Project Detailed Design
Due: Tue Nov 1, 2022 11:59pm
Team 9
Angela Tseng
Yogesh Boricha
Abdul Shaik
Sadaf Davre

Section 1: Requirements

Oftentimes simply looking at a hashtag is not enough in understanding the purpose, meaning, and context surrounding the use of the hashtag. Even when the hashtag is searched in common search engines, there can be difficulty finding reliable new sources that explain it properly. Our project's goal is to address this ambiguity by providing reputable news sources to explain the meaning behind important trending hashtags.

A) Upon visiting our website, the user will be presented with a list of the top 30 hashtags currently trending on Twitter. These hashtags will be clickable links that take the user to a list of news articles related to that hashtag.

B) Alternatively, the user can type a hashtag of their choosing in a search bar on our website and search to look for relevant news sources that mention that hashtag.

C) Only articles that have been determined trustworthy will be displayed. Trustworthiness is based on various parameters, for example; incorrect grammar usage, incorrect spelling, language crudeness, and random symbols usage. The fewer the errors in the article, the greater the possibility that the article is trustworthy, and the higher the chance that the article will be displayed in the results.

D) The user can then select a search result and read any of the suggested news articles. The article will help the user learn relevant context about the hashtag while removing the sources that may be less reliable.

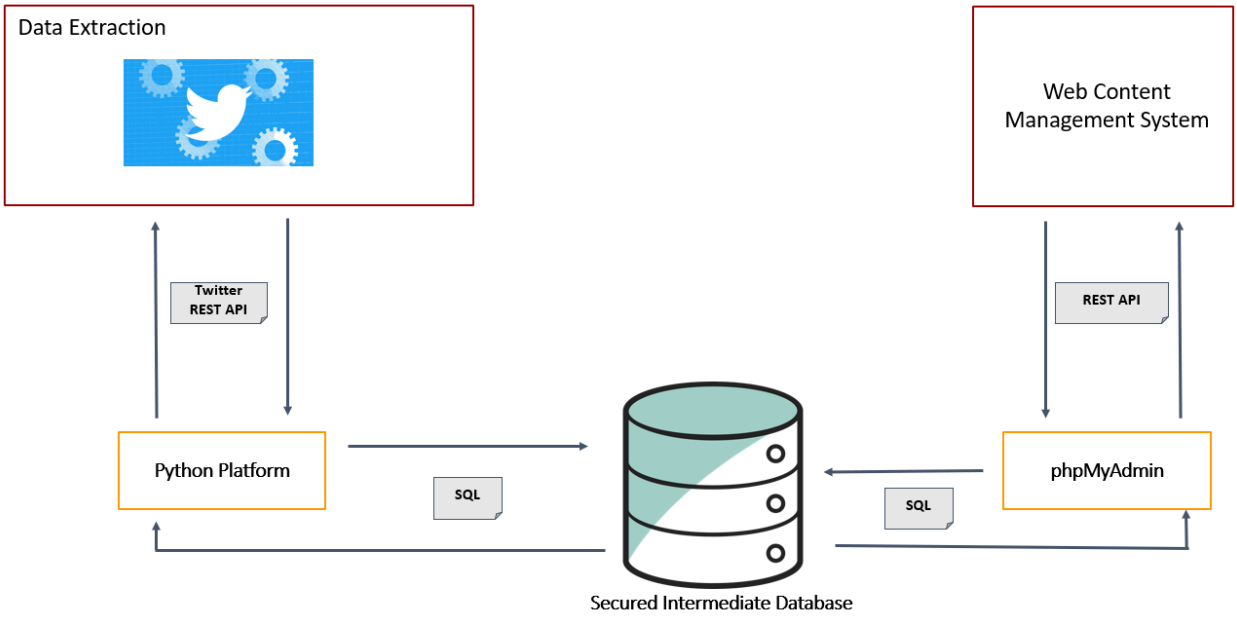
E) Ideally, the website will show the user which sources are worth reading and not worth reading.

An example would be the #MeToo movement. Upon first glance, the hashtag means very little. A quick search in a regular search engine for news articles takes the user to a plethora of articles about current events or trends in the movement, with no context on how useful the site will be for the user to understand the hashtag itself. However, when a user enters #MeToo in the search bar of our website, various news articles determined to be trustworthy will be displayed in the search results. With the filtered results, the user can peruse the articles more efficiently by eliminating the less useful news.

For users who are less intent on understanding a specific hashtag and simply wish to browse, the front page of the website will present them with the current top 30 hashtags. This list will be updated every hour. If they see something that interests them in the list, the user can click on the linked hashtag to view the associated news articles that have been filtered through our program and determined to be more trustworthy.

Section 2: Implementation

The project is mainly based on integration of three modules:



1. Python script which includes the following features:
    a. data extraction
    b. news scrapping
    c. machine learning code for real news detection

2. An SQL Database which stores historical as well as up-to-date data for recent trends and its information

3. A website which is accessible by a user online

1. Python
   System Requirements- Google Colab (Python 3.6.9)
   Libraries used - Tweepy, pygooglenews, pandas, sklearn

a. Data Extraction & ETL:

First step is to extract the data from a popular social media platform. In this case, we are going to start the project from Twitter since it is one of the most well known social media platforms closely connected to the spread of fake news. Twitter's Developer Platform enables us to harness the power of Twitter's open, global, real-time and historical platform within our application. The platform provides us with 3 different products i.e. Twitter API, Twitter Ads API, Twitter for Websites out of which we are going to use Twitter API for information extraction.

The Twitter API is a set of programmatic endpoints that can be used to understand or build the conversation on Twitter. This allows us to find and retrieve, engage with, or create a variety of different resources including the following: tweets, users, spaces, direct messages, lists, trends, media and places.

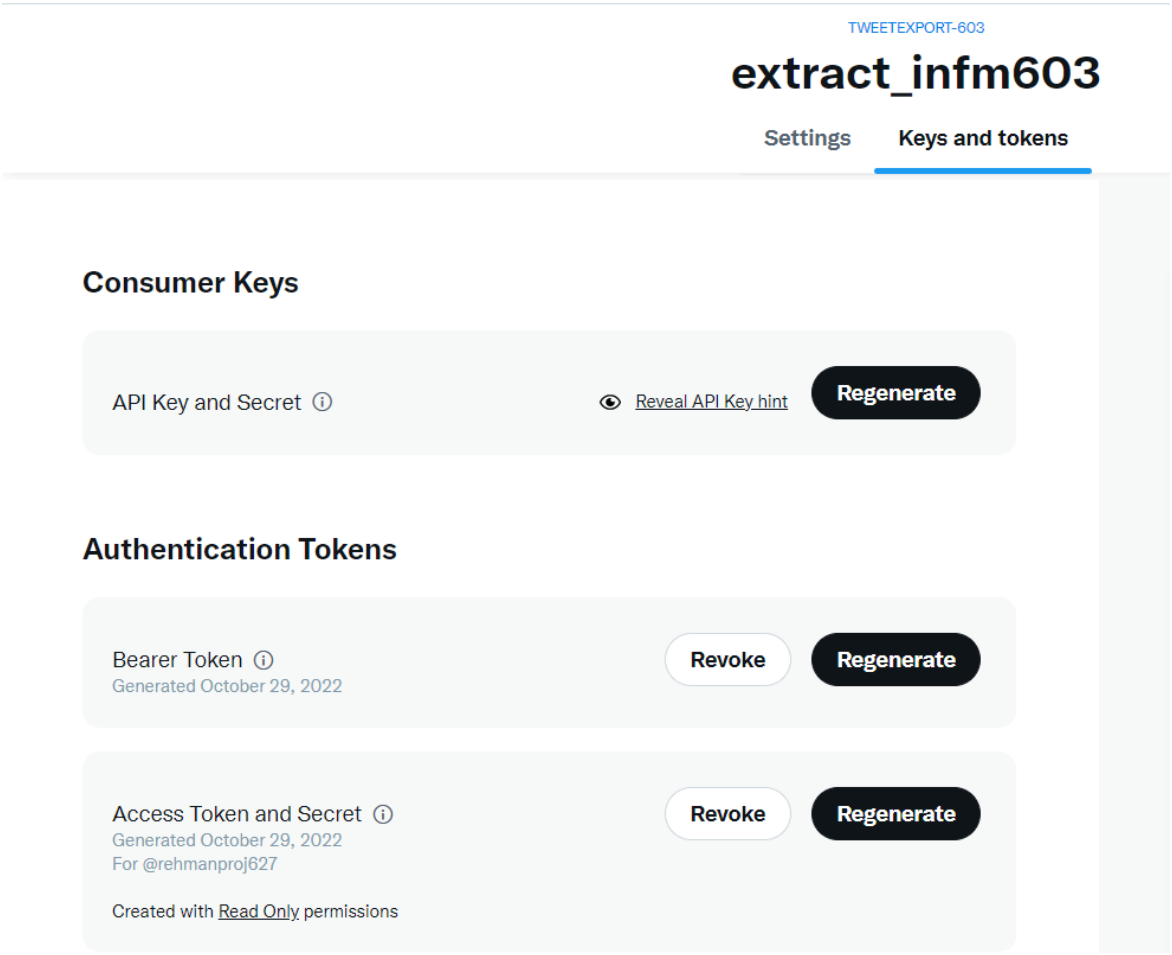
Our primary concern here is extracting the top 30 trends which are present on Twitter and establish their meaning in front of the user. Twitter allows us to mine the data of any user using Twitter API. Over the years, the Twitter API has grown by adding additional levels of access for developers and academic researchers to be able to scale their access to enhance and research the public conversation.

All Twitter API access requires a developers account, which can be created quickly by signing up. Essential access will be available immediately, and Elevated access can be requested.

Essential: Free, instant access to the Twitter API. Includes 500k Tweets/month and a single App environment. 1 App, 1 Project.
Elevated: Free access up to 2M Tweets/month, and 3 App environments. 3 Apps, 1 Project. Requires an approved developer account application.

The endpoints which we plan to use from tweepy library are associated with v1.1 API. So, Elevated access is required to make requests to the v1.1 version of the Twitter API.

When we can securely access our account, the first thing to do is get the consumer key, consumer secret, access key and access secret from twitter developer available easily for each user. These keys will help the API for authentication.



Then, the get_place_trends() method of the API class in the Tweepy module is used to fetch the top 50 trending topics for a specific location. The location is also supposed to be declared inside the code and for this we use the variable WOEID (Where On Earth IDentifier).

WOEID = 2347579 #Code for the State of Maryland

This returns an object of class JSON, which can be easily parsed for relevant details. This is going to be stored in a list and later utilized for scrapping top news articles associated with the entries.

b. News scraping

For this purpose, we are going to use PyGoogleNews, created by the NewsCatcher Team. It acts like a Python wrapper for an unofficial Google News API. It is based on one simple trick: it exploits a lightweight Google News RSS feed. When web-scraping news articles with this library, for every trend we capture using Tweepy, we get a number of data points listed below:

Title - contains the Headline for the article
Link - the original link for the article
Published - the date on which it was published
Summary - the article summary
Source - the website on which it was published
Sub-Articles - list of titles, publishers, and links that are on the same topic

We store this information in a csv file to feed the final machine learning algorithm which performs fake news detection.

c. Real News Detection

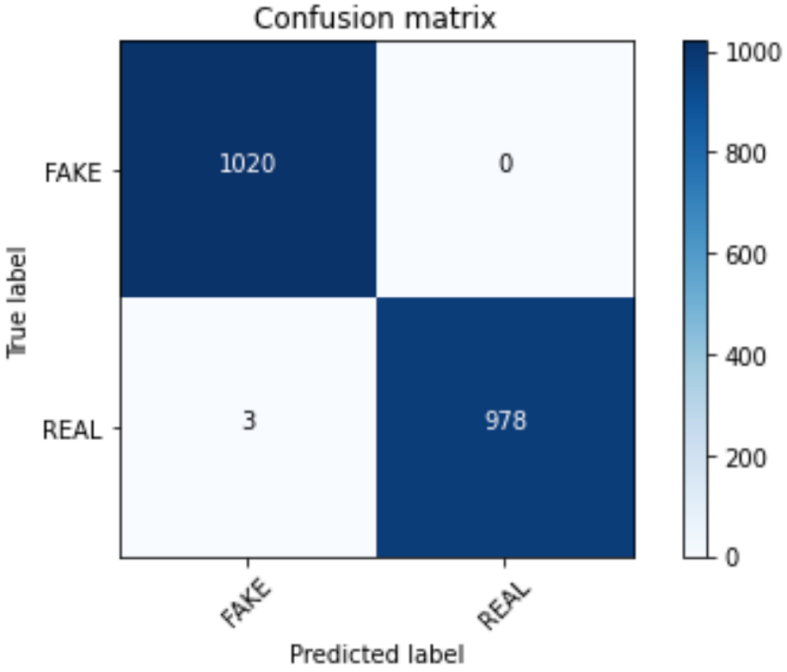
From a machine learning standpoint, fake news detection is a binary classification problem; hence we can use traditional classification methods or state-of-the-art Neural Networks to deal with this problem. The most fundamental part of a machine learning project is data- here we use the information and data we get from the news scraping module of “PyGoogleNews” (in a csv file) and feed it into a pre-trained machine learning model. The entire process will follow below given trajectory:

Step1: Load data from Kaggle to Google Colab.
These are 2 sets of fake news and real news stored in 2 separate csv files.

Step2: Text preprocessing.
The datasets are relatively clean and organized. For the sake of training speed, we are using the first 5000 data points in both datasets to build the model.

Step3: Model training and validation.
Next, we use the TF-IDF vectorizer to convert each token to a vector, aka, vectorize tokens or word embedding. This process calculates how important a word is by considering both the frequency of that word in a document and other documents in the same corpus. After we get the results, we pass it through a classification algorithm; here we use Online Passive-Aggressive Algorithms.

```
Confusion matrix, without normalization
```



We further refine the model by passing it through a confusion matrix to have a look at the False Positives and False Negatives.

Step4: Pickle and load model.
Once we have a prototype, we will pickle (save) the model and vectorizer so as to use them for actual news detection on the csv file which we generated earlier. Since the data extraction is going to happen in real-time, we need a time-saving process and pickle greatly reduces redundant processing.

Step5: Implementation
Finally, we pass our dataset into the machine learning algorithm and get a boolean value for whether it is authentic or not. We store this value along with the news article and use it for historical processing and further visualization on the website.

2. An SQL database:

- We are using MySQL as the database management system as it is relational, fast, reliable, scalable, and open source. We will use SQL to program and design for managing data held in a relational database management system. Localhost is the database server that we are going to utilize. MySQL has storage engines like MyISAM, Blackhole, Archive, Merge, Merge, CSV which will help in storing

and retrieving historical data. The MyISAM storage engine is suitable for non-transactional environments like Data Warehouses, where huge tables are there with minimal write operations. The security features of MySQL such as MySQL Enterprise Firewall, MySQL Enterprise Encryption, MySQL Transparent Data Encryption (TDE), etc. Eventually, MySQL is a good choice for storing all the data we get by extracting it from Twitter.

- We are going to use phpMyAdmin is a free and open source administration tool for MySQL. phpMyAdmin is written in PHP and it has become one of the most popular MySQL administration tools, especially for web hosting services. It brings MySQL to the web. It will help us Import data from CSV and SQL and export data to various formats: CSV, SQL, XML, PDF, etc. All the twitter extracted data stored in MySQL needs to be modified in a particular way and edited, in that case phpMyAdmin helps us to:

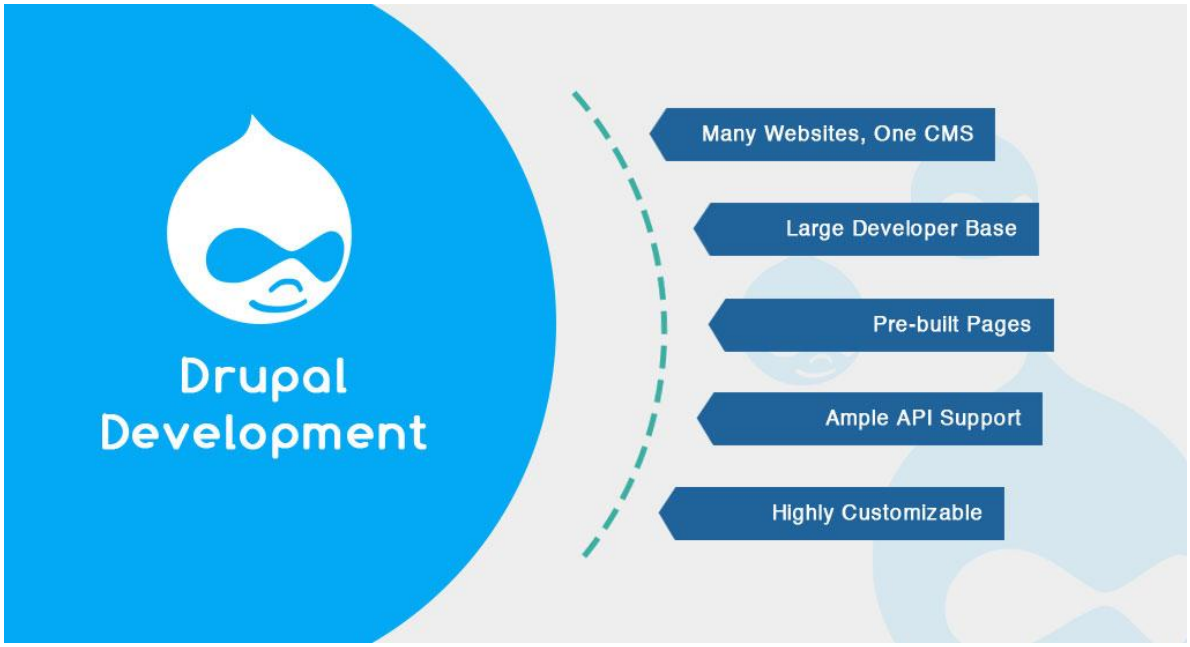
1. Browse and drop databases, tables, views, fields and indexes.
2. Create, copy, drop, rename and alter databases, tables, fields and indexes.
3. Execute, edit and bookmark any SQL-statement, even batch-queries.
4. Manage MySQL user accounts and privileges. phpMyAdmin is a PHP script meant for giving users the ability to interact with their MySQL databases.

3. Creating a website

We are going to use Drupal for making a website.

Operating system: Unix-like, Windows
Browser requirements: Google Chrome, Firefox, Safari, Microsoft Edge, Opera
Type: Content management system
Database server requirements: MySQL
Supported versions for Drupal 9 or 10: MySQL/Percona 5.7.8+
Required configuration: InnoDB as the primary storage engine

- Drupal is an open-source CMS platform that small-to-large organizations can use freely without fear of vendor lock-in, low cost of implementation due to enormous amounts of freely-available community code, and robust and flexible architecture ready for the enterprise. It is highly scalable. Drupal has an advanced permission control system that allows for easier user management and control. Drupal improves website performance through caching, especially for logged-in users, reducing page load times and resource requirements



- We are going to make a user interface in which we get a search bar directly while opening the Drupal site. We can search for the top trends in the search bar. We had extracted the trends and hashtags from twitter. These trends will update on the website on a daily basis to get a fresh variety of news or information. By clicking on the trend which was searched, it will direct us to a new page with the top news articles based on that trend. These news articles would be authenticated articles.
- Drupal itself generally operates with a default MySQL configuration. Create a MySQL database for use with Drupal.
  Creating a MySQL database and user account for DrupalCreate a Drupal MySQL user account.
  1. Create a MySQL database for use with Drupal.
  2. Create a Drupal MySQL user account.
  3. Restore a Drupal database backup from a MySQL dump on one server into the new Drupal MySQL database on this server.

Limitations:

- Unfamiliarity with the technologies associated with this project may prove to be a limitation in execution. Machine learning and data extraction are both fairly new concepts that are used in this project and require more research to learn about.
- In addition, after completing this process there is a possibility that detecting the accuracy of the fake news may not be as accurate as desired.

Section 3: System Testing

System testing will be focused on three different parts; the usability of the website, the ability of the code to match the hashtag with relevant articles, and the effectiveness of the suggested articles.

With the website, the user should be able to navigate between pages smoothly without encountering errors or confusion about the location of important interactions. In addition, the format should be designed simply so the search results are easily accessible.

The program matching articles and determining the trustworthiness of the article will be the most difficult to implement. First, the article must contain the hashtag to be considered a relevant search result. Second, the content must be parsed to determine the validity and reliability of the article.

Finally, the effectiveness of the validation system can be tested by clicking on the article as a user and determining how helpful a kept suggestion is compared to an eliminated one. This process cannot be automated, since it requires a human to look over the content and make a decision on how useful they find the article. In addition, different individuals may find different articles more useful than others.