Clara Pena - u186416
Yuyan Wang – u199907

Project Report Part 1

Project GitHub URL: https://github.com/yuyanwang03/IRWA

Project GitHub Tag: IRWA-2024-part-1

The steps required to execute the Python Jupyter Notebook are detailed in the README.md file. To improve the user experience and reduce the execution time, we are also providing the preprocessed data in a CSV file. This allows users to bypass the preprocessing stage, which otherwise takes approximately 10 minutes, and jump directly to creating the visualizations.

Preprocessing Documents

To preprocess the documents effectively, we developed a helper function called `preprocess_text`, which is responsible for processing the actual content of each tweet. This function takes three arguments: the tweet's text, an optional language code (a two-character ISO 639-1 language code), and an optional language mapping dictionary. The primary reason for including a language code and mapping is to enable language-specific preprocessing, ensuring that each tweet is handled in its original language. It would be inappropriate to apply English stopword removal or tokenization to a tweet written in another language, as it could lead to inaccurate results.

The `preprocess_text` function implements basic NLP operations tailored to the detected language of the tweet. By processing the text in its native language, we aim to provide a more accurate and contextually relevant preprocessing pipeline. However, we acknowledge certain limitations to this approach. Specifically, the language in question must be supported by both NLTK's `SnowballStemmer` and `word_tokenize` functions. This restricts the preprocessing pipeline to only 13 languages, as indicated in the code. While a possible alternative would be to integrate a translation API (such as Google Translate) to convert all content into English, this method poses potential limitations, including API restrictions on the number of queries allowed per minute or per authorization key. Therefore, we opted to work within the languages supported by NLTK's stemmer and tokenizer.

One important preprocessing step we implemented is the removal of mentions, identified by the "@" symbol. These mentions often refer to users or accounts but do not contribute meaningfully to the query or analysis. For instance, a tweet like "@ReallySwara @rohini_sgh watch full video here…" becomes "watch full video here…" after preprocessing, which removes irrelevant terms such as usernames. Most mentions are not likely to be useful for our queries unless they refer to official accounts (e.g., "@CountryGovernment"). In such cases, the ID of the mentioned account could be stored and analyzed later to assess its relevance to the query. However, our focus remains on the text content, so mentions are excluded for the time being.

Another feature of the preprocessing step is the removal of emojis. Emojis might be useful for sentiment analysis, but since this is not the focus of our project, they are discarded. Emojis can also be represented by character sequences, which may not be filtered out in basic text preprocessing. Therefore, using the `emoji` library, we ensure that these are cleaned from the text.

Beyond these specific operations, the program also applies standard text preprocessing techniques. This includes converting all characters to lowercase, removing punctuation, removing stopwords, and applying stemming. Stemming is particularly important for reducing words to their base form, allowing for more consistent analysis across variations of the same word. We use NLTK's

`SnowballStemmer`, which supports a wide range of languages and offers tailored stopword removal and tokenization capabilities for languages other than English. Additionally, we explicitly remove certain irrelevant terms, such as URLs identified by the word “https”, to further clean the text of noise that does not contribute to the analysis.

The `preprocess_text` function is then called within the `process_tweets` function, which processes each tweet in the dataset line by line. In cases where the language detection fails, an “unsure” label is assigned to the tweet, and the content is processed as if it were in English. This ensures that no tweets are left out of the analysis, even if the language is not clearly identified. The processed output includes key metadata such as the tweet’s content, date, hashtags, likes, retweets, URL, and an additional column for the detected language.

In addition to these standard preprocessing steps, we made a deliberate choice to retain hashtags in their original form (including the “#” symbol), rather than breaking them into their component words. Our analysis of the dataset revealed that hashtags often contain valuable context that could be lost if they were split. For instance, the hashtag #FarmersProtest conveys a specific meaning that goes beyond the individual terms “Farmers” and “Protest”. If we were to split the hashtag, the term “Protest” could become associated with other unrelated topics, potentially leading to incorrect query matches. An example of this is the hashtag #WorldSupportIndianFarmers, which, when broken into separate words, might incorrectly signal relevance to a query about the “Indian Government”, when in fact the tweet may not pertain to government-related matters.

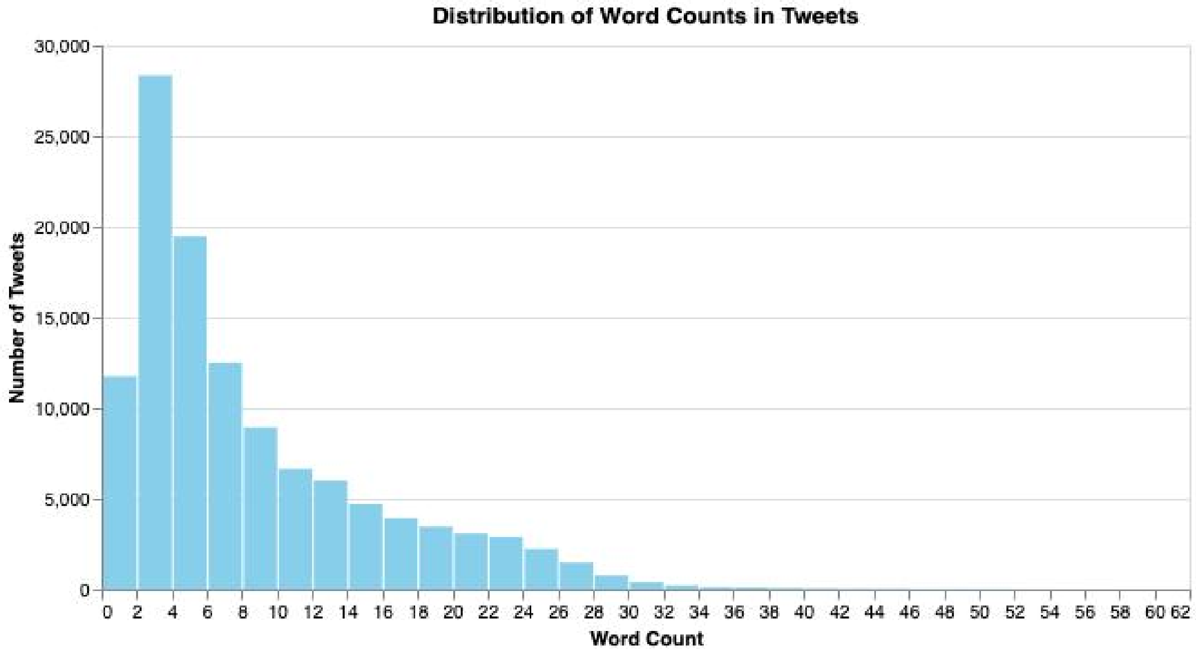
Further evidence supporting our decision to retain hashtags can be seen in our analysis of relevant and non-relevant hashtags. In one DataFrame, we observed that the hashtag #ModiIgnoringFarmersDeaths frequently co-occurs with #HumanRights in relevant tweets. Although #HumanRights also appears in non-relevant contexts, its co-occurrence with #ModiIgnoringFarmersDeaths indicates that it is more likely to be relevant when both hashtags are present together. This contextual relationship would be lost if the hashtags were split into individual words. By retaining them intact, we preserve these associations and enhance the accuracy of our information retrieval system.

In summary, our preprocessing pipeline was designed to balance flexibility and accuracy, considering both the language of the tweet and the specific structure of social media content such as mentions and hashtags. While the decision to retain hashtags and handle tweets in their native languages adds complexity to the preprocessing, it ultimately enhances the relevance and quality of the data analysis. The use of language-specific tokenization, stopword removal, and stemming ensures that each tweet is processed in a manner that respects its linguistic context, allowing for more precise analysis and information retrieval.

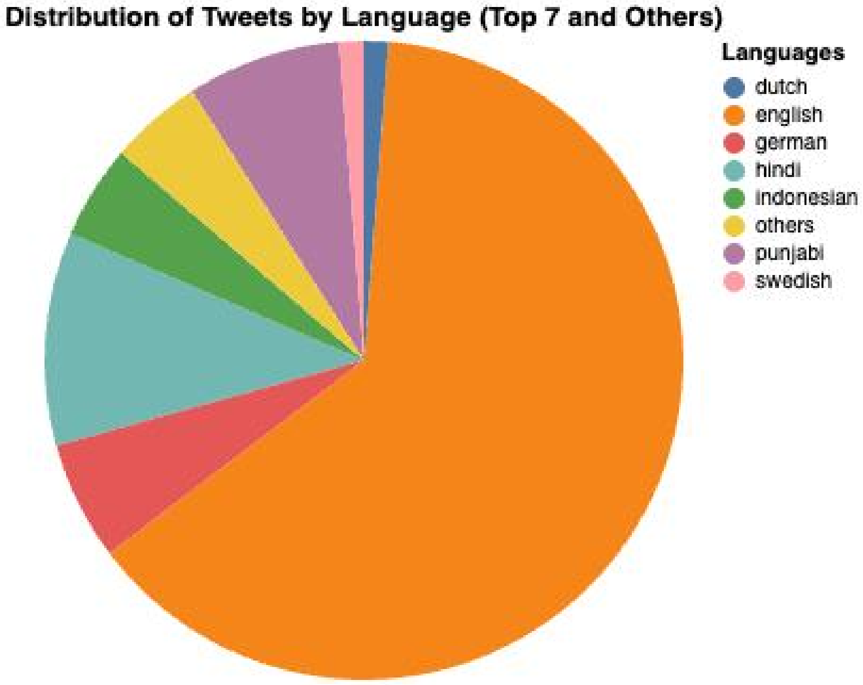
Explanatory Data Analysis

In addition to processing the tweets, we performed an exploratory analysis to gain a better understanding of the content and the structure of the tweets’ dataset we have used in the project. The aim of this analysis is to mainly provide some statistics, identify patterns and provide some insights into the data. We have examined several aspects of the tweets such as tweet characteristics, language usage and word distributions. We would recomend you see these visualizations from the Jupyter Notebook because we’re providing tooltips that could display content and exact numbers when hovering over the component in the plot.

In the code, we first started by analyzing the word count of each tweet and plotting the histogram for the distribution across the tweets in the dataset. The word count was computed for each tweet by splitting the content of the tweet into individual words.



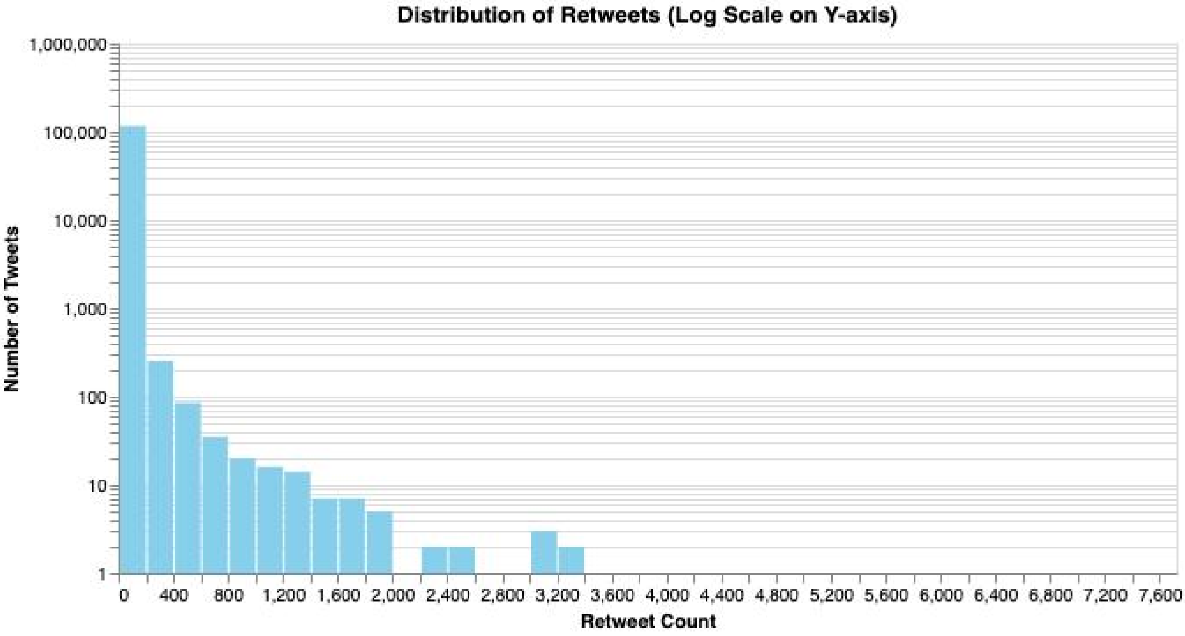
As we can observe in the histogram, most of the tweets have a relatively low word count (keep in mind that those that are not accounting for stop words), between 2 to 8 words that suggests that users keep their tweets short and concise, maybe using links, hashtags or images instead of more text. This fact completely aligns with the short messages typically seen on Twitter due to its character limit. Also, as the word count increases, we can see that there are less number of tweets, indicating that users rarely write longer tweets (more than 10 words, approximately).



We have processed the tweets to label them by its language and have more information about them. This has allowed us to get the seven most used languages in the dataset and plot a pie chart showing how many tweets there are of each language. The pie chart displays the proportion of tweets of the top 7 most frequently used languages in the dataset together with a label called ‘others’, which includes all the remaining languages.

We have expected that English will be the dominant language in the dataset, which can suggest that most of the users are from English speaking regions or they just prefer to post tweets in English. Other significant languages are Hindi and Punjabi and most likely they represent that there are specific regions where there is a high Twitter activity. The 'others' label are all the languages that that make up a small percentage of the dataset and, even though there may be a wide range of languages, none of them are as relevant as the top seven.

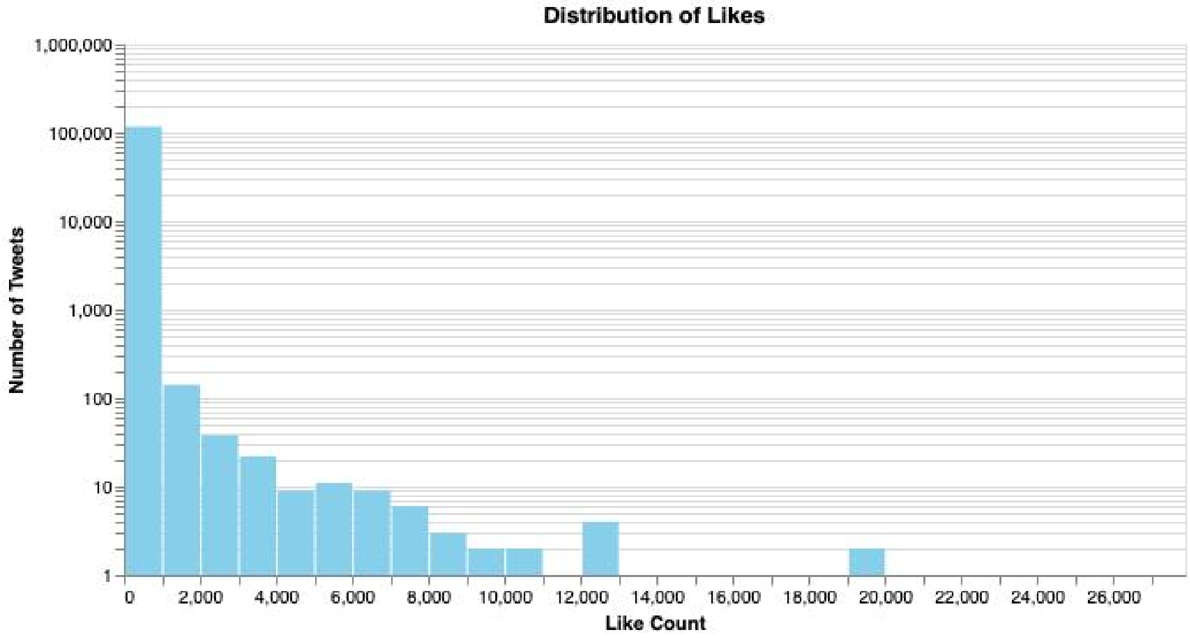
For the following two analysis, we have used the log-scale because it helped to visualize the distribution better as there is a large variation between the data. In the first one, we counted the number of retweets to check the overall engagement with the tweets in our dataset. Applying the logarithmic scale allowed us to see more clearly the wide range of retweet activity from tweets that have little to no interaction to those that went viral, which helps to reveal the typical trends and outliers of the dataset.



It can be observed that the distribution is skewed to the right indicating that there are some tweets with a very large number of retweets, but most of them have very few retweets. Therefore, we can suggest that retweets are a highly skewed phenomenon because only a small number of tweets receive a large amount of attention. This is logical because in social media, it is a common pattern and, it can be due to several reasons, like the quality of the content or the popularity of the user.

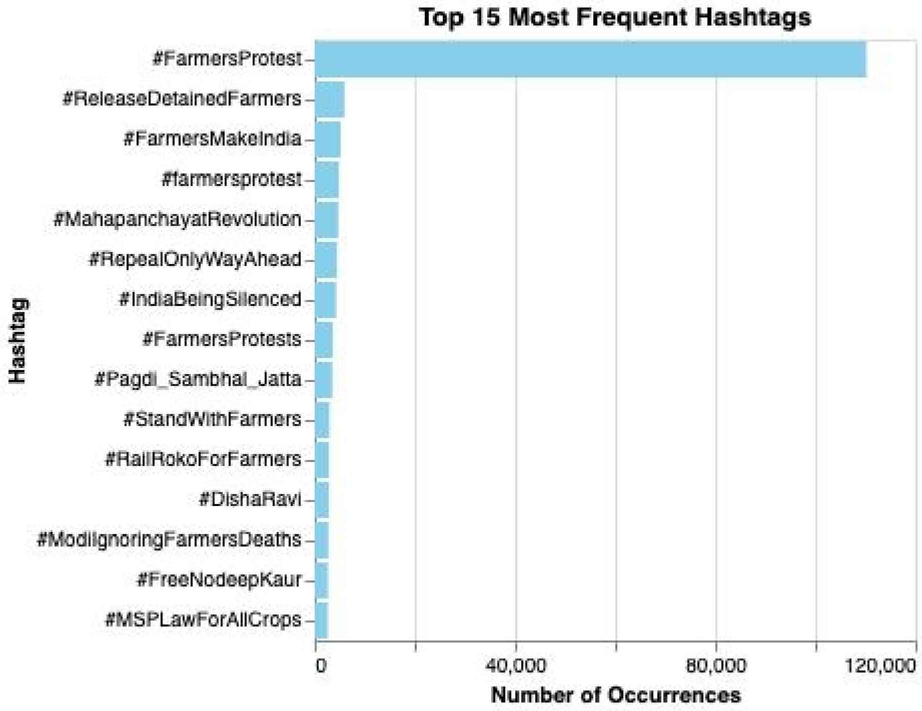
In the second analysis, we have counted the number of like each tweet received. Similar to the previous histogram, we used a logarithmic scale to help observe the distribution more clearly. This approach makes it a lot easier to observe the patterns in the data and draw more meaningful conclusions given the wide range of like counts across the tweets.

The histogram is also skewed to the right, which means that there are a few tweets with a very large number of likes while most of them have fewer likes. We have a peak of tweets at between 19.000 and 20.000 likes but it is acting as an outlier as I mentioned before, the majority of the tweets are concentrated in the lower count range. As well as before, the histogram suggests that this a skewed phenomenon with some tweets receiving a disproportionate amount of likes, and it is also a common pattern in social media.

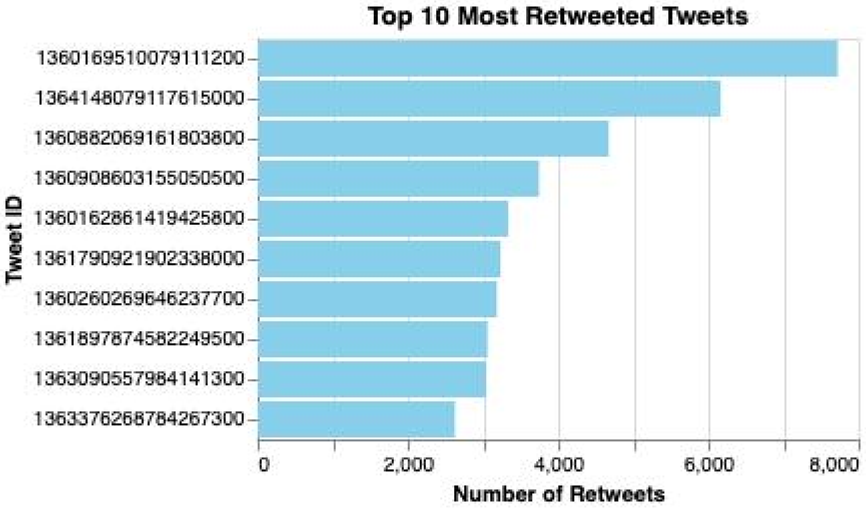


The following bar chart shows the top fifteen most frequently used hashtags in the tweets of the dataset.

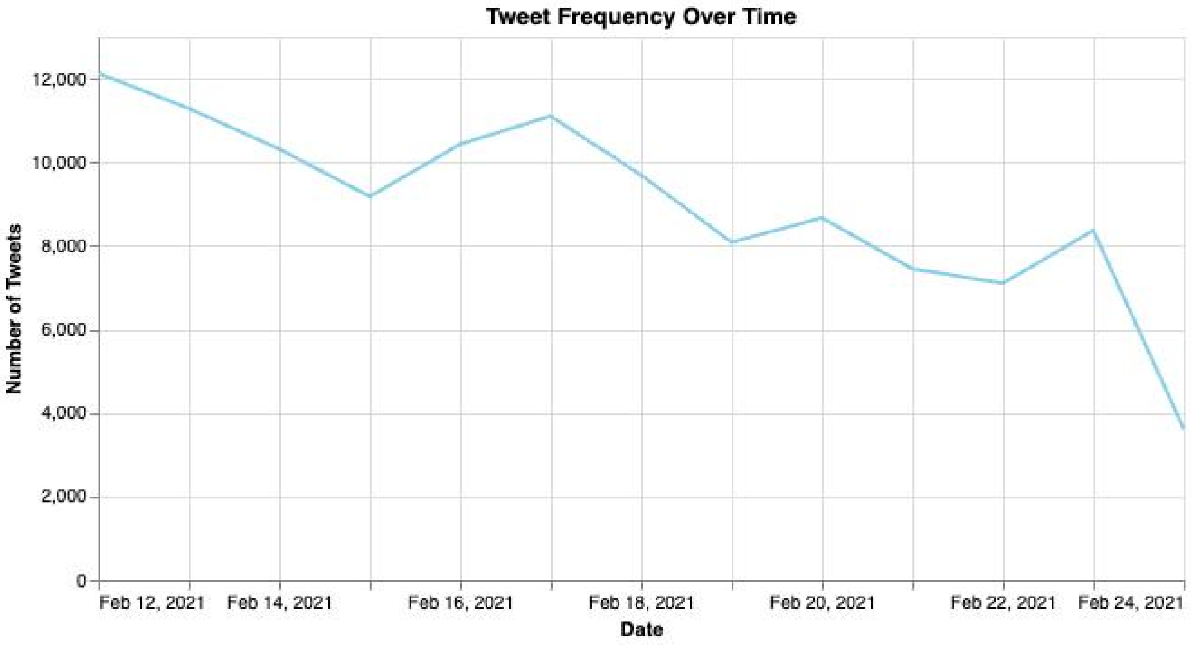
The hashtag that appears on most tweets, by a great difference to the other hashtags, is #FarmersProtest. Several other hashtags that are retaled to the Farmers' Protest hashtag are also between the most frequent ones, such as #ReleaseDetainedFarmers or #FarmersMakeIndia. The remaining hashtags cover other topics like calls for action, criticism of the government, and support for the farmers.



We have also made a bar chart to get the top ten most retweeted tweets. We can observe that the number of retweets in the ranking significantly varies, as its range goes from 8.000 retweets for the most retweeted tweet, and over 3.000 retweets for the least of the top ten. In this way we can identify the most engaging content. Moreover, we identify the tweets by its ID and if the mouse goes on top of the bar chart (inside the notebook), we can see additional information such as the content of the tweet and the language it is written in.



We have found that the frequency of tweets over time has been decreasing over time, as the line chart below shows. There was an initial peak around February 12, 2021 and since then it has gradually decreased. Also, we can see that the tweet frequency fluctuated slightly as time wen by, with some days having very subtle increases or decreases. As the chart ends on February 24, 2021, it is a bit difficult to get definite conclusions about the long-term trend.

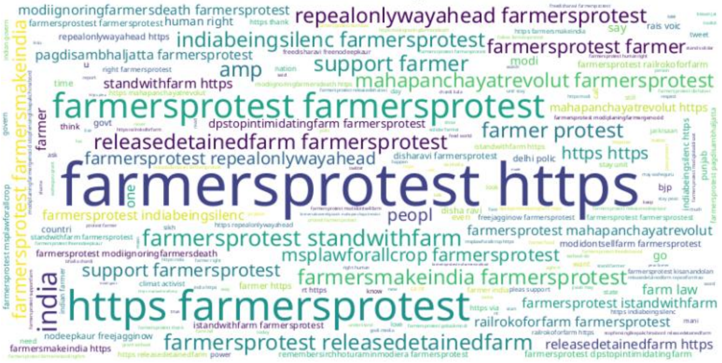


Finally, to finish with the analysis, we have generated wordclouds containing the most repeated words over each of the to seven most used languages. These wordclouds share some common patterns and characteristics that give an insight into the overall content of the tweets dataset we where given.

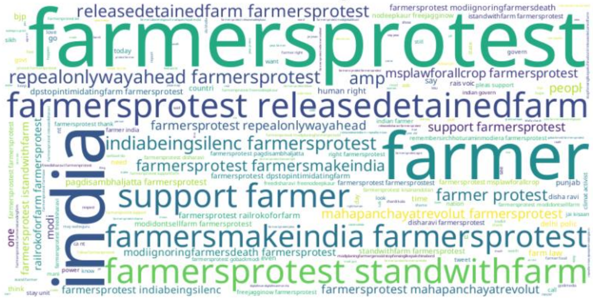
If the wordclouds below are observed, we can see that across all languages, the word ‘farmersprotest’ is the most common term, which indicates that the main focus of the tweets are about the Farmer’s Protest and that it is a global concern as there are discussions in different languages of the same topic.

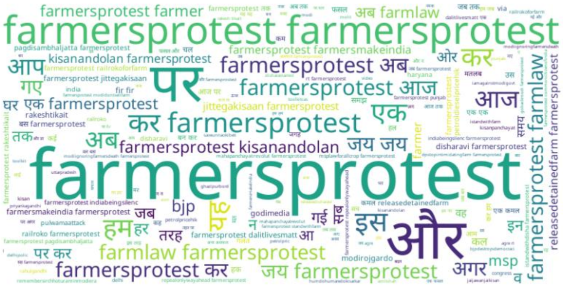
Clara Pena - u186416
Yuyan Wang – u199907

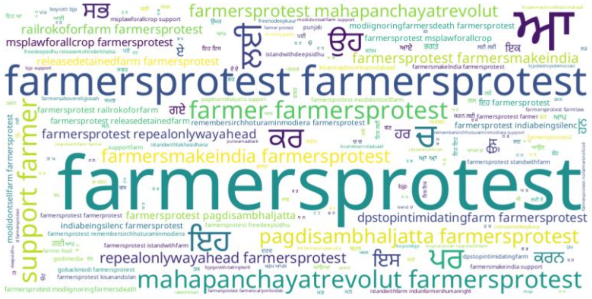
One common word we can observe through the wordclouds was ‘https’, which means that the tweets contain links to articles, videos or external resources related to the protest. This shows that the users are sharing relevant information to spread awareness. However, since this word is not really providing meaningful information regarding the content, we removed it during our preprocessing phase. Following is a wordcloud example before we did the removal of the https word.

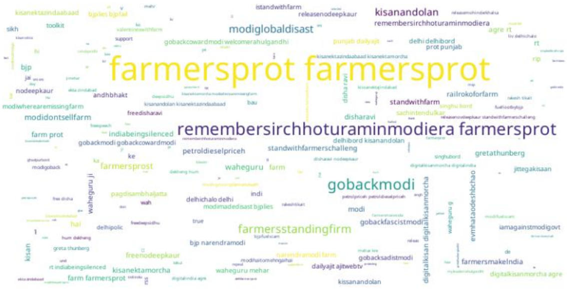

Word Cloud for English

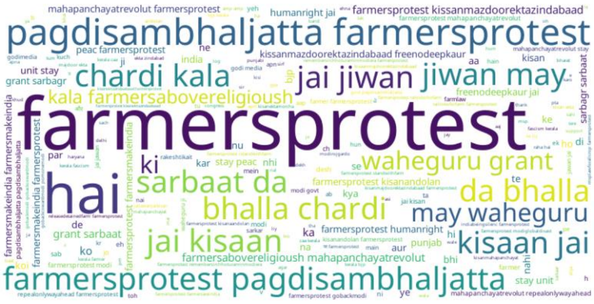
Even though the tweets are in different languages the content of them seems to be revolving around the same core topics, meaning that the protest is being discussed internationally with translations or multilingual content that help to propagate the same message.

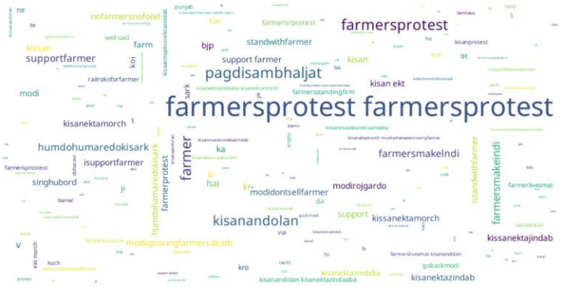

Word Cloud for English


Word Cloud for Hindi


Word Cloud for Punjabi

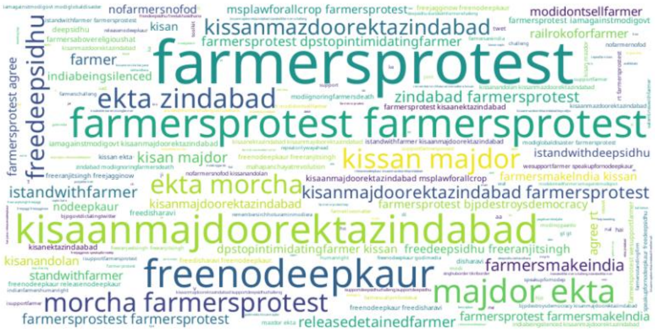

Word Cloud for German


Word Cloud for Indonesian


Word Cloud for Swedish

Clara Pena - u186416
Yuyan Wang – u199907

Word Cloud for Dutch


farmersprotest
kisaanmajdoorektazindabad
farmersprotest farmersprotest
freenodeepkaur
majdor ekta
kissan majdor
ekta zindabad
ekta morcha
morcha farmersprotest
kissanmazdoorektazindabad
freedeepsidhu
kisanmajdoorektazindabad farmersprotest
istandwithfarmer
zindabad farmersprotest
farmersprotest dpstopintimidatingfarmer
msplawforallcrop farmersprotest
nofarmersnofod
modidontsellfarmer
railrokoforfarmer
farmersmakeindia
farmersprotest farmersmakeindia
releasedetainedfarmer
farmersprostest farmersprotest
standwithfarmer
kisanandolan
nodeepkaur
indiabeingsilenced
farmer
kisan majdor
istandwithdeepsidhu
dpstopintimidatingfarmer kissan
freedeepsidhu freeranjitsingh
farmersprotest bjpdestroysdemocracy
farmersmakeindia kissan
kisanmajdoorektazindabad
farmersprotest agree
agree