817
1    offer   0.577

Document 8876:
     Word  TF-IDF
0  clever   0.476
1    core   0.356
2   track   0.356
3  action   0.339
4   right   0.306

Document 8877:
         Word  TF-IDF
0     growing   0.500
1  competitor   0.472
2     crowley   0.453
3  foursquare   0.384
4    facebook   0.354

Document 8878:
       Word  TF-IDF
0     tight   0.844
1      work   0.345
2  employee   0.211
3      used   0.190
4    friend   0.172

Document 8879:
```

```
            Word  TF-IDF
0  readwriteweb   0.565
1     reporting   0.512
2         might   0.383
3       network   0.249
4         today   0.235

Document 8880:
        Word  TF-IDF
0  endeavor   0.486
1   tweeted   0.439
2   appears   0.382
3   preview   0.329
4      fast   0.303

Document 8881:
            Word  TF-IDF
0  entrepreneur   0.437
1         young   0.437
2     wordpress   0.413
3      tweeting   0.353
4       session   0.273

Document 8882:
        Word  TF-IDF
0   applink   0.430
1     dwnld   0.395
2   welcome   0.383
3  anywhere   0.371
4      ride   0.363

Document 8883:
          Word  TF-IDF
0    minimize   0.481
1        tear   0.460
2        wear   0.446
3  commercial   0.446
4      laptop   0.308

Document 8884:
        Word  TF-IDF
0  waitthat   0.443
1     fruit   0.400
2     setup   0.372
3       gon   0.317
4      hear   0.303

Document 8885:
       Word  TF-IDF
0    agree   0.605
1  totally   0.595
2      put   0.529

Document 8886:
      Word  TF-IDF
0  easier   0.451
```

```
1      hi   0.418
2  little   0.395
3    hope   0.347
4    find   0.321

Document 8887:
      Word  TF-IDF
0   coynes   0.475
1  covered   0.444
2    recap   0.410
3     went   0.379
4     wish   0.351

Document 8888:
   Word  TF-IDF
0  wish   0.891
1  said   0.455

Document 8889:
           Word  TF-IDF
0         djroe   0.483
1  ummmmawesome   0.483
2      midnight   0.383
3       learned   0.353
4      downtown   0.241

Document 8890:
      Word  TF-IDF
0  awesome   0.623
1      got   0.582
2      via   0.478
3     link   0.211

Document 8891:
      Word  TF-IDF
0  selling   0.589
1   chance   0.578
2    booth   0.491
3      new   0.280

Document 8892:
           Word  TF-IDF
0        retail   0.481
1          peep   0.478
2  presentation   0.413
3          nice   0.387
4         think   0.344

Document 8893:
      Word  TF-IDF
0  optimal   0.545
1   lanyrd   0.521
2  account   0.436
3   access   0.376
4      way   0.317
```

```
Document 8894:
     Word  TF-IDF
0    size   0.578
1  opened   0.510
2     pop   0.374
3     got   0.364
4     get   0.298

Document 8895:
         Word  TF-IDF
0        wanw   0.537
1  evangelist   0.485
2      former   0.475
3    kawasaki   0.409
4         guy   0.290

Document 8896:
      Word  TF-IDF
0  ireport   0.445
1      cnn   0.378
2    share   0.360
3    whats   0.352
4    photo   0.320

Document 8897:
     Word  TF-IDF
0     cnn   0.397
1   share   0.378
2   whats   0.370
3  friend   0.340
4   photo   0.336

Document 8898:
        Word  TF-IDF
0     lookin   0.430
1      north   0.411
2  inventory   0.367
3       word   0.330
4       sold   0.287

Document 8899:
       Word  TF-IDF
0   science   0.569
1  deciding   0.545
2       art   0.381
3   release   0.371
4   product   0.309

Document 8900:
    Word  TF-IDF
0   temp   0.898
1  store   0.440

Document 8901:
     Word  TF-IDF
0  wheres   0.619
```

```
1   setup   0.600
2  anyone   0.404
3     get   0.305

Document 8902:
         Word  TF-IDF
0  ubiquitous   0.399
1          mo   0.399
2      faster   0.377
3    expected   0.335
4         ago   0.323

Document 8903:
       Word  TF-IDF
0  debating   0.562
1    rather   0.503
2    taking   0.398
3    laptop   0.389
4      take   0.350

Document 8904:
    Word  TF-IDF
0  black   0.770
1  white   0.638

Document 8905:
        Word  TF-IDF
0  apartment   0.495
1       wire   0.495
2      white   0.342
3       look   0.291
4       know   0.277

Document 8906:
      Word  TF-IDF
0     said   0.498
1     able   0.492
2  opening   0.345
3      get   0.305
4     sxsw   0.298

Document 8907:
        Word  TF-IDF
0      gmail   0.440
1     delete   0.415
2  surprised   0.387
3    curious   0.382
4     button   0.353

Document 8908:
     Word  TF-IDF
0   texas   0.677
1   thats   0.642
2  austin   0.359

Document 8909:
```

```
        Word  TF-IDF
0        bpm   0.420
1  surprised   0.400
2        run   0.360
3      sxswi   0.353
4       folk   0.327

Document 8910:
       Word  TF-IDF
0  location   0.903
1    future   0.298
2       say   0.269
3   retweet   0.121
4      link   0.096

Document 8911:
       Word  TF-IDF
0  raffling   0.472
1  livetapp   0.472
2   entered   0.409
3   checkin   0.352
4     tweet   0.293

Document 8912:
     Word  TF-IDF
0   alpha   0.480
1   byfor   0.480
2  tester   0.480
3   least   0.356
4  update   0.287

Document 8913:
      Word  TF-IDF
0      zip   0.576
1  chatter   0.549
2   switch   0.530
3     time   0.292

Document 8914:
       Word  TF-IDF
0  chilling   0.495
1      hair   0.442
2   station   0.415
3    maggie   0.379
4       may   0.311

Document 8915:
       Word  TF-IDF
0  partying   0.617
1    lustre   0.554
2     pearl   0.544
3      link   0.132

Document 8916:
       Word  TF-IDF
0   thirsty   0.435
```

```
1    gadget   0.369
2  anywhere   0.367
3    charge   0.357
4     juice   0.353

Document 8917:
      Word  TF-IDF
0  ispyart   0.602
1     send   0.469
2    photo   0.380
3      use   0.352
4     free   0.286

Document 8918:
       Word  TF-IDF
0    zaggle   0.436
1       zms   0.436
2  jonathan   0.395
3   texting   0.383
4    showed   0.378

Document 8919:
            Word  TF-IDF
0       artifact   0.357
1  partiespanels   0.357
2    devastation   0.357
3   recollection   0.357
4        current   0.316

Document 8920:
       Word  TF-IDF
0      bubo   0.457
1  sbsw2011   0.457
2      owle   0.457
3    coupon   0.422
4      code   0.288

Document 8921:
     Word  TF-IDF
0    foam   0.524
1  window   0.379
2     wan   0.375
3   bring   0.357
4      na   0.340

Document 8922:
         Word  TF-IDF
0     yelping   0.417
1      walkin   0.417
2  navigating   0.385
3      random   0.345
4     welcome   0.342

Document 8923:
      Word  TF-IDF
0  arsense   0.656
```

```
1  arwords   0.656
2   anyone   0.372

Document 8924:
         Word  TF-IDF
0          db   0.327
1       nosql   0.327
2  opensource   0.317
3          s2   0.317
4      worked   0.296

Document 8925:
      Word  TF-IDF
0      umm   0.439
1     shit   0.410
2  winning   0.398
3   killer   0.367
4     keep   0.320

Document 8926:
          Word  TF-IDF
0         prob   0.650
1    designing   0.443
2        would   0.405
3          say   0.383
4  application   0.264

Document 8927:
         Word  TF-IDF
0    nutshell   0.448
1  underneath   0.434
2         gym   0.388
3        gold   0.383
4        kind   0.365

Document 8928:
     Word  TF-IDF
0   perma   0.628
1    came   0.490
2  result   0.450
3     wow   0.404

Document 8929:
        Word  TF-IDF
0      kudos   0.487
1       iron   0.441
2    clearly   0.435
3  smackdown   0.425
4        dev   0.369

Document 8930:
       Word  TF-IDF
0  terrible   0.531
1    anyway   0.520
2   popular   0.471
3   concept   0.462
```

```
4      link   0.111

Document 8931:
        Word  TF-IDF
0  billboard   0.458
1    hacking   0.437
2     square   0.437
3       real   0.346
4      video   0.281

Document 8932:
        Word  TF-IDF
0       news   0.561
1         z6   0.502
2  40075959p   0.388
3      valid   0.332
4       code   0.303

Document 8933:
      Word  TF-IDF
0  breathe   0.500
1     sigh   0.452
2     kill   0.405
3     plan   0.347
4   relief   0.321

Document 8934:
     Word  TF-IDF
0   bring   0.540
1    game   0.434
2   ipad2   0.423
3    know   0.410
4  people   0.388

Document 8935:
          Word  TF-IDF
0  influencers   0.473
1         core   0.337
2       action   0.321
3         sell   0.320
4         shop   0.280

Document 8936:
       Word  TF-IDF
0  bassquot   0.375
1    melody   0.375
2  quotdope   0.375
3     heavy   0.359
4   invades   0.331

Document 8937:
       Word  TF-IDF
0      sway   0.487
1    prefer   0.404
2    chrome   0.380
3  explorer   0.380
```

```
4   browser   0.368

Document 8938:
      Word  TF-IDF
0    donut   0.390
1  massage   0.390
2   lenewz   0.377
3       fo   0.368
4   behind   0.290

Document 8939:
       Word  TF-IDF
0  portable   0.592
1     thank   0.480
2   charger   0.455
3     ready   0.441
4      link   0.133

Document 8940:
         Word  TF-IDF
0  compelling   0.459
1     writing   0.405
2   excellent   0.372
3     leaving   0.358
4         dev   0.322

Document 8941:
     Word  TF-IDF
0   cheer   0.329
1   spark   0.329
2  lustre   0.296
3   pearl   0.291
4    door   0.291

Document 8942:
         Word  TF-IDF
0      schema   0.479
1  navigation   0.428
2   interface   0.422
3   designing   0.403
4       check   0.331

Document 8943:
       Word  TF-IDF
0     magic   0.766
1  hmmmtaxi   0.449
2    appear   0.430
3    austin   0.162

Document 8944:
            Word  TF-IDF
0            xml   0.446
1          skier   0.446
2  visualization   0.421
3             3d   0.357
4           data   0.307
```

```
Document 8945:
              Word  TF-IDF
0  festivalexplorer   0.588
1            solves   0.588
2           finally   0.436
3              sxsw   0.246
4       application   0.215

Document 8946:
       Word  TF-IDF
0        il   0.436
1   nowhere   0.235
2  lexpress   0.235
3        os   0.235
4     primo   0.235

Document 8947:
            Word  TF-IDF
0  quotpopupquot   0.594
1      marketing   0.538
2          smart   0.522
3          store   0.248
4           link   0.156

Document 8948:
     Word  TF-IDF
0  hilton   0.739
1    meet   0.674

Document 8949:
    Word  TF-IDF
0  store   0.846
1   link   0.532

Document 8950:
        Word  TF-IDF
0   captured   0.408
1     lesson   0.360
2      sheen   0.348
3  attention   0.338
4      whole   0.315

Document 8951:
      Word  TF-IDF
0   script   0.498
1  reading   0.399
2     film   0.391
3  sitting   0.388
4     folk   0.358

Document 8952:
     Word  TF-IDF
0   going   0.757
1  social   0.589
2    link   0.283
```

```
Document 8953:
              Word  TF-IDF
0  bloggersketchup   0.467
1     unexpectedly   0.467
2            quiet   0.433
3         compared   0.422
4            night   0.293

Document 8954:
        Word  TF-IDF
0  spectacle   0.536
1       line   0.485
2   enjoying   0.423
3      block   0.355
4       long   0.346

Document 8955:
       Word  TF-IDF
0     asset   0.429
1      corp   0.429
2  personal   0.377
3      care   0.360
4     stuff   0.310

Document 8956:
             Word  TF-IDF
0  taptuquotapple   0.426
1           taptu   0.426
2       videoquot   0.426
3              hd   0.408
4          shared   0.370

Document 8957:
         Word  TF-IDF
0    disgrace   0.490
1  everywhere   0.357
2        home   0.338
3        feel   0.334
4     instead   0.334

Document 8958:
     Word  TF-IDF
0   bryce   0.401
1    cent   0.401
2  spider   0.401
3   manor   0.401
4  secret   0.324

Document 8959:
    Word  TF-IDF
0    pic   0.586
1  store   0.570
2   sxsw   0.396
3  popup   0.377
4   link   0.179
```

```
Document 8960:
              Word  TF-IDF
0  entrepreneurial   0.470
1       collective   0.435
2         together   0.360
3             mind   0.358
4              put   0.323

Document 8961:
        Word  TF-IDF
0  epicenter   0.445
1        pdx   0.445
2    madness   0.379
3       drop   0.353
4     retail   0.336

Document 8962:
          Word  TF-IDF
0     argument   0.472
1  parenthesis   0.472
2       settle   0.472
3       opened   0.346
4          use   0.264

Document 8963:
      Word  TF-IDF
0      app   0.515
1  society   0.396
2      war   0.356
3     heat   0.356
4    group   0.315

Document 8964:
    Word  TF-IDF
0   mifi   0.543
1  solid   0.543
2   rock   0.379
3    far   0.369
4     cc   0.361

Document 8965:
         Word  TF-IDF
0    cornered   0.517
1        easy   0.394
2  apparently   0.377
3      market   0.371
4        demo   0.335

Document 8966:
          Word  TF-IDF
0       lovely   0.558
1  integration   0.548
2      awesome   0.353
3       thanks   0.350
4          get   0.270
```

```
Document 8967:
         Word  TF-IDF
0      appits   0.366
1       solid   0.351
2    resource   0.351
3  scheduling   0.351
4    planning   0.308

Document 8968:
     Word  TF-IDF
0   tenet   0.872
1  design   0.489

Document 8969:
       Word  TF-IDF
0     glued   0.558
1  included   0.516
2  expected   0.468
3  everyone   0.352
4       day   0.282

Document 8970:
          Word  TF-IDF
0        music   0.507
1        legal   0.414
2          5th   0.387
3  downloading   0.366
4      torrent   0.362

Document 8971:
      Word  TF-IDF
0      day   0.432
1    newly   0.427
2      one   0.416
3    added   0.345
4  amazing   0.302

Document 8972:
       Word  TF-IDF
0  behavior   0.314
1  swarming   0.314
2  flocking   0.314
3    collab   0.314
4     basis   0.304

Document 8973:
         Word  TF-IDF
0   realistic   0.401
1  explaining   0.388
2         gee   0.379
3  experiment   0.364
4         bot   0.348

Document 8974:
          Word  TF-IDF
```

```
0    regularly   0.336
1  programming   0.334
2    interrupt   0.334
3    scheduled   0.331
4          try   0.319

Document 8975:
    Word  TF-IDF
0  booth   0.952
1   link   0.307

Document 8976:
          Word  TF-IDF
0  impedimenta   0.399
1          lug   0.370
2    traveling   0.331
3      leaving   0.311
4        light   0.309

Document 8977:
        Word  TF-IDF
0        may   0.461
1  launching   0.443
2     called   0.344
3    network   0.324
4     circle   0.313

Document 8978:
         Word  TF-IDF
0        mall   0.487
1  absolutely   0.478
2   beautiful   0.451
3       track   0.364
4         guy   0.297

Document 8979:
   Word  TF-IDF
0  mean   0.702
1   way   0.573
2   get   0.422

Document 8980:
       Word  TF-IDF
0  acludont   0.444
1    thrown   0.444
2       dad   0.425
3      maes   0.329
4    maggie   0.325

Document 8981:
      Word  TF-IDF
0    hubby   0.389
1   effing   0.376
2  towards   0.352
3   lineup   0.347
4     wife   0.347
```

```
Document 8982:
     Word  TF-IDF
0   fairy   0.357
1    mail   0.323
2  mother   0.323
3     sat   0.284
4  1230pm   0.279

Document 8983:
       Word  TF-IDF
0  appeared   0.420
1  prepares   0.420
2    motley   0.420
3      fool   0.379
4     fight   0.318

Document 8984:
        Word  TF-IDF
0         ck   0.542
1  developed   0.489
2    support   0.430
3     friend   0.331
4       show   0.300

Document 8985:
         Word  TF-IDF
0  apparently   0.621
1       major   0.399
2     network   0.360
3      social   0.335
4      launch   0.326

Document 8986:
       Word  TF-IDF
0   knowing   0.540
1      fine   0.505
2  actually   0.424
3      idea   0.385
4     could   0.355

Document 8987:
          Word  TF-IDF
0      trouble   0.519
1          pak   0.519
2      anybody   0.458
3  downloading   0.458
4  application   0.205

Document 8988:
        Word  TF-IDF
0      crazy   0.513
1       post   0.456
2        pic   0.440
3     coming   0.414
4  temporary   0.344
```

```
Document 8989:
         Word  TF-IDF
0     culture   0.549
1  documented   0.525
2       crazy   0.395
3      doodle   0.386
4        much   0.343

Document 8990:
      Word  TF-IDF
0  geekest   0.426
1   hidden   0.426
2   nowyes   0.426
3    queue   0.302
4    place   0.265

Document 8991:
       Word  TF-IDF
0     party   0.489
1       six   0.445
2    lounge   0.391
3      bing   0.341
4  location   0.323

Document 8992:
          Word  TF-IDF
0           rt   0.386
1  interesting   0.380
2         look   0.332
3     possibly   0.298
4        major   0.285

Document 8993:
          Word  TF-IDF
0    localmind   0.551
1         luck   0.484
2    available   0.361
3  interesting   0.359
4         look   0.313

Document 8994:
     Word  TF-IDF
0   given   0.490
1   cover   0.458
2      ok   0.421
3    tool   0.395
4  better   0.361

Document 8995:
         Word  TF-IDF
0      ipaded   0.656
1  officially   0.521
2      others   0.411
3        sxsw   0.274
4       store   0.197
```

```
Document 8996:
      Word  TF-IDF
0   others   0.578
1  getting   0.561
2    ipad2   0.494
3    store   0.278
4     link   0.175

Document 8997:
      Word  TF-IDF
0   sorted   0.609
1  finally   0.451
2     data   0.401
3     stop   0.394
4     next   0.332

Document 8998:
        Word  TF-IDF
0  enchanted   0.536
1   kawasaki   0.440
2      stuff   0.400
3       talk   0.318
4        guy   0.312

Document 8999:
      Word  TF-IDF
0   earned   0.468
1    train   0.423
2    lunch   0.371
3  brought   0.368
4     took   0.345

Document 9000:
        Word  TF-IDF
0      adapt   0.360
1  appealing   0.360
2   overcome   0.344
3   sounding   0.344
4    somehow   0.318

Document 9001:
         Word  TF-IDF
0         aps   0.413
1      mobile   0.392
2         wmy   0.383
3  compatible   0.365
4  suggestion   0.331

Document 9002:
          Word  TF-IDF
0     concrete   0.359
1  fortunately   0.359
2       injury   0.359
3      bicycle   0.333
4         ouch   0.333
```

```
Document 9003:
     Word  TF-IDF
0    help   0.455
1  please   0.446
2     pal   0.319
3      4g   0.277
4     cab   0.269

Document 9004:
       Word  TF-IDF
0       pst   0.386
1    planet   0.339
2  minister   0.336
3     prime   0.333
4    lonely   0.330

Document 9005:
     Word  TF-IDF
0  planet   0.505
1  lonely   0.492
2   guide   0.418
3    week   0.372
4    free   0.289

Document 9006:
         Word  TF-IDF
0        take   0.396
1   virtually   0.343
2      prompt   0.328
3  creativity   0.328
4      memory   0.318

Document 9007:
       Word  TF-IDF
0   nonipad   0.520
1  thousand   0.444
2    tablet   0.356
3      seen   0.336
4       two   0.332

Document 9008:
          Word  TF-IDF
0  successquot   0.407
1     drowning   0.394
2      fucking   0.369
3     panelist   0.363
4       insane   0.349

Document 9009:
       Word  TF-IDF
0       ten   0.361
1     named   0.361
2      thru   0.344
3   discovr   0.323
4  musthave   0.323
```

```
Document 9010:
          Word  TF-IDF
0       moment   0.576
1   apparently   0.523
2       minute   0.501
3         line   0.311
4        store   0.216

Document 9011:
    Word  TF-IDF
0  block   0.541
1    wut   0.408
2   grew   0.395
3   past   0.325
4    min   0.316

Document 9012:
          Word  TF-IDF
0   apparently   0.611
1        block   0.532
2        tweet   0.499
3  application   0.307

Document 9013:
            Word  TF-IDF
0      francisco   0.414
1            sfo   0.414
2  international   0.359
3            san   0.348
4      hopefully   0.344

Document 9014:
        Word  TF-IDF
0  upgrading   0.610
1   thinking   0.513
2      crazy   0.474
3      right   0.374

Document 9015:
       Word  TF-IDF
0    tonite   0.571
1   display   0.562
2  industry   0.455
3     party   0.297
4      link   0.249

Document 9016:
        Word  TF-IDF
0     smoked   0.465
1     hahaha   0.445
2   painting   0.431
3  yesterday   0.337
4       much   0.290

Document 9017:
```

```
      Word  TF-IDF
0   legacy   0.386
1     tron   0.358
2  stopped   0.335
3    audio   0.335
4     sync   0.329

Document 9018:
         Word  TF-IDF
0       using   0.400
1    magnetic   0.335
2  brilliance   0.335
3   impressed   0.303
4        pure   0.303

Document 9019:
    Word  TF-IDF
0  nifty   0.406
1  cream   0.357
2    ice   0.357
3   food   0.310
4   band   0.306

Document 9020:
      Word  TF-IDF
0     full   0.482
1     room   0.463
2  staring   0.281
3     hold   0.273
4      air   0.261

Document 9021:
         Word  TF-IDF
0  apparently   0.598
1        shop   0.450
2         set   0.441
3   temporary   0.397
4      austin   0.296

Document 9022:
          Word  TF-IDF
0  applebreeds   0.428
1        midst   0.428
2      madness   0.365
3       discus   0.365
4          pre   0.365

Document 9023:
              Word  TF-IDF
0           quotad   0.518
1  preferencesquot   0.518
2       apparently   0.435
3            think   0.323
4              see   0.298

Document 9024:
```

```
              Word  TF-IDF
0  spanishspeaking   0.464
1            scout   0.450
2            trend   0.393
3            based   0.349
4               gt   0.317

Document 9025:
         Word  TF-IDF
0     schtuff   0.432
1       quake   0.398
2      timely   0.398
3  absolutely   0.391
4      finder   0.365

Document 9026:
       Word  TF-IDF
0     cream   0.496
1       ice   0.496
2  townquot   0.312
3     rated   0.299
4      amys   0.282

Document 9027:
     Word  TF-IDF
0  andoid   0.352
1    yall   0.345
2  choice   0.308
3   award   0.308
4     job   0.301

Document 9028:
         Word  TF-IDF
0   thingquot   0.665
1  quottheyre   0.665
2        line   0.311
3        link   0.136

Document 9029:
          Word  TF-IDF
0  demonstrate   0.398
1          mar   0.379
2     thousand   0.366
3         town   0.309
4          top   0.303

Document 9030:
    Word  TF-IDF
0    yay   0.762
1  thank   0.648

Document 9031:
    Word  TF-IDF
0  maybe   0.965
1   link   0.261
```

```
Document 9032:
      Word  TF-IDF
0     yeah   0.736
1  nothing   0.677

Document 9033:
      Word  TF-IDF
0      yep   0.428
1  believe   0.411
2       oh   0.343
3      yes   0.338
4     wait   0.334

Document 9034:
      Word  TF-IDF
0   marisa   0.485
1      aka   0.448
2    meyer   0.415
3     yelp   0.406
4  showing   0.346

Document 9035:
     Word  TF-IDF
0  picked   0.586
1     1st   0.532
2     yes   0.488
3     one   0.367

Document 9036:
        Word  TF-IDF
0        dah   0.590
1        doo   0.590
2        win   0.303
3  favorited   0.295
4      thats   0.199

Document 9037:
           Word  TF-IDF
0     expecting   0.583
1       exactly   0.570
2  presentation   0.418
3           yes   0.401

Document 9038:
        Word  TF-IDF
0     mayhem   0.449
1     mobile   0.425
2       hoot   0.356
3  hootsuite   0.317
4       blog   0.302

Document 9039:
        Word  TF-IDF
0      meill   0.396
1     flying   0.358
2      shoot   0.358
```

```
3  currently   0.343
4     direct   0.339

Document 9040:
      Word  TF-IDF
0     void   0.380
1  provide   0.334
2     fill   0.334
3    claim   0.325
4      biz   0.322

Document 9041:
        Word  TF-IDF
0  analytics   0.806
1   standard   0.503
2        use   0.312

Document 9042:
       Word  TF-IDF
0  birthday   0.397
1  reminder   0.375
2  saturday   0.371
3  friendly   0.364
4      dear   0.357

Document 9043:
      Word  TF-IDF
0  sokinda   0.400
1  useless   0.361
2  groupme   0.347
3  texting   0.336
4  signing   0.321

Document 9044:
       Word  TF-IDF
0      zomg   0.587
1  everyone   0.401
2      look   0.362
3     first   0.353
4       got   0.347

Document 9045:
       Word  TF-IDF
0  favorite   0.600
1   getting   0.482
2    coming   0.461
3    thanks   0.441

Document 9046:
       Word  TF-IDF
0     added   0.626
1  seatmate   0.388
2    become   0.351
3     plane   0.316
4    laptop   0.260
```

```
Document 9047:
     Word  TF-IDF
0   whats   0.596
1  detail   0.560
2    need   0.445
3  circle   0.365

Document 9048:
       Word  TF-IDF
0  preorder   0.461
1    amazon   0.439
2    kindle   0.439
3       bet   0.404
4       man   0.362

Document 9049:
       Word  TF-IDF
0      pick   0.532
1      used   0.484
2  tomorrow   0.444
3       buy   0.416
4       one   0.335

Document 9050:
          Word  TF-IDF
0        check   0.775
1  application   0.562
2         link   0.290

Document 9051:
   Word  TF-IDF
0  look   0.630
1  know   0.599
2   get   0.495

Document 9052:
    Word  TF-IDF
0   tmrw   0.690
1  could   0.420
2    buy   0.400
3  popup   0.274
4    new   0.230

Document 9053:
     Word  TF-IDF
0  xperia   0.479
1   later   0.412
2    play   0.380
3  detail   0.346
4   could   0.336

Document 9054:
      Word  TF-IDF
0  flannel   0.387
1   skinny   0.374
2     jean   0.357
```

```
3    beard   0.340
4    shirt   0.327

Document 9055:
   Word  TF-IDF
0  give   0.588
1   yet   0.585
2  away   0.559

Document 9056:
           Word  TF-IDF
0      reaction   0.483
1  announcement   0.464
2      realtime   0.459
3         would   0.319
4          love   0.309

Document 9057:
     Word  TF-IDF
0     cbs   0.439
1  scvngr   0.425
2     spy   0.405
3    quot   0.309
4    give   0.295

Document 9058:
          Word  TF-IDF
0  presentquot   0.378
1       meyers   0.362
2       quotdo   0.350
3         seth   0.350
4         xmas   0.341

Document 9059:
       Word  TF-IDF
0  knowthey   0.603
1     sayin   0.494
2      sell   0.379
3     place   0.376
4   product   0.327

Document 9060:
          Word  TF-IDF
0  tweetcaster   0.335
1     keywords   0.335
2          zip   0.324
3         hide   0.309
4   appreciate   0.309

Document 9061:
      Word  TF-IDF
0  padless   0.487
1  realize   0.404
2     gave   0.322
3     away   0.298
4    still   0.281
```

```
Document 9062:
       Word  TF-IDF
0     oryou   0.537
1     kiosk   0.498
2  purchase   0.419
3      find   0.321
4     ipad2   0.287

Document 9063:
      Word  TF-IDF
0  lugging   0.619
1     save   0.463
2   laptop   0.415
3     back   0.391
4      get   0.277

Document 9064:
        Word  TF-IDF
0  gathering   0.491
1    nothing   0.368
2      front   0.340
3       tech   0.331
4       geek   0.317

Document 9065:
     Word  TF-IDF
0   soooo   0.626
1  friday   0.442
2    sale   0.418
3   think   0.338
4  mobile   0.297

Document 9066:
                 Word  TF-IDF
0  disproportionately   0.448
1            stocking   0.448
2  influencerhipsters   0.448
3                 bet   0.346
4               thank   0.306

Document 9067:
       Word  TF-IDF
0    quotif   0.419
1    afford   0.373
2  consider   0.365
3    attend   0.356
4      able   0.333

Document 9068:
      Word  TF-IDF
0      try   0.455
1     many   0.404
2  tonight   0.393
3    event   0.392
4     make   0.348
```

```
Document 9069:
                Word  TF-IDF
0  quotgourdoughsquot   0.522
1                 ate   0.499
2                  em   0.445
3                sure   0.329
4                many   0.320

Document 9070:
      Word  TF-IDF
0  hooting   0.463
1    grown   0.429
2      men   0.418
3    bunch   0.389
4     made   0.293

Document 9071:
        Word  TF-IDF
0       cord   0.433
1     pocket   0.403
2  essential   0.393
3       mine   0.381
4      extra   0.381

Document 9072:
       Word  TF-IDF
0       rad   0.524
1      stop   0.418
2     ready   0.405
3  tomorrow   0.400
4      case   0.367

Document 9073:
      Word  TF-IDF
0  charger   0.619
1   friend   0.585
2     best   0.525

Document 9074:
          Word  TF-IDF
0  boomersquot   0.472
1     quotyour   0.452
2          mom   0.386
3       center   0.365
4   convention   0.360

Document 9075:
       Word  TF-IDF
0  quotyour   0.562
1  ipadquot   0.555
2       mom   0.480
3   session   0.382

Document 9076:
       Word  TF-IDF
```

```
0  quotyour   0.433
1  ipadquot   0.428
2       acc   0.395
3       mom   0.370
4  ballroom   0.365

Document 9077:
        Word  TF-IDF
0  convinced   0.460
1     switch   0.444
2   coverage   0.417
3         pr   0.413
4       back   0.315

Document 9078:
          Word  TF-IDF
0          cue   0.427
1         hype   0.427
2    regularly   0.325
3  programming   0.323
4    interrupt   0.323

Document 9079:
              Word  TF-IDF
0          lavelle   0.510
1  quotpapyrussort   0.510
2         ipadquot   0.418
3              lol   0.388
4             nice   0.310

Document 9080:
          Word  TF-IDF
0    quotmight   0.389
1  playstation   0.389
2  essentially   0.360
3         xbox   0.351
4    todayquot   0.351

Document 9081:
        Word  TF-IDF
0  incorrect   0.489
1       rate   0.443
2     launch   0.403
3     report   0.361
4        say   0.281

Document 9082:
        Word  TF-IDF
0  screaming   0.597
1    running   0.482
2       shop   0.342
3        guy   0.337
4       open   0.306

Document 9083:
       Word  TF-IDF
```

```
0        ie   0.594
1  location   0.414
2    future   0.410
3    around   0.390
4       say   0.371

Document 9084:
       Word  TF-IDF
0  location   0.910
1    future   0.300
2       say   0.271
3      link   0.096

Document 9085:
         Word  TF-IDF
0  stabilizer   0.390
1         cam   0.345
2        mode   0.345
3  suggestion   0.313
4       image   0.313

Document 9086:
        Word  TF-IDF
0  benmcgraw   0.381
1   gmailcom   0.381
2  lightning   0.365
3      stage   0.312
4      email   0.292

Document 9087:
         Word  TF-IDF
0         yup   0.495
1       third   0.487
2  suggestion   0.458
3         yet   0.365
4          cc   0.363

Document 9088:
         Word  TF-IDF
0  everywhere   0.968
1        link   0.251

Document 9089:
          Word  TF-IDF
0         wave   0.397
1    regularly   0.348
2    interrupt   0.347
3  programming   0.347
4    scheduled   0.343

Document 9090:
        Word  TF-IDF
0  physician   0.516
1  operating   0.269
2        fda   0.269
3   dataquot   0.258
```

```
4     relies   0.258

Document 9091:
         Word  TF-IDF
0  complained   0.387
1     yorkers   0.387
2    attended   0.342
3        fell   0.330
4     verizon   0.288

Document 9092:
      Word  TF-IDF
0  checkin   0.511
1     test   0.508
2    offer   0.506
3       rt   0.454
4     link   0.136
```

- Printed the top 5 most informative words in each tweet based on TF-IDF scores which plays a role in showing words that have a meaning in each tweet

In [ ]:

```
X_tfidf_dense = tfidf_matrix.toarray()
def print_top_features(document_index, top_n=10):
    scores = X_tfidf_dense[document_index]
    vocabulary = tfidf_vect.get_feature_names_out()
    print("Vocabulary:", vocabulary)
    feature_scores = [(vocabulary[i], scores[i]) for i in range(len(vocabulary))]

    feature_scores.sort(key=lambda x: x[1], reverse=True)


    print(f"Top {top_n} features for document {document_index}:")
    for term, score in feature_scores[:top_n]:
        print(f"{term}: {score:.4f}")
    print()


for i in range(min(5, tfidf_matrix.shape[0])):
    print_top_features(i)

vocabulary = tfidf_vect.get_feature_names_out()
print("Vocabulary:", vocabulary)

df_vectorization = pd.DataFrame(X_tfidf_dense, columns=vocabulary)
print(df.head())
```

```
Vocabulary: ['0310apple' '100' '103011p' ... 'zuckerberg' 'zynga' 'zzzs']
Top 10 features for document 0:
hr: 0.4134
plugin: 0.4134
station: 0.3669
upgrade: 0.3633
dead: 0.3511
tweeting: 0.3333
3g: 0.3247
need: 0.2275
0310apple: 0.0000
100: 0.0000

Vocabulary: ['0310apple' '100' '103011p' ... 'zuckerberg' 'zynga' 'zzzs']
Top 10 features for document 1:
appreciate: 0.4787
likely: 0.4300
giving: 0.3358
also: 0.3319
design: 0.3041
awesome: 0.3032
know: 0.2806
free: 0.2462
application: 0.1983
0310apple: 0.0000

Vocabulary: ['0310apple' '100' '103011p' ... 'zuckerberg' 'zynga' 'zzzs']
Top 10 features for document 2:
sale: 0.6098
wait: 0.5611
also: 0.5598
0310apple: 0.0000
100: 0.0000
103011p: 0.0000
1045am3: 0.0000
10am: 0.0000
10k: 0.0000
10mins: 0.0000

Vocabulary: ['0310apple' '100' '103011p' ... 'zuckerberg' 'zynga' 'zzzs']
Top 10 features for document 3:
year: 0.5942
crashy: 0.5268
festival: 0.3867
hope: 0.3558
app: 0.3053
0310apple: 0.0000
100: 0.0000
103011p: 0.0000
1045am3: 0.0000
10am: 0.0000

Vocabulary: ['0310apple' '100' '103011p' ... 'zuckerberg' 'zynga' 'zzzs']
Top 10 features for document 4:
booksconferences: 0.3615
mullenweg: 0.3615
```

```
fri: 0.3222
wordpress: 0.3222
matt: 0.3094
oreilly: 0.3094
tim: 0.2854
stuff: 0.2608
tech: 0.2434
marissa: 0.1945

Vocabulary: ['0310apple' '100' '103011p' ... 'zuckerberg' 'zynga' 'zzzs']
                                          tweet_text  \
0  [3g, hr, tweeting, dead, need, upgrade, plugin...
1  [know, awesome, application, likely, appreciat...
2                                 [wait, also, sale]
3          [hope, year, festival, crashy, year, app]
4  [great, stuff, fri, marissa, mayer, tim, oreil...

  emotion_in_tweet_is_directed_at  \
0                          iPhone
1              iPad or iPhone App
2                            iPad
3              iPad or iPhone App
4                          Google

  is_there_an_emotion_directed_at_a_brand_or_product  \
0                                   Negative emotion
1                                   Positive emotion
2                                   Positive emotion
3                                   Negative emotion
4                                   Positive emotion

                                      original_tweet
0  .@wesley83 I have a 3G iPhone. After 3 hrs twe...
1  @jessedee Know about @fludapp ? Awesome iPad/i...
2  @swonderlin Can not wait for #iPad 2 also. The...
3  @sxsw I hope this year's festival isn't as cra...
4  @sxtxstate great stuff on Fri #SXSW: Marissa M...
```

In [ ]:

```
print (df.head())
```

```
                                          tweet_text  \
0  [3g, hr, tweeting, dead, need, upgrade, plugin...
1  [know, awesome, application, likely, appreciat...
2                                  [wait, also, sale]
3          [hope, year, festival, crashy, year, app]
4  [great, stuff, fri, marissa, mayer, tim, oreil...

  emotion_in_tweet_is_directed_at  \
0                          iPhone
1              iPad or iPhone App
2                            iPad
3              iPad or iPhone App
4                          Google

  is_there_an_emotion_directed_at_a_brand_or_product  \
0                                   Negative emotion
1                                   Positive emotion
2                                   Positive emotion
3                                   Negative emotion
4                                   Positive emotion

                                      original_tweet
0  .@wesley83 I have a 3G iPhone. After 3 hrs twe...
1  @jessedee Know about @fludapp ? Awesome iPad/i...
2  @swonderlin Can not wait for #iPad 2 also. The...
3  @sxsw I hope this year's festival isn't as cra...
4  @sxtxstate great stuff on Fri #SXSW: Marissa M...
```

In [ ]:

```
#initializing sentiment analyze
from nltk.sentiment import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
sent = SentimentIntensityAnalyzer()
def get_sentiment_scores(text):
    if isinstance(text, list):
        text = ' '.join(text)

    return sent.polarity_scores(text)
def get_sentiment_label(scores):
    compound = scores['compound']
    if compound >= 0.05:
        return 'positive'
    elif compound <= -0.05:
        return 'negative'
    else:
        return 'neutral'
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
```

VADER compound score + label extraction

In [ ]:

```
# Applied sentiment analysis with the fixed function
df['sentiment_scores'] = df['tweet_text'].apply(get_sentiment_scores)

# Extracted the compound score
df['compound_score'] = df['sentiment_scores'].apply(lambda x: x['compound'])
```

```
#Applied sentiment labeling
df['sentiment_label'] = df['sentiment_scores'].apply(get_sentiment_label)

# Display the results
print(df[['tweet_text', 'compound_score', 'sentiment_label']].head())
```

```
                                          tweet_text  compound_score  \
0  [3g, hr, tweeting, dead, need, upgrade, plugin...         -0.6486   
1  [know, awesome, application, likely, appreciat...          0.9100   
2                                [wait, also, sale]          0.0000   
3          [hope, year, festival, crashy, year, app]          0.7269   
4  [great, stuff, fri, marissa, mayer, tim, oreil...          0.6249   

  sentiment_label  
0        negative  
1        positive  
2         neutral  
3        positive  
4        positive  
```

- Aplied vader compound score + labeling
- compound_score is a single number that summarized the tweet's emotional tone
- sentiment_label translated the score to positive, negative or neutral.

used the above to compare with ML models

Modeling

Logistic regression model

In [ ]:
```
#Splitting the df into training and testing data to an algorithm
y = df.is_there_an_emotion_directed_at_a_brand_or_product
X_train, X_test, y_train, y_test = train_test_split(tfidf_matrix, y, test_size=0.3,
log_reg = LogisticRegression(random_state=42)
log_reg.fit(X_train, y_train)

y_pred = log_reg.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
                                    precision    recall  f1-score   support

                      I can't tell       0.00      0.00      0.00        41
                  Negative emotion       0.73      0.04      0.08       182
No emotion toward brand or product       0.67      0.88      0.76      1585
                  Positive emotion       0.64      0.44      0.52       920

                          accuracy                           0.66      2728
                         macro avg       0.51      0.34      0.34      2728
                      weighted avg       0.65      0.66      0.62      2728
```

- The model had a accuracy of 66% meaning it predicted the sentiments for 2 out of every 3 tweets.This metric is misleading because the recall for "I_cant_tell" and "Negative_emotion" are zero and 4% respectively this means that the model could barely tell actual and negative sentiments. With low macro-average f1 score indicating the model is not reliable across all sentiment categories.
- There is a clear class imbalance shown by the two classes "i cant tell and negative emotion" which had 41 and 182 instances in the whole document.

In [ ]:
```
#Splitting the data into training and testing data to an algorithm
y = df['sentiment_label']
X_train, X_test, y_train, y_test = train_test_split(tfidf_matrix, y, test_size=0.3,
log_reg = LogisticRegression(random_state=42)
log_reg.fit(X_train, y_train)

y_pred = log_reg.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

    negative       0.97      0.32      0.48       325
     neutral       0.78      0.90      0.84      1179
    positive       0.85      0.88      0.86      1224

    accuracy                           0.82      2728
   macro avg       0.87      0.70      0.73      2728
weighted avg       0.83      0.82      0.81      2728
```

- Used a different approach of sentiment_label which improved the accuracy to 82% and macro average 73% which was better than the prior but still was biased

Random Forest

In [ ]:
```
param_grid = {
    "criterion":["gini","entropy"],
    "max_depth":[None,5,10,15],
    "min_samples_split":[2,10],
    "min_samples_leaf":[1,4,8]
}
```

In [ ]:
```
forest = RandomForestClassifier(
    n_estimators=200,
    criterion='gini',
    max_depth=20,
    random_state=42,
    n_jobs=-3
)
grid_search = GridSearchCV(
    estimator=forest,
    param_grid=param_grid,
    cv=5,
    scoring='accuracy',
    n_jobs=-1
)
grid_search.fit(X_train,y_train)


y_pred = grid_search.predict(X_test)
print(metrics.classification_report(y_test, y_pred))
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

```
              precision    recall  f1-score   support

    negative       0.94      0.42      0.58       325
     neutral       0.77      0.97      0.86      1179
    positive       0.94      0.83      0.88      1224

    accuracy                           0.84      2728
   macro avg       0.88      0.74      0.77      2728
weighted avg       0.86      0.84      0.83      2728

Accuracy: 0.843108504398827
```

- Random Forest Improved over Logistic Regression with Accuracy of 84% and macro avg of 77%, by capturing more complex patterns in TF-IDF features.added value as interpretable model for sentiment classification. However, its lack of sequential awareness limited its ability to capture deeper meaning in tweets.

Multiclass Neural Network

In [ ]:
```
# Prepared inputs and labels
X = df['tweet_text'].astype(str).values
y = df['sentiment_label'].values
```

```
# Encoded sentiment labels
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# Tokenized and pad text
tokenizer = Tokenizer(num_words=10000, oov_token="<OOV>")
tokenizer.fit_on_texts(X)
X_seq = tokenizer.texts_to_sequences(X)
X_pad = pad_sequences(X_seq, maxlen=100, padding='post', truncating='post')

# Trained/tested split
X_train, X_test, y_train, y_test = train_test_split(X_pad, y_encoded, test_size=0.2
```

In [ ]:

```
# Encode labels
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
y_categorical = to_categorical(y_encoded)  # One-hot encode the labels

# Splited the data
X_train, X_test, y_train, y_test = train_test_split(tfidf_matrix.toarray(), y_categ

# Defined model
model = Sequential()
model.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(32, activation='relu'))
model.add(Dense(y_categorical.shape[1], activation='softmax'))  # Output layer for

# Compile model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy
# Train model
model.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_test, y_tes

# Predict
y_pred_prob = model.predict(X_test)
y_pred = y_pred_prob.argmax(axis=1)
y_true = y_test.argmax(axis=1)

# Evaluation
print("\nClassification Report for Multiclass Neural Network:\n")
print(classification_report(y_true, y_pred, target_names=label_encoder.classes_))
```

```
Epoch 1/5
199/199 ━━━━━━━━━━━━━━━━━━━━ 3s 10ms/step - accuracy: 0.5235 - loss: 0.9815 - val_ac
curacy: 0.7386 - val_loss: 0.6241
Epoch 2/5
199/199 ━━━━━━━━━━━━━━━━━━━━ 2s 8ms/step - accuracy: 0.8386 - loss: 0.4234 - val_acc
uracy: 0.8383 - val_loss: 0.4358
Epoch 3/5
199/199 ━━━━━━━━━━━━━━━━━━━━ 2s 11ms/step - accuracy: 0.9631 - loss: 0.1421 - val_ac
curacy: 0.8512 - val_loss: 0.4101
Epoch 4/5
199/199 ━━━━━━━━━━━━━━━━━━━━ 2s 9ms/step - accuracy: 0.9887 - loss: 0.0556 - val_acc
uracy: 0.8570 - val_loss: 0.4331
Epoch 5/5
199/199 ━━━━━━━━━━━━━━━━━━━━ 2s 8ms/step - accuracy: 0.9960 - loss: 0.0283 - val_acc
uracy: 0.8596 - val_loss: 0.4576
86/86 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step

Classification Report for Multiclass Neural Network:

              precision    recall  f1-score   support

    negative       0.81      0.62      0.71       325
     neutral       0.85      0.88      0.87      1179
    positive       0.88      0.90      0.89      1224

    accuracy                           0.86      2728
   macro avg       0.85      0.80      0.82      2728
weighted avg       0.86      0.86      0.86      2728
```

In [ ]:
```python
print(y)
```

```
['negative' 'positive' 'neutral' ... 'neutral' 'negative' 'neutral']
```

- The Accuracy of 86%, macro average f1 score 0f 82% the neural network model performed well offering improved acccuracy over prior models but wasn't able to fully capture 'tone, sarcasm or negation' and it was unable to process word order.

LSTM- Long-short term memory

In [ ]:
```python
# Tokenization
tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(df['tweet_text'])
X_seq = tokenizer.texts_to_sequences(df['tweet_text'])
X_pad = pad_sequences(X_seq, maxlen=100)

# Encode labels
le = LabelEncoder()
y = le.fit_transform(df['sentiment_label'])
y_cat = to_categorical(y)

# Split data
X_train, X_test, y_train, y_test = train_test_split(X_pad, y_cat, test_size=0.2, st
```

```
# Build LSTM Model
model_lstm = Sequential([
    Embedding(input_dim=5000, output_dim=128, input_length=100),
    LSTM(128),
    Dropout(0.5),
    Dense(64, activation='relu'),
    Dropout(0.3),
    Dense(y_cat.shape[1], activation='softmax')
])

model_lstm.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['acc
model_lstm.summary()

# Train
history = model_lstm.fit(X_train, y_train, epochs=5, batch_size=32, validation_data

# Evaluate
y_pred = model_lstm.predict(X_test)
y_pred_classes = y_pred.argmax(axis=1)
y_true = y_test.argmax(axis=1)

print(classification_report(y_true, y_pred_classes, target_names=le.classes_))
```

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	?	0 (unbuilt)
lstm (LSTM)	?	0 (unbuilt)
dropout (Dropout)	?	0
dense (Dense)	?	0 (unbuilt)
dropout_1 (Dropout)	?	0
dense_1 (Dense)	?	0 (unbuilt)

Total params: 0 (0.00 B)

Trainable params: 0 (0.00 B)

Non-trainable params: 0 (0.00 B)

```
Epoch 1/5
228/228 ━━━━━━━━━━━━━━━━━━━━ 43s 173ms/step - accuracy: 0.5651 - loss: 0.8961 - val_
accuracy: 0.8224 - val_loss: 0.4739
Epoch 2/5
228/228 ━━━━━━━━━━━━━━━━━━━━ 40s 176ms/step - accuracy: 0.8953 - loss: 0.3065 - val_
accuracy: 0.8851 - val_loss: 0.3545
Epoch 3/5
228/228 ━━━━━━━━━━━━━━━━━━━━ 43s 185ms/step - accuracy: 0.9683 - loss: 0.1127 - val_
accuracy: 0.8895 - val_loss: 0.3820
Epoch 4/5
228/228 ━━━━━━━━━━━━━━━━━━━━ 79s 171ms/step - accuracy: 0.9830 - loss: 0.0641 - val_
accuracy: 0.8955 - val_loss: 0.4286
Epoch 5/5
228/228 ━━━━━━━━━━━━━━━━━━━━ 40s 168ms/step - accuracy: 0.9863 - loss: 0.0557 - val_
accuracy: 0.8939 - val_loss: 0.4967
57/57 ━━━━━━━━━━━━━━━━━━━━ 3s 46ms/step
              precision    recall  f1-score   support

    negative       0.79      0.66      0.72       206
     neutral       0.89      0.93      0.91       797
    positive       0.92      0.92      0.92       816

    accuracy                           0.89      1819
   macro avg       0.87      0.84      0.85      1819
weighted avg       0.89      0.89      0.89      1819
```

In [ ]:
```python
y = df['sentiment_label']
# Encoded labels
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
y_categorical = to_categorical(y_encoded)

# Tokenization and padding already done earlier as X_pad
X_train, X_test, y_train, y_test = train_test_split(X_pad, y_categorical, test_size

# LSTM Model
model = Sequential()
model.add(Embedding(input_dim=10000, output_dim=64, input_length=100))
model.add(LSTM(64, return_sequences=False))
model.add(Dropout(0.3))
model.add(Dense(32, activation='relu'))
model.add(Dense(y_categorical.shape[1], activation='softmax'))  # For multiclass

# Compile and Train
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy
model.summary()

model.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_test, y_tes

# Predict and Evaluate
y_pred_prob = model.predict(X_test)
y_pred = y_pred_prob.argmax(axis=1)
y_true = y_test.argmax(axis=1)

from sklearn.metrics import classification_report
```

```
print("\nClassification Report for LSTM Model:\n")
print(classification_report(y_true, y_pred, target_names=label_encoder.classes_))
```

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	?	0 (unbuilt)
lstm_1 (LSTM)	?	0 (unbuilt)
dropout_2 (Dropout)	?	0
dense_2 (Dense)	?	0 (unbuilt)
dense_3 (Dense)	?	0 (unbuilt)

```
Total params: 0 (0.00 B)
Trainable params: 0 (0.00 B)
Non-trainable params: 0 (0.00 B)
```

```
Epoch 1/5
199/199 ━━━━━━━━━━━━━━━━━━━━ 20s 85ms/step - accuracy: 0.5350 - loss: 0.9529 - val_a
ccuracy: 0.7573 - val_loss: 0.6148
Epoch 2/5
199/199 ━━━━━━━━━━━━━━━━━━━━ 12s 62ms/step - accuracy: 0.8475 - loss: 0.4227 - val_a
ccuracy: 0.8512 - val_loss: 0.4116
Epoch 3/5
199/199 ━━━━━━━━━━━━━━━━━━━━ 20s 61ms/step - accuracy: 0.9489 - loss: 0.1502 - val_a
ccuracy: 0.8790 - val_loss: 0.3918
Epoch 4/5
199/199 ━━━━━━━━━━━━━━━━━━━━ 21s 64ms/step - accuracy: 0.9871 - loss: 0.0536 - val_a
ccuracy: 0.8842 - val_loss: 0.4092
Epoch 5/5
199/199 ━━━━━━━━━━━━━━━━━━━━ 12s 62ms/step - accuracy: 0.9885 - loss: 0.0444 - val_a
ccuracy: 0.8871 - val_loss: 0.4973
86/86 ━━━━━━━━━━━━━━━━━━━━ 2s 24ms/step

Classification Report for LSTM Model:

              precision    recall  f1-score   support

    negative       0.88      0.60      0.72       325
     neutral       0.88      0.92      0.90      1179
    positive       0.90      0.93      0.91      1224

    accuracy                           0.89      2728
   macro avg       0.89      0.82      0.84      2728
weighted avg       0.89      0.89      0.88      2728
```

- The LSTM was the most effective model for sentiment analysis. It captured both meaning and structure, providing reliable sentiment classification for complex, short-form text like tweets. It was balanced across all emotions and handled confusing/emotional tweets better than the other models with Accuracy score of 89%, macro average f1 score of 84%

Conclusion

- After building and evaluating multiple machine learning and deep learning models, LSTM demonstrated the highest effectiveness in understanding language in a better way, making it the most suitable model for sentiment analysis on social media text. Future enhancements could include more training data, use of pre-trained embeddings like GloVe or BERT, or expanding into bidirectional LSTMs for even deeper context learning.

Recommendations

Data Imbalance:

The model struggles to detect negative sentiment reliably due to a significant under-representation (only 326 instances). To improve recall for the negative class, consider oversampling the negatives, undersampling the majority classes, or generating synthetic samples using methods like SMOTE. Handling of Missing Emotional Target Data: With about 64% of entries missing in the emotional target field, simply replacing missing values with "None" is insufficient. A more nuanced, multi-stage process is suggested—first identifying if a tweet targets any brand, then pinpointing the specific brand, and finally assessing the sentiment. This refined approach could yield deeper insights into brand-specific sentiment. Combat Neural Network Overfitting: The current neural network shows near-perfect performance on training data but plateaus on validation data, signaling overfitting. Integrate further regularization techniques (e.g., additional dropout layers, L1/L2 regularization) and early stopping to improve generalization on unseen data.

Explore Ensemble Methods:

Merging the high precision of the logistic regression model (which, however, has low recall for negatives) with other algorithms that better capture negative sentiment could help balance the precision-recall trade-off. An ensemble approach may provide more robust and reliable sentiment classifications across all sentiment categories.

To enhance sentiment analysis performance, it is crucial to balance the dataset through sampling or synthetic data generation, adopt a multi-stage classification process for more refined brand and sentiment detection, and address overfitting in the neural network with stronger regularization measures and early stopping. Moreover, integrating ensemble methods that combine models with complementary strengths could further balance precision and recall, leading to a more reliable analysis