Graph Mining

Muzammil Mushtaq

Task 1 - The Story

Deutsche Welle (DW) is a German international broadcaster that operates television, radio, and online services.

- **News Coverage**: Deutsche Welle covers a wide range of news topics, including international affairs, politics, business, science, technology, culture, and sports. The news coverage aims to provide a balanced and comprehensive view of global events.
- **Global Perspectives**: Deutsche Welle is known for offering diverse perspectives on global issues. The organization aims to present news and information from an international viewpoint, fostering a better understanding of different cultures and opinions.

In this project, our goal is to analyze various features of DW.com. To achieve this, we initially collect a dataset comprising news articles published from October 1, 2023, to December 31, 2023. The key attributes extracted for each article include:

- *Link*: The actual URL of the article.
- *Publication_Date*: The date of publication of the article.
- *Category*: The classification of the article, such as Politics, Conflict, Science, Culture, etc.
- *Region*: The geographical region to which the news is related, such as US, Germany, Europe, Asia, India, China, etc.
- *Title*: The headline of the article.
- *Text*: The complete text content of the article.
- *Summary*: A concise summary of the article.
- *Sub-heading*: If the article is written, the sub-headings used within it.
- *Related-Topics*: The related topics provided at the end of each article, assisting readers in exploring further related subjects.

Tasks

The primary objective of this project is to identify and examine the correlation among the *Region*, *Category*, and *Related Topics*. Our hypothesis posits that DW highlights genuine global issues across various categories and regions without bias. This aligns with DW's overarching mission of disseminating news from around the world to every individual, emphasizing their commitment to providing comprehensive coverage without discrimination.

In substantiating our hypothesis, we postulated that critical roles should be played by features such as *Region*, *Category*, and *Related Topics*. Our approach involves employing Graph Mining techniques, utilizing the NetworkX packages. This methodology allows us to extract information regarding various graph properties, identify central measures, and perform community clustering.

In [1]:
```
'''                          Read The News Articles Dataset of DW
'''
import pandas as pd
df = pd.read_excel('dw_news_output.xlsx')
print ('Number of Articles avaialable on given Dataset :',
len(df['Link']))
df.head()
```

```
Number of Articles avaialable on given Dataset : 3538
```

Out[1]:

	Link	Publication_Date	Category	Region	Title	Summary
0	https://www.dw.com/en/israel-evacuates-city-ne...	2023-10-20	Conflicts	Israel	Israel evacuates city near Lebanon border	\nIsrael is planning to evacuate the city of K...
1	https://www.dw.com/en/israel-to-let-aid-into-g...	2023-10-20	Conflicts	Middle East	Israel to let aid into Gaza, but bombing conti...	\nIsrael has confirmed that it will allow a li...
2	https://www.dw.com/en/hundreds-arrested-for-vi...	2023-10-20	Conflicts	Germany	Hundreds arrested for violating protest ban in...	\nIn Berlin, pro-Palestinian demonstrations ha...
3	https://www.dw.com/en/hamas-learning-about-dro...	2023-10-20	Politics	Israel	Hamas: Learning about drone warfare from the w...	\nDrones have long been part of modern warfare...
4	https://www.dw.com/en/protesters-heckle-dutch-...	2023-10-20	Politics	South Africa	Protesters heckle Dutch royals at Cape Town sl...	\nSecurity guards held back protesters as they...

In [3]:
```
'''             Data type of the Following Attributes
'''
df.dtypes
```

Out[3]:
```
Link                        object
Publication_Date    datetime64[ns]
Category                    object
Region                      object
Title                       object
Summary                     object
Text                        object
SubHeadings                 object
Related_topics              object
dtype: object
```

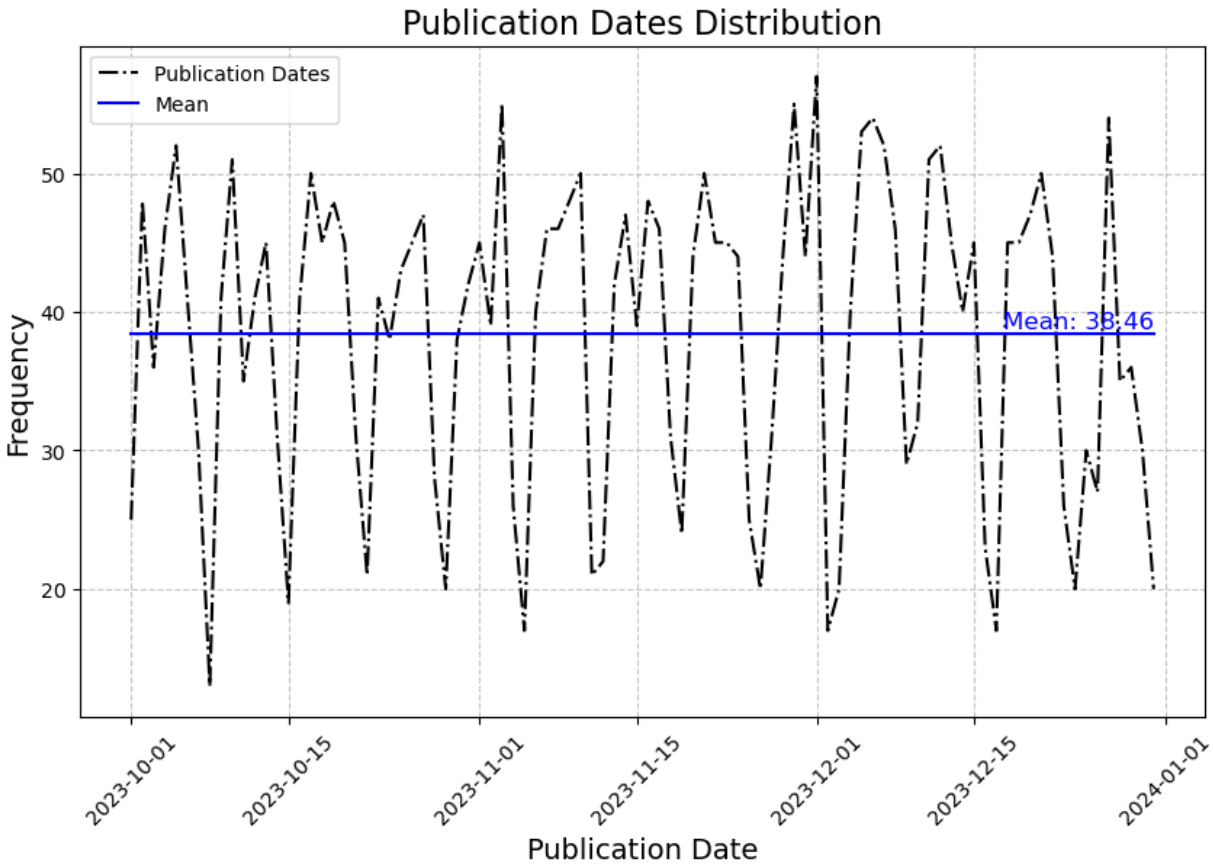
Task 3

Initial Data Analysis: In this phase, we aim to visualize fundamental plots to glean insights from the dataset's internal dynamics.

- *Time Series*: Illustrates the daily count of published articles within the specified timeframe.
- *Category Bar Plot*: Displays the top 10 categories with the highest article count.
- *Region Bar Plot*: Exhibits the top 10 regions with the highest frequency of articles.
- *Region-Category Bar Plot*: Highlights the most frequent combinations of regions and their corresponding categories.
- *Heat Map*: Depicts the top 20 counts of related topics over the given time period.

In [3]:

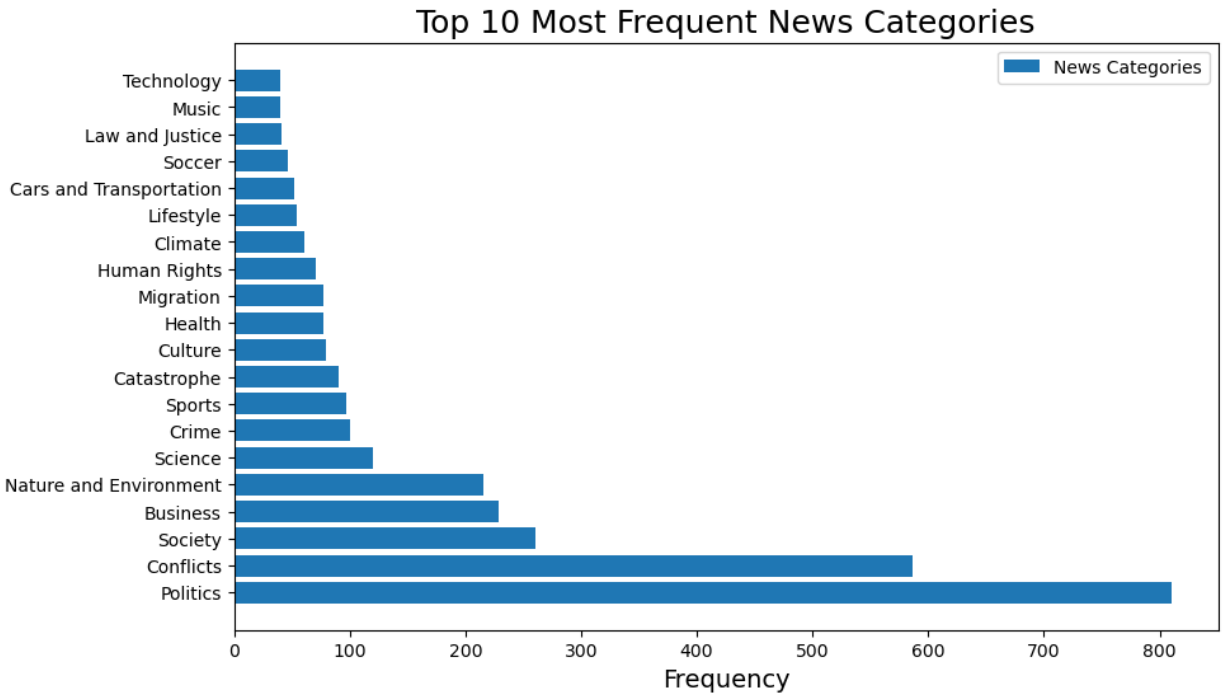
```
from matplotlib import pyplot as plt
import numpy as np
%matplotlib inline

dates = df['Publication_Date'].value_counts().sort_index()
plt.figure(figsize=(10, 6))
plt.plot(dates.index, dates, 'k-.', label='Publication Dates')
plt.plot(dates.index, [np.mean(dates)] * len(dates), 'b-', label='Mean')

plt.title('Publication Dates Distribution', fontsize=16)
plt.xlabel('Publication Date', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.7)

plt.text(dates.index[-1], np.mean(dates), f'Mean: {np.mean(dates):.2f}',
ha='right', va='bottom', color='blue', fontsize=12)
plt.legend()
plt.show()
```



Discussion:

The daily count of published articles fluctuates, ranging from a minimum of 5 to a maximum of 55. On average, there are 39 articles published each day. The patterns of highs and lows in the number of articles exhibit weekly variations.

In [5]:
```
plt.figure(figsize=(10, 6))
categories = df['Category'].value_counts().head(20)
plt.barh(categories.index, categories, label='News Categories')
plt.xlabel('Frequency', fontsize=14)
plt.title('Top 10 Most Frequent News Categories', fontsize=18)
plt.legend()
plt.show()
```

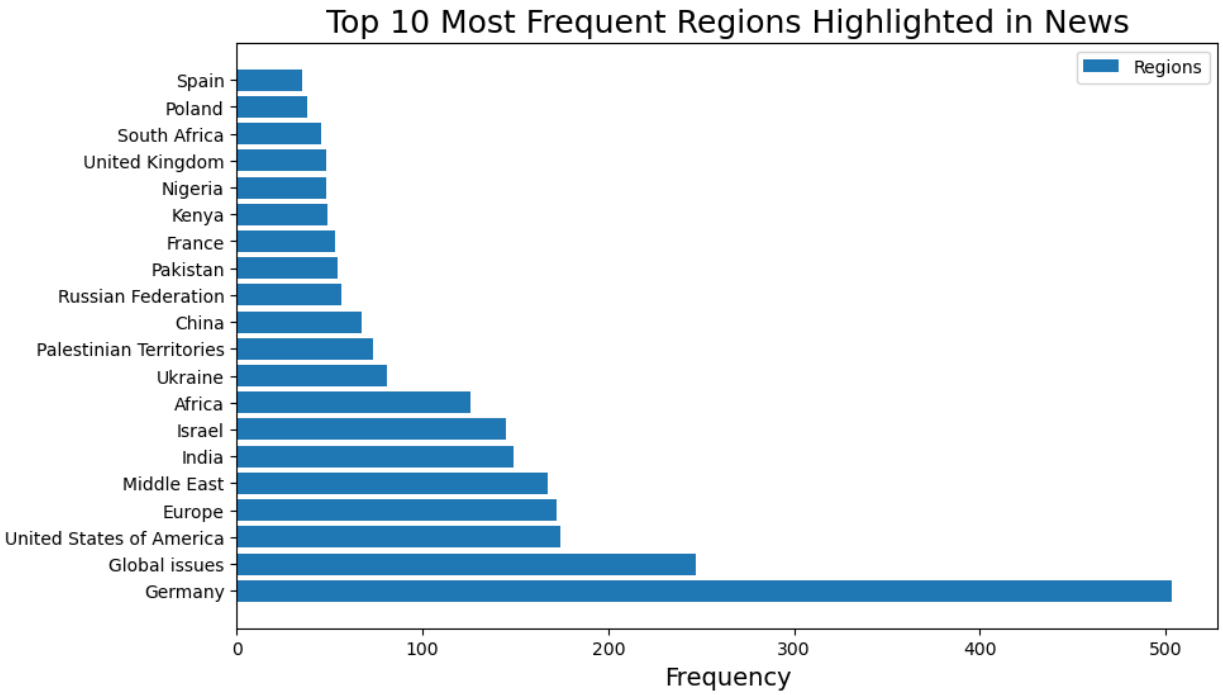


Discussion:

Among the 3538 articles, the predominant categories are Politics and Conflicts, collectively constituting over 40% of all article classifications.

In [6]:

```
plt.figure(figsize=(10, 6))
regions = df['Region'].value_counts().head(20)
plt.barh(regions.index, regions, label='Regions')
plt.xlabel('Frequency', fontsize=14)
plt.title('Top 10 Most Frequent Regions Highlighted in News', fontsize=18)
plt.legend()
plt.show()
```



Discussion:

Germany stands out as the most frequently mentioned region in news articles, encompassing 12% of all mentioned regions. However, other regions are predominantly associated with specific periods. For instance, the current news landscape is marked by the Israel and Palestine War.

In [7]:

```
# Find the counts of each group
group_counts = df.groupby(['Region',
'Category']).size().reset_index(name='Count')

# Sort by count in descending order
sorted_counts = group_counts.sort_values(by='Count', ascending=False)
print (len(sorted_counts['Count']))
# Select the top 20 rows
top_20 = sorted_counts.head(20)
top_20
```

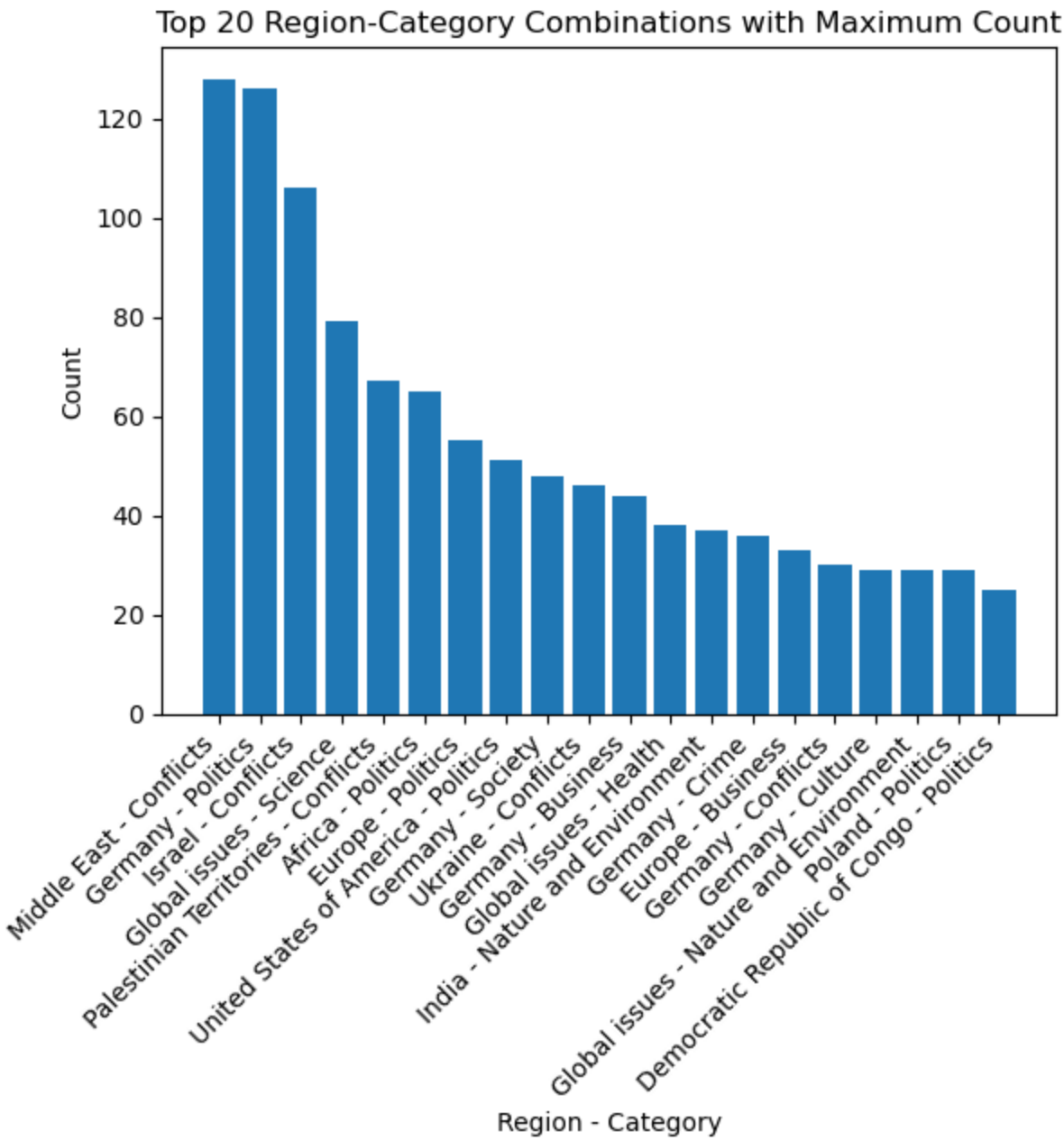
```
1056
```

Out[7]:

	Region	Category	Count
599	Middle East	Conflicts	128
311	Germany	Politics	126
453	Israel	Conflicts	106
353	Global issues	Science	79
716	Palestinian Territories	Conflicts	67
25	Africa	Politics	65
252	Europe	Politics	55
1020	United States of America	Politics	51
316	Germany	Society	48
968	Ukraine	Conflicts	46
288	Germany	Business	44
344	Global issues	Health	38
418	India	Nature and Environment	37
294	Germany	Crime	36
237	Europe	Business	33
292	Germany	Conflicts	30
295	Germany	Culture	29
350	Global issues	Nature and Environment	29
741	Poland	Politics	29
210	Democratic Republic of Congo	Politics	25

In [8]:

```
# Find the counts of each group
group_counts = df.groupby(['Region',
'Category']).size().reset_index(name='Count')
sorted_counts = group_counts.sort_values(by='Count', ascending=False)
top_20 = sorted_counts.head(20)

# Plot 'Region' and 'Category' with the maximum count
plt.bar(top_20['Region'] + ' - ' + top_20['Category'], top_20['Count'])
plt.xlabel('Region - Category')
plt.ylabel('Count')
plt.title('Top 20 Region-Category Combinations with Maximum Count')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better
readability
plt.show()
```



Discussion

In our analysis, we grouped regions with their respective categories, revealing that *Middle East - Conflicts* is the most prevalent theme in news articles. Following closely is *Germany - Politics* as the second most prominent category. Subsequent distributions show a gradual decrease.

In [4]:

```
'''                Reformat the *Related_topics* from string to List
'''
import re
def parse_json(x):
    match = re.search(r'\[\'([^\']*)\'\]', x)
    if match:
        my_list = [str(item.strip()) for item in
```

```
match.group(1).split(',')]
        return my_list

df['Related_topics_list'] = df['Related_topics'].apply(parse_json)
print (df['Related_topics_list'])
```

```
0             [Israel, Lebanon, Hezbollah, Israel at war]
1       [Antonio Guterres, Hosni Mubarak, Egypt, Israe...
2                          [Berlin, Israel, Israel at war]
3       [Gaza, Palestinian territories, Hamas, Israel ...
4                                                    None
                              ...
3533    [Floods in Germany, Arctic, COP27: Everything ...
3534                                [Máxima\n, Argentina]
3535    [Uyghur community, BRICS, Philippines, Taiwan,...
3536                            [LGBTQ+ rights in Africa]
3537                                [Máxima\n, Argentina]
Name: Related_topics_list, Length: 3538, dtype: object
```

In [5]:

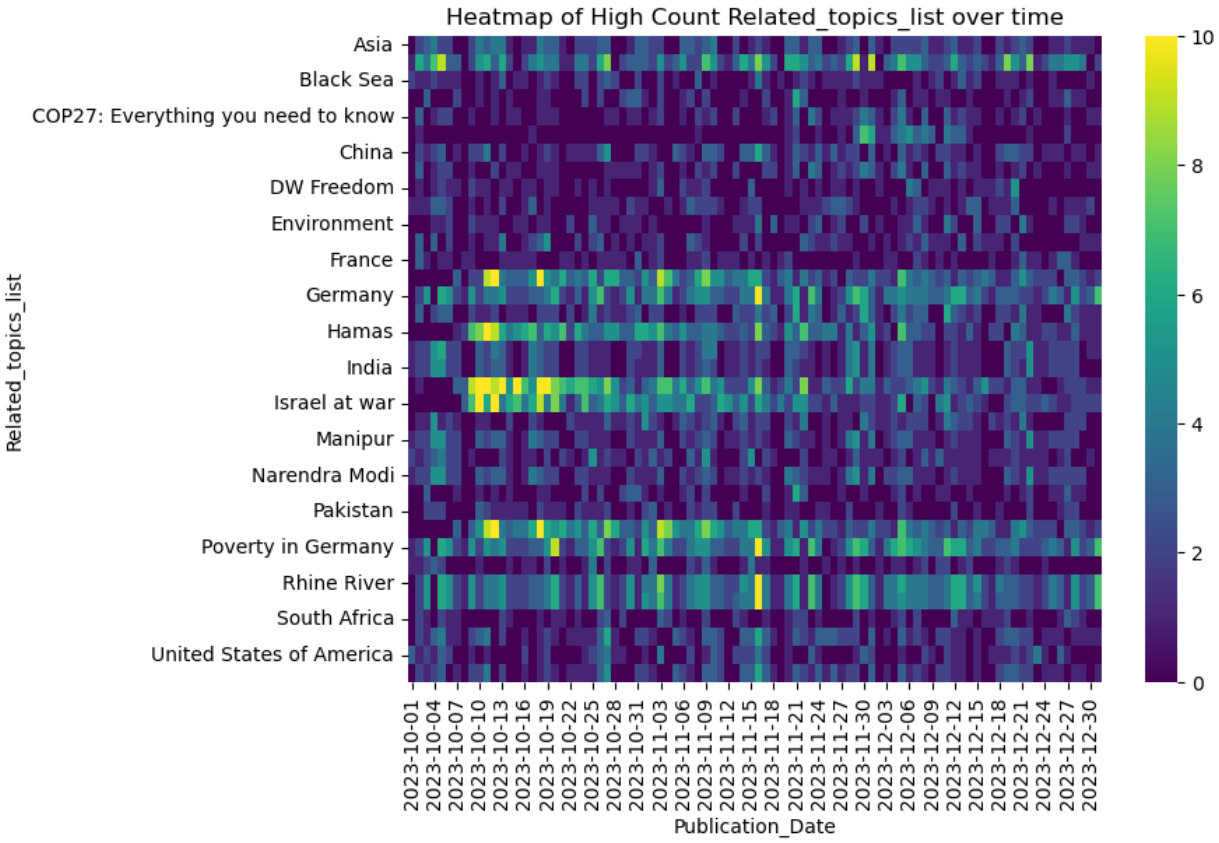
```
'''                          Re-datatype the Publication_Date
'''
print ('Before adjust the format of Dates : ', df['Publication_Date'][0])
df['Publication_Date'] = df['Publication_Date'].dt.date
print ('After adjust the format of Dates : ', df['Publication_Date'][0])
```

```
Before adjust the format of Dates :  2023-10-20 00:00:00
After adjust the format of Dates :  2023-10-20
```

In [11]:

```
from matplotlib.colors import ListedColormap
import numpy as np
import seaborn as sns
import json
def heatmap(variable, count_threshold):
    explode_variable = df.explode(variable)
    counts = explode_variable[variable].value_counts()
    count_threshold = count_threshold

    high_count_values = counts[counts > count_threshold].index
    filtered_df =
explode_variable[explode_variable[variable].isin(high_count_values)]
    cross_tab_filtered = pd.crosstab(filtered_df[variable],
filtered_df['Publication_Date'])

    plt.figure(figsize=(8, 6))
```

```
    sns.heatmap(cross_tab_filtered, annot=False, cmap='viridis', fmt='d',
vmin=0, vmax=10)
   # plt.colorbar(cax=None, ax=None, cbar_kws={"ticks":[]})  # Set ticks
to an empty list to remove axis values
    plt.title('Heatmap of High Count '+ variable+' over time')
    plt.show()
heatmap('Related_topics_list', 60)
```



Discussion:

The heatmap depicting the count of the top 20 *Related Topics* indicates that *Israel* and *Israel at war* consistently exhibited high frequency from October 10, 2023, to October 19, 2023. During the same period, topics related to *Hamas* and *Palestinian territories* were also prominent. The topic of *BRICS* appears to be a recurrent discussion approximately once every month. On November 15, 2023, the subjects of *Poverty in Germany* and *Mr. Robert Habeck* were also addressed.

Task 4-Graph Properties

Bring the dataset into the form that you need for the experiments.

• Compute and explore various graph properties and interpret the results.

First we implement the attributes of *Region* and *Category*

In [6]:
```
import pandas as pd
import itertools
import networkx as nx
from community import community_louvain
from d3graph import d3graph, vec2adjmat
from networkx.algorithms.community import girvan_newman
```

In [8]:
```
'''                    Make the pair of two attributes i.e., Region and
Category
'''
pairs = []
try:
    for x, y in itertools.product(df.Region, df.Category):
        if x != y:    # Not Included the Same values
            pairs.append((x, y))
except:
    pairs.append(None)
```

In [9]:
```
def nodes_edges_subgraph(pairs, x, y, degrees):
    Pairs = pd.DataFrame(pairs, columns=[x, y])
    print ('Total length of the connections between Region and Category is
: ',len(Pairs[x]))
    print (50*'*')
    # Count occurrences of unique pairs
    edges = pd.DataFrame(pairs, columns=[x, y]).groupby([x,
y]).size().reset_index(name='weight')
    #edges.sort_values(by=['weight'],ascending=False).head(10)

    # Creating a network graph using NetworkX
    G = nx.from_pandas_edgelist(edges, x, y, 'weight')

    # subgraphing
    subgraph = G.subgraph([node for node, degree in
G.degree(weight='weight') if degree > degrees])

    num_nodes_subgraph = subgraph.number_of_nodes()
    print("Number of nodes in the subgraph:", num_nodes_subgraph)
```

```
    num_edges_subgraph = subgraph.number_of_edges()
    print("Number of edges in the subgraph:", num_edges_subgraph)
    print (50*'*')

    subgraph_edges = pd.DataFrame(subgraph.edges(data="weight"), columns=
["source", "target", "n"])
    adjmat = vec2adjmat(subgraph_edges.source, subgraph_edges.target,
subgraph_edges.n)
    d3 = d3graph()
    d3.graph(adjmat)
    d3.show()
    return subgraph
```

In [10]:

```
'''                          For NetworkX analysis, first we implement the
"Region" and "Category" attributes
'''
region_category_network = nodes_edges_subgraph(pairs, 'Region',
'Category', 10000)
region_category_network
```

```
Total length of the connections between Region and Category is :  12502906
**************************************************
```

```
[d3graph] INFO> Set directed=True to see the markers!
Number of nodes in the subgraph: 159
Number of edges in the subgraph: 5820
**************************************************
```

```
[d3graph] INFO> Keep only edges with weight>0
[d3graph] INFO> Number of unique nodes: 159
[d3graph] INFO> Slider range is set to [0, 294758]
[d3graph] INFO> Write to path: [C:\Users\AKTRAD~1\AppData\Local\Temp\tmpcfaz9pa4\d3gr
aph.html]
[d3graph] INFO> File already exists and will be overwritten: [C:\Users\AKTRAD~1\AppDa
ta\Local\Temp\tmpcfaz9pa4\d3graph.html]
```

Out[10]:

```
<networkx.classes.graph.Graph at 0x2822867b890>
```

In [11]:

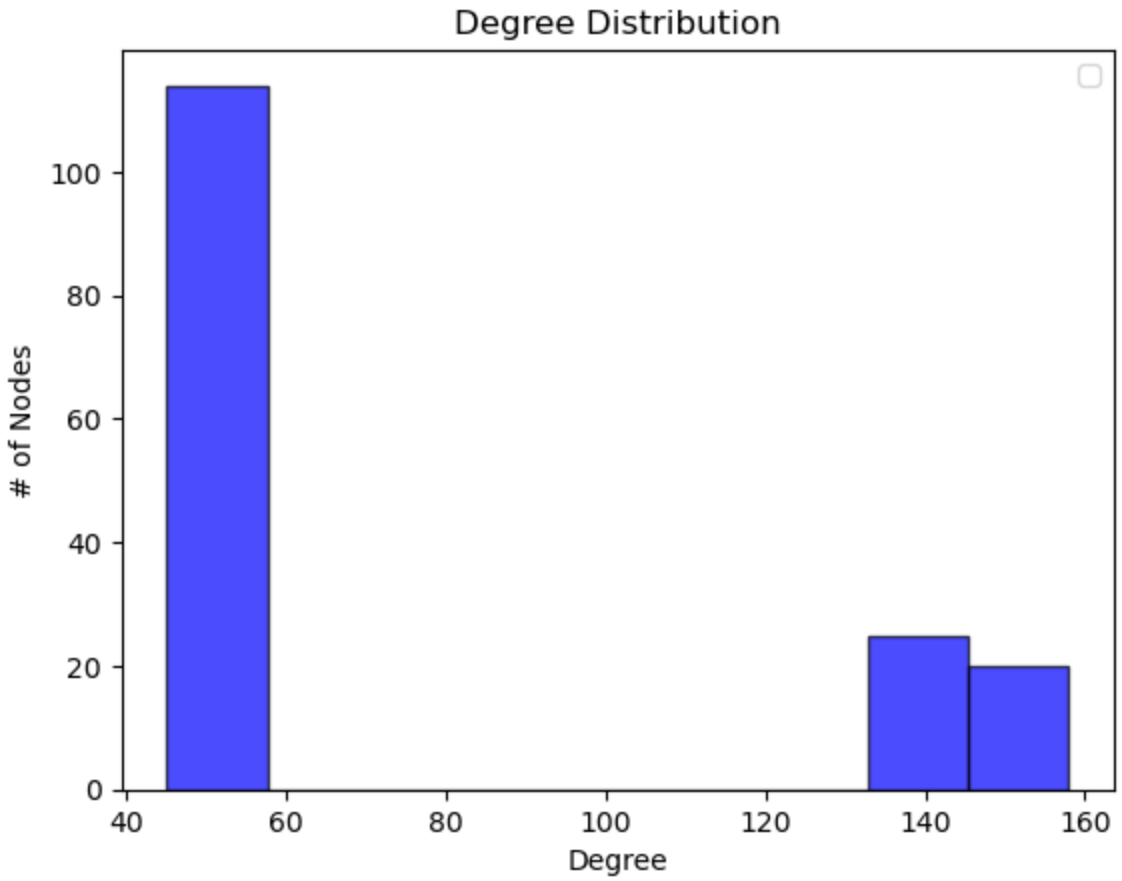
```
'''              Degree Distribution & Wordcloud of Nodes
'''
import networkx as nx
import matplotlib.pyplot as plt
from wordcloud import WordCloud
%matplotlib inline
def degree_distribution(subgraph):
```

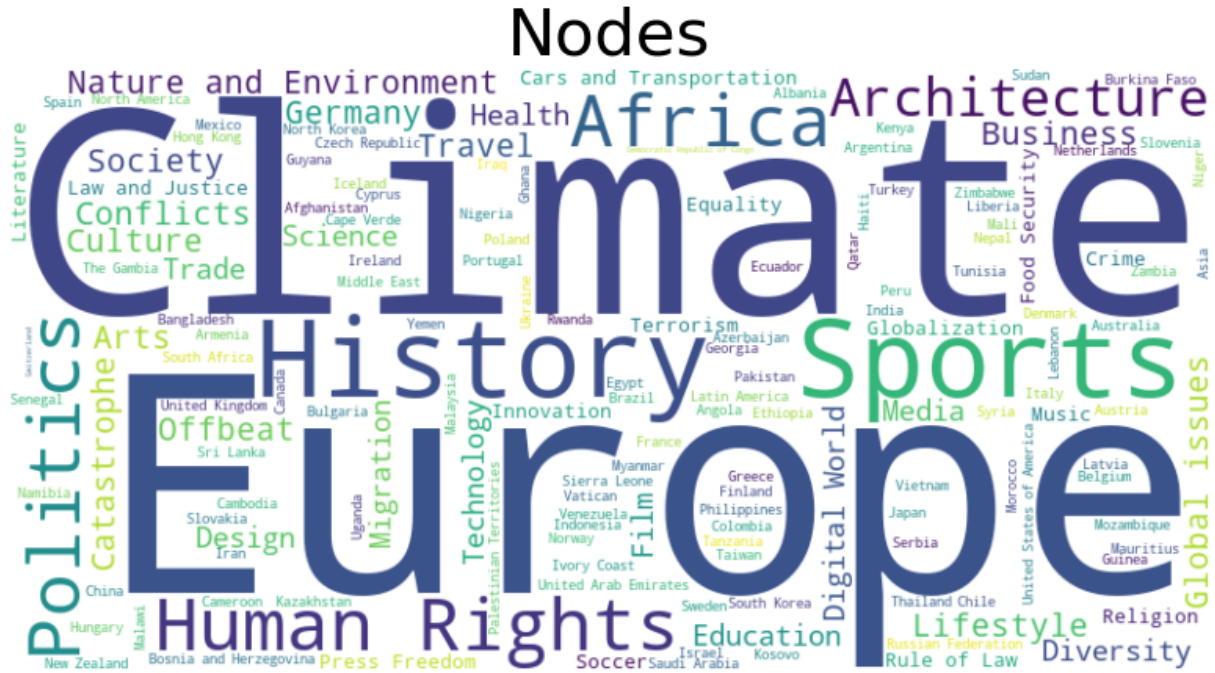
```
    # Calculate degree distribution
    degrees = [degree for _, degree in subgraph.degree()]
    nodes = [node for node in subgraph.nodes()]

    # Plot degree distribution
    plt.hist(degrees, bins='auto', color='blue', edgecolor='black',
alpha=0.7)
    plt.xlabel('Degree')
    plt.ylabel('# of Nodes')
    plt.title('Degree Distribution')
    plt.legend()
    plt.show()


#*************************************************************************
    # Wordcloud of Nodes

    # First convert two lists into Dataframe
    nodes_degrees = pd.DataFrame(
        {'Nodes': nodes,
         'Degrees': degrees
        }
    )
    # Combine the words and their corresponding values into a dictionary
    word_values_dict = dict(zip(nodes_degrees['Nodes'],
nodes_degrees['Degrees']))
    # Generate the word cloud
    wordcloud = WordCloud(width=800, height=400,
background_color='white').generate_from_frequencies(word_values_dict)

    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.title('Nodes', fontsize=30)
    plt.axis('off')
    plt.show()
```

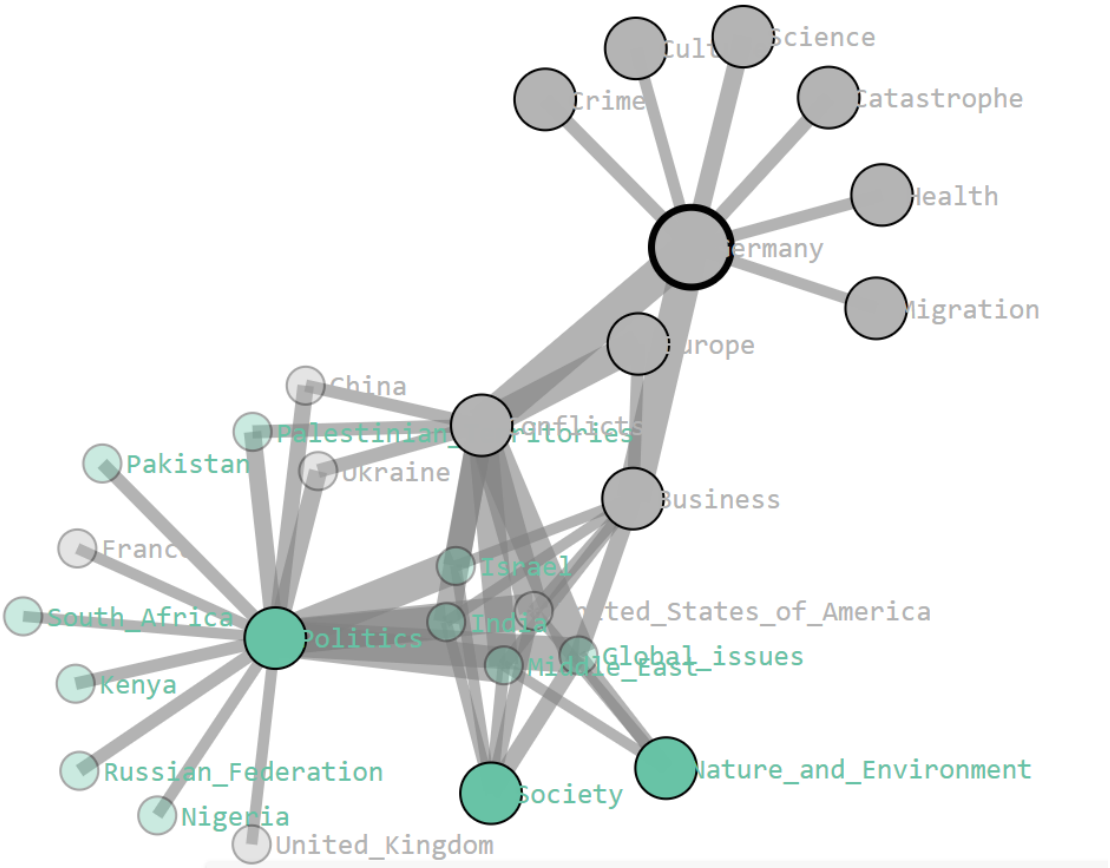
In [12]:

```
degree_distribution(region_category_network)
```

```
[d3graph] WARNING> No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with no argument.
```







In [29]:

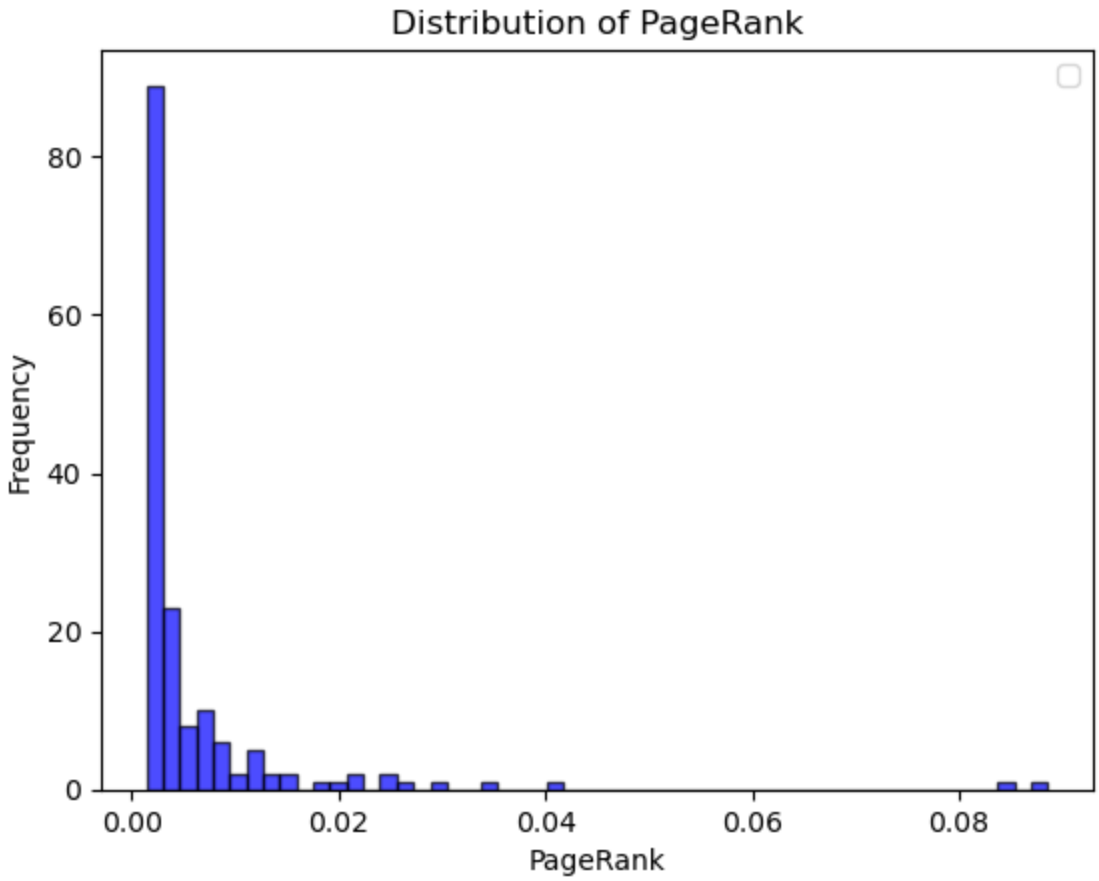
```
'''                         Pagerank, Eccentricity & their Wordcloud of
Nodes
'''

def PageRank_Eccentricity(function, subgraph, Text, weight):

    graph_properties = function(subgraph, weight=weight)
    print('\n'+Text+':\n')
    nodes, values = [], []
    for node, value in graph_properties.items():
        nodes.append(node)
        values.append(value)


#*************************************************************************
    # Histogram Plot for the distribution of various graph properties
    plt.hist(values, bins='auto', color='blue', edgecolor='black',
alpha=0.7)
    plt.xlabel(Text)
```

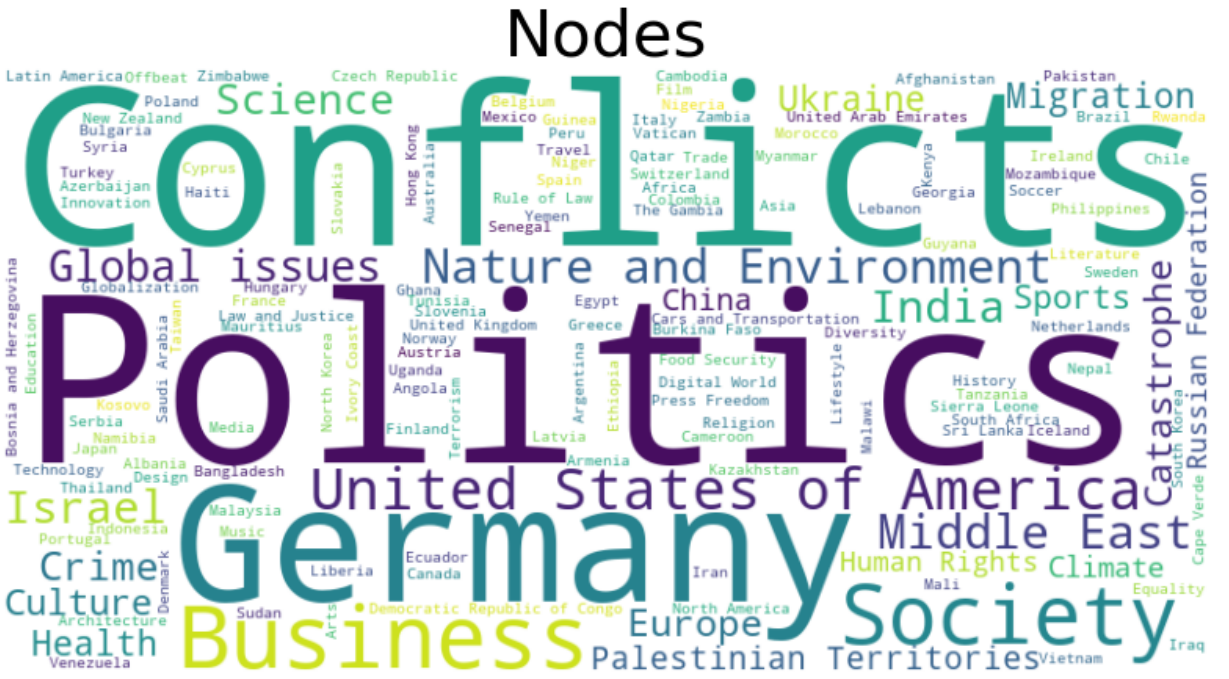
```
    plt.ylabel('Frequency')
    plt.title('Distribution of '+Text)
    plt.legend()
    plt.show()



#**************************************************************************
    # Wordcloud of Nodes

    # First convert two lists into Dataframe
    nodes_Values = pd.DataFrame(
        {'Nodes': nodes,
         'Values': values
        }
    )
    # Combine the words and their corresponding values into a dictionary
    word_values_dict = dict(zip(nodes_Values['Nodes'],
nodes_Values['Values']))
    # Generate the word cloud
    wordcloud = WordCloud(width=800, height=400,
background_color='white').generate_from_frequencies(word_values_dict)

    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.title('Nodes', fontsize=30)
    plt.axis('off')
    plt.show()
    print (200*'*')
```

In [31]:

```
PageRank_Eccentricity(nx.pagerank, region_category_network, 'PageRank',
'weight')
#PageRank_Eccentricity(nx.eccentricity, region_category_network,
'Eccentricity', 'weight')
```

```
[d3graph] WARNING> No artists with labels found to put in legend.  Note that artists
whose label start with an underscore are ignored when legend() is called with no argu
ment.
:PageRank
```






```
********************************************************************************
********************************************************************************
******************************
```

Discussion:

1. **Generating Pairs:**
   - First we create pairs of unique combinations between the "Region" and "Category"

2. **Creating a Network Graph:**
   - A network graph `G` is constructed using NetworkX, where nodes represent regions and categories, and edges represent the connections between them based on the count of occurrences.
3. **Subgraphing:**
   - A subgraph `subgraph` is created by selecting nodes with a degree greater than 10,000.
4. **Visualizing the Subgraph:**
   - The subgraph edges are converted into an adjacency matrix, and a D3.js-based graph visualization is generated using the `d3graph` utility.
5. **Degree Distribution Plot & WordCloud of Nodes:**
   - The code calculates the degree distribution of the subgraph and plots a histogram. Meanwhile, the topological importance of each node is visualized in a WordCloud in which higher degrees have bigger front.
6. **Other Graph Properties:**
   - We have also explored additional graph properties, namely *PageRank* and *Eccentricity*. As part of this analysis, we have created distributions and WordClouds based on their higher values.

In short, we have provided the insights about the connections between regions and categories of the News Articles, the structure of the network, and the distribution of node degrees and centrality within a subgraph.

In graph theory, the degree of a node refers to the number of edges connected to that node. It is a crucial metric that provides insights into the structure and connectivity of a network. Given that the "Number of nodes" exhibits an inverse relationship with the "Degree," in our scenario, the number of nodes has decreased to approximately 20, coinciding with the highest degrees of 158. This decline follows a gradual and consistent pattern.

Also, we have generated the wordcloud of the all the Nodes in which the bigger front words correspond to the higher degree of Nodes. In our observation, we have identified that *Category* exhibits higher degree in comparison to *Region*. Notably, categories such as **Politics, Culture, Conflicts, Nature & Environment, Arts**, etc., play a more crucial role in network connections. However, we have observed only three *Regions*—**Germany, Africa, and Europe**—with elevated centrality.

For a clearer visual representation, refer to the accompanying diagram where categories demonstrate a more central influence, albeit with the support of regions. Germany stands out as an exceptional case.

PageRank serves as a centrality metric, indicating the significance of nodes within a graph. Nodes with elevated PageRank values are deemed more central and influential in the network. Notably, nodes such as **Germany, Politics, Conflicts, United States, Business, Society** hold

particular importance, as evident in the provided network visualization. This prominence is attributed to the algorithm's ability to consider both the quantity and quality of connections a node possesses. In essence, PageRank offers a nuanced evaluation of a node's influence by factoring in the influence of its neighbors.

In [ ]:

```
```

Task 5 - Central Nodes

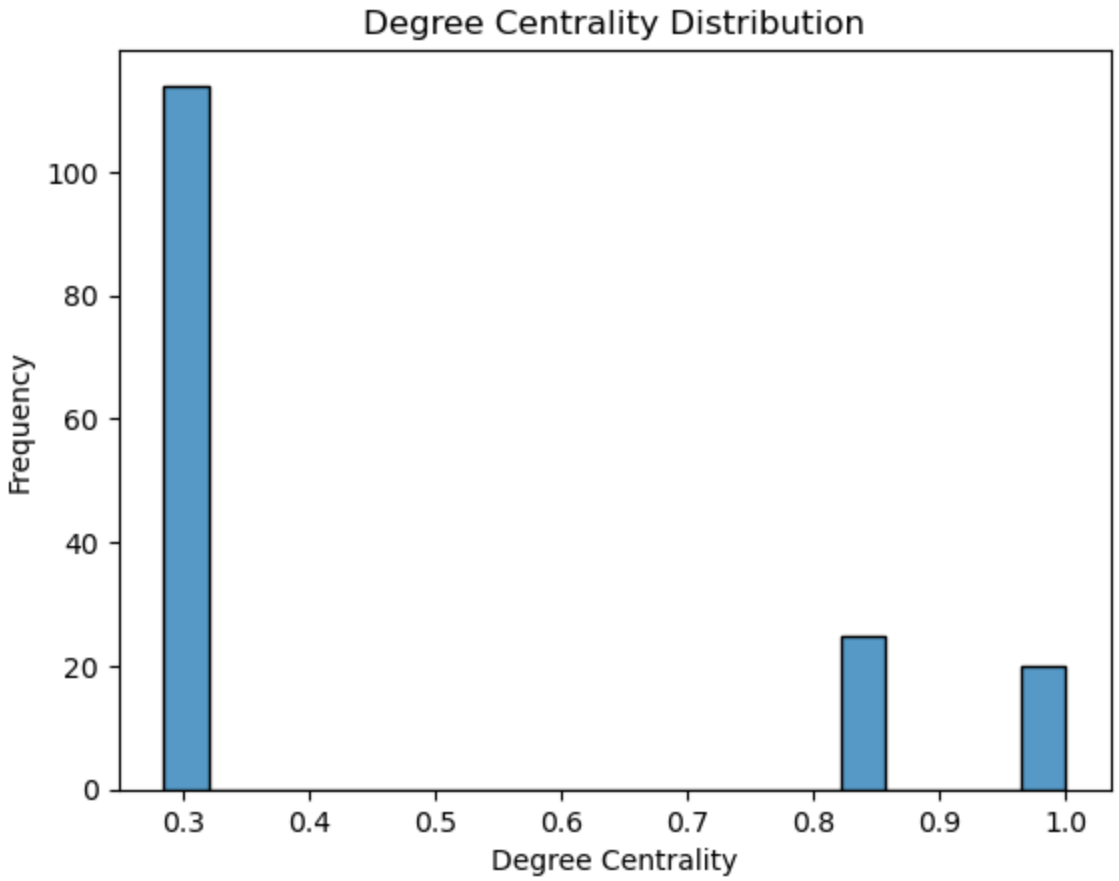
In [13]:

```
'''                    Compute the various Central Nodes
'''
import matplotlib.pyplot as plt
import seaborn as sns
def Centrality(function, subgraph, threshold_centrality_min,
threshold_centrality_max, Text):
    centrality = function(subgraph)

#*************************************************************************
    '''                    Distribution of Centrality
    '''
    sns.histplot(list(centrality.values()), kde=False, bins=20)
    plt.title(Text+" Distribution")
    plt.xlabel(Text)
    plt.ylabel("Frequency")
    plt.show()

#*************************************************************************
    '''                    WordCloud of Centrality
    '''
    sorted_nodes = sorted(centrality, key=centrality.get)
    sorted_values = [centrality[node] for node in sorted_nodes]
    # Convert two lists into Dataframe
    nodes_values_centrality = pd.DataFrame(
        {'Nodes': sorted_nodes,
         'Values': sorted_values
        }
    )
    # Combine the words and their corresponding values into a dictionary
    word_values_dict = dict(zip(nodes_values_centrality['Nodes'],
```

```
nodes_values_centrality['Values']))
    # Filter values
    filtered_values = {key: value for key, value in
word_values_dict.items() if value > threshold_centrality_min and value <=
threshold_centrality_max}
    # Generate the word cloud
    wordcloud = WordCloud(width=800, height=400,
background_color='white').generate_from_frequencies(filtered_values)

    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.title('Higher '+Text, fontsize=30)
    plt.axis('off')
    plt.show()
    #************************************************************
    print (100*'*')
```

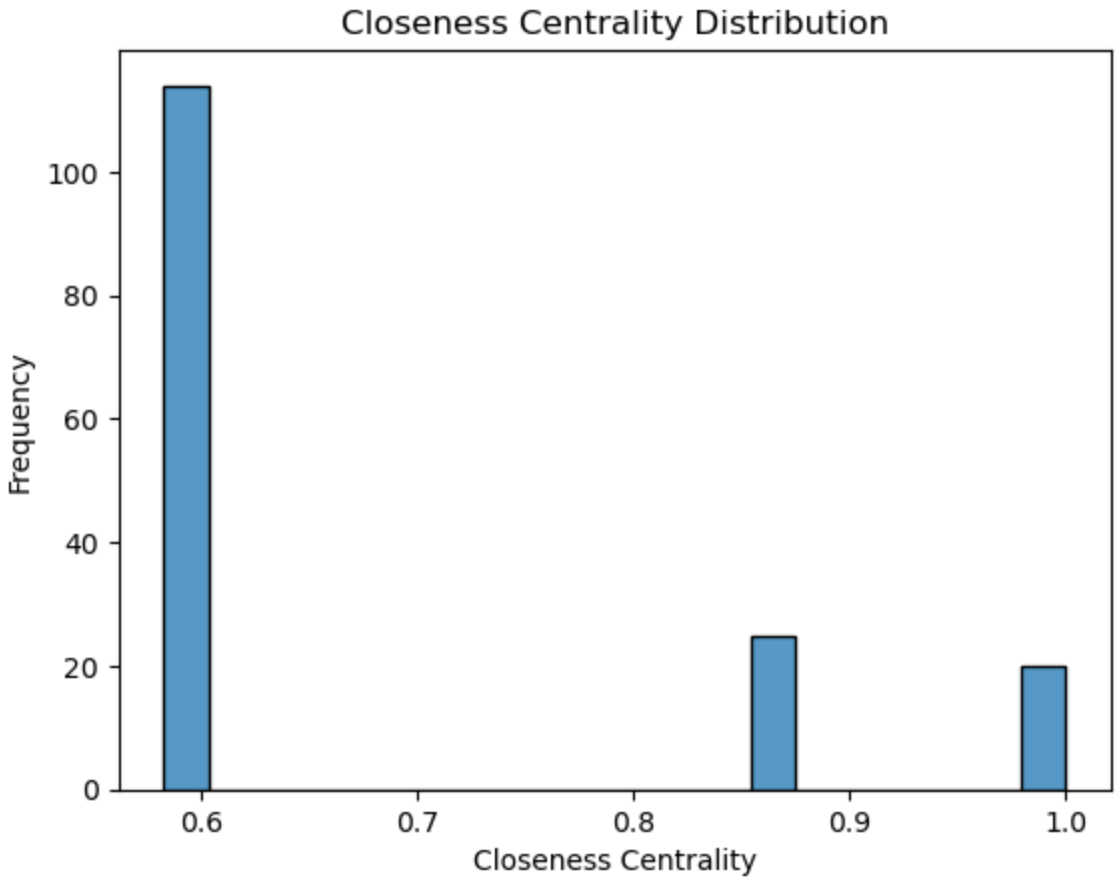
In [31]:

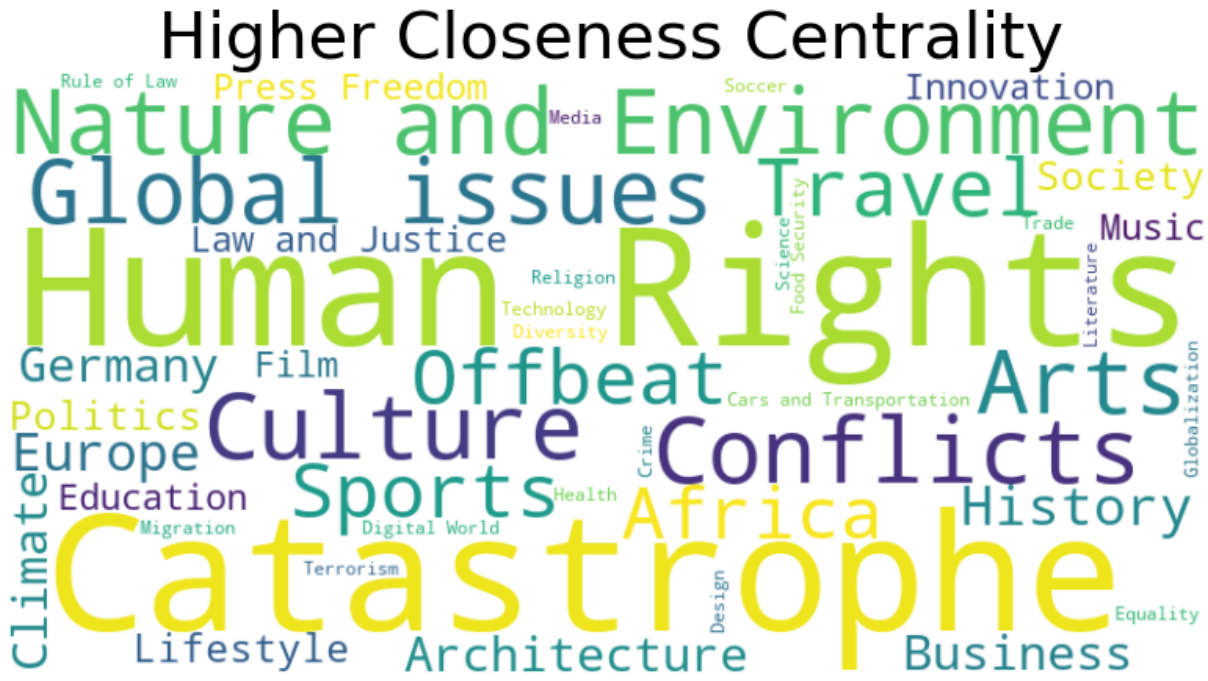
```
Centrality(nx.degree_centrality, region_category_network, 0.8, 1.0,
'Degree Centrality')
Centrality(nx.closeness_centrality, region_category_network, 0.8, 1.0,
'Closeness Centrality')
Centrality(nx.betweenness_centrality, region_category_network, 0.010, 1.0,
'Betweenness Centrality')
```


Degree Centrality Distribution
Frequency
Degree Centrality
0
20
40
60
80
100
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

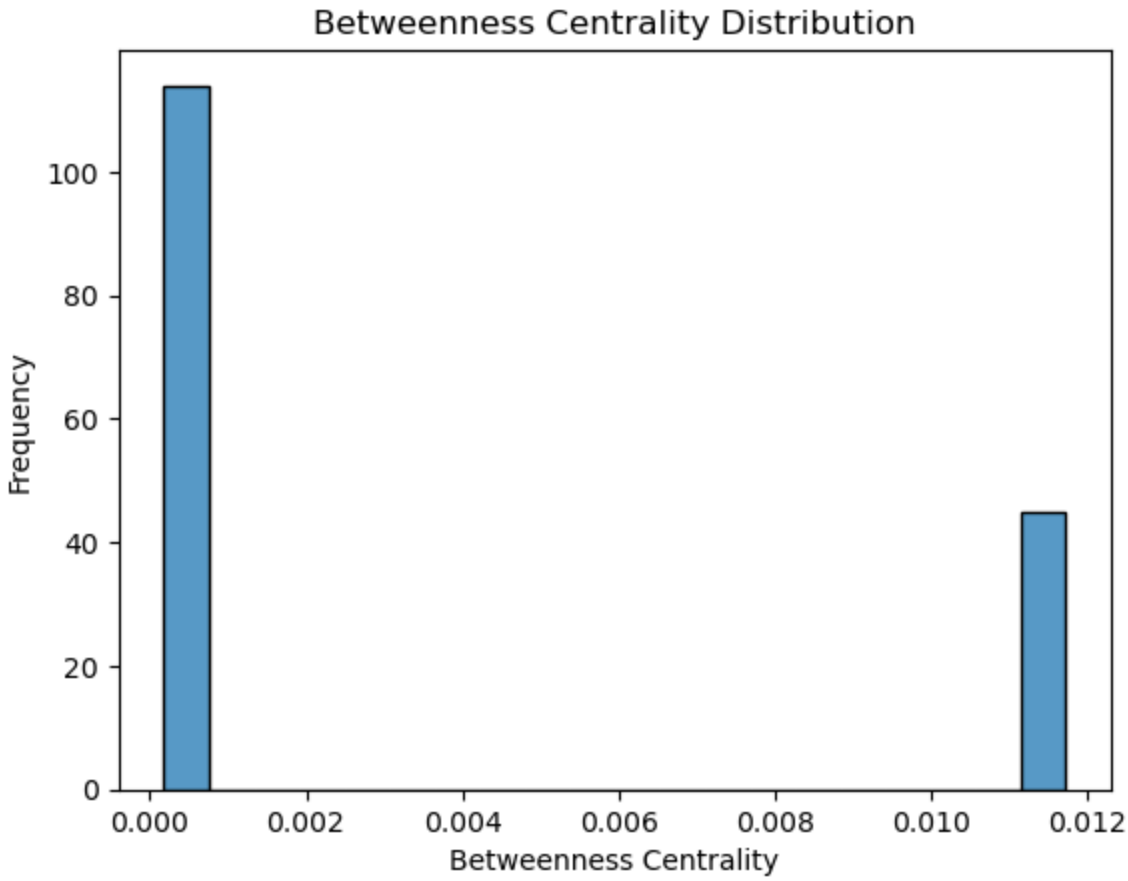

Higher Degree Centrality
Human Rights
Catastrophe
Nature and Environment
Global issues
Conflicts
Sports
Culture
Arts
Business
Architecture
History
Africa
Europe
Germany
Climate
Travel
Lifestyle
Offbeat
Politics
Press Freedom
Education
Society
Music
Technology
Terrorism
Trade
Religion
Law and Justice
Globalization
Migration
Food Security
Cars and Transportation
Digital World
Innovation
Diversity
Health
Crime
Media
Design
Film
Science
Literature
Rule of Law
Soccer
Equality

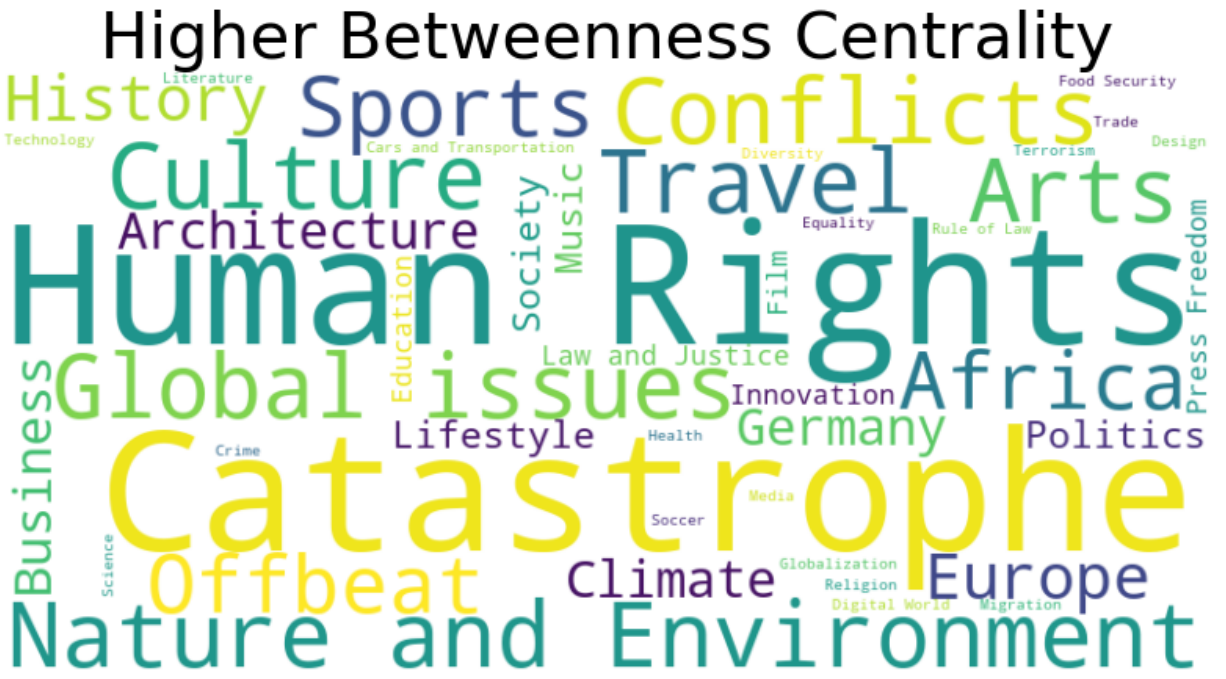
```
**********************************************************************************
***************
```


Closeness Centrality Distribution
Frequency
Closeness Centrality
0
20
40
60
80
100
0.6
0.7
0.8
0.9
1.0


Higher Closeness Centrality
Rule of Law
Press Freedom
Soccer
Innovation
Nature and Environment
Media
Global issues
Travel
Society
Law and Justice
Trade
Music
Human Rights
Religion
Science
Food Security
Literature
Technology
Diversity
Germany
Film
Offbeat
Arts
Politics
Culture
Cars and Transportation
Conflicts
Europe
Crime
Globalization
Education
Sports
Health
Africa
History
Climate
Migration
Digital World
Catastrophe
Terrorism
Design
Equality
Lifestyle
Architecture
Business

```
**********************************************************************************
***************
```






```
**********************************************************************************************************
```

Discussion:

We have implemented the `Degree Centrality`, `Closeness Centrality`, and `Betweenness Centrality`. Each centrality serves a unique purpose:

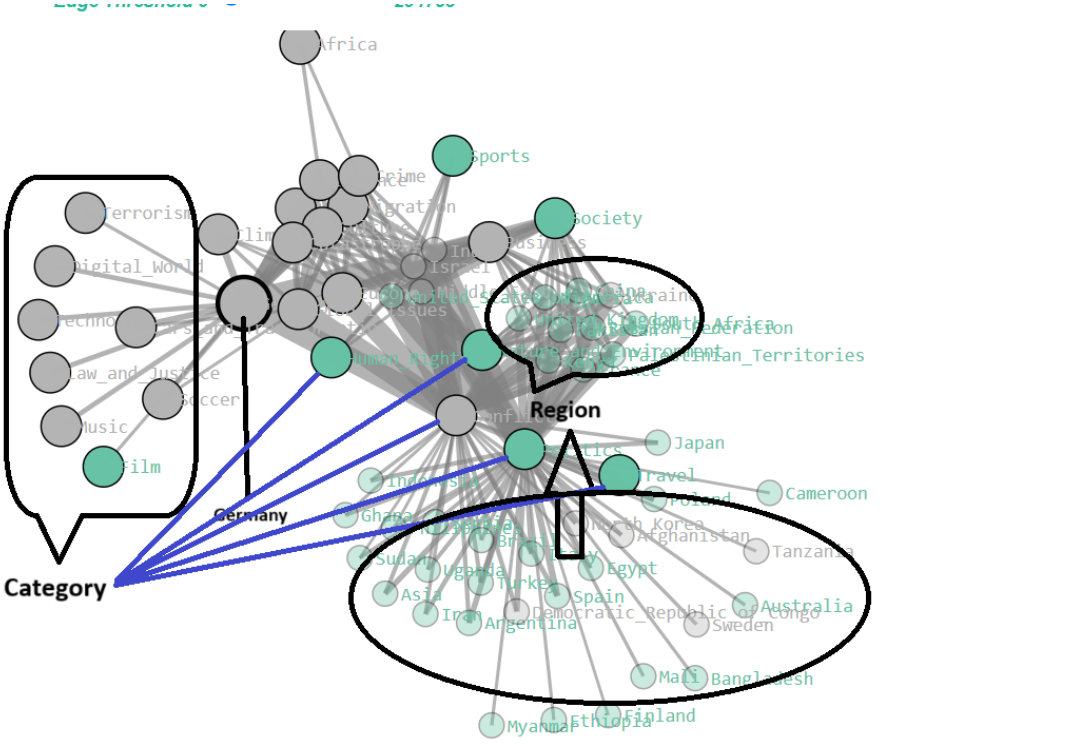
Degree Centrality: Highlights nodes with many direct connections, important for overall network structure.

Closeness Centrality: Emphasizes nodes that can efficiently reach other nodes, crucial for rapid communication or information spreading.

Betweenness Centrality: Identifies nodes controlling information flow, essential for maintaining network connectivity and resilience.

In our scenario, we observe a noteworthy similarity between the sets of nodes highlighted by Degree and Closeness centrality. These nodes predominantly pertain to the "Category" attribute, indicating that nodes associated with the "Category" attribute play a central role and are closely connected to other nodes in the network.

Conversely, the set of nodes exhibiting high Betweenness centrality differs, even though they share the same attribute. These nodes serve as bridges or critical points, facilitating communication between other nodes. The distinction in sets suggests that certain nodes within the "Category" attribute, while not necessarily having the highest local connectivity or the quickest access to all other nodes, play a crucial role in maintaining overall network connectivity and facilitating communication between different parts of the network.



The analysis of central nodes in the context of our dataset supports our initial hypothesis regarding **Deutsche Welle's** approach to news coverage. It affirms that **Deutsche Welle** provides inclusive coverage without discrimination, emphasizing global news and addressing sensitive issues across various regions and categories, notably in the domains of Politics and Conflicts.

Examining the network image, we discern key central nodes such as *Germany, Conflicts, Politics, Human Rights, Global issues,* and *Catastrophe*. These nodes occupy pivotal positions and are closely connected to other nodes, particularly those associated with the **Region** attribute. This configuration indicates that the identified central nodes serve as critical hubs, facilitating connectivity with nodes from different regions, thereby reinforcing the global and comprehensive nature of **Deutsche Welle's** news reporting.

Task 6 - Community Discovery

In [14]:

```
'''                    Community detection using Louvain method
'''
partition_louvain =
community_louvain.best_partition(region_category_network)



# Print communities detected by Louvain method
print("\nCommunities (Louvain Method):")
desired_value = 1
keys_for_value = [key for key, value in partition_louvain.items() if value
== desired_value]
print ('Total Nodes in the Community 1 : ',len(keys_for_value))
print ('List of Nodes belong to Community 1 :','\n','\n',keys_for_value)
print (200*'*')
#**************************************************************************
desired_value = 0
keys_for_value = [key for key, value in partition_louvain.items() if value
== desired_value]
print ('Total Nodes in the Community 0 : ',len(keys_for_value))
print ('List of Nodes belong to Community 0 :','\n','\n',keys_for_value)
```

```
Communities (Louvain Method):
Total Nodes in the Community 1 :  67
List of Nodes belong to Community 1 :

 ['Education', 'Film', 'Europe', 'Climate', 'Greece', 'Bosnia and Herzegovina', 'The
Gambia', 'Cambodia', 'Diversity', 'Kosovo', 'Africa', 'Haiti', 'Digital World', 'Desi
gn', 'Angola', 'Chile', 'Science', 'Technology', 'Ivory Coast', 'South Africa', 'Medi
a', 'Kenya', 'Architecture', 'Trade', 'Sweden', 'Catastrophe', 'Migration', 'Health',
'Kazakhstan', 'Burkina Faso', 'Cars and Transportation', 'Slovenia', 'Religion', 'Tun
isia', 'Lebanon', 'Soccer', 'Zambia', 'Music', 'Global issues', 'Literature', 'Terror
ism', 'Business', 'Press Freedom', 'Colombia', 'Finland', 'Guinea', 'Crime', 'Sri Lan
ka', 'Vietnam', 'North America', 'Nepal', 'Equality', 'Culture', 'Syria', 'Innovatio
n', 'Law and Justice', 'Thailand', 'Zimbabwe', 'Rule of Law', 'Food Security', 'Austr
ia', 'New Zealand', 'Globalization', 'United Kingdom', 'Arts', 'North Korea', 'German
y']
********************************************************************************
********************************************************************************
****************************
Total Nodes in the Community 0 :  92
List of Nodes belong to Community 0 :

 ['Mozambique', 'Iraq', 'Malawi', 'Venezuela', 'Malaysia', 'Pakistan', 'Iceland', 'Ir
eland', 'Peru', 'Poland', 'Australia', 'History', 'United Arab Emirates', 'Morocco',
'Ghana', 'Sports', 'Czech Republic', 'Philippines', 'Human Rights', 'Qatar', 'Banglad
esh', 'China', 'Canada', 'Senegal', 'Hungary', 'Armenia', 'Uganda', 'Japan', 'Myanma
r', 'Spain', 'Sudan', 'Politics', 'Ethiopia', 'Sierra Leone', 'Ukraine', 'Latin Ameri
ca', 'Italy', 'Middle East', 'Nature and Environment', 'Cameroon', 'Slovakia', 'Lifes
tyle', 'Albania', 'Cyprus', 'Afghanistan', 'Asia', 'Norway', 'Society', 'Iran', 'Rwan
da', 'Bulgaria', 'Conflicts', 'Mauritius', 'Vatican', 'Guyana', 'Latvia', 'Nigeria',
'Namibia', 'Serbia', 'Argentina', 'Mali', 'Israel', 'Egypt', 'South Korea', 'Palestin
ian Territories', 'France', 'Belgium', 'Mexico', 'United States of America', 'Ecuado
r', 'Denmark', 'Indonesia', 'Azerbaijan', 'Hong Kong', 'Taiwan', 'Offbeat', 'Georgi
a', 'Saudi Arabia', 'Travel', 'Tanzania', 'Russian Federation', 'India', 'Liberia',
'Netherlands', 'Turkey', 'Cape Verde', 'Niger', 'Brazil', 'Yemen', 'Portugal', 'Democ
ratic Republic of Congo', 'Switzerland']
```

In [39]:

```
'''                    Community detection using Girvan Newman
'''
from networkx.algorithms.community import girvan_newman

# Community detection using Girvan-Newman algorithm
communities_generator = girvan_newman(region_category_network)
partition_girvan_newman = next(communities_generator)

print("\nGirvan-Newman Community Detection Results:")
for i, community_set in enumerate(partition_girvan_newman):
    if len(community_set) > 1:
        print(f"Community {i + 1}: {community_set}")
```

```
        print ('Total Number of Nodes in the given community :
',len(community_set))
    else:
        print(f"Community {i + 1} has only 1 Node")
    print (200*'*'+'\n')
```

```
Girvan-Newman Community Detection Results:
Community 1: {'Mozambique', 'Iraq', 'Malawi', 'Venezuela', 'Education', 'Malaysia',
'Film', 'Europe', 'Climate', 'Pakistan', 'Iceland', 'Greece', 'Bosnia and Herzegovin
a', 'Ireland', 'Peru', 'Poland', 'The Gambia', 'Australia', 'History', 'Cambodia', 'D
iversity', 'United Arab Emirates', 'Morocco', 'Ghana', 'Sports', 'Czech Republic', 'P
hilippines', 'Human Rights', 'Kosovo', 'Qatar', 'Bangladesh', 'China', 'Haiti', 'Afri
ca', 'Digital World', 'Canada', 'Senegal', 'Design', 'Hungary', 'Angola', 'Armenia',
'Uganda', 'Myanmar', 'Spain', 'Chile', 'Sudan', 'Science', 'Politics', 'Technology',
'Ethiopia', 'Sierra Leone', 'Ukraine', 'Ivory Coast', 'Latin America', 'South Afric
a', 'Italy', 'Media', 'Kenya', 'Middle East', 'Architecture', 'Nature and Environmen
t', 'Trade', 'Sweden', 'Catastrophe', 'Cameroon', 'Slovakia', 'Migration', 'Lifestyl
e', 'Albania', 'Cyprus', 'Afghanistan', 'Asia', 'Health', 'Kazakhstan', 'Norway', 'So
ciety', 'Iran', 'Burkina Faso', 'Rwanda', 'Bulgaria', 'Cars and Transportation', 'Slo
venia', 'Religion', 'Conflicts', 'Mauritius', 'Tunisia', 'Lebanon', 'Vatican', 'Guyan
a', 'Latvia', 'Nigeria', 'Soccer', 'Zambia', 'Namibia', 'Serbia', 'Music', 'Global is
sues', 'Argentina', 'Mali', 'Literature', 'Terrorism', 'Business', 'Press Freedom',
'Israel', 'Egypt', 'Colombia', 'Finland', 'South Korea', 'Palestinian Territories',
'France', 'Guinea', 'Crime', 'Belgium', 'Mexico', 'Sri Lanka', 'Vietnam', 'United Sta
tes of America', 'Ecuador', 'North America', 'Denmark', 'Nepal', 'Equality', 'Cultur
e', 'Indonesia', 'Azerbaijan', 'Syria', 'Hong Kong', 'Taiwan', 'Offbeat', 'Georgia',
'Innovation', 'Saudi Arabia', 'Travel', 'Tanzania', 'Law and Justice', 'Russian Feder
ation', 'Thailand', 'India', 'Zimbabwe', 'Rule of Law', 'Food Security', 'Netherland
s', 'Liberia', 'Turkey', 'Cape Verde', 'Austria', 'New Zealand', 'Niger', 'Globalizat
ion', 'United Kingdom', 'Brazil', 'Arts', 'Yemen', 'North Korea', 'Portugal', 'Democr
atic Republic of Congo', 'Switzerland', 'Germany'}
Total Number of Nodes in the given community :  158
**********************************************************************************
**********************************************************************************
******************************

Community 2 has only 1 Node
**********************************************************************************
**********************************************************************************
******************************
```

Discussion:

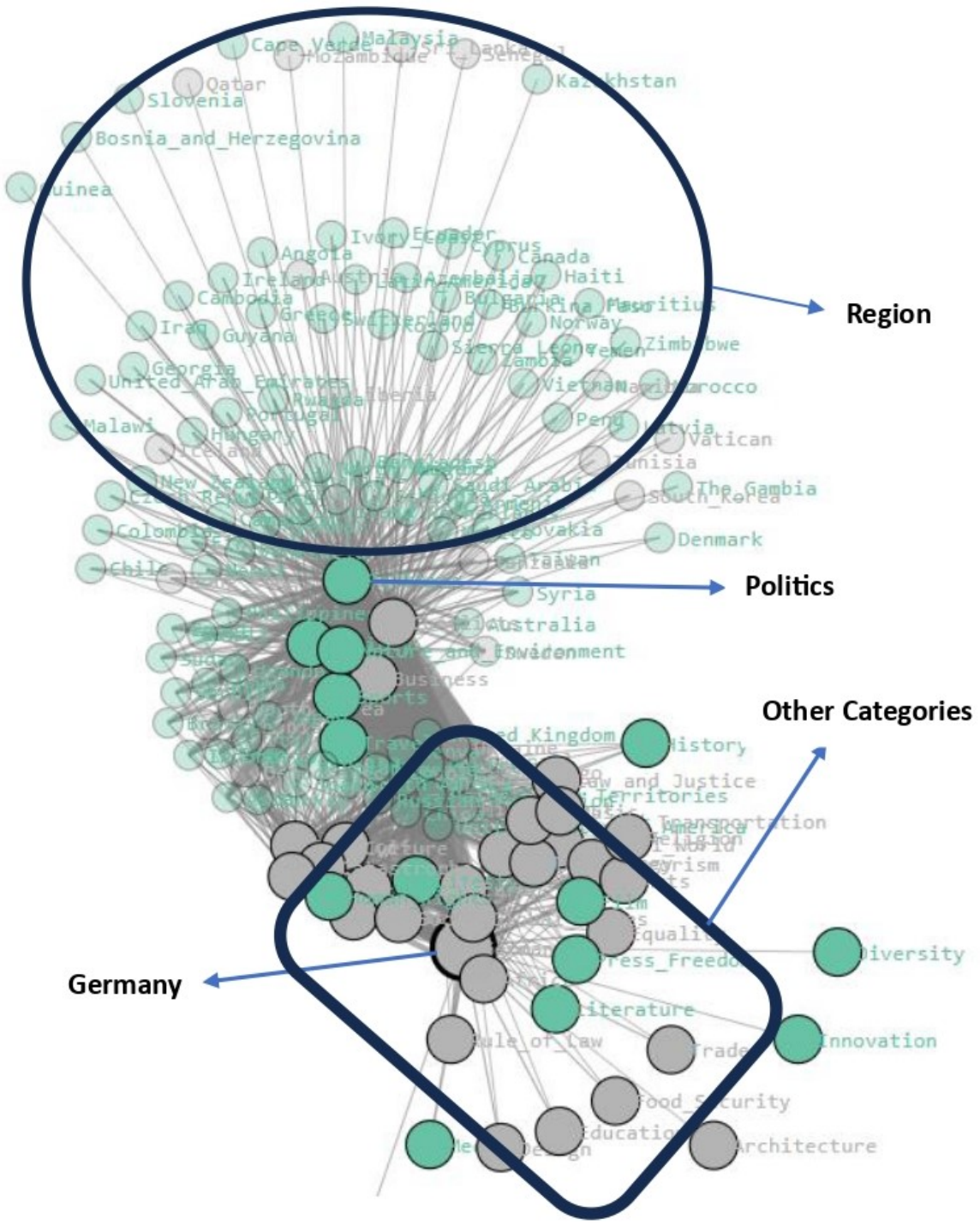
In our exploration of community detection, we applied two prominent algorithms: the Louvain method and the Girvan-Newman Algorithm.

Louvain Method: Widely recognized for uncovering communities or clusters within a network, the Louvain method is a function provided by the python-louvain library. This algorithm focuses on optimizing the modularity of a network partition. Its objective is to achieve a division of

nodes into communities that maximizes the density of edges within communities while minimizing the density of edges between communities.

In our specific context, where we have 159 nodes representing the *Region* and *Category* attributes of news articles, the Louvain method effectively split the network into two distinct communities. The first community, denoted as community = 1, comprises 88 nodes, primarily emphasizing regions. Notably, the role of *Politics* is pivotal in this community, aligning with the overarching theme of international news falling under the category of Politics. This aligns with our hypothesis that Deutsche Welle (DW) excels in presenting unbiased coverage of global politics.

Conversely, community = 0, the second community identified by the Louvain method, is centered around various categories linked with the region = *Germany*. This finding reinforces our initial hypothesis that DW diligently covers diverse news categories within the region of Germany. Examining the network image, the oval area corresponds to the predominantly political community, while the square area reflects the community centered on Germany-related categories.



However, the analysis also unveils complex nodes that bridge both communities. These nodes exhibit intricate interconnections between categories and regions, as depicted in the network image. These intricacies suggest a nuanced relationship between general categories and Germany, potentially representing connections between Germany and the broader world.

Central node analysis further substantiates this observation. Categories such as *Sport, Business, Nature & Environment, Travel, Conflicts* emerge prominently in the central nodes. These categories exhibit higher degree centrality and closeness centrality, indicating their significance

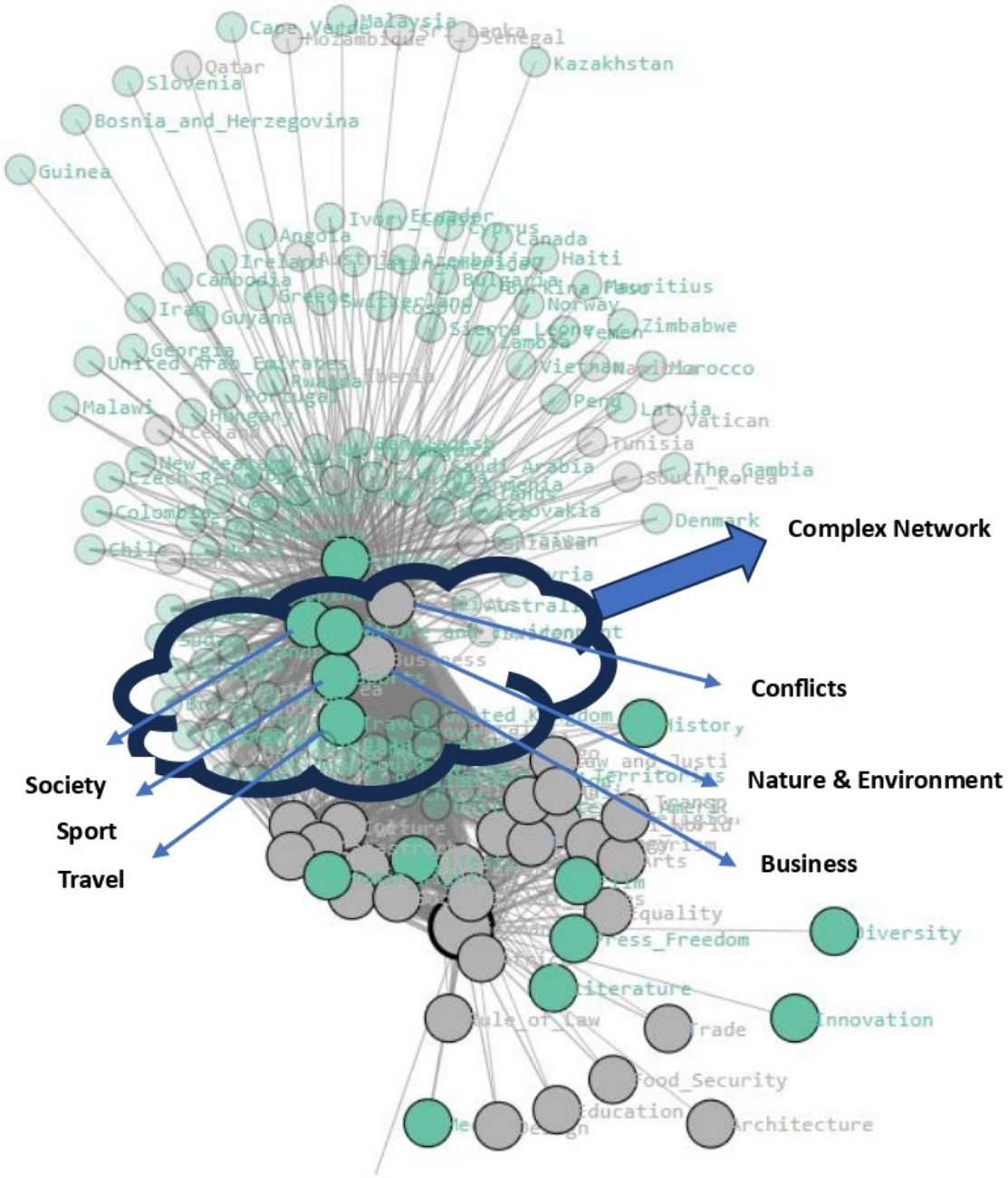
in connecting various nodes. This aligns with the notion that these categories may have direct or indirect relationships with Germany, contributing to their central role in the network.

Girvan-Newman Algorithm Overview: The Girvan-Newman algorithm employs a divisive strategy by iteratively eliminating edges with high betweenness centrality, progressively fragmenting the graph into disconnected components. In the specific instance mentioned, we obtained the initial partition from the algorithm's generator, reflecting the community structure at a specific stage.

Observations and Complexity: Our investigation revealed a complexity in the algorithm's performance as it struggled to adequately form communities based on the provided `subgraph`. The outcome consisted of two communities, wherein the first community encompassed all nodes, while the second comprised only a single node.

Conceptual Reasoning: This unexpected result underscores the sensitivity of the Girvan-Newman algorithm to the structural characteristics of the input graph. The algorithm's reliance on betweenness centrality to identify edges for removal may not always align with the desired community structure, especially in cases where the graph has distinct properties or the chosen subgraph exhibits atypical connectivity patterns. Adjustments to the algorithm's parameters or consideration of alternative community detection methods might be necessary in such scenarios to obtain more meaningful and representative results.

In summary, the Louvain method successfully identified distinct communities in the DW news network, shedding light on the interconnectedness between global and regional news coverage. The algorithm's outcomes align with our hypotheses and provide valuable insights into the thematic structure of DW's news dissemination.



Task 7 - Conclusion & Future Work

Task 1:

- Our project is centered around the analysis of the Deutsche Welle (DW) website, a prominent German international broadcaster with operations in television, radio, and online services. Our objective is to scrutinize various features of DW.com, focusing on news articles published between October 1, 2023, and December 31, 2023. The primary aim is to investigate the correlation among the Region, Category, and Related Topics. Our hypothesis suggests that DW presents unbiased global issues across diverse categories and regions, aligning with its mission of delivering comprehensive news coverage globally.

Task 3:

- Visualizations constructed to depict the daily article count and highlight primary regions and categories revealed fluctuations in daily counts, ranging from a minimum of 5 to a maximum of 55 articles. On average, 39 articles are published each day, with Politics and Conflicts constituting over 40% of all article classifications among the 3538 articles. Germany emerges as the most frequently mentioned region, encompassing 12% of all mentioned regions.

Task 4:

- Graph properties applied to *Region* and *Category* attributes provided insights into the connections between regions and categories, network structure, and distribution of node degrees and centrality within a subgraph. Notably, our observation indicates that the "Category" attribute exhibits higher degrees compared to "Region," with categories like Politics, Culture, Conflicts, Nature & Environment, Arts, playing pivotal roles in network connections, a trend also corroborated by Pagerank analysis.

Task 5:

- Centrality analyses, encompassing Degree Centrality, Closeness Centrality, and Betweenness Centrality, reveal that nodes associated with the "Category" attribute play central roles and exhibit close connections within the network. The intersection of nodes highlighted by Degree and Closeness centrality predominantly pertains to the "Category" attribute, underscoring its pivotal role.

Task 6:

- In the realm of community detection, the Louvain method and Girvan-Newman Algorithm successfully identified distinct communities. The Louvain method delineated communities based on categories surrounded by the region = Germany and regions surrounded by the category = Politics. These outcomes align with our hypotheses, offering valuable insights into the thematic structure of DW's news dissemination.

***Future Work**

- The analysis, limited to *Region* and *Category* attributes, upholds our initial hypothesis regarding Deutsche Welle's unbiased news coverage. It underscores DW's commitment to inclusive and comprehensive news reporting on global issues across regions and categories, particularly in Politics and Conflicts. Acknowledging project limitations, future analyses could delve into attributes such as *Text, Related Topics, Summary* to further scrutinize our initial hypothesis and provide a more comprehensive understanding.

In [ ]: