Topic Modelling Arabic Using Different Natural Language Processing Techniques

April 2022

Contents

List of Figures iii

List of Tables iv

1 Proposed Approach 1
- 1.1 Obtaining Data . . . 2
  - 1.1.1 Tweepy Tire . . . 2
  - 1.1.2 Application Tier . . . 3
  - 1.1.3 Database Tier . . . 4
  - 1.1.4 Parsing Tweets Tier . . . 4
- 1.2 Data Prepossessing . . . 5
- 1.3 Building the LDA model for Arabic Tweets . . . 5
- 1.4 Model Evaluation and Testing . . . 7

2 Implementation 9
- 2.1 Obtaining Data . . . 9
- 2.2 Data Preprocessing . . . 12
  - 2.2.1 EDA . . . 12
  - 2.2.2 Filtering out all data which where retweets and duplicates: . . . 14
  - 2.2.3 Removing all words starting with "@" which represents the mentioned users within tweets: . . . 15
  - 2.2.4 Cleaning the tweets by removing links, decoding HTML tags and substituting accents: . . . 15
  - 2.2.5 Removing non Arabic characters and punctuation marks . . . 15
  - 2.2.6 Extract Hasthtags from words . . . 15
  - 2.2.7 Removing extra spaces within tweets: . . . 15
  - 2.2.8 Reduce repeated substring . . . 15
  - 2.2.9 Removing Arabic Stop Words: . . . 16
  - 2.2.10 Diacritization of Arabic tweets: . . . 16
  - 2.2.11 Reducing Orthographic Ambiguity (Normalisation) : . . . 16
  - 2.2.12 Arabic Words tokenization: . . . 17
  - 2.2.13 Morphological Disambiguation: . . . 17
  - 2.2.14 Performing Lemmatization on the processed texts: . . . 17
- 2.3 Model Building an training . . . 19
  - 2.3.1 Building and train the model . . . 20

3 Testing and Evaluation 21

4 Conclusion **25**
4.1 Summary 25
4.2 Future work 25

Bibliography **26**

List of Figures

1.1 Approach of correlation of mining of Twitter data 1
1.2 Conceptual Diagram for obtaining Data step 2
1.3 LDA topic modelling sample [1] 6
1.4 LDA Diagram source Lee et al. (2018) [2] 6

2.1 Couch Database on local host 9
2.2 Application Tier Snapshot 10
2.3 Example of one of the retrieved tweets 11
2.4 Example of one of the retrieved tweets saved as a JSON into the database 11
2.5 The text of the tweet 11
2.6 Tweets Distribution 12
2.7 Word Cloud for terms before cleaning 13
2.8 Word Cloud for terms before lemmatization 13
2.9 Most Frequent Terms 13
2.10 Tweets statistics before cleaning 13
2.11 Word cloud of terms after cleaning 18
2.12 Tweets records and statics after cleaning 19

3.1 The number of topics against Coherence scores 22
3.2 The main interface for the Visualisation 23
3.3 The final result for visualisation of each topic with their keywords 24

List of Tables

3.1 Coherence scores table 21
3.2 Each topic with the assigned terms 22
3.3 Words for each topic with their assigned probabilities 23

Listings

2.1 Create CouchDBs . . . 10
2.2 Stream API . . . 10
2.3 Parsing Tier function . . . 11
2.4 Delete duplicated and retweeted tweets . . . 14
2.5 preprocess function . . . 14
2.6 Reduce Repeated Substring Function . . . 16
2.7 Removing Stop Words Function . . . 16
2.8 Strip Tashkeel Function . . . 16
2.9 Normalisation Function . . . 17
2.10 Word Tokenizer . . . 17
2.11 Lemmatization Function . . . 18
2.12 Building LDA Dictionary . . . 19
2.13 Train LDA Model function . . . 20
2.14 LDA Results . . . 20

Chapter 1

Proposed Approach

This chapter will discuss topic modelling by using LDA algorithm methodology in our study and how can we perform it. This study was performed by following steps as illustrated in 1.1

- Data Obtaining
- Data Prepossessing
- Building the LDA model for Arabic Tweets
- Model evaluation and testing

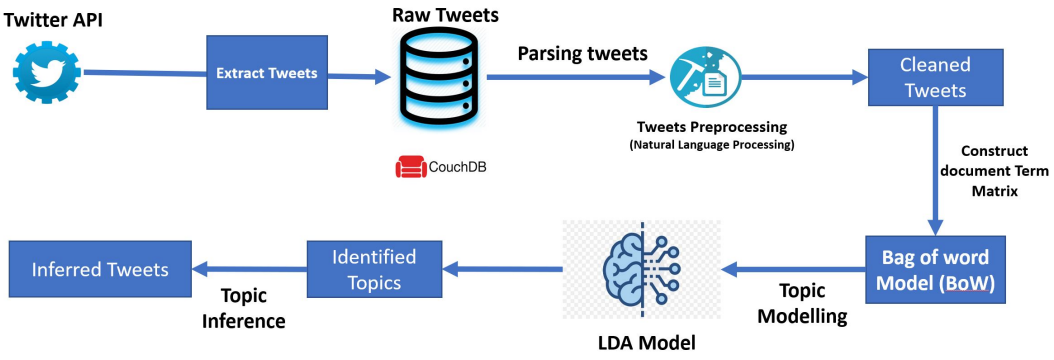

FIGURE 1.1: Approach of correlation of mining of Twitter data

1.1 Obtaining Data

The data is collected from Twitter social media platform. The users can share their thoughts and opinions on different social issues, matters and events by posting about 280 characters long posts. Twitter data can be accessed by making calls to Twitter APIs. These API calls require a developer access token. We build a harvester by python programming language to retrieve raw Tweets. We used **Tweepy**[3] which is a python library for accessing the Twitter API.

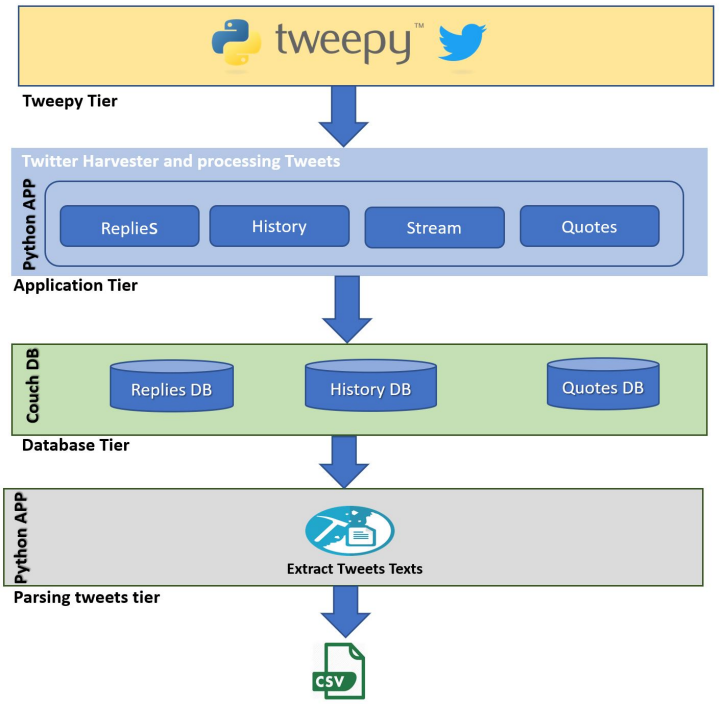

FIGURE 1.2: Conceptual Diagram for obtaining Data step

We filtered all tweets containing the word,صحتك,كورونا,سياسة,الأرصاد,تقنية,رياضة,تربية الأبناء. Also, we retrieved the tweets from common trends in Saudi Arabia during a specific period to retrieve a massive amount of the dataset.

The streaming API returns tweets, as well as several other types of messages (e.g. a tweet deletion notice, user update profile notice, etc.), all in **JSON** format. We use Python libraries `JSON` for parsing the data and pandas for data manipulation. The Fig 1.2 illustrates the conceptual diagram for obtaining the tweets tires and how we can store them into **CouchDB** [4] database.

1.1.1 Tweepy Tire

This tier, when we can use the Tweepy library to access the tweets via the internet. Also, we can retrieve users and their profile contents.

1.1.2 Application Tier

The application tier consists of Twitter data harvesting and processing jobs. The process comprises four scheduled jobs that allow obtaining tweets from different sources. Below is a description of each of these sub-processes.

1.1.2.1 Current tweets:

The Twitter streaming API allowed the process to capture all the tweets with origin Saudi Arabia in a continuous and real-time way. Given that the tweet can be a new, a reply or a retweet, this will determine to which database the tweet is stored. If it is a new tweet, the system will store it in the `historic_db`, if it is a reply or a retweet, it will be stored in the `reply_db` which will be used for further processing.

1.1.2.2 Historic tweet:

The search API allows access to the latest seven days of history when a query is submitted. For this specific case, a query was created that would allow access to tweets based on Saudi Arabia's location to complement the information not collected by the. Like the current tweets subprocess, it has different types of tweets (new, reply, or retweet) and given this type will determine its destination in the database.

1.1.2.3 Tweets by user:

Since the previous process only can access the last seven days of history, another process was generated to access more history based on each user and their history.

A user base was created with the previously stored tweets that allowed access to their history and the search API was used for this purpose.

The tweets will be stored in the basis as the last two processes. Reply and Retweet: From the two previous processes, replies or retweets to other tweets are obtained, generating a parent-child relationship.

Given the above, a process is created to search for the parent tweet and add its information to the child tweet.

1.1.3 Database Tier

Apache CouchDB is an open-source document-oriented NoSQL database, implemented in Erlang. CouchDB uses multiple formats and protocols to store, transfer, and process its data; it uses JSON to store data, JavaScript as its query language using MapReduce, and HTTP for API. Unlike a relational database, a CouchDB database does not store data and relationships in tables. Instead, each database is a collection of independent documents, and each document is assigned a unique key.

Couch DB provides a set of RESTful APIs for reading and updating (add, edit, delete) database documents. Easy to use replication, Couch DB allows users to copy, share and synchronize the data between databases and machines. For this solution, three databases are created and below are the name and purposes of the three databases.

1.1.3.1 History database

This database stores raw tweets which are ready to process for Topic Modelling. This database stores tweets from sources like streaming, history, retweets and replies to tweets. Among the fields stored in each document, it can be found the date, text and location of the tweets(not always available).

1.1.3.2 Replies database

Given that many tweets are replies to tweets or retweets, these documents have to be pre-processed to be able to get the root tweet and add that text to the current text of the tweet. replies_db is meant to support this process and keep a record of these kinds of tweets.

1.1.3.3 Quote database

It is used to save the quoted tweets. A quote tweet is a retweet with an added comment. They can be used as extra tweets for topic modelling.

1.1.4 Parsing Tweets Tier

In this tier, we extracted the text tweets from the database and skipped unwanted details such as the author of the tweet, number of retweets, and creation time. After that, we convert the raw tweets records into CSV files to be applicable for the next steps.

1.2 Data Prepossessing

As a part of Natural Language Processing techniques as described in 1.1. We first need to process the dataset by converting the text file, which is in `CSV` format, to a data frame consisting of tweet id and the corresponding tweet as separate columns. We have to do the following steps to clean the dataset.

- Filtering out all data which were retweets and duplicates.
- Cleaning the tweets by removing links, decoding `HTML` tags and substituting accents.
- Removing all words starting with "`@`" which represents the mentioned users within tweets.
- Removing non-Arabic characters and punctuation marks.
- Extract Hasthtags from words.
- Removing extra spaces within tweets.
- Reduce repeated substring
- Removing Arabic Stop Words.
- Discretization of Arabic tweets.
- Reducing Orthographic Ambiguity and normalization.
- Arabic Words tokenization.
- Morphological Disambiguation
- Performing Lemmatization on the processed texts.

1.3 Building the LDA model for Arabic Tweets

We decided to build our model with one of the most popular topic modelling techniques, which is the LDA model. Latent Dirichlet allocation (LDA) is a generative topic model for text documents [5]. By Using topic modelling, we can discover the abstract "topics" that appear in the collections of documents.

Each record contains a variety of words, and each topic can be linked to a particular word. The purpose of the LDA is to identify the subject matter of a document based on the words it contains. Similar-themed documents are assumed to use the same set of words. In this way, the

documents can map the probability distribution of latent topics. In our study, every tweet will be considered a document.

There are two fundamental principles that guide the LDA model as illustrated in Fig 1.4:

- **Each document consists of various of topics:**
  Using a three-topic model, we can say that a document is 70% about topic A, 30% about topic B, and 0% about topic C."
- **Each particular topic comprises several words:**
  A topic can be thought of as a probabilistic distribution of multiple words.

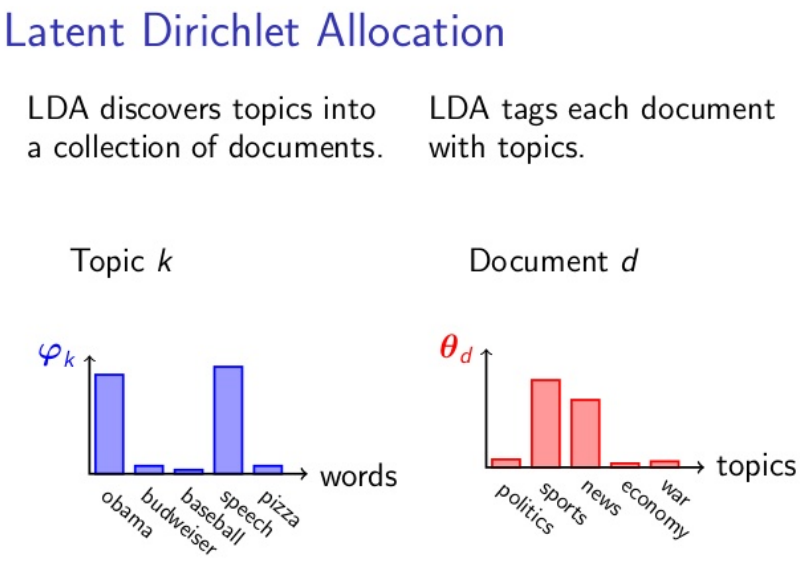

FIGURE 1.3: LDA topic modelling sample [1]

LDA is an imaginary generative process, as depicted in the plate diagram 1.4

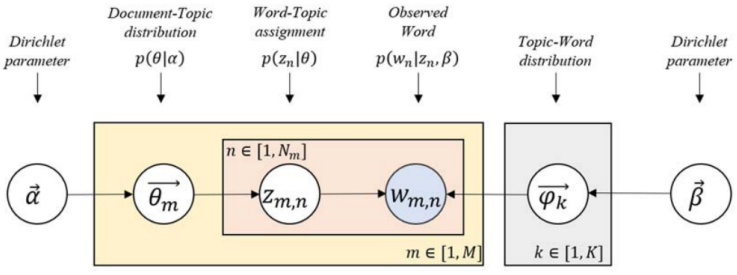

FIGURE 1.4: LDA Diagram source Lee et al. (2018) [2]

- M denotes the number of documents.

- N is the number of words in a given document (document i has N_i words).
- $\vec{\theta_m}$ is the expected topic proportion of document$m$, which is generated by a Dirichlet distribution parameterized by $\vec{\alpha}$ (e.g., in a two topic model $\theta_m = [0.3, 0.7]$ means document m is expected to have 30% topic 1 and 70% topic 2).
- $\vec{\phi_k}$ is the word distribution of topic k, which is generated by a Dirichlet distribution parameterized by $\vec{\beta}$
- $z_{m,n}$ is the topic for the nth word in document m, one word are assigned to one topic.
- $w_{m,n}$ is the word in the nth position word of document

The only observed variable in this graphical probabilistic model is $w_{m,n}$, so it is "latent". To actually infer the topics in a corpus, we imagine the generative process as follows. LDA assumes the following generative process for a corpus D consisting of MM documents each of length N_i:

1. Generate $\vec{\theta_i} \sim \text{Dir}(\vec{\alpha})$, where $i \in \{1, 2, ..., M\}$. $\text{Dir}(\vec{\alpha})$ is a Dirichlet distribution with symmetric parameter$\vec{\alpha}$where $\vec{\alpha}$ is often sparse.
2. Generate $\vec{\phi_k} \sim \text{Dir}(\vec{\beta})$, where $k \in \{1, 2, ..., K\}$ and $\vec{\beta}$ is typically sparse.
3. For the nth position in document m, where $n \in \{1, 2, ..., N_m\}$ and $m \in \{1, 2, ..., M\}$.
   (a) Choose a topic $z_{m,n}$ for that position which is generated from$z_{m,n} \sim \text{Multinomial}(\vec{\theta_i})$
   (b) Fill in that position with word $w_{m,n}$ which is generated from the word distribution of the topic picked in the previous step $w_{i,j} \sim \text{Multinomial}(\phi_{z_{m,n}})$

In our study we decided to build the LDA algorithm which is developed by `Gensim` [6] as explained in Section 2.3.

1.4 Model Evaluation and Testing

It has been suggested in the NLP community that topic models be evaluated using coherence measures. As an intrinsic evaluation method for topic models, topic coherence has been proposed.

Top-word pairwise word similarities are used to calculate the average or median of these pairwise word similarities. Topic modelling does not use any external data when determining word similarity.

The Topic Coherence Metric is used in our study to evaluate the outputted topics for each tweets cluster.

$$C(t; V^{(t)}) = \sum_{m=1}^{M} \sum_{l=1}^{m-1} log \frac{D(v_m^{(t)}, v_l^{(t)}) + 1}{D(v_l^{(t)})}$$

Where $D(v)$is the document frequency of word type (unique words) v, and $D(v, v')$ is the co-document frequency of word types v and v', and $V(t)$ is a list of the M most probable terms within the topic t.

In other words, a higher topic coherence score indicates that the most meaningful words in the documents used to develop a topic are more likely to be found together.This metric is developed and by `Gensim` library [6] .

Chapter 2

Implementation

This chapter will talk about our approach steps in our research. We used python programming language to develop our model.Also, we used various data science libraries such as `NLTK`, `Gensim`,`pyarabic` and `CAMeL Tools`. `CAMeL Tools` which is a suite of Arabic natural language processing tools developed by the CAMeL Lab at New York University Abu Dhabi.

2.1 Obtaining Data

FIGURE 2.1: Couch Database on local host

We decided to save the tweets into **CouchDB** as mentioned in section 1.1. Our purpose is to retrieve a huge amount of tweets, which may help us get the pattern of topics within Tweets in the Saudi Arabia region. We could retrieve around 500,000 tweets and store them with their details into **CouchDB** database. Fig 2.1 shows the database after creation by the code in 2.1 which contains the retrieved tweets.We used Tweepy library to access Twitter API and we use filter to stream all tweets containing the word,صحتك,كورونا,سياسة,الأرصاد,تقنية,رياضة,تربية الأبناء. The tweets were retrieved between (20 August 2021 -25 August 2021). The number of records is 24k tweets.

However, to obtain a large number of tweets, we retrieved the tweets containing trended hashtags during the Saudi National Day between (23 September 2021 and 24 September 2021). We built our model based on the large dataset, which was 500k records.

```
import couchdb
#Local Server
couch = couchdb.Server('http://couchdb:aa2447@localhost:5984')

#Create databases

couch.create('db_arabic_historic')
couch.create('db_arabic_streamer')
couch.create('db_arabic_replies')
couch.create('db_arabic__quoted')
couch.create('db_arabic__user')
couch.create('db_user_tweets')
couch.create('db_tweets_quoted')
```

Listing 2.1: Create CouchDBs

The code 2.2 implements the harvester and retrieves the tweets by using the Tweepy library, and 2.2 shows an example for retrieving tweets process.

```
def tweetProcessor(api):
    my_stream_listener = MyStreamListener()
    my_stream = tweepy.Stream(auth=api.auth, listener=my_stream_listener)
    while True:
        try:
            keywords=Arabic_word_lists
            my_stream.filter(track=kewyords,is_async=True)

        except:
            continue
            harvestTweets()
```

Listing 2.2: Stream API

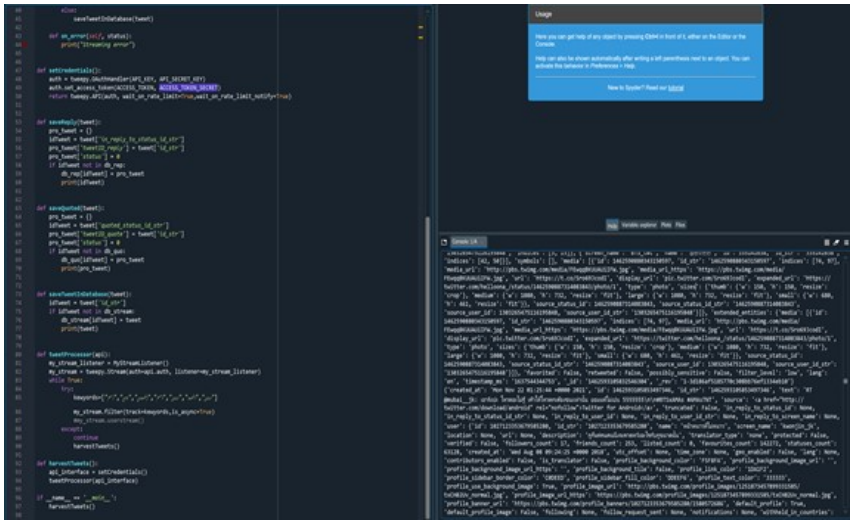

FIGURE 2.2: Application Tier Snapshot

After we retrieved the tweets, we have to parsing tweets from `JSON` format and convert them into CSV files to be applicable for next process which is Data Prepossessing.

FIGURE 2.3: Example of one of the retrieved tweets

The tweet with status id=1435173084718305285 can be displayed on Twitter website as shown in Fig 2.3. After saving the tweets within the database as illustrated in Fig 2.4,in parsing tweets tier we have to extract the text from JSON files and produce CSV files for preprocessing.The code for the main function for parsing tweets `getMainText()`can be seen in code 2.3 and the extracted text can be seen in Fig 2.5

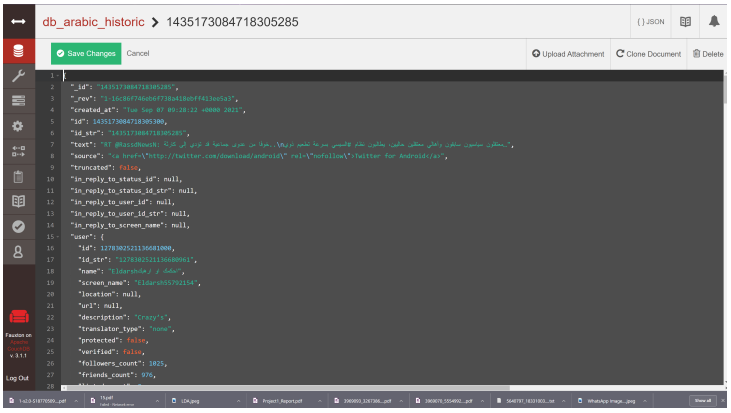

FIGURE 2.4: Example of one of the retrieved tweets saved as a JSON into the database

خوفًا من عدوى جماعية قد تؤدي إلى كارثة.. معتقلون سياسيون سابقون وأهالي معتقلين حاليين، يطالبون نظام #السيسي بسرعة تطعيم ذويهم ضد فيروس #كورونا؛ حفاظًا على أرواحهم، مطالبين أيضًا بالإفراج عن كل السجناء فوق عمر 60 عامًا https://t.co/5JAvRoWpDK

FIGURE 2.5: The text of the tweet

```
def getMainText(tweet):
     text=""
     if('retweeted_status'  in tweet):
        if("extended_tweet" in tweet["retweeted_status"] ):
            text =  tweet["retweeted_status"]["extended_tweet"]["full_text"]
            #print(text)
        else:
            text=tweet['text']
        elif("extended_tweet" in tweet ):
        if("full_text" in tweet["extended_tweet"]):
            text=tweet["extended_tweet"]["full_text"]
    else:
        text=tweet['text']
    return text
```

Listing 2.3: Parsing Tier function

2.2 Data Preprocessing

We generated a CSV file consisting of roughly 506k tweets by 100k users, as you can observe in 2.6.Upon further analysis We also observed that many texts were short. Before cleaning the dataset, we performed exploratory data analysis (EDA) to gain insight into the records.

	index
count	506284.00000
mean	253141.50000
std	146151.74618
min	0.00000
25%	126570.75000
50%	253141.50000
75%	379712.25000
max	506283.00000

FIGURE 2.6: Tweets Distribution

2.2.1 EDA

From the figures 2.8,2.9 and 2.10.It can be seen that there are many stop words within the dataset. You can notice the size of من and في and a lot of unwanted characters such as "". Also,

FIGURE 2.7: Word Cloud for terms before cleaning

FIGURE 2.8: Word Cloud for terms before lemmatization

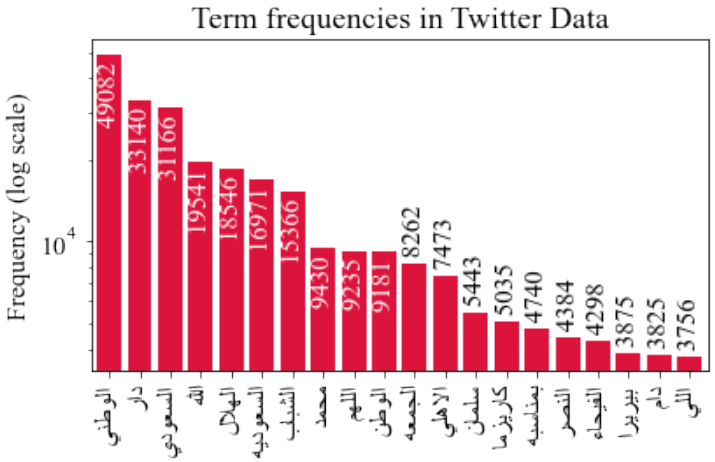

FIGURE 2.9: Most Frequent Terms

	tweet	stopwords	emoji_count	avg_words
0	#اليوم_الوطني ي اهل الخير تكفون زوجي مريض بالسرطان منتشر بجسمة مقعدعايش على الكيماوي وكل همة سداد الدين يشهدالله تم فصلة من العمل باصابة عمل خارج مؤقت بسبب كورونا من السجن هوعائلنا بعدالله مالناالحدتراكمت علية الديون وعاجز عن سدادهاعتذادخل ونسكن ايجارتكفون والله حالنا صعب https://t.co/Kj6qKtYJel	9	0	5.795455
1	قررت فلندا إلغاء جميع قيود كورونا بعد تطعيم 80٪ من جميع الأشخاص فوق سن 12 بالكامل :RT @_AAN90	6	0	4.277778
2	مع سعودي فلورز اختار اللي يناسبك واستعد تنبهر بعد التنفيذ تواصل معنا على الارقام التالية 0534305248 0500888750 #الدمام #الدفاع_الجوي_السعودي #الامارات #الدفاع_الجوي #القبول_الموحد #الكويت_البوسنة #الوطن #بس_حبيت_اقول #كورونا #صباح_الخير #نطالب_الملك_بالدراسه_عن_بعد #يوم_الجمعة https://t.co/vc2HToUgBS	4	0	9.066667
3	استمتع بالخصوصية في حديقتك الخاصة🛀نصنع الجمال والتميز والرفاهية عيش جوك واستمتع بالرفاهية🏠💎 وتواصل معنا عبر الارقام التالية:👇 0500888750 0590756444 #الدمام #الدفاع_الجوي_السعودي #بس_حبيت_اقول #صباح_الخير #عمرو_دياب #قاطعوا_ام_بي_سي_مصر #كورونا #يوم_الجمعة #درعا #درعا_البلد https://t.co/Fjm6DlE4IH	2	4	8.645161
4	مشروع أرضيات الخرسانة المطبوعة فيلا خاصه بالقصيم استمتع معنا بأرضيات مميزة👍 للتواصل على الارقام التالية 0500888750 0590756444 0592662299 0534305248 #الدمام #الدفاع_الجوي_السعودي #كورونا #بس_حبيت_اقول #يوم_الجمعة #الاحد #القبول_الموحد #قاطعوا_ام_بي_سي_مصر #عمرو_دياب https://t.co/WTQliq0Dsz	2	1	9.033333
5	اتوقع كل المؤشرات تقول اننا مقبلين على ازمة ماليه الحكومة ماعندها فلوس الشعب مفلس النفط يوم صاعد يوم نازل تضخم كورونا افلست شركات كثيره ف كل المؤشرات تقولك داشين بالطوفه قريبا	7	0	4.677419

FIGURE 2.10: Tweets statistics before cleaning

some words have the same root or include prefixes such as ال with السعوديه and السعودي which they have the same root which is سعودي.

2.2.2 Filtering out all data which where retweets and duplicates:

We filtered out all data which where retweets and duplicates via the code in the code listings 2.4. Line (1) is used to remove duplicated retweets and the second line(1) filters retweeted tweets. We deleted them because it may lead to building a biased model due to the frequent terms within the redundant tweets.

```
Data=pd.DataFrame(Data["tweet"].unique()).reset_index().drop('index',axis=1)
Data=Data[~Data['tweet'].str.startswith('RT')]
```

Listing 2.4: Delete duplicated and retweeted tweets

We developed a function including the other preprocessing steps, which are described in section 1.2. The function is implemented in code listings 2.5 which includes cleaning the text of the tweets.The function is implemented by Regular Expressions library `regex` which is comes with Python.

It helps us extract the matched pattern during the preprocessing approach.

```
def preprocess(tw):
    #remove users from tweets
    tw['Tweet_clean'].replace("(@[A-Za-z0-9]+)","",regex=True,inplace=True)
    #-----
    #Html coding
    tw["Tweet_clean"] = tw['Tweet_clean'].apply(lambda x: html.unescape(x))
      #remove  HTTPS links
    tw["Tweet_clean"].replace(r'https?://[A-Za-z0-9./]*','', regex=True,inplace=True)

    #------
    #remove non arabic character
    tw['Tweet_clean'].replace("([^0-9\u0621-\u064a\ufb50-\ufdff\ufe70-\ufefc])"," ",
    regex=True,inplace=True)
    #------
    #Stripping everything but alphanumeric chars from Hasthags
    tw["Tweet_clean"] .replace(r'#(\w+)', '' ,regex=True,inplace=True)
    #------
     # remove extra space
    tw["Tweet_clean"].replace("(\\r|)\\n$", '', regex=True,inplace=True)
    #remove digits
    tw["Tweet_clean"] .replace(r'(\d+)', '' ,regex=True,inplace=True)

    #---------------
    #remove empty lines from texts
    tw['Tweet_clean'].replace(r"^ +| +$", r"", regex=True, inplace=True)

    tw = tw.reset_index(drop=True)
    return tw
```

Listing 2.5: preprocess function

2.2.3 Removing all words starting with "@" which represents the mentioned users within tweets:

From the coding list 2.5 from function `preprocess()` at **the second line** ,you can observe that we replaced all words which start with @ with space.This function will remove all the mentioned users from the tweets.

2.2.4 Cleaning the tweets by removing links, decoding HTML tags and substituting accents:

As a part of NLP preprocessing steps,we deleted the included HTML tags and links within tweets .We first decoded HTML tags as explained in coding list 2.5 from function `preprocess()` at **the sixth line**.After we decoded HTML codes, we could able to detect the pattern for HTML tags and deleted them as it can be seen at **the eighth line**.

2.2.5 Removing non Arabic characters and punctuation marks

It is important to remove all Non-Arabic characters since we focus on Arabic texts.As illustrated in coding list 2.5 from function `preprocess()` at **the twelfth line**,this step will strip the whole non Arabic characters and keep Arabic words only.Also, this step will remove Arabic punctuation marks. Also, this step will include removing **emojis** which are considered unwanted items.

2.2.6 Extract Hasthtags from words

To extract the term from Hashtags which are important to detect the most important keywords into tweets, we stripped the symbol from the words which include it and kept the term.This step can be seen in coding list 2.5 from function `preprocess()` at **the sixteenth line**.

2.2.7 Removing extra spaces within tweets:

For removing extra space and new line from text and digits from tweets,it can be seen in coding list 2.5 from function `preprocess()` from (line 18 to 25)

2.2.8 Reduce repeated substring

We have to remove repeating characters within terms to extract the primary terms. We used **Maha** [7] library, which is a text processing library specially developed to deal with Arabic text to perform this task. The implementation can be seen in list 2.6

```
from maha.cleaners.functions import reduce_repeated_substring
Data['Tweet_clean'] = Data['Tweet_clean'].apply(reduce_repeated_substring)
```

Listing 2.6: Reduce Repeated Substring Function

2.2.9 Removing Arabic Stop Words:

Because we are dealing with texts, most of them are written in colloquial language.We decided to used around 14k stop words from various sources.They contains `NLTK` [8],`pyarabic`[7] libraries and the largest list of Arabic stop words on Github which contains collection misses many of these common spelling inconsistencies[9].

```

def remove_stop_words(text):
    tk = TweetTokenizer(strip_handles=True, reduce_len=True, preserve_case = False)
    text = simple_word_tokenize(text)

    text = [word for word in text if word not in Merged_stop_words and len(word)>1 ]
    text = ' '.join(word.strip() for word in text)
    return text
```

Listing 2.7: Removing Stop Words Function

2.2.10 Diacritization of Arabic tweets:

We removed the text's diacritics, which will reduce some serious data sparsity.Here are the list of the Arabic dicritics which we should strip from tweets , ّ"Tashdid",َ"Fatha",ً"Tanwin Fath",ُ" Damma",ٌ"Tanwin Damm",ِ"Kasra",ٍ"Tanwin Kasr",ْ"Sukun",ـ"Tatwil/Kashida".

This step can be performed by using function `strip_tashkeel()` from library `pyarabic` as it can be seen in listing 2.8

```

def remove_diacritics(text):

    text=araby.strip_tashkeel(text)
    return text
Data['Tweet_clean'] = Data['Tweet_clean'].apply(remove_diacritics)
```

Listing 2.8: Strip Tashkeel Function

2.2.11 Reducing Orthographic Ambiguity (Normalisation) :

To account for a number of common spelling inconsistencies across dialects (and, more broadly, for the gap between spoken and written Arabic), the next step is to eliminate orthographic

ambiguity. `CAMeL_Tools` [10] accomplishes this by omitting specific symbols from specific letters (the dots from the teh-marbuta ة and the hamza from the alef أإآ).

```
from camel_tools.utils.normalize import normalize_alef_maksura_ar
from camel_tools.utils.normalize import normalize_alef_ar
from camel_tools.utils.normalize import normalize_teh_marbuta_ar

def ortho_normalize(text):
    text = normalize_alef_maksura_ar(text)
    text = normalize_alef_ar(text)
    text = normalize_teh_marbuta_ar(text)
    return text

Data['Tweet_clean'] = Data['Tweet_clean'].apply(ortho_normalize)
```

Listing 2.9: Normalisation Function

2.2.12 Arabic Words tokenization:

The following step is a word tokenizer. This is required in order to enter our text into the functions of our next steps.We used simple_word_tokenize() which is included within `CAMeL_Tools` library as it can be seen in listings 2.10.

```

from camel_tools.tokenizers.word import simple_word_tokenize
Data["Tokens"] =Data["Tweet_clean"].apply(simple_word_tokenize)
```

Listing 2.10: Word Tokenizer

2.2.13 Morphological Disambiguation:

The striped diacritics in 2.2.10 create a new problem. Now that we know only the root characters, it is impossible to tell which of the many possible words. For example, "and" و with our contract' وبعقدنا could be interpreted as: 'and with our contract / necklace / psychoses' or 'and he stresses us out.'

To solve this contract/necklace `CAMeL_Tools` 'morphological analyzer' before extracting the lemmas for our proceeding texts.

2.2.14 Performing Lemmatization on the processed texts:

After performing Morphological Disambiguation in step 2.2.13 we can get all of the lemmas of our Arabic text using the function in listing 2.11

```
from camel_tools.disambig.mle import MLEDisambiguator
def get_lemmas(tokenized_text):
    print(tokenized_text)
    disambig = mle.disambiguate(tokenized_text)
    lemmas = [d.analyses[0].analysis['lex'] for d in disambig]
    return lemmas
```

Listing 2.11: Lemmatization Function

After performing the steps in Section 1.2,the number of records became **100k** .Also, now we are ready to build our model for topic modelling.The proceed data can be seen in Fig 2.12 which contains the cleaned records and the most frequent terms are shown in Fig 2.11 .

FIGURE 2.11: Word cloud of terms after cleaning

	tweet	stopwords	emoji_count	avg_words	Tweet_clean	Tokens	number_char	number_Tokens	word_count	char_count	avg_char_per_word
0	اليوم_الوطني ي اهل الخير تكفون زوجي مريض بالسرطان منتشر بجسمة# مقعدعايش على الكيماوي وكل همة سداد الدين يشهدالله تم فصلة من العمل باصابة عمل خارج مؤقت بسبب كورونا من السجن هوعائلنا بعدالله مالنااحدتراكمت علية الديون وعاجز عن سدادهاماعندنادخل ونسكن ايجارتكفون والله حالنا صعب https://t.co/Kj6qKtYJel	0	0	5.795455	وَطَنِيّ أَهْل خَيْر كَفَى زَوْج مَرِيض سَرَطان مُنْتَشِر جِسْم مقعدعايش كِيمِيائِيّ هَمّ سَداد دِين يشهدالله فَصْل عَمَل إِصابَة عَمَل خارِج مُؤَقَّت كَوَّر سِجْن هوعائلنا بعدالله مالنااحدتراكمت دَيْن عاجِز سدادهاماعندنادخل سَكَّن ايجارتكفون الله حال صَعْب	[وَطَنِيّ, أَهْل, خَيْر, كَفَى, زَوْج, مَرِيض, سَرَطان, مُنْتَشِر, جِسْم, مقعدعايش, كِيمِيائِيّ, هَمّ, سَداد, دِين, يشهدالله, فَصْل, عَمَل, إِصابَة, عَمَل, خارِج, مُؤَقَّت, كَوَّر, سِجْن, هوعائلنا, بعدالله, مالنااحدتراكمت, دَيْن, عاجِز, سدادهاماعندنادخل, سَكَّن, ايجارتكفون, الله, حال, صَعْب]	259	34	34	259	6.647059
7	نداء انساني مريض كورونا في مستشفى الجامعة الاميركية بحاجة ماسة إلى بلازما من شخص متعافي من كورونا منذ اقل من AB positive أو A positive فئة دم ثلاثة أشهر للتواصل... 03997473 70359863	0	0	4.687500	نِداء إِنْسانِيّ مَرِيض كَوَّر مُسْتَشْفَى جامِعَة أَمْرِيكِيّ حاجَة ماسّ بلازما فِئَة دَم شَخْص مُتَعافِي كَوَّر أَقَلّ ثَلاث شَهْر تَواصُل	[نِداء, إِنْسانِيّ, مَرِيض, كَوَّر, مُسْتَشْفَى, جامِعَة, أَمْرِيكِيّ, حاجَة, ماسّ, بلازما, فِئَة, دَم, شَخْص, مُتَعافِي, كَوَّر, أَقَلّ, ثَلاث, شَهْر, تَواصُل]	140	19	19	140	6.421053
8	أمر الرئيس الجزائري عبدالمجيد تبون اليوم ... بالإبقاء على الحدود البرية والبحرية والجوية مغلقة إلى حين الانتصار على فيروس كورونا المستجد.....	0	0	5.761905	أَمْر رَئِيس جَزائِرِيّ عَبْدالمَجِيد تبون إِبْقاء حَدّ بَرِّيّ بَحْرِيَّة جَوِّيّ مُغْلَق اِنْتِصار فَيْرُوس كَوَّر مُسْتَجِدّ	[أَمْر, رَئِيس, جَزائِرِيّ, عَبْدالمَجِيد, تبون, إِبْقاء, حَدّ, بَرِّيّ, بَحْرِيَّة, جَوِّيّ, مُغْلَق, اِنْتِصار, فَيْرُوس, كَوَّر, مُسْتَجِدّ]	127	15	15	127	7.533333
10	عبدالعزيز_بن_فهد اللهم لك الحمد والشكر تم اليوم اخذ الجرعة الأولى من لقاح# كورونا لسمو سيدي اسأل الله أن ينفع بها وأن يحفظ لنا أميرنا الغالي https://t.co/yTbG1OSnQM	0	0	5.346154	عَبْدالعَزِيز فَهْد اللَّهُمَّ حَمْد شُكْر أَخْذ جُرْعَة أَوَّل لَقاح كَوَّر سُمُوّ سَيِّد سَأَل الله نَفَع حَفِظ أَمِير غالِيّ	[عَبْدالعَزِيز, فَهْد, اللَّهُمَّ, حَمْد, شُكْر, أَخْذ, جُرْعَة, أَوَّل, لَقاح, كَوَّر, سُمُوّ, سَيِّد, سَأَل, الله, نَفَع, حَفِظ, أَمِير, غالِيّ]	129	18	18	129	6.222222
12	عشان كورونا صح @hu_x22 @1i_nll هه	0	0	13.333333	عُشّ كَوَّر صَحّ هه	[عُشّ, كَوَّر, صَحّ, هه]	19	4	4	19	4.000000
13	لا تخلي حرصك على صحتك يخليك تصاب بالوسواس #الصحي .. الأمراض أمر وارد لكل شخص يعيش في هذه #الحياة #مختبرات #مختبرات_الرياض https://t.co/EENmukOvRs	0	0	5.636364	تَخَلِّي حِرْص صِحَّة أَخْلَى أَصاب وَسْواس صِحِّيّ مَرَض أَمْر وارِد شَخْص عاش حَياة مُخْتَبَر مُخْتَبَر رِياض	[تَخَلِّي, حِرْص, صِحَّة, أَخْلَى, أَصاب, وَسْواس, صِحِّيّ, مَرَض, أَمْر, وارِد, شَخْص, عاش, حَياة, مُخْتَبَر, مُخْتَبَر, رِياض]	111	16	16	111	6.000000
14	أكثرنا كان متحمس جدا للهسنه وخاصه بعد انقطاع فتره كبيره عن الدوام الحضوري بسبب اوضاع كثيرة واكثي مازاد حماسنا ان بنرجع نداوم 100٪ وبقرارات حلوة وتمر سنتنا بسلام مثل سنواتنا قبل كورونا بس بعد قرارات أمس الي قتلت حماسنا وتمنينا يكون مجرد حلم! #قرارات_دفعة_٢٠٢٢	0	0	4.886364	أَكْثَر مُتَحَمِّس للهسنه خاصّ اِنْقِطاع راوَنْد كَبِير دَوام الحضوري وَضْع كَثِير واكثي مازاد حَماس بنرجع داوَم قَرار حَلّ مَرّ سَنَة سَلام سَنَة كَوَّر قَرار قَتَل حَماس تَمَنَّى مُجَرَّد حُلْم قَرار دَفْع	[أَكْثَر, مُتَحَمِّس, للهسنه, خاصّ, اِنْقِطاع, راوَنْد, كَبِير, دَوام, الحضوري, وَضْع, كَثِير, واكثي, مازاد, حَماس, بنرجع, داوَم, قَرار, حَلّ, مَرّ, سَنَة, سَلام, سَنَة, كَوَّر, قَرار, قَتَل, حَماس, تَمَنَّى, مُجَرَّد, حُلْم, قَرار, دَفْع]	207	31	31	207	5.709677
15	الى متى يستمر فيروس كورونا؟ تبلغ مدة حضانة الفيروس من 2 إلى 5 أيام بعد الإصابة، في حين أن بعض الناس قد يستغرقون وقتًا طويلاً للتعافي، فمن المقدر أن متوسط الوقت الذي يستغرقه الفيروس في الجسم هو أسبوعين شركة تعقيم من كورونا @alnqaa1 تعقيم منازل تعقيم البيت الشركة المصنع https://t.co/huAkjMmO7B	0	0	4.860000	مَتَى اِسْتَمَرّ فَيْرُوس كَوَّر بَلَغ مُدَّة حِضانَة فَيْرُوس إِصابَة ناس اِسْتَغْرَق طَوِيل تَعافِي مُقَدَّر مُتَوَسِّط اِسْتَغْرَق فَيْرُوس جِسْم أُسْبُوع شَرِكَة تَعْقِيم كَوَّر تَعْقِيم مَنْزِل تَعْقِيم بَيْت شَرِكَة مَصْنَع	[مَتَى, اِسْتَمَرّ, فَيْرُوس, كَوَّر, بَلَغ, مُدَّة, حِضانَة, فَيْرُوس, إِصابَة, ناس, اِسْتَغْرَق, طَوِيل, تَعافِي, مُقَدَّر, مُتَوَسِّط, اِسْتَغْرَق, فَيْرُوس, جِسْم, أُسْبُوع, شَرِكَة, تَعْقِيم, كَوَّر, تَعْقِيم, مَنْزِل, تَعْقِيم, بَيْت, شَرِكَة, مَصْنَع]	229	28	28	229	7.214286
16	شركة تعقيم بالرياض هل يمكن أن تنتقل فيروسات كورونا من شخص إلى آخر؟ نعم، يمكن أن تنتقل بعض سلالات الفيروس من شخص إلى آخر، بالاتصال عن قُرب مع الشخص المصاب عادةً، كما يحدث في سياق الأسرة أو العمل أو في مراكز الرعاية الصحية مثلاً. شركة تعقيم منازل تعقيم مكاتب @Alnqaa_Care https://t.co/s0Yw6kjbNY	0	0	4.764706	شَرِكَة تَعْقِيم رِياض اِنْتَقَل فَيْرُوس كَوَّر شَخْص آخَر اِنْتَقَل سُلالَة فَيْرُوس شَخْص آخَر اِتِّصال شَخْص مُصاب عادَة حَدَث سِياق أُسْرَة عَمَل مَرْكَز رِعايَة صِحِّيّ مَثَّل شَرِكَة تَعْقِيم مَنْزِل تَعْقِيم مَكْتَب	[شَرِكَة, تَعْقِيم, رِياض, اِنْتَقَل, فَيْرُوس, كَوَّر, شَخْص, آخَر, اِنْتَقَل, سُلالَة, فَيْرُوس, شَخْص, آخَر, اِتِّصال, شَخْص, مُصاب, عادَة, حَدَث, سِياق, أُسْرَة, عَمَل, مَرْكَز, رِعايَة, صِحِّيّ, مَثَّل, شَرِكَة, تَعْقِيم, مَنْزِل, تَعْقِيم, مَكْتَب]	222	30	30	222	6.433333
18	😂😩 اي والله لما جاني كورونا كانت اخف بعد @Mmmmm949494	0	0	4.400000	الله جانِي كَوَّر خَفَى	[الله, جانِي, كَوَّر, خَفَى]	25	4	4	25	5.500000

FIGURE 2.12: Tweets records and statics after cleaning

2.3 Model Building an training

2.3.0.1 Create the Dictionary and Corpus needed for Topic Modeling

We use the parallelized Latent Dirichlet Allocation (LDA) from **Gensim** for building the LDA model. Firstly, we need to create the Dictionary and Corpus needed for Topic Modelling. The two main inputs to the LDA topic model are the dictionary(id2word) and the corpus.

```
documents = Data["Tokens"].values
texts =documents
#Building Dictionary for LDA
id2word = corpora.Dictionary(texts)
corpus = [id2word.doc2bow(text) for text in texts]
```

Listing 2.12: Building LDA Dictionary

Gensim creates a unique id for each word in the document. The produced corpus from Code 2.12 **id2word** are (word_id, word_frequency) which is c alled Bag of Words. For example, (0, 1) above implies, word id 0 occurs once in the first document. Likewise, word id 1 occurs twice and so on. This is used as the input by the LDA model.

2.3.1 Building and train the model

Everything required to train the LDA model is available now. In addition to the corpus and dictionary, we also need to provide the number of topics.

Apart from that, `alpha` α and `eta` η are hyperparameters that affect the sparsity of the topics. According to the `Gensim` docs, both defaults to 1.0/(num_topics) prior.

`chunksize` is the number of documents to be used in each training chunk.

`update_every` determines how often the model parameters should be updated and `passes` is the total number of training passes.

We trained the model by the code into listing 2.13. We set a list of a number of topics to select the best model for our study.

The results of the final model will be explained in Chapter 3 as the results can be seen in 3.3.

```
#numb_topics can be 2,3,4,5,6,7,8,9,10,11,12,13,14,15,based
on the coherent score testing.
lda = LdaMulticore(corpus=corpus,
                   id2word=id2word,
                   num_topics=numb_topics,
                   random_state=100,
                   chunksize=100,
                   passes=10,
                   per_word_topics=True
                  )
```

Listing 2.13: Train LDA Model function

```

lda.print_topics()
```

Listing 2.14: LDA Results

Chapter 3

Testing and Evaluation

Topic modelling is a form of unsupervised learning, so the set of possible topics is unknown. After building several LDA models with different numbers of topics (k), we select the one with the highest coherence score as explained in Section 1.4 as the optimum number of topics. Choose a "k" at the end of rapid growth in topic coherence to find meaningful and interpretable topics.

We calculated topic coherence for LDA topic models using Gensim's `CoherenceModel()` function.The results can be seen in Table 3.1.We achieved the highest **coherence score = 0.432296** when the number of topics is 9 for our model.The results are illustrated in Tbale (3.1) and the Fig (3.1).After we selected the $k = 9$,we assigned it to the `num_topics()` in listing 2.13 in Chapter 2.

Number of Topics (k)	**Coherence Score**
2	0.33201
3	0.361511
4	0.375995
5	0.357321
6	0.391509
7	0.317913
8	0.392247
9	**0.432296**
10	0.362245
11	0.347553
12	0.359228
13	0.346675
14	0.392998
15	0.36078

TABLE 3.1: Coherence scores table

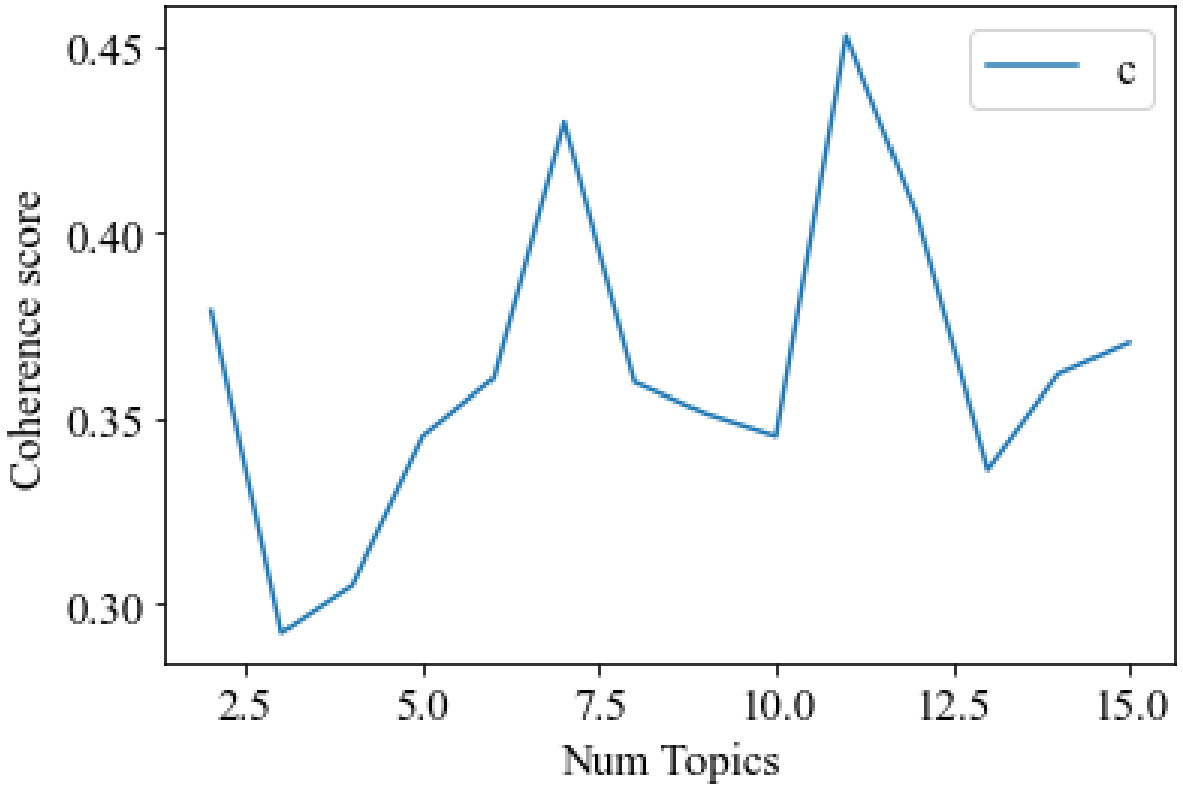

FIGURE 3.1: The number of topics against Coherence scores

The result of the words with their assigned probabilities for each topic can be printed by the method `print_topics ()` in Listing 2.14 in Chapter 2.The output can be seen in Table (3.3).The Table display Words for each topic with their assigned probabilities.Table 3.2 the assigned words for each topic.

Topic Number	Words
0	[هِلال، وَطَنِيّ، سَعُودِيّ، شَبّ، دار، تَخْفِيض، العبداللطيف، مَفْرُوشات، وَطَن، رِياض]
1	[سَعُودِيّ، وَطَنِيّ، اللّه، كارفور، مُحَمَّد، دار، سَيِّد، كؤد، وَلِي، اللّٰهُمَّ]
2	[هِلال، اللّه، نَصْر، شَبّ، حال، دار، جَدِيد، باطِن، لاعِب، مُباراة]
3	[شَبّ، كُوَيْت، هِلال، خَضِير، اللّه، وَطَنِيّ، حال، تَسْجِيل، حِساب، سَفَر]
4	[سَلَّم ، صل، مُحَمَّد، اللّٰهُمَّ، نَبِيّ، اللّه، جُمْعَة، باراك، صَلَّى، صَلاة]
5	[وَطَنِيّ، صِحَّة، وَطَن، اللّه، فَيْرُوس، لَقاح، سَعُودِيّ، رَقْم ، جُرْعَة، هِلال]
6	[جُمْعَة، اللّه، اللّٰهُمَّ، وَطَنِيّ، وَطَن، الل، سَعُودِيّ، رَحِم، مُحَمَّد، جَعَل]
7	[سَعُودِيّ، وَطَنِيّ، دار، وَطَن، عِزّ، دام، اللّه، أَمْن، مَمْلَكَة، أَهْلِيّ]
8	[وَطَنِيّ، سَعُودِيّ، دار، تَعْلِيم ، مَجْد، وَطَن، يَوْم، خَيْر، مُشارَكَة، مُناسَبَة]

TABLE 3.2: Each topic with the assigned terms

Topic_Number	Words	Weight
0	[هِلال، وَطَنِيّ، سَعُودِيّ، شَبّ، دار، تَخْفِيض، العبداللطيف، مَقْرُوشات، وَطَن، رِياض]	[0.040430978, 0.03379977, 0.031848613, 0.02951761, 0.021386767, 0.012204995, 0.010964618, 0.010833869, 0.008594224, 0.008028839]
1	[سَعُودِيّ، وَطَنِيّ، اللّه، كارفور، مُحَمَّد، دار، سَيِّد، كوْد، وَلِي، اللّهُمَّ]	[0.030022703, 0.029111644, 0.026326293, 0.026056195, 0.021457698, 0.019146556, 0.011840816, 0.01027253, 0.009629028, 0.0094303405]
2	[هِلال، اللّه، نَصْر، شَبّ، حال، دار، جَدِيد، باطِن، لاعِب، مُبارَاة]	[0.012778848, 0.012453038, 0.012284182, 0.01053993, 0.010408824, 0.009565501, 0.008741282, 0.007595001, 0.0074849073, 0.006577454]
3	[شَبّ، كَوِيْت، هِلال، خَضِير، اللّه، وَطَنِيّ، حال، تَسْجِيل، حِساب، سَفَر]	[0.011094921, 0.008893895, 0.008307676, 0.00788375, 0.0075734435, 0.007339966, 0.0072028087, 0.0064331447, 0.006402384, 0.00574607]
4	[سَلَّم، صل، مُحَمَّد، اللّهُمَّ، نَبِيّ، اللّه، جُمْعَة، بارك، صَلِّى، صَلاة]	[0.04455367, 0.040600777, 0.039024595, 0.03700333, 0.033825383, 0.030247724, 0.027761135, 0.01356362, 0.012640123, 0.01195126]
5	[وَطَنِيّ، صِحّة، وَطَن، اللّه، فَيْرُوس، لَقاح، سَعُودِيّ، رَقْم، جُرْعَة، هِلال]	[0.011931988, 0.009373382, 0.0074697877, 0.0070286035, 0.006326058, 0.0053108972, 0.0050615096, 0.004837527, 0.004317959, 0.0042437683]
6	[جُمْعَة، اللّه، اللّهُمَّ، وَطَنِيّ، وَطَن، ال، سَعُودِيّ، رَحِم، مُحَمَّد، جَعَل]	[0.023672549, 0.023390044, 0.020094931, 0.014384721, 0.014134624, 0.0112051675, 0.009971517, 0.008835785, 0.008063997, 0.0067415107]
7	[سَعُودِيّ، وَطَنِيّ، دار، وَطَن، عِزّ، دام، اللّه، أَمْن، مَمْلَكَة، أَهْلِيّ]	[0.047256194, 0.038882796, 0.03481146, 0.030511713, 0.020627398, 0.017798103, 0.017667485, 0.013242955, 0.012328967, 0.010795315]
8	[وَطَنِيّ، سَعُودِيّ، دار، تَعْلِيم، مَجْد، وَطَن، يَوْم، خَيْر، مُشارَكَة، مُناسَبَة]	[0.10158062, 0.09095499, 0.053436708, 0.015652603, 0.012843374, 0.012296804, 0.009406656, 0.008495404, 0.008090902, 0.007831353]

TABLE 3.3: Words for each topic with their assigned probabilities

Finally,We use `LDAvis` [11] library , a web-based interactive visualization of topics estimated using LDA.`pyLDAVis`, developed by Gensim, is the most frequently used visualisation tool for visualising the data contained in a topic model.

You can see the main interface for the visualisation in Figure 3.2. The bubbles on the left illustrate the overall topic distribution, while the sky blue bars on the right illustrate the overall term frequency distribution. When a selected bubble is being more larger, it indicates the selected terms relevant to that topic.

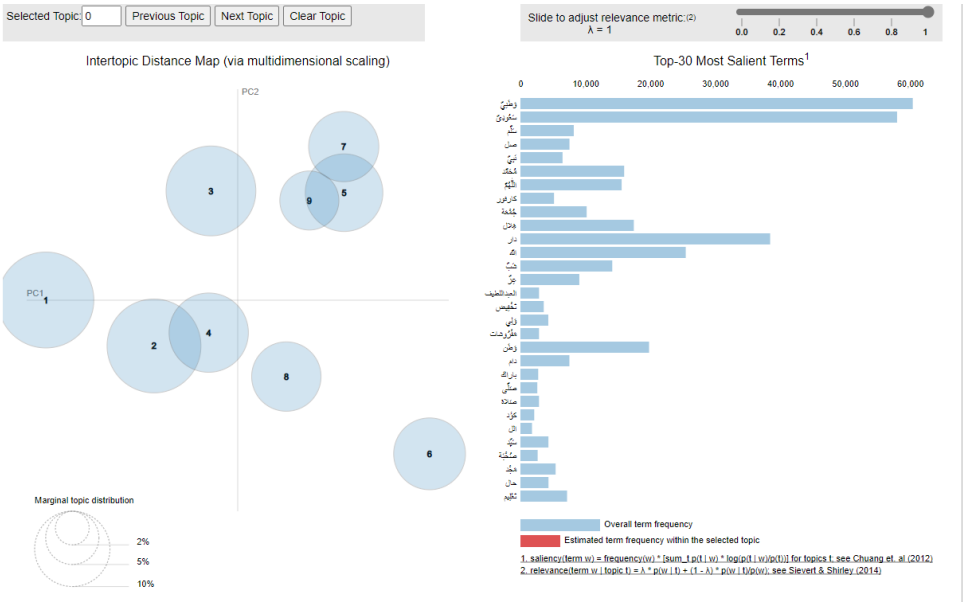

FIGURE 3.2: The main interface for the Visualisation

Also, you can notice from Fig (3.3), the red bubble on the left side represents the currently selected topic, which is Topic 3. The red bars on the right illustrate the estimated term frequencies of the top-30 most important keywords that comprise Topic 3.

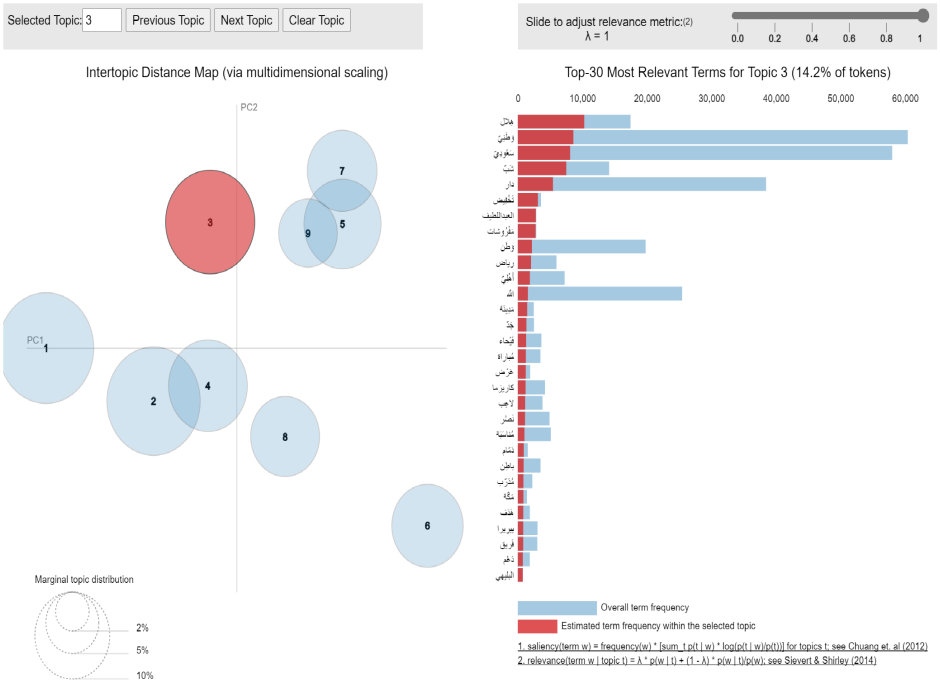

FIGURE 3.3: The final result for visualisation of each topic with their keywords

Chapter 4

Conclusion

4.1 Summary

The massive amount of digital information overgrowing over the internet due to computer science technology advancement has increased over the last two decades. Therefore, there is a need for a tool to search, organize and understand the enormous digital quantity of information. Topic modeling is a technique used to summarize, understand, and organize a vast amount of textual data.

In this project, we have investigated various issues and developed various methodologies for performing Topic Modelling for Arabic corpora. It may help the NLP researchers focus on the Saudi dialect.

It will help extract the most common trended topic on social media platforms. Also, we developed a Twitter harvester connected to NO SQL database, which may act as a repository for Social Media data records. We also illustrated many NLP techniques to manipulate Arabic texts and avoid challenges such as unwanted items.

4.2 Future work

This study can be used with Arabic Sentiment Analysis [12] and Author Attribution approaches[13]. The researchers can detect the specific pattern of the authors and link them to various topics. Also, It will help to cluster users who used to tweet about a specific field.

Bibliography

[1] wp content. Lda, 2016. URL http://nlpx.net/wp/wp-content/uploads/2016/01/LDA_image2.jpg. [Online; accessed August 27, 2021].

[2] Junseok Lee, Ji-Ho Kang, Sunghae Jun, Hyunwoong Lim, Dongsik Jang, and Sangsung Park. Ensemble modeling for sustainable technology transfer. Sustainability, 10:2278, 07 2018. doi: 10.3390/su10072278. URL http://dx.doi.org/10.3390/su10072278.

[3] Joshua Roesslein. Tweepy: Twitter for python! URL: https://github.com/tweepy/tweepy, 2021.

[4] Apache couchdb. URL:https://couchdb.apache.org/, 2021.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. J. Mach. Learn. Res., 3(null):993–1022, mar 2003. ISSN 1532-4435.

[6] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 3(2), 2011.

[7] Mohammad Al-Fetyani. Maha Processing Library, 9 2021. URL https://github.com/TRoboto/Maha.

[8] Steven Bird, Ewan Klein, and Edward Loper. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.", 2009. URL https://www.nltk.org/.

[9] Tarek BAZINE Mohamed Taher Alrefaie. Largest list of arabic stop words on github. https://github.com/mohataher/arabic-stop-words. Accessed: 2021-09-30.

[10] Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. CAMeL tools: An open source python toolkit for Arabic natural language processing. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 7022–7032, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.868.

[11] Carson Sievert and Kenneth Shirley. Ldavis: A method for visualizing and interpreting topics. 06 2014. doi: 10.13140/2.1.1394.3043. URL http://dx.doi.org/10.13140/2.1.1394.3043.

[12] Jingyi Ye, Xiaojun Jing, and Jia Li. Sentiment Analysis Using Modified LDA, pages 205–212. 01 2018. ISBN 978-981-10-7520-9. doi: 10.1007/978-981-10-7521-6_25. URL http://dx.doi.org/10.1007/978-981-10-7521-6_25.

[13] Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. Authorship attribution with latent dirichlet allocation. pages 181–189, 01 2011.