ANALYSIS OF TWITTER AND NEWS DATASET TRENDING WORDS

Barah
Dept of Artificial Intelligence and Machine Learning,
Lambton college,
Toronto, Canada
c0860531@mylambton.ca

Catharin Jose
Dept of Artificial Intelligence and Machine Learning,
Lambton college,
Toronto, Canada
c0860087@mylambton.ca

Danny Jose
Dept of Artificial Intelligence and Machine Learning,
Lambton college,
Toronto, Canada
c0864600@mylambton.ca

Sri Bindu Chintakayala
Dept of Artificial Intelligence and Machine Learning,
Lambton college,
Toronto, Canada
c0857498@mylambton.ca

Gurveer Kaur
Dept of Artificial Intelligence and Machine Learning,
Lambton college,
Toronto, Canada
c0830495@mylambton.ca

***Abstract*— In this study, we used hashtag iPhones to collect 10,000 tweets from the dataset in order to determine trending terms. To locate comparable terms, we employ three clusters. We are exploring and showing the data freely, as well as categorizing and evaluating the groups.**

I. INTRODUCTION

In this project we are making an analysis of two datasets. One is on the news dataset and the other analysis on the twitter data. On this dataset, we are performing the natural language processing to analyse and find the trends. Initially we will collect the dataset from twitter and then perform all the preprocessing steps such as removing stopwords, finding trends, handing the languages. Then we will be analyzing the dataset by plotting graphs and creating clusters to get more insights into the dataset

By importing different libraries like Pandas, matplotlib and seaborn. Python's Pandas package is used to manipulate data collections. It offers tools for data exploration, cleaning, analysis, and manipulation. With the aid of Pandas, we can examine large data sets and draw conclusions based on statistical principles. Pandas can organize disorganized data sets, making them understandable and useful. For Python and its numerical extension NumPy, Matplotlib is a cross-platform data visualization and graphical charting package. As a result, it presents a strong open-source substitute for MATLAB. The APIs (Application Programming Interfaces) for matplotlib allow programmers to include graphs in GUI applications. The way a Python matplotlib script is written makes it possible to create a visual data plot in the majority of cases with just a few lines of code. Python's Seaborn package is mostly used to create statistical visuals. On top of matplotlib, Seaborn is a data visualization framework that is tightly connected with Python's Pandas data structures. The core of Seaborn is visualization, which aids in data exploration and comprehension. With the help of pandas, the dataset is loaded. In the second part of this project, we collected tweets from Twitter with help of Tweepy.

For use in development and education, Steven Bird, Edward Loper, and Ewan Klein created the Natural Language Toolkit, an open-source library for the Python programming language. It is appropriate for linguists without extensive programming experience, engineers and researchers who need to delve into computational linguistics, students, and educators because it includes a hands-on guide that introduces topics in computational linguistics as well as Python programming fundamentals. More than 50 corpora and lexical sources are included in NLTK, including Lin's Dependency Thesaurus, Problem Report Corpus, Open Multilingual Wordnet, and Penn Treebank Corpus.

II. DATA COLLECTION

A large portion of machine learning use cases involve natural language processing, but this requires a lot of data. Here, in the first part of the assignment, we have a dataset with the shape of (421993, 19) and in the second part, we fetched tweets from Twitter using tweepy and stored the data in a file. We must first utilize the Twitter API in order to gather tweets, thus we must register as developers on the Twitter applications website. From there, we obtained the consumer secret, consumer key, access token, and access token secret. Tweepy is a fantastic tool for gaining access to the Twitter API. Each application we'll build today starts out by requiring us to use Tweepy to create an API object that we can use to call functions from. We must first authenticate using our developer information before we can create the API object. Import Tweepy and add authentication data for that.

In the creation of the Twitter API, pagination is heavily utilized. iterating over user lists, direct messages, timelines, and other channels We need to provide a page/cursor argum ent in each of our queries to implement pagination. The prob lem here is that there is a lot of boilerplate code required onl y to manage the pagination loop. To simplify and reduce co de requirements for pagination, Tweepy includes the Cursor object. We create a data frame from the tweets after we have collected them.

Any NLP system requires a large amount of data for tra ining and testing. They are classified into two types of datas ets: valid and wrong (erroneous) data. Finding and obtaining a set of proper data is usually not an issue because correct t exts are available from various sources, albeit they may cont ain some errors. On the other side, it is difficult to obtain dat

a that contains errors such as typos, mistakes, and misspellin gs. This type of data is often gathered through a time-consu ming manual method that necessitates human annotation. Cr eating the erroneous dataset is one approach to acquire it fas ter. However, this raises the question of how to manufacture faulty sentences that correlate to genuine human errors.

III. SAVING DATA

For the second part of the project, to save the data, we created an excel file for the twitter data. We first used "df. to CSV" to create a data frame from the obtained data and store it as a . CSV file.

IV. CLEANING DATA

Both corporations and individuals should prioritize data cleanup. Data management includes data cleaning. Individuals and organizations amass a lot of private data over time! The information eventually gets antiquated. Data cleaning is the act of going through all of the information in a database and updating or removing any that is inaccurate, duplicated, badly structured, or irrelevant (source). Data cleansing is mainly concerned with updating, correcting, and consolidating data to make sure your system is as efficient as possible, even if it can and may entail removing information (source). Data cleaning is often done all at once, and if the information has been accumulating for years, it may take some time. Data cleaning should be done often for this reason.

On the dataset, we performed lot of data cleaning process su ch as below:
A) Stopword Removal:
Stop words are typical terms inside phrases that do not contr ibute value and may thus be removed when cleaning for NL P prior to analysis in English, among many other prominent languages.

b) Punctuation Removal:
There are several punctuations in the title text. Punctuation i s frequently unnecessary because it adds no value or meanin g to the NLP model. There are 32 punctuations in the "string " library.

c) Lemmatize/ Stem:
The process of reducing a word to its root form is known as stemming and lemmatizing. The major goal is to eliminate v ariants of the same term, lowering the corpus of words in the model. The distinction between stemming and lemmatizing is that stemming removes the end of the word without consi dering its context. Lemmatizing, on the other hand, evaluate s the context of the word and shortens it into its basic form d epending on the dictionary meaning. Stemming is a quicker method than Lemmantizing. As a result, there is a trade-off between speed and precision.

d) Other steps:
Based on the data, more cleaning procedures can be done. We have handled numbers, urls, emoji's and html tags withi n the dataset to enrich them.

To remove URLs from text, use the sub() function, as in result = re. sub(r'httpS+', '', my string). The re. sub() function replaces any URLs in the string with empty strings to eliminate them.

At times, since we are handling data from the internet there are chances of having data in different languages. It is observed that the dataset we are having contains data in different languages.

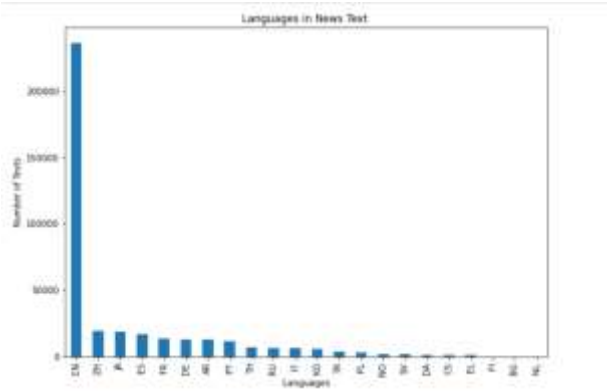

Figure 1: Languages present in the news data

Although, texts in English are more in number there are soo many other language texts in part 1 of the project's dataset. We filtered the text data by setting the language preference to English.

In the second part of the project, we imported detect from langdetect which helped to detect tweets only in English.

Finding missing values and handling missing values is a very important part in cleaning the data. If there are missing values present in the dataset then there are very less chances of accuracy. We have different methods to handle missing values. Such as mean, median, impute and so on. We used the mean method to handle the missing values. As we have text data in which we will have to remove stop words, punctuation and duplicates from the text in order to identify the most trending words in the dataset. For the process of cleaning. Firstly, we drop duplicates. Once that is done we convert the data into lowercase for punctuation and stopword removal.

For stop word removal, we imported stop words from nltk. corpus. Specifying the language of the stop word is also mandatory as we are using text data in English stop words in English can be imported. Tokenization is another important step that has been handled in the process. A token is the smallest possible word in a text. there are two types of tokenization. Sentence tokenization and word tokenization. It can be done by importing tokenize library.

We must eliminate any duplicates and numbers from the tweets in order to clean them up. To do that, we used the 'import re' command to import Regular Expressions. A regular expression is a group of characters that may be used to search for text strings that include specific letters, words, or character patterns. The "re"

module, which is pre-installed so you don't have to, imports Regular Expressions (REs, regexes, or regex patterns) into Python. We may search for matches in a string by using a number of functions in the re-module. Here, we used re. sub() to eliminate duplicates and other errors from the dataset. The re. sub() function, which denotes a substring, returns the string with replacement values.

V. DATA ANALYSIS

Finding Frequency of Words:

A frequency distribution for experiment results. A frequency distribution tracks the number of times an experiment outcome has happened. A frequency distribution, for example, might be used to record the frequency of each word type in a document. A frequency distribution is formally defined as a function that maps each sample to the number of times that sample occurred as an outcome.

In general, frequency distributions are created by running a series of experiments and incrementing the count for a sample every time it is the result of an experiment.

In the first part, to find the most trending words we used freqdist() which is a frequency distribution. It helps to get the frequency of each and every word in the text

	Trending_words	count
0	iphone	3185
1	apple	798
2	pro	513
3	phone	510
4	new	490
5	case	394
6	amp	270
7	max	269
8	user	268
9	get	265
10	ios	262
11	best	250
12	android	239
13	dynamic	201
14	portrait	195
15	free	194
16	mobile	189
17	app	177
18	buy	158
19	like	141

Figure 2: Frequency of the words in the dataset

Term Frequency- The way term frequency works is by examining how frequently a specific term is used in relation to the document.

Inverse document frequency examines how frequently (or infrequently) a term appears in the text.

the TF-IDF The frequency of a phrase across texts is negatively correlated with its significance. A term's frequency in a document is revealed by TF, while its relative rarity among the collection of documents is revealed by IDF. We may get our final TF-IDF value by averaging these numbers. The more relevant or important a term is, the higher its TF-IDF score; as a term becomes less relevant, its TF-IDF score will decrease until it is zero.

$$\text{tf-idf}(t) = \text{tf}(t, d) \times \text{idf}(t)$$

In text analysis, the cosine similarity measure of similarity is frequently used to assess document similarity. To calculate the cosine similarity, we use the formula below.

Similarity = (A.B) / (||A||.||B||)

When A and B are vectors, similarity is calculated as (A.B) / (||A||.||B||):

A and B are the dot product (A.B): It is calculated as the element-wise product of A and B added together.

The L2 norm of A is ||A||: It is calculated as the square root of the sum of the squares of the vector A's elements.

VI. CLUSTERING AND RESULTS

The k-means clustering method is an unsupervised machine learning methodology for identifying data item groupings in a dataset. There are other clustering algorithms, but k-means is one of the oldest and most often used. These characteristics make k-means clustering in Python relatively simple to implement, even for inexperienced programmers and data scientists.

We are using 3 clusters to divide what our dataset is talking about.

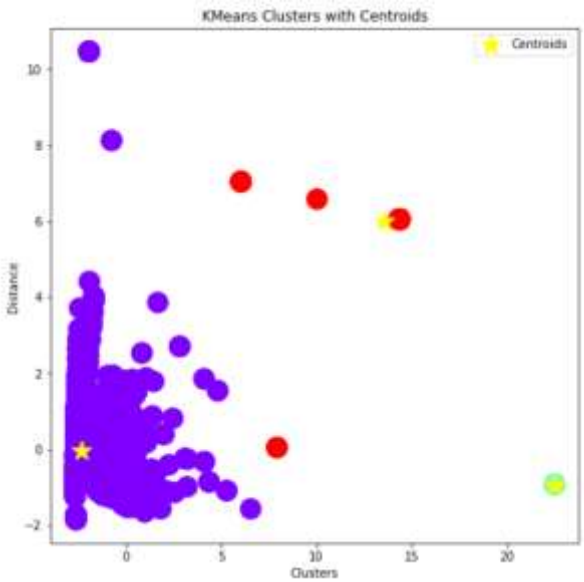

Figure 3: Kmeans clusters with centroid

We use TF-IDF which converts textual data into numerical vectors by taking into account the frequency of each word in

the document, the total number of words in the document, the total number of documents, and the number of documents that include each unique word.

On top of that cosine similarity is implemented to find a similarity measure that is frequently used in text analysis to measure document similarity.

Then will be performing the principal component analysis (PCA) which will reduce the dimensionality of a dataset with a large number of related variables while retaining as much variance as possible in the data. PCA discovers a collection of new variables from which the original variables are simply linear combinations

Here we get cluster0 which is in purple, as a biggest cluster among the three. The top five words among within the cluster0 are:

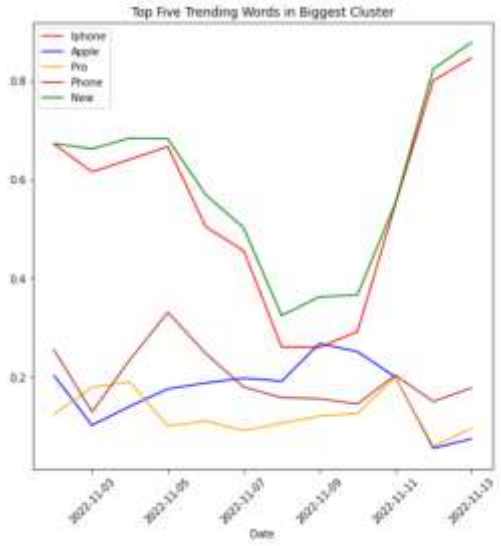

Figure 4: Trending words in cluster0

Now when we find the trends of the top 3 words within the largest cluster as shown below:

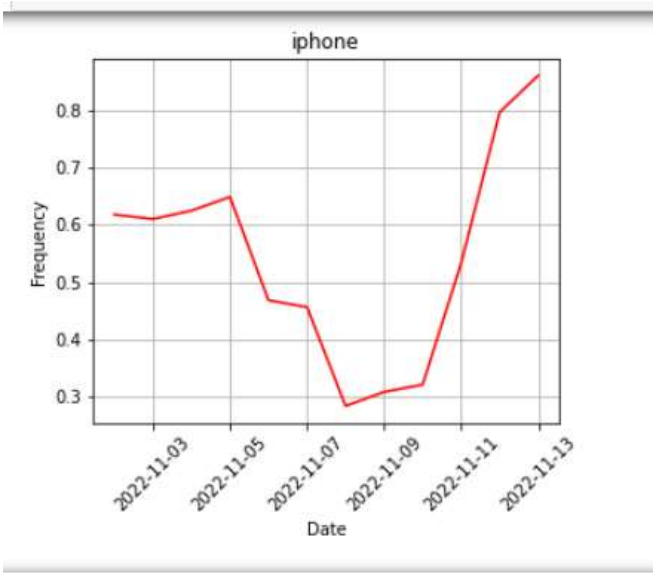

Figure 5: Word iphone in cluster0

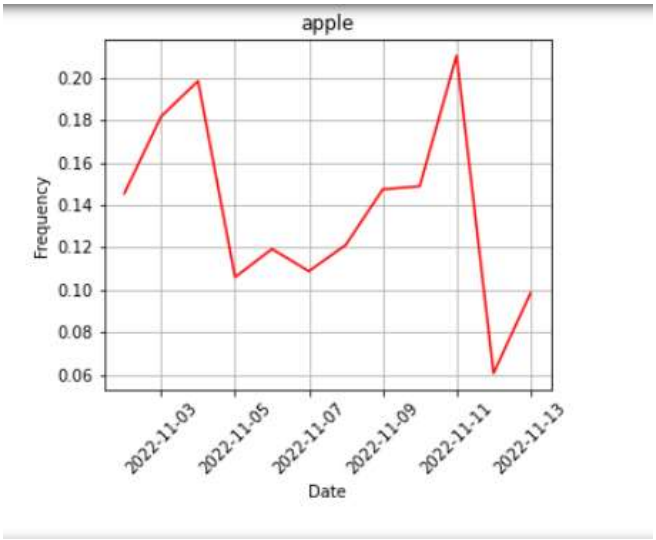

Figure 6: Second frequent word in cluster0

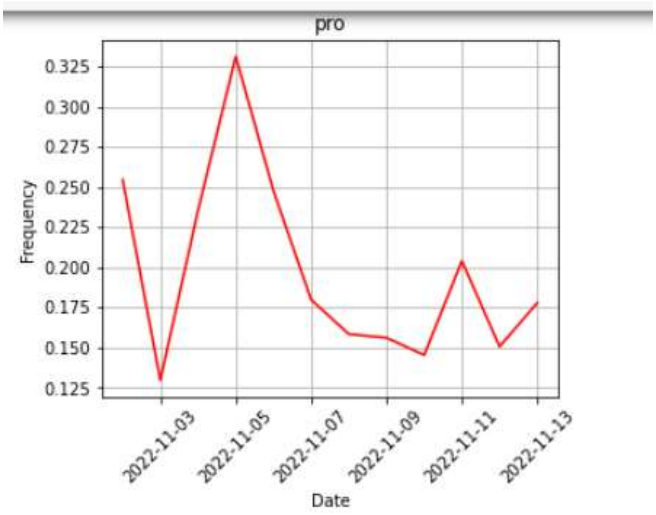

Figure 7: Third trending word in cluster0

These are the trends for the largest cluster which is 0. If we combine all the clusters into one and find the the trending words iphone and apple will remain in the top five words among all words. When we plot the graph based on time the data is generated for the top five words among all cluster we endup something like below:

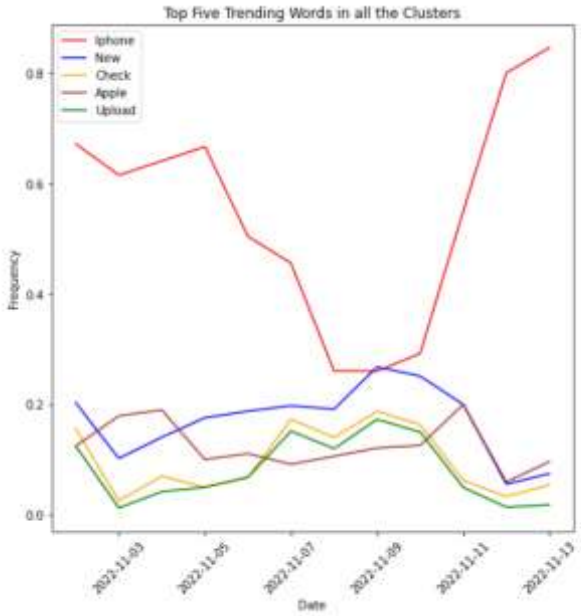

Figure 8: Top five trending words among all cluster

Similarly five trending words in the news dataset are news, top, share, update and say. When we plot the trend with respect to the date and frequency, we will get as below

Fiure

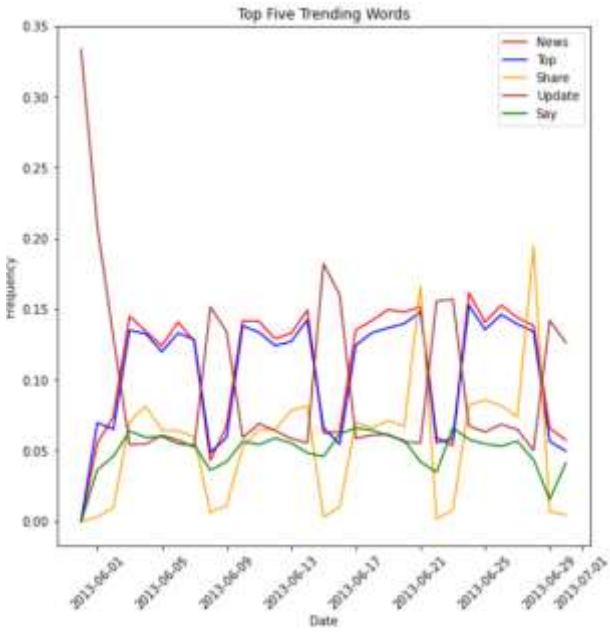

Figure 9: Top five trending words in the news dataset.

CONCLUSION

In this project, we analyzed news and Twitter data to identify the top 20 trending words. The top three trending words in the news dataset are news (32616 words), top (29812 words), and share (15912 words). We used two methods to identify the top trending words in the Twitter data. We extracted the top trending words from the largest cluster and the entire dataset. iPhone, Apple, and Pro are the top three trending words from the largest cluster. While the top three trending words in the entire dataset are iPhone, new, and check. In both scenarios, the most popular trending word was iPhone.

REFERENCES

[1] Arvai , K. (2022, September 1). *K-means clustering in Python: A practical guide*. Real Python. Retrieved November 13, 2022, from https://realpython.com/k-means-clustering-python/

[2] Hrkút, P., & Toth, Š. (2020, March). *Data collection for Natural Language Processing Systems - ResearchGate*. Retrieved November 14, 2022, from https://www.researchgate.net/publication/339645099_Data_Collection_for_Natural_Language_Processing_Systems

[3] Kuo, C. (n.d.). *Dimension reduction techniques with python*. Retrieved November 14, 2022, from https://towardsdatascience.com/dimension-reduction-techniques-with-python-f36ca7009e5c

[4] Kilmen, S. (2022, January 16). *Text vectorization using python: TF-IDF*. Okan Bulut. Retrieved November 13, 2022, from https://okan.cloud/posts/2022-01-16-text-vectorization-using-python-tf-idf/#:~:text=The%20TF%2DIDF%20vectorization%20transforms,documents%20including%20each%20unique%20word.

[5] Kilmen, S., & Okan , B. (2022, January 16). *Text vectorization using python: TF-IDF*. Okan Bulut. Retrieved November 13, 2022, from https://okan.cloud/posts/2022-01-16-text-vectorization-using-python-tf-idf/#:~:text=The%20TF%2DIDF%20vectorization%20transforms,documents%20including%20each%20unique%20word.

[6] M, R. (2022, July 12). *What is Matplotlib in python? how to use it for plotting?* ActiveState. Retrieved November 13, 2022, from https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/

[7] Raghunathan, D. (n.d.). *NLP in Python-Data Cleaning - towardsdatascience.com*. Retrieved November 14, 2022, from https://towardsdatascience.com/nlp-in-python-data-cleaning-6313a404a470