PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

SAAD DAHLEB BLIDA 01 UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

MASTER'S INTELLIGENT SYSTEMS ENGINEERING

NATURAL LANGUAGE PROCESSING

REPORT

CREATING AN ARABIC FAKE NEWS DETECTION DATASET USING WEB-SCRAPING

Made by :

ABDELATIF MEKRI

NAHLA YASMINE MIHOUBI

Academic year : 2023-2024

Contents

1. Introduction

In an era inundated with vast volumes of digital content, discerning fact from fiction has become an increasingly daunting challenge. With the proliferation of misinformation and disinformation, the need for robust mechanisms to detect and combat fake news has never been more pressing. Against this backdrop, the project at hand addresses this critical concern by constructing a comprehensive dataset for detecting fake news in Arabic.

2. Objective

The overarching objective of this project is to construct a dataset for the detection of fake news within the Arabic-speaking context. Beyond mere identification, the dataset aims to enable categorization of articles based on their thematic domains. By delineating between fake and authentic news stories and categorizing them by topic, the project seeks to offer a nuanced perspective on the dissemination of fake news across diverse subjects.

The task entails a multifaceted approach to dataset construction, underpinned by meticulous data gathering and systematic categorization efforts.

In summary, this project is a collaborative effort aimed at combating the scourge of misinformation through data-driven insights. By constructing a dataset filled with contextual relevance and thematic diversity, we endeavor to equip stakeholders with the tools necessary to navigate the intricate landscape of news veracity in the Arabic-speaking world.

3. Implementation

Given the variation in web architecture and implementation across different websites, and in pursuit of achieving optimal results, scraping each website individually was a necessary step. Moreover, there are instances where it was advantageous to split a single website into distinct parts for more efficient scraping.

To ensure a diverse range of topics and sources, we have curated a selection of websites for scraping, in addition of the ones suggested by the supervisor, each offering unique perspectives and content. These include:

-Misbar (https://misbar.com/)

-BBC Arabic (www.bbc.com/arabic/topics)

-No Rumors (http://norumors.net/)

-Verify-Sy (https://verify-sy.com/)

-Fatabyyano (https://fatabyyano.net/)

-CNN Arabia (https://arabic.cnn.com/)

-Aljazeera (https://www.aljazeera.net/)

-Matsda2ch (https://matsda2sh.com)

We employed Python programming language along with the Selenium library and Beautiful soup for web scraping. Selenium is a powerful tool for automating web browsers, making it suitable for dynamic web pages.

In the process of setting Up the Environment We installed the necessary Python libraries, including Selenium and CSV, and configured the Chrome WebDriver for browser automation.

3.1. CNN Arabia

CNN Arabia (https://arabic.cnn.com/) is a real news reliable website , therefore the aim of scrapping this website is to get a good amount of real news to balance out the existance of sevral fake news carrying websites .

3.1.1. Methodology:

Navigating to the Website

Using Selenium, we automated the process of opening the CNN sites.

```
urls_categories = [
    ('https://arabic.cnn.com/middle-east', 'middle-east'),
    ('https://arabic.cnn.com/travel', 'travel'),
    ('https://arabic.cnn.com/sport','sport'),
    ('https://arabic.cnn.com/science-and-health', 'science-and-health'),
    ('https://arabic.cnn.com/entertainment', 'entertainment'),
    ('https://arabic.cnn.com/style', 'style'),
    ('https://arabic.cnn.com/world', 'world')
]
```

Due to the site's architecture , we were able to use the distinct URLs and use them for the extraction of topics or as named here categories

Iterating Through Categories and Loading More Articles: The dynamique nature of the website consitisting of dynamically generating new content after every scroll , we aimed to implement a simulation of the human scrolling for a number of times in order to get some sort of a nice amount of scrapped articles for each category as follows :

```
for _ in range(30):
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    time.sleep(2)  # Adjust sleep time as needed
```

So that for each of the predefined categories we get the same treatment .

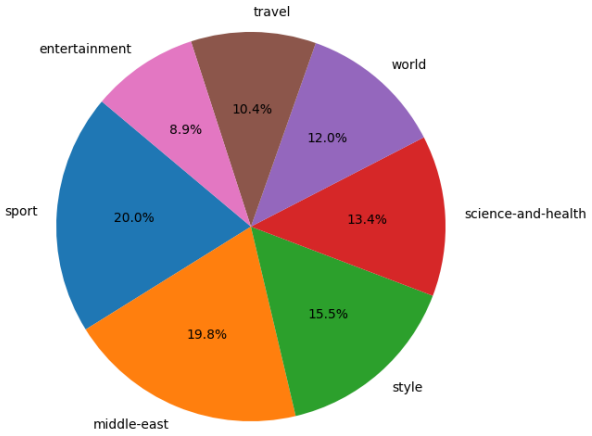
Extracting Article Information: For each article, we collected the title, publication date, URL, and content. We handled cases where the content was not available gracefully.

The extraction of the attributes was from the article container on the main page and on the artile page itself opened in the new tab (case of the artile_content) , as for the label it was set by default to real since all the news on this website is real .

Saving Data to CSV: We saved the extracted data to a CSV file for further analysis. Each row in the CSV file represents an article, with columns for category, title, date, URL, and content.
The saving progress was done after each scrapping all the articles of every category (resp Topic) allowing the scrappignof the next one till reaching the end of the URLs .

3.1.2. Results

We run the scrolling simulation 30 times in order to get a finer data , we can see on the piechart the distribution of the articles topics in this part of the dataset :



3.2. Misbar

Misbar (https://misbar.com/) is a real and fake news checking website ,but with a majority of fake news existing among the different pages and section of it , therefore the aim of scrapping this website is to get a good amount of fake news with topics and the level of fakeness of each article .

3.2.1. Methodology:

In the Misbar website we find two main links that are beneficial for the scraping operation , the [https://misbar.com/editorial] and [https://misbar.com/factcheck]

Navigating to the Website

Using Selenium, we automated the process of opening the Misbar site.

```
# Open the webpage
driver.get('https://misbar.com/editorial')
```

Due to the site's architecture , we were able to use the unique URL to the editorial page and use for the extraction of topics or as named here categories

Iterating Through Categories and Loading More Articles: The dynamique nature of the website consitisting of giving the user the option of viewing 14 possible articles of the same category at the same time and getting more of the content by clicking on the 'Show-more' button , we aimed to implement a simulation of the human scrolling for a number of times for each category as follows :

```
# Iterate through each category
for category_element in category_elements :
    # Click on the category element
    category_element.click()

    # Wait until the page finishes loading
    WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CSS_SELECTOR, 'div.articles_card--container')))

    # Initialize counter for tracking containers
    container_counter = 0

    # Initialize set to keep track of URLs already scraped
    scraped_urls = set()
            # Loop to click "Load more" button 10 times
    for _ in range(40): #for more data make it bigger
```

So that for each of the category found in the <li> element we get the same treatment .

Extracting Article Information: For each article, we collected the title, publication date, URL, and content. We handled cases where the content was not available or some sort of error accured during the process .
Finding News Articles: Using a CSS selector, we locate the container holding each news article.
Extracting Information: For each article, we extract its URL, title, publication date, and category name (if available).
Content Extraction: We try to extract the article's content, handling exceptions if content is not available.
Switching Tabs: After extracting data from an article, we close its tab and return to the main page.

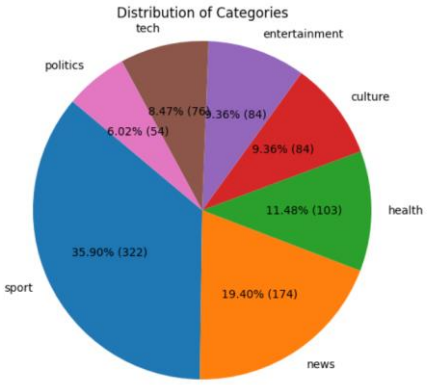
Saving Data to CSV: We saved the extracted data to a CSV file for further analysis. Each row in the CSV file represents an article, with columns for category, title, date, URL, and content.

The only difference between the two approches for the two subwebsites is that for [https://misbar.com/factcheck] is we saved the fake news and the correction of fakeness for each one (if available) .

3.2.2. Results

	Category	Title	Label	Date	URL	Content
count	896	896	896	896	896	896
unique	7	725	1	566	875	830
top	sport	مضلل	fake	27 نوفمبر 2022	Content or Correction not available	... تحقق "مسبار" من الادعاء وتبين أنه مضلل، إذ إن
freq	322	156	896	7	19	4

As we can see , the distribution of the various topics on the website visually presented in a piechart .



3.3. AlJazeera

3.3.1. Overview

Scraping Aspects: Extracting articles from various categories such as politics, business, culture, etc.

Unique Features: Al Jazeera offers a diverse range of news categories covering global events, with in-depth analysis and reporting.

3.3.2. Scraping Process

Utilized BeautifulSoup and requests libraries to scrape the Al Jazeera website.

Implemented functions to extract article titles, links, and full text from each category page.

```python
def scrape_articles_from_category(category_url, category):
    response = requests.get(category_url)
    soup = BeautifulSoup(response.text, 'html.parser')
    articles = soup.find_all('a', class_='u-clickable-card__link')

    data = []
    for article in articles:
        title = article.find('span').text.strip()
        article_url = article['href']
        full_text = get_article_text(article_url)
        data.append({'Category': category, 'Title': title, 'Link': article_url, 'Text': full_text})

    return data
```

3.3.3. Results

Extracted articles from various categories such as politics, business, culture, etc.

Each article includes its title, category, link, and full text content.

As we can see the data obtained from this website are represented as :

```
df
```

	Category	Title	Link	Text
0	politics	إيران وإسرائيل.. ما بعد قصف القنصلية وخيارات الرد	/politics/2024/4/7/%d9%82%d8%b5%d9%81-%d8%a5%d...	في محاولة لاستقراء قصف إسرائيل لقنصلية إيران ب...
1	politics	الغارديان: على حلفاء إسرائيل أن "يقولوا لها "كفى	/politics/2024/4/6/%d8%a7%d9%84%d8%ba%d8%a7%d8...	قال الكاتب كريس ماكغريل -في مقال رأي بصحيفة ال...
2	politics	شاهد: آثار تدمير الدعم السريع للكنائس في أم درمان	/programs/2024/4/6/%d8%b4%d8%a7%d9%87%d8%af-%d...	نشرت المقاومة الشعبية بأم درمان مشاهد للدمار ا...
3	politics	فضيحة بتر أطراف أسرى فلسطينيين تكشف المستور في...	/politics/2024/4/6/%d9%81%d8%b6%d9%8a%d8%ad%d8...	نابلس- رسم الاحتلال الإسرائيلي بعدوانه على الأ...
4	politics	نيوزويك: شعبية بايدن تنخفض بشكل كارثي بسبب إسر...	/politics/2024/4/6/%d9%86%d9%8a%d9%88%d8%b2%d9...	قالت مجلة نيوزويك إن شعبية الرئيس جو بايدن في...
...	...	...	...	...
163	family	كيف تتعامل مع سلوك استخفاف شريك العمر بك؟	/family/2024/3/30/%d9%83%d9%8a%d9%81-%d8%aa%d8...	رغم أن شريك العمر (الزوج أو الزوجة) هو الشخص ا...

```
df.describe()
```

	Category	Title	Link	Text
count	168	168	168	168
unique	12	165	165	165
top	politics	دعوى أميركية ضد هواوي تتجه للمحاكم مطلع 2026	/ebusiness/2024/4/5/%D8%AF%D8%B9%D9%88%D9%89-%...	تتجه دعوى جنائية أقامتها وزارة العدل الأميركية...
freq	14	2	2	2

3.4. BBC Arabic

3.4.1. Overview

Scraping Aspects: Retrieving headlines, articles, and news categories from the BBC News website.

Unique Features: BBC News offers a wide range of news categories with a focus on both domestic and international news coverage.

3.4.2. Scraping Process

Utilized BeautifulSoup and requests libraries to scrape the BBC News website.

```python
import time
import pandas as pd
from selenium import webdriver
from bs4 import BeautifulSoup

def web_driver():
    options = webdriver.ChromeOptions()
    options.add_argument("--verbose")
    options.add_argument('--no-sandbox')
    options.add_argument('--headless')
    options.add_argument('--disable-gpu')
    options.add_argument("--window-size=1920,1200")
    options.add_argument('--disable-dev-shm-usage')
    driver = webdriver.Chrome(options=options)
    return driver
```

Implemented functions to extract headlines, article summaries, and categories.

```python
for article in articles:
    title = article.text.strip()
    link = article.find('a')['href']
    driver.get(link)  # Visit the article page to extract text
    time.sleep(2)  # Wait for the article page to load
    article_soup = BeautifulSoup(driver.page_source, 'html.parser')
    date_element = article_soup.find('time', class_='bbc-1eu2r82 e1mklfmt0')
    date = date_element.text.strip() if date_element else 'Date not found'
    text_paragraphs = article_soup.find_all('p', class_='bbc-1gjryo4 e17g058b0')
    text = ' '.join([p.text.strip() for p in text_paragraphs])
    data.append({'Title': title, 'Link': link, 'Date': date, 'Text': text})
```

3.4.3. Results

Retrieved headlines, article summaries, and categories from various sections of the BBC News website.

Each article includes its headline, summary, category, and link to the full article.

```
df_Economic
```

	Category	Title	Link	Date	Text
0	Economic	فيديو باندا و"الورواري".. بائعا فواكه يحرجان ...	https://www.bbc.com/arabic/articles/cmm3779yqq4o	Date not found	
1	Economic	ما أسباب ارتفاع معدل الانتحار بين الشباب الأمر...	https://www.bbc.com/arabic/articles/cv2rlg7vzm2o	قبل 7 ساعة	تحذير: يحتوي المقال على ** قصص وحقائق قد يجدها...
2	Economic	ماذا تعرف عن المليارديرة الفيتنامية التي حكم ع...	https://www.bbc.com/arabic/articles/c0v01zxw0vlo	أبريل/ نيسان 12 2024 ،09:20 GMT	تعد محاكمة خضعت لها ملياردیرة فيتنامية، الأكثر...
3	Economic	هل سيصبح 75 عاما سن التقاعد، وهل بات ذلك مستحي...	https://www.bbc.com/arabic/articles/c4nrg84r2m3o	أبريل/ نيسان 10 2024	بات الناس يعيشون عمرا أطول، والحياة أصبحت تزدا...

This process was done to each of the chosen topics to scrap on the BBC Arabic website , at the end the data was merged and cleaned in one file as follows :

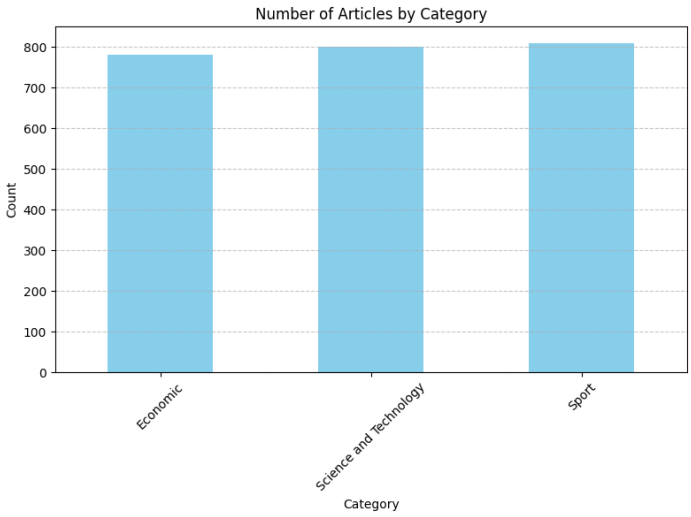
```python
# Concatenate the dataframes along the rows
df_merged = pd.concat([df_Science, df_Sport, df_Economic], ignore_index=True)

# Optionally, you can reset the index of the merged dataframe
df_merged.reset_index(drop=True, inplace=True)

summary_stats = df_merged.describe()
print(summary_stats)
```

The data summary :

	Category	Title	Link	Date	Text
count	2435	2435	2435	2435	2435
unique	3	2389	2390	884	2389
top	Sport	ما أسباب ارتفاع معدل الانتحار بين الشباب الأمر...	https://www.bbc.com/arabic/articles/cv2rlg7vzm2o	19 يناير/ كانون الثاني 2023	تحذير: ** يحتوي المقال على قصص وحقائق قد يجدها...
freq	819	2	2	17	2



3.5. VERIFY-SY

3.5.1. Overview

Scraping Aspects: Extracting news articles and categorizing them based on verification status.

Unique Features: VERIFY-SY focuses on verifying news articles and categorizing them as true, false, misleading, etc.

3.5.2. Scraping Process

Utilized BeautifulSoup and requests libraries to scrape the VERIFY-SY website.

Implemented functions to extract article titles, links, and verification statuses.

The architecture of the site allows to extract the category of fakeness depending on the url .

```
# List of URLs to scrape along with their corresponding category names
urls_categories = [
    ('https://verify-sy.com/all/18?page=1', 'كذب'),
    ('https://verify-sy.com/all/53?page=1', 'تضليل'),
    ('https://verify-sy.com/all/54?page=1', 'نظرية-المؤامرة'),
    ('https://verify-sy.com/all/55?page=1', 'كذب-باسم-العلم'),
    ('https://verify-sy.com/all/56?page=1', 'خطأ'),
    ('https://verify-sy.com/all/57?page=1', 'انحياز'),
    ('https://verify-sy.com/all/58?page=1', 'تلاعب-بالحقائق'),
    ('https://verify-sy.com/all/59?page=1', 'عنوان-مضلل'),
    ('https://verify-sy.com/all/60?page=1', 'سخرية'),
    ('https://verify-sy.com/all/61?page=1', 'خارج-السياق'),
    ('https://verify-sy.com/all/62?page=1', 'غير-مؤكد'),
    ('https://verify-sy.com/all/165?page=1', 'مؤكد'),
]
```

3.5.3. Results

Extracted news articles from VERIFY-SY along with their titles, links, and verification statuses.

Each article is categorized based on its verification status, providing valuable insights into the credibility of the news.

for echa article of each page of each url in the urls_categories , we focused on detecting the the article container and extracting the necessary attributes , title , url , label , content

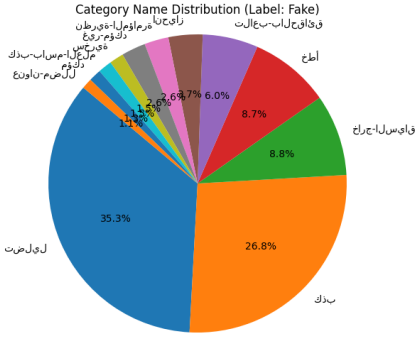
```
        # Output the articles for the current category
        data = []
        for article in news_articles:
            title = article.find_element(By.CSS_SELECTOR, 'h3.list-title-ca').text
            article_url = article.find_element(By.CSS_SELECTOR, 'a').get_attribute('href')

            # Get article content and date from URL
            article_content, article_date = get_article_info(article_url)

            if article_content is not None and article_date is not None:
                # Set the label to 'fake' by default
                label = 'fake'
                # For the URL 'https://verify-sy.com/all/165', set the label to 'real'
                if url == 'https://verify-sy.com/all/165':
                    label = 'real'

                # Append data to the list
                data.append([title, label, category_name, article_content, article_date])
```

The piechart shows the distribution of each category of the fake news label



After extraction and cleaning of the data , here is a head of the obtained data .

```
                                article_content           article_date
0                                article_content           article_date
1  \n\n2024 نيسان 13 السبت   تشرت حسابات بمواقع التواصل الاجتماعي، خبرا...
2  \n\n2024 نيسان 02 الثلاثاء ادعت مواقع وحسابات على منصات التواصل الاجت...
3  \n\n2024 نيسان 01 الاثنين  ادعت حسابات على منصات التواصل الاجتماعي أن...
4  \n\n2024 آذار 20 الأربعاء  ادعت شبكات إخبارية وحسابات على منصات التوا...

Shape of the cleaned data: (1947, 5)
```

Note that the website doesn’t have topics on it , we had to use a model in order to detect the topic of each of the articles content

```
# Load the CSV file into a DataFrame with specified column names
df = pd.read_csv('verify-sy-bs-RAW-data.csv', names=['title', 'label', 'category_name', 'article_content', 'article_date'])
# Initialize the topic classification pipeline
topic_pipe = pipeline("text-classification", model="Ammar-alhaj-ali/arabic-MARBERT-news-article-classification")

# Function to classify the text into one of the predefined topics
def classify_topic(text):
    # Split the text into smaller chunks
    max_chunk_length = 512  # Maximum sequence length supported by the model
    chunks = [text[:max_chunk_length]]
    ...

# Add a new column "TOPIC" to the DataFrame and populate it with topics predicted from the article content
df['TOPIC'] = df['article_content'].apply(classify_topic)

# Save the DataFrame with the new column back to a CSV file with the same column names
df.to_csv('verify-sy-bs-data-with-topics.csv', index=False, header=['title', 'label', 'category_name', 'article_content', 'article_date','TOPIC'])
```

The data shape after adding topics :

```
# Display the shape of the cleaned data
print("\nShape of the cleaned data:", df.shape)

Shape of the cleaned data: (1947, 6)
```

3.6. Fatabyyano

3.6.1. Overview

Scraping Aspects: Scraping Arabic news articles and categorizing them based on labels.

Unique Features: Fatabayyano provides Arabic-language news articles with labels indicating their authenticity.

3.6.2. Scraping Process

Utilized BeautifulSoup and Selenium with Chrome WebDriver to scrape the Fatabayyano website.

Implemented functions to extract article titles, links, labels, and dates.

The data is noted to be all fake , so it was not a problem labeling it after .

```
def scrape_fatabyyano(num_pages=2):
    driver = web_driver()
    base_url = 'https://fatabyyano.net/'
    data = []

    for page in range(1, num_pages + 1):
        if page > 1:
            url = f'https://fatabyyano.net/page/{page}/'
        else:
            url = base_url
```

3.6.3. Data Obtained

Extracted articles along with their titles, links, labels, and dates.

```
Label
120        زائف
2          زائف جزئي
324        مضلل
1          مفبرك
dtype: int64
```

	Title	Link	Label	Date	Fake News	Fact Ne
0	لا، لم تحظر اليابان لقاحات كوفيد 19. ولا صحة ل...	https://fatabyyano.net/371617829383-2/	زائف	2024-04-12T01:59:53+03:00	عاجل_الان: اليابان# تحظر لقاحات كورونا بعد زيا...	حث المباشر\n موقع "وزارة صحة والعمل و
1	هذه الصورة تعود لاستعراض عسكري في كوريا الشمال...	https://fatabyyano.net/هذه-الصورة-تعود-لاستعرا...	مضلل	2024-04-12T01:44:52+03:00	not found	not fou

3.7. Matsda2ch

3.7.1. Overview

This platform is an independent specialized platform in verification and news fact-checking. Its aim is to combat the torrent of false or misleading news, whether intentionally disseminated for political or ideological bias, or incidentally for the purpose of attracting audiences to the pages or websites publishing rumors.

3.7.2. Scrapping P

- Extract Links from Page: This function extracts the URLs of individual news articles from a given page. It uses the requests library to retrieve the HTML content of the page and BeautifulSoup for parsing the HTML and extracting relevant information. Specifically, it targets the elements with the class "media", which encapsulate the links to the articles.

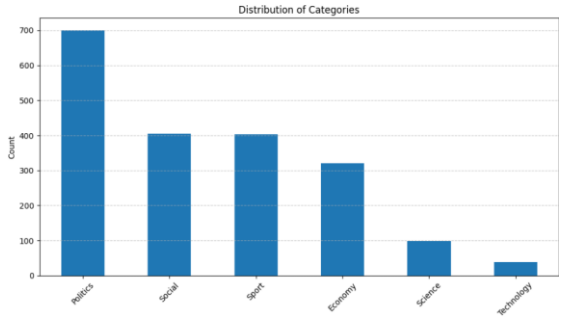
- Scrape Classes from Link: This function scrapes specific classes from the content of a news article page. It retrieves the title, publication time, category, fake news content, and fact news content. Similar to the previous function, it uses requests and BeautifulSoup to fetch and parse the HTML content of the article page.

3.7.3. Results

The statisitics and shape of the obtained data :

	Title	Time	Category	Link	Fake News	Fact News
count	1968	1968	1968	1968	1968	1968
unique	1968	1027	6	1968	1806	1922
top	مو سيلفا" من هو المتهم بترويج أخبار" ... كاذبة عن	May. 19, 2021	Politics	https://matsda2sh.com/details/13210/-%D9%85%D9...	[]	[]
freq	1	22	700	1	150	38

```
Category
Politics      700
Social        405
Sport         404
Economy       321
Science        99
Technology     39
```



As noticed thehighest category destribution in this scrapping process is politics while the least of it istechnology .

4. Retrieving the Data:

The process of data collection was done on different work enviroments and different machines which would not be idea to settle on an exact syntax of how the different attributes are set to be and so on .

At the end of the scrapping process , some of the output files run through the cleaning phase ensuring the reliability of the obtained data , while others were kept in their original output forms , some of the attributes were missing from some of the tables and was not an issue of the scrapping tool but the missing informations on the websites .

4.1. Handeling the missing information

We aimed to use two different models at the beggening of our merging tool in order to cover the probelems mentioned earlier .

4.1.1. The FATABAYYANO missing topics

We tend to use a predefined tuned model under the name "Ammar-alhaj-ali/arabic-MARBERT-news-article-classification" in orrder to set the necessary topics , the model is well tuned to give accurate classification for arabic text forms , and therefore was the best option to go with .

```
# Initialize the topic classification pipeline
topic_pipe = pipeline("text-classification", model="Ammar-alhaj-ali/arabic-MARBERT-news-article-classification")

# Function to classify the text into one of the predefined topics
def classify_topic(text):
    # Split the text into smaller chunks
    max_chunk_length = 512  # Maximum sequence length supported by the model
    chunks = [text[:max_chunk_length]]
```

4.1.2. The Misbar-Editorial missing Labelings

In this case of editorials the form of text extracted had enough of information to tell if the article in hands is real or fake , but it was not yet labeled or tagged by the Misbar website as it was done to the Misbar-Factcheck page .

Using the previous model was not the right call , as this model didn’t have the ability to handle the detection of real/fake aspect or a sentiment analysis of it ;

We opted to use the ‘asafaya/bert-base-arabic' model as it had a great capacity of distincting between the real and fake type of news .

The model also had a limitation to the chunk sizes andtherefore the text was split to multiple chunks for an accurate labeling .

```python
# Initialize the real/fake classification pipeline for Arabic
fake_pipe = pipeline("text-classification", model="asafaya/bert-base-arabic")

# Function to classify the text into real or fake
def classify_fakeness(text):
    # Split the text into smaller chunks
    max_chunk_length = 512  # Maximum sequence length supported by the model
    chunks = [text[i:i+max_chunk_length] for i in range(0, len(text), max_chunk_length)]

    # Predictions
    fake_statuses = []
```

3.8. No Rumors

Due to a fatal error in the website , the operation of scrapping was not ;possible ; therefore there are scrapping mecanisms implemented and no data related to this website.

4.2. Handeling the CSV files

The final files of the scrapping were 8 files with different columns names , significations and values , therefore a pretreatment would come along way in order to have usable data .

4.2.1. Setting annotation

To handle the csv files , this step was crucial , or the work of merging would have been impossible .

```python
# Add 'Origine' column to each DataFrame
df1['Origine'] = 'Aljazeera'
df2['Origine'] = 'BBC'
df3['Origine'] = 'CNN'
df4['Origine'] = 'Fatabyyano'
df5['Origine'] = 'Matsd24'
df6['Origine'] = 'MISBAR'
df7['Origine'] = 'MISBAR'
df8['Origine'] = 'VerifySy'

# Rename columns as per the required attributes
df1.rename(columns={'Category': 'Topic', 'Text': 'Article_content','Link':'Article_url'}, inplace=True)
df2.rename(columns={'Category': 'Topic', 'Text': 'Article_content','Link':'Article_url','Date':'Article_date'}, inplace=True)
df3.rename(columns={'title': 'Article_date', 'label': 'Topic', 'article_date': 'Article_url', 'category_name': 'Article_content','article_url':'Label','article_content':'Article_correction'}, inplace=True)
df4.rename(columns={'Date': 'Article_date', 'Fake News': 'Article_content','TOPIC':'Topic','Link':'Article_url','Fact News':'Article_correction'}, inplace=True)
df5.rename(columns={'Title': 'Title', 'Time': 'Article_date', 'Category': 'Topic','Link':'Article_url', 'Fake News': 'Article_content','Fact News':'Article_correction'}, inplace=True)
df6.rename(columns={'Date': 'Article_date','Title':'Title', 'Content': 'Article_content','URL':'Article_url','Label':'Label','Category':'Topic'}, inplace=True)
df7.rename(columns={'Category': 'Topic','Title': 'Title','Label':'Label','Date': 'Article_date','URL':'Article_url', 'Content': 'Article_content'}, inplace=True)
df8.rename(columns={'TOPIC': 'Topic', 'article_date': 'Article_date', 'article_content': 'Article_content','title': 'Title','label':'Label','category_name':'Article_correction'}, inplace=True)
```

4.2.2. Value checking

This operation was done in the number of CSV files .

```
DF2 - BBC

df2['Label'] = 'real'
df2['Article_correction'] = ''
# DataFrame 2 (df2)
df2.head()

✓ 0.0s
```

4.3. Merging the data

After the previous process , we only had to merge the data accordingly .

```python
# Concatenate DataFrames
merged_df = pd.concat([df1[['Title', 'Label', 'Topic', 'Origine', 'Article_date', 'Article_content', 'Article_correction']],
                       df2[['Title', 'Label', 'Topic', 'Origine', 'Article_date', 'Article_content', 'Article_correction']],
                       df3[['Title', 'Label', 'Topic', 'Origine', 'Article_date', 'Article_content', 'Article_correction']],
                       df4[['Title', 'Label', 'Topic', 'Origine', 'Article_date', 'Article_content', 'Article_correction']],
                       df5[['Title', 'Label', 'Topic', 'Origine', 'Article_date', 'Article_content', 'Article_correction']],
                       df6[['Title', 'Label', 'Topic', 'Origine', 'Article_date', 'Article_content', 'Article_correction']],
                       df7[['Title', 'Label', 'Topic', 'Origine', 'Article_date', 'Article_content', 'Article_correction']],
                       df8[['Title', 'Label', 'Topic', 'Origine', 'Article_date', 'Article_content', 'Article_correction']]],
                      ignore_index=True)

# Write merged DataFrame to CSV
merged_df.to_csv('ARABIC-NEWS-CLASSIFICATION-MERGED.csv', index=False)
```

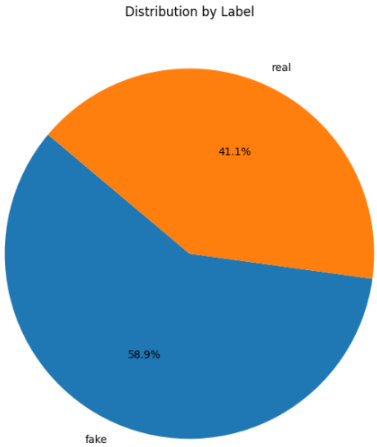
4.4. The results :

The form of the obtained data :

	Title	Label	Topic	Origine	Article_date	Article_content	Article_correction
0	54 عاما في القضاء.. نهاية حزينة لمعركة مقدسي ح...	real	politics	Aljazeera		القدس المحتلة- لن يتمكن المقدسي سامي درويش في ...	
1	خبراء إيرانيون يشرحون لماذا لا تستعجل طهران با...	real	politics	Aljazeera		طهران- منذ الهجوم الإسرائيلي على القنصلية الإي...	
2	فلسطيني يقود فريق خبراء بإيطاليا لبناء قضية إب...	real	politics	Aljazeera		غادر المحامي الفلسطيني راجي صوراني قطاع غزة رف...	
3	مستوطنون يهاجمون قرى نابلس بحماية من جنود الاح...	real	politics	Aljazeera		نابلس- لليوم الثاني على التوالي، تتعرض قرية دو...	
4	جدل قانوني حول ترشح جاكوب زوما لانتخابات جنوب ...	real	politics	Aljazeera		بريتوريا- تقدمت اللجنة المستقلة للانتخابات في ...	

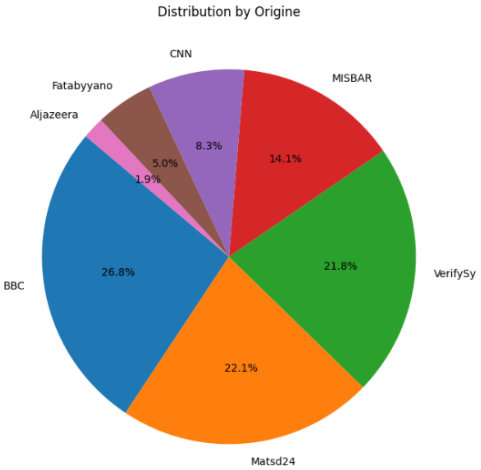
The data destribution by label and by origine :

41% of the obtained data accross the websites is real , while the rest of it is represented as fake .



The difference of articles distribution by origine website , this also would varry on how many pages were to be scrapped and the used methods of each scrapper .

```
Label
fake    5257
real    3662
dtype: int64
########################
Origine
Aljazeera      168
BBC           2389
CNN            741
Fatabyyano     446
MISBAR        1261
Matsd24       1968
VerifySy      1946
dtype: int64
```



The data and the implimented scrappers can be found on [THE REPOSITORY](#) .