Hajj And Umrah Tweets Data Analysis

Special Topics I

Group Members:

Khalil Alsulaimani (Group Leader) **ID:439007816**

Nawaf Noor Aldeen **ID:439003406**

Yasser Alharbi **ID:439003070**

Supervisor: Dr. Mohammed Algamidi

Table Of Content

Chapter 1: Introduction

Introduction:

Social media has taken over the world in every aspect from our daily lives to business, therefore, the world has evolved and changed from the past where governments and components depended on surveys to know the public opinion of a certain subject to now depending on social media.

The advantage of this evolution is a larger sample population which will give a better more detailed public opinion, moreover, with the advancements in Artificial Intelligence (AI) help make the process automated with grater statistics and numbers.

However, with every evolution there are some new challenges, these vairy from social media platform, however, some common problems occur and one of the main ones is filtering out useless meaningless social media responses that do not benefit the study.

These advantages and disadvantages are what made the new field of data analysis such an important and integral part of society as it handles all the data and gives the results researchers need and are looking for.

As for this project, we will be using data analysis on hajj and umrah to learn the techniques and process of data analysis to find the public opinion about hajj and umrah our goal is to take an initial set of one hundred thousand tweets to analysis.

Purpose Of This Document:

This document details the entire process of analyzing tweets about hajj and umrah from the first stages to the product.

Data Analysis:

Is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis [1].

Tools:

Data analysis uses many fields to achieve the desired results, therefore, it depends on a lot of different tools and frameworks, we will be using the following tools in our project:

Python:

Python is a general-purpose language, which means it is designed to be used in a range of applications, including data science, software and web development, automation, and generally getting stuff done [2].

Which makes it the perfect tool for this project as python has a lot of sophisticated libraries for AI and Data analysis which will be integral to our project and save us a lot of time with better results.

Libraries:

SnScrape:

SnScrape is a scraper for social networking services (SNS). It scrapes things like user profiles, hashtags, or searches and returns the discovered items, e.g., the relevant posts [3], which makes it a powerful library which we will be using to gather data.

Pandas:

Pandas is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language[4].

XlsxWriter:

XlsxWriter can be used to write text, numbers, formulas, and hyperlinks to multiple worksheets, and it supports features such as formatting and many more[5].

PyCharm And VSCode:

PyCharm and VSCode are both IDEs which are Integrated Development Environments which are the places where you can write Python code and test.

TweetBinder:

Tweet Binder is an excellent and cost-effective tool for measuring campaign hashtag performance. It is especially handy when it comes to historical reports. Tweet Binder is the ideal tool for tracking, facilitating, and amplifying the online conversation [7].

Summary:

To conclude, we discussed the change of the world in the sense of general opinion metrics and how they have changed in recent years, how the new advantages have their disadvantages and disadvantages, lastly, we listed all the tools we will be using in the project even.

Chapter 2: Data Gathering

Data Gathering:

To analysis data firstly you need the data this is where data gathering comes in, there are many different approaches to gathering data from twitter we have tried two methods both worked however had varying results.

Method 1: Twitter API:

Firstly, we made a twitter developers account to be able to access the twitter API, afterwards twitter asked us some questions such as what we are going to use the API for, are we students or a private company etc., after completing the sign up and questions we waited for twitters approval for us to use the API which came days later, however, we would come to find out that there is a limit of three thousand tweets we can gather, which would make reaching our goal of one hundred thousand tweets impossible, therefore, we abandoned this method and started looking for alternatives.

Method 2: SnScrape Library:

The second method we used which gave us positive results and met the requirements was using the SnScrape library, as it gave us more control over the number of tweets we can search for, as well as the ability to choose what information from the tweet we want to save.

Code:

```
import snscrape.modules.twitter as sntwitter
import pandas as pd
tweets = []
limit = 500000


query = '(مكة OR العمره OR الحج OR العمرة OR مكه OR منى OR مزدلفة OR مزدلفه OR المشاعر المقدسة OR عرفة OR عرفه OR ,

for tweet in sntwitter.TwitterSearchScraper(query).get_items():

    if len(tweets) == limit:
        break
    else:
        tweets.append([(tweet.date).strftime("%m/%d/%Y, %H:%M:%S"),tweet.user.username,
        tweet.user.verified,tweet.content,
        tweet.lang,tweet.replyCount,
        tweet.retweetCount,tweet.likeCount,tweet.quoteCount])

df = pd.DataFrame(tweets,columns=['Date','User','isVerified','Tweet','Language','ReplyCounter','RetweetCount
writer = pd.ExcelWriter('Tweets.xlsx',engine='xlsxwriter')
df.to_excel(writer,sheet_name='Sheet1')
writer.save()
print(df)
```

The code set up starts of by importing “SnScrape” twitter module, then importing the panda and remaining it as pd for ease of use in the code, lastly, we set an empty array called tweets which we will use to save the tweets information in and set a maximum value of tweets to read to make sure the code doses not run for infinity.

Our method of gathering the data depend on using keywords which ae mentioned in tweets, instead, of hashtags as it will guarantee more data this was done with the following code:

```
query = '(مكة OR العمره OR الحج OR العمرة OR مكه OR منى OR مزدلفة OR مزدلفه
OR الكعبة OR الحرام المسجد OR الحرم OR عرفه OR عرفة OR المقدسة المشاعر
OR الاضحى OR الجمرات OR المروه OR المروة OR الصفا OR الكعبه OR المقدسه المشاعر
نمرة OR نمره)'
```

The query included the major keywords that are related to the hajj and umrah topic, furthermore, we used the advanced search twitter feature to achieve better results, now everything is ready we used the query and collected the tweets data by looping over them in a for each loop and before saving any data we check if the array has met the limit we set if it has we end the code and start the data transfer to an excel file.

If the limit has not been reached, we add a tweet to the array and add the following information from the tweet:

1. Date of tweet
2. Username
3. Verification status true if verified false if not
4. Tweet language
5. Replay count
6. Retweet count
7. Like count
8. Quote count

The majority of the information is straightforward to gather, however, the date of tweets needed further processing to enable python to manage the date format which we did by using “strftime” command to parse the date to the accepted format, this concludes the scraping process.

Afterwards, we need to transfer the tweets to excel for further processing later, firstly, the date format needed further processing to the accepted excel date format which was achieved by the “dateframe” command from the “panda” library, moreover, the “dateframe” was critical to the process by allowing us to transform the data from a 1-Diminsoal array which is not useful to a 2-Dimsoinal array which is what we need.

Lastly, the tweets where ready to be transferred to the excel file and this was achieved by the “pandas” command “ExcelIWriter” which we received the tweet and using the “XlsxWriter” library we set the engine to be used in the transfer, to close the code we used the “writer. Save” command to save the work and end the code, our result was 202 thousand tweets.

Summary:

To summarize, data gathering is the first step in data analysis which is what we achieved by the end of this chapter, furthermore, we explained the code we used in detailed and provided a code snippet which delivers the requirements for the project.

Chapter 3: Data Filtering

Introduction:

When gathering a large amount of data, it is critical to filter the data especially when its source is a social media site, as you cannot guarantee all the data is connected to the keywords or hashtags you used as using the unfiltered data will give an inaccurate result.

Attempt 1 Process:

The filtering process can be split into three categories:

1- Duplicated tweets:

To remove duplicated tweets, we used the built in excel feature of finding duplicate tweets in the tweet column and deleting them.

2- Removing entire tweet based on some criteria:

a- Length:

We made new column called length and used the excel function =LEN(TRIM('tweet'))-LEN (SUBSTITUTE ('tweet'," ","""))+1' which calculated the length of the tweets cell value then we removed any tweets with length that is less then 3 as those tweets do not contain any valuable information.

b- Keywords:

We removed any tweets containing the following keywords as we know they are irrelevant to the study as a lot of accounts use the popular hashtags or keyword to post Ads:

1- #AD
2- وظائف
3- إعلان
4- رواتب
5- راتب
6- وظيفة

We removed these tweets by using the built in excel feature of filtering based on keywords in the tweets column which returned the rows which contained the keyboard which we then deleted.

c- Language:

As we retrieved the tweets language in the data gathering phase, we can use that column to remove any tweets that are not I Arabic as we used Arabic keywords and hashtags, therefore, any tweets that tweeter recognized as another language where either gibberish tweets, or tweets that only contained hashtags.

3- Replacing tweets content with empty space but keeping the rest of the tweet:

As tweets sometimes contain emojis or some other symbols that have no grater meaning to the study its best to remove them from the tweets which we did by using the built in excel feature of find and replace which searches the file for the inputted character and replaced it with the inputted character which in our case was an empty space.

We also removed some punctuation such as:

1- Full stops (.)
2- Comma (,)
3- Plus sign (+)
4- Doller sign ($)
5- Percent sign (%)
6- Explanation marks (!)
7- Square brackets ([])
8- Quotation marks (‘’)
9- Special character (|)
10- Special character (^)
11- Special character (&)
12- Special character (`)

The majority pf these methods were found using the internet such as the referenced website [7].

Result:

After the data gathering phase, we accumulated 202 thousand tweets, now after filtering we got that down to 154750 tweets that are set to be used with a model to understand the setting of the tweet, this will help us obtain better results in later stages [8].

Attempt 2 Process:

After the first attempt, and the group discussions done at the lecture we learned some new valuable information on how to improve on our filtering process to gain better results, therefore, we refilter the data from the first stage, however, this time we filtered the data using python.

Set up:

Firstly, we saved the data from the tweet's column inside a dataFrame from the "pandas" library which will processing the tweets easier shown by the following code snapshot:

```
# import pandas lib as pd
import pandas as pd
array=[]
# setting the 3rd row as header.
df = pd.read_excel('Tweets-filterd2.xlsx', sheet_name = 0)
```

Filtering:

Using Regular expressions known as RegEx (a string of text that lets you create patterns that help match, locate, and manage text [9]) which we will be using to find certain patterns in the tweets then either removing that section of text or removing the whole text that contains that pattern.

Tweets Removal:

We removed any rows that there tweets contained the same keywords we used in the first attempt using the following RegEx:

```
df=df [~(df.Tweet.str.contains(u'AD|وظائف|إعلان|رواتب|راتب|وظيفة', na=False))]
#Deleting rows that have these words
```

Filtering The Tweets:

Using RegEx to find matching patterns then when the match is found we replace it with an empty space, therefore, removing any characters we do not want shown by the code snippet below:

```
# This Code For Filter Data Set
df['Tweet']=df['Tweet'].str.replace(r'@[a-zA-Z_0-9]{0,}','')#Deleting Mention
df['Tweet']=df['Tweet'].str.replace(r'#[^\s]{0,}','')   #Deleting Hashtag
df['Tweet']=df['Tweet'].str.replace(r'http\S+','')      #Deleting URL Links
df['Tweet']=df['Tweet'].str.replace(r'[\.,\+\$\%\!\[\]\'\"\|\^\&\`\:]','')
#Deleting Special Charecter
df['Tweet']=df['Tweet'].str.replace(r'([^\u0621-\u064A0-9\s]+)','') #Deleting
any char not arabic or number
```

This code resulted in pristinely clean code, as we removed hashtags, URLs, mentions, special characters and emojis all together giving us tweets ready for the later stages of the data analysis, as well as, helping us remove any duplicate tweets that only different based on the said removed content.

Result:

In the end we went down from 202 thousand tweets to 133 thousand which is a higher percentage, of deleted and filtered tweets than attempt 1 where we only used the excel tools, lastly, we will be using the data filtered from attempt 2 as it is drastically better and cleaner.

Summery:

In this chapter we discussed the importance of data filtering, described what criteria and methods we used to filter our tweets data, moreover, we showed our two attempts at filtering the tweets which gave us drastically different results, which shows how important this stage.

Chapter 4: Tweet Stemming

Introduction:

Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. [10].

In the arabic language we get the root of the word which carries meaning and cannot be reduced to anything smaller which will help in the later stages of the data analysis.

Attempt 1 Tashaphyne Library:

Set Up:

Using the 'Tashaphyne' library which contains all the necessary functions and methods to get the root of an arabic word.

Firstly, we imported the library and saved its constructor in a variable to ease its use in the code, then using the same technique as before we saved the excel in a dataFrame and accessed the tweets column and made sure we only received the not null values.

Execution:

The code can be split into two phases as we used nested loops, the first loop we took the tweets and split it by the space character so we can take each word by itself and saved it inside a variable called 'x'.

Inside the second loop is where the rooting of the word is executed using the 'Tashaphyne' library we firstly check if the word is longer then 3 letters as rooting words less then that is a waste of resources as they are already in the rooted form, therefore, we add the word in the sentence as it is an continue to the next word , we retrieve the root by calling the get_root() method on every word in the sentence, lastly, we append the result in an array which will contain the new rooted sentence then added to a new column called "RootTweet" in the excel file.

```
# This Code For Extracting Root For Each Tweet
#make propre display for unicode
import pyarabic.arabrepr
arepr = pyarabic.arabrepr.ArabicRepr()
repr = arepr.repr

from tashaphyne.stemming import ArabicLightStemmer
ArListem = ArabicLightStemmer()

df = df[df['Tweet'].notnull()]
for i in df['Tweet']:
    x=[]
    x=i.split(sep=' ')
    sentence=''
    for j in x:
        if(len(j)>3):
            ArListem.light_stem(j)
            sentence+=ArListem.get_root()
            if(x[len(x)-1]!=j): sentence+=" "
        else:
            sentence+=j
            sentence+=" "
    array.append(sentence)

df["RootTweet"]=array
df.to_excel('Tweets-filterd2.xlsx')
```

Result:

The resulting tweets were very mixed some tweets were rooted well , however, a large portion of the tweets were returned in a bad form of rooting, as well as, facing problems with words that where less than 3 letters, therefore, we added an if statement in the code to make sure we didn't take in any words less than that length.

Attempt 2 Farasa API:

In this attempt we signed up to the Farasa website and received an API key along side some code from the website which we used to send the tweets via the API to get stemmed and wait for the APIs response of the stemmed tweet, however, after sending three requests we were met with a 'you reached maximum requests limit' message , therefore, we abandoned this API and looked for another solution.

Attempt 3 Farasa Library:

After the bad results of the first attempt, we did more research to find any better tools and found the Farasa tool which uses machine learning to stem the word.

Code:

This attempts code is very similar to the first attempt, however, the end result was better, but, it was not a smooth sailing attempt, as for the first attempt with this library we didn't set the Faras Stemmer to interactive mode which made the code work for 19 hours straight without giving us the final results, therefore, we looked into it even more and found the interactive flag which sped up the processes exponentially and gave us the optimal results in a matter of minutes, which we saved in a new column in a new version of the excel sheet to save all the work even if we didn't get the results we wanted.

```
#make propre display for unicode
import pyarabic.arabrepr
arepr = pyarabic.arabrepr.ArabicRepr()
repr = arepr.repr
from farasa.stemmer import FarasaStemmer

stemmer = FarasaStemmer(interactive=True)

from tashaphyne.stemming import ArabicLightStemmer
ArListem = ArabicLightStemmer()

df = df[df['Tweet'].notnull()]

for i in df['Tweet']:

    array2.append(stemmer.stem(i.strip())

    x=[]
    x=i.split(sep=' ')
    sentence=''
    for j in x:
        if(len(j)>3):
            ArListem.light_stem(j)
            sentence+=ArListem.get_root()
            if(x[len(x)-1]!=j): sentence+=" "
        else:
            sentence+=j
            sentence+=" "
    array.append(sentence)

df["RootTweet"]=array
df["SteemTweet"]=array2
df.to_excel('Tweets-filterd2+Rooted.xlsx')
```

Results:

The results from this library where immaculate and very clean which is what we were, aiming for and thanks to the interactive mode which sped up the stemming process for us.

Comparison between the attempts:

The difference between the attempts is substantial as a Faras Stemmer gave us the most clean and well stemmed results we can hope for, however, the Tashaphyne Library was lack luster to say the least as it caused us a lot of problems and resulted in weak and incorrect results.

Summery:

In conclusion, of this chapter we discussed what stimming is and provided the attempts we did to achieve the stimmed versions of our tweets, as well as, comparing between the attempts lastly, we saved the new results in new files for both attempts in an effort to save all of our work incase it is ever needed down the line.

Chapter 5: Data Labeling

Introduction:

Data labeling is the process of identifying raw data (images, text files, videos, etc.) and adding one or more meaningful and informative labels to provide context so that a machine learning model can learn from it [11]. This means that to achieve better results we categorize the tweets we gathered, this process is usually done manually which makes it the hardest and yet the most important part of the preprocessing process as the labels highly effect the end product.

Process:

Firstly, as we started labeling, we saw that there were still some left-over duplicated tweets, therefore, we did a bit more data cleansing and removed those duplicated tweets which bought our total tweet count down to 112681 tweets.

For our labeling process we will have two labels a categories label and the tone of the tweet, and each label had the following subcategories:

Category labels:

We used numbers to represent each category as that will help us avoid any spelling errors, and the labels where:

1 = Place Of Stay

2 = Security

3 = Food and Drink

4 = Practicing Religion

5 = Health Care

6 = Transport

7 = Hajj Groups

8 = Catering

9 = Weather

10 = Hospitality

11 = Services

12 = Unrelated To Hajj And Umrah

Tweet tone:

We used numbers in this label as well as it will make handling the code much easier later, the labels are:

0 = negative tone

1 = positive tone

2 = neutral tone

Results Of Attempt 1:

Due to the manual nature of data labeling its very time consuming and tedious, therefore, labeling all the data inside a week was near impossible, we did our best efforts and labeled a total of 2000 tweets.

Results Of Attempt 2:

After we saw how slow attempt 1 was we decided to use a way that is faster , as we saw from the first 2000 tweets they were 99% positive, therefore, we looked up keywords for each of the 11 categories we have and assigned a positive tone to them plus the category related to that key word, afterwards, we double checked and only needed to change the tone if the tweet was natural or negative which saved a lot of time giving us the ability to label 63 thousand tweets, lastly as we used numbering in attempt 1 we filtered and changed those numbers to the corresponding titles.

References:

[1] https://www.guru99.com/what-is-data-analysis.html

[2] https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

[3] https://github.com/JustAnotherArchivist/snscrape

[4] https://pandas.pydata.org/

[5] https://pypi.org/project/XlsxWriter/

[6] https://www.tweetbinder.com/

[7] https://www.simplilearn.com/tutorials/excel-tutorial/excel-data-cleaning#:~:text=One%20of%20the%20easiest%20ways,dataset%20that%20has%20duplicate%20values

[8]https://www.sciencedirect.com/science/article/pii/S2405844021002966

[9] https://www.computerhope.com/jargon/r/regex.htm

[10] https://www.geeksforgeeks.org/introduction-to-stemming/

[11] https://aws.amazon.com/sagemaker/data-labeling/what-is-data-labeling/