In [30]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [31]:
```python
data = pd.read_csv("C:/Users/[redacted]Desktop/appstore_instagram_reviews_anonymized.csv")
```

In [32]:
```python
data.head()
```

Out[32]:

	appId	country	date	id	score	text	title	url	userName	userUrl	version
0	389801252	DE	2024-03-20T22:37:03-07:00	11067766035	1	When I want to share any stories, posts or ree...	List of friends in weird order	https://itunes.apple.com/de/review?id=38980125...	988858b9f96ee718616ad5dc2496ff911ff86bd790caad...	https://itunes.apple.com/de/reviews/id159880897	323.0.0
1	389801252	DE	2024-03-20T22:17:05-07:00	11067726970	4	I have been using the app for a week now. I ca...	Great companion	https://itunes.apple.com/de/review?id=38980125...	06dfad50909f5bc85a0a4198d91da9e5d5ae9954e8bba6...	https://itunes.apple.com/de/reviews/id1333052389	323.0.0
2	389801252	DE	2024-03-20T16:08:40-07:00	11066866484	1	Instagram, was ist nur mit euch los? Es ist fr...	#Instagram #Fehler #Verbesserung dringend nötig	https://itunes.apple.com/de/review?id=38980125...	4b8d13b92ce712f53651cb538b1c079daa782136547a20...	https://itunes.apple.com/de/reviews/id711608090	323.0.0
3	389801252	DE	2024-03-20T15:54:39-07:00	11066832683	1	Die neue Schriftart ist so unglaublich hässlich.	Hässliche Layout	https://itunes.apple.com/de/review?id=38980125...	52acf5c35b9fa5653f8fb7b9ab7ec305e0ead4f07708bc...	https://itunes.apple.com/de/reviews/id830638504	323.0.0
4	389801252	DE	2024-03-20T14:59:23-07:00	11066699753	1	Null zufrieden darum gelöscht	Instagram absoluter Müll geworden sorry aber e...	https://itunes.apple.com/de/review?id=38980125...	f31334bd164a58cffae342f04040b0f7cb2ad9416ed0b0...	https://itunes.apple.com/de/reviews/id1193916555	323.0.0

In [63]:
```python
# Scénario : Instagram has given you the task to draw insights from this data. It would like to know its top markets, what are the markets it needs
# to target to improve its reviews. It would like to understand how the ratings evolved since the beginning date
```

In [34]:
```python
average_country_data = data.groupby('country')['score'].mean().reset_index()
```

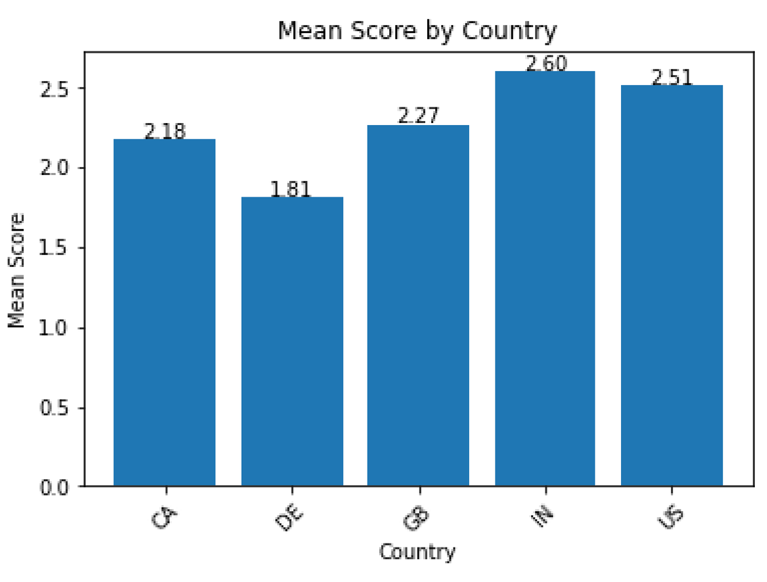
In [35]:
```python
print(average_country_data)
```

```
  country  score
0      CA   2.18
1      DE   1.81
2      GB   2.27
3      IN   2.60
4      US   2.51
```

In [36]:
```python
fig, average_plot = plt.subplots()
bars = average_plot.bar(average_country_data['country'], average_country_data['score'])


for bar in bars:
    height = bar.get_height()
    average_plot.text(bar.get_x() + bar.get_width() / 2, height - 0.02, f'{height:.2f}',
                      ha='center', va='bottom')

average_plot.set_ylabel('Mean Score')
average_plot.set_xlabel('Country')
average_plot.set_title('Mean Score by Country')
plt.xticks(rotation=45)
plt.show()
```



In [37]:
```python
#This first plot shows India has the highest rankings whereas Germany has the lowest. Therefore, Instagram should target Germany and see why the rankings are this low
```

In [38]:
```python
#For Germany we will try to understand how the reviews evolved, then we will try clustering methods to group reviews based on their similitude to extract the common feelings users have. It will help Instagram understand what it s
```

In [39]:
```python
#First, we notice the reviews were drawn from one month only, therefor we draw the conclusion it wouldn't be interesting to draw insights from the time
```

In [40]:
```python
#We will use the latent dirichlet allocation
```

In [41]:
```python
comments_title_de = []

titles_de = data[data['country'] == 'DE']['title']
```

In [42]:
```python
print(comments_title_de)
```

```
[]
```

In [43]:
```python
import re
emoji_pattern = re.compile("["
                           u"\U0001F600-\U0001F64F"
                                u"\U0001F300-\U0001F5FF"
                                u"\U0001F680-\U0001F6FF"
                                u"\U0001F1E0-\U0001F1FF"
                                u"\U00002702-\U000027B0"
                                u"\U000024C2-\U0001F251"
                                u"\U0001F900-\U0001F9FF"
                                u"\U0001FA00-\U0001FA6F"
                                u"\U0001FA70-\U0001FAFF"
                                u"\U00002600-\U000026FF"
                                u"\U00002B05-\U00002B07"
                                u"\U00002B1B-\U00002B1F"
                                u"\U00002B50"
                                u"\U00002B55"
                            "]+", flags=re.UNICODE)
```

In [44]:
```python
titles_de = [emoji_pattern.sub(r'', string) for string in titles_de]
```

In [45]:
```python
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk import word_tokenize
```

```
[nltk_data] Downloading package stopwords to C:\Users\LAMBERT
[nltk_data]     Matthieu\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to C:\Users\LAMBERT
[nltk_data]     Matthieu\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [46]:
```python
stop_words_de = stopwords.words('german')
stop_words_en = stopwords.words('english')


filtered_phrases = []
for phrase in titles_de:

    tokens = word_tokenize(phrase)

    filtered_words = [word for word in tokens if word.lower() not in stop_words_de and word.lower() not in stop_words_en and word.isalpha()]

    filtered_phrases.append(filtered_words)

    import itertools

flat_filtered_phrases = list(itertools.chain.from_iterable(filtered_phrases))
```

In [47]:
```python
print(flat_filtered_phrases)
```

```
['List', 'friends', 'weird', 'order', 'Great', 'companion', 'Instagram', 'Fehler', 'Verbesserung', 'dringend', 'nötig', 'Hässliche', 'Layout', 'Instagram', 'absoluter', 'Müll', 'geworden', 'sorry', 'echt', 'Bleibt', 'ganze', 'Zei
t', 'hängen', 'Schlechte', 'Qualität', 'story', 'INSTAGRAM', 'Unnötige', 'Veränderung', 'Like', 'Anzahl', 'Kommentaren', 'Nothing', 'works', 'Unnötiges', 'feature', 'App', 'immer', 'schlechter', 'bekommt', 'viele', 'Fake', 'Bewer
tungen', 'Alte', 'Beiträge', 'angezeigt', 'glaub', 'wohl', 'spinne', 'Bruh', 'Immer', 'Probleme', 'hängen', 'abstürzen', 'Seit', 'Update', 'Kommentar', 'Darkmode', 'isnt', 'dark', 'anymore', 'Story', 'Aufrufe', 'Update', 'Deaktiv
ierung', 'Spam', 'Filter', 'macht', 'App', 'kaputt', 'Insta', 'mal', 'geil', 'Werbung', 'möglich', 'Mobile', 'Daten', 'DISABLED', 'ACCOUNT', 'Aufploppen', 'Profil', 'privat', 'trotzdem', 'User', 'Story', 'anschauen', 'außerhalb',
'Freundesliste', 'Schöne', 'Bilder', 'immer', 'gern', 'gesehen', 'Reels', 'mehr', 'teilen', 'Weiterhin', 'Müll', 'Naja', 'Videos', 'hängen', 'Buggy', 'GIF', 'schon', 'gut', 'Funktioniert', 'mehr', 'Fotos', 'überbelichtet', 'Gif
s', 'Many', 'functions', 'working', 'immer', 'schlechter', 'Anmeldecode', 'lost', 'option', 'Upload', 'highest', 'quality', 'Konstante', 'Spam', 'Story', 'messed', 'Töne', 'trotz', 'Stummmodus', 'bad', 'Save', 'time', 'use', 'Liv
e', 'Status', 'GIF', 'Button', 'Algorithmus', 'kaputt', 'Reels', 'seit', 'letzten', 'Update', 'komplett', 'verbuggt', 'Unglaublich', 'schlecht', 'Seit', 'letzen', 'Update', 'Probleme', 'Intransparent', 'anstrengend', 'ständig',
'GIFS', 'verstehe', 'App', 'mehr', 'Fix', 'Sound', 'GIFs', 'Kommentaren', 'Funktioniert', 'Hoher', 'Speicherbedarf', 'Reels', 'haken', 'andauernd', 'destek', 'App', 'einwandfrei', 'großes', 'tötet', 'Awful', 'recommend', 'peopl
e', 'ask', 'upload', 'contacts', 'Extravaganza', 'finest', 'GIFs', 'Button', 'Eigentlich', 'Sterne', 'Kommentar', 'gelöscht', 'Problem', 'Naja', 'Warum', 'WARUM', 'nehmt', 'GIF', 'Button', 'schämt', 'Weshalb', 'Instagram', 'Frühe
r', 'besser', 'Hindernisse', 'Hürden', 'Fehler', 'seit', 'update', 'Leave', 'TikTok', 'alone', 'views', 'App', 'algorithm', 'function', 'getting', 'worse', 'worse', 'Unzufrieden', 'Storys', 'laden', 'hoch', 'Инст', 'подписываетс
я', 'на', 'людей', 'без', 'моего', 'ведома', 'Bewertung', 'schlecht', 'Störung', 'GIFs', 'insta', 'Kommentaren', 'deaktiviert', 'puedo', 'cambiar', 'entre', 'cuentas', 'works', 'better', 'Safari', 'Unverschämtheit', 'Since', 'Ap
p', 'ok', 'Text', 'einfügen', 'möglich', 'Problem', 'Upload', 'Fotos', 'Zensur', 'ACHTUNG', 'love', 'new', 'timet', 'system']
```

In [48]:
```python
import gensim
```

In [49]:
```python
import gensim.corpora as corpora
```

In [50]:
```python
documents = [flat_filtered_phrases]
dictionary = corpora.Dictionary(documents)
```

In [51]:
```python
texts =  documents
```

In [52]:
```python
corpus = [dictionary.doc2bow(text) for text in texts]
```

In [53]:
```python
print(corpus[:1][0][:40])
```

```
[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 6), (7, 1), (8, 1), (9, 1), (10, 1), (11, 1), (12, 1), (13, 1), (14, 1), (15, 1), (16, 1), (17, 3), (18, 1), (19, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 2), (26,
1), (27, 1), (28, 2), (29, 1), (30, 1), (31, 2), (32, 3), (33, 1), (34, 3), (35, 1), (36, 1), (37, 1), (38, 1), (39, 1)]
```

In [54]:
```python
num_topics = 20
lda_model = gensim.models.LdaMulticore(corpus = corpus, id2word = dictionary, num_topics = num_topics)
```

In [55]:
```python
print(lda_model.print_topics())
doc_lda = lda_model[corpus]
```

```
[(0, '0.008*"App" + 0.007*"Update" + 0.006*"Button" + 0.006*"GIF" + 0.006*"hängen" + 0.006*"Story" + 0.006*"GIFs" + 0.006*"immer" + 0.006*"Instagram" + 0.006*"Kommentaren"'), (1, '0.005*"geworden" + 0.005*"gesehen" + 0.005*"gut"
+ 0.005*"großes" + 0.005*"glaub" + 0.005*"highest" + 0.005*"getting" + 0.005*"geil" + 0.005*"gelöscht" + 0.005*"hoch"'), (2, '0.012*"App" + 0.009*"Update" + 0.009*"Kommentaren" + 0.008*"Instagram" + 0.008*"Fotos" + 0.008*"mehr" +
0.007*"Probleme" + 0.007*"immer" + 0.007*"GIFs" + 0.007*"GIF"'), (3, '0.011*"App" + 0.009*"Update" + 0.008*"Reels" + 0.008*"hängen" + 0.008*"Button" + 0.008*"immer" + 0.008*"GIF" + 0.007*"mehr" + 0.007*"Story" + 0.007*"Instagra
m"'), (4, '0.014*"App" + 0.011*"Update" + 0.011*"Story" + 0.009*"hängen" + 0.009*"GIF" + 0.008*"immer" + 0.008*"mehr" + 0.008*"Button" + 0.008*"GIFs" + 0.008*"Reels"'), (5, '0.013*"App" + 0.012*"Button" + 0.011*"Update" + 0.011
*"immer" + 0.010*"Kommentaren" + 0.010*"hängen" + 0.009*"Story" + 0.009*"mehr" + 0.009*"Müll" + 0.008*"GIF"'), (6, '0.023*"App" + 0.013*"Update" + 0.013*"Instagram" + 0.013*"Story" + 0.012*"Reels" + 0.011*"hängen" + 0.011*"Kommen
taren" + 0.011*"GIFs" + 0.011*"mehr" + 0.010*"Button"'), (7, '0.008*"Update" + 0.008*"App" + 0.007*"hängen" + 0.007*"GIFs" + 0.007*"Button" + 0.007*"Reels" + 0.006*"Kommentaren" + 0.006*"mehr" + 0.006*"immer" + 0.006*"GIF"'), (8,
'0.013*"App" + 0.012*"Update" + 0.010*"GIF" + 0.010*"hängen" + 0.009*"mehr" + 0.009*"Reels" + 0.009*"GIFs" + 0.009*"Kommentaren" + 0.008*"Button" + 0.008*"immer"'), (9, '0.011*"App" + 0.009*"Update" + 0.008*"GIFs" + 0.007*"mehr"
+ 0.007*"Instagram" + 0.007*"hängen" + 0.007*"Story" + 0.007*"Kommentaren" + 0.007*"immer"'), (10, '0.023*"App" + 0.013*"Update" + 0.011*"immer" + 0.010*"GIF" + 0.010*"mehr" + 0.010*"Kommentaren" + 0.010*"Story" +
0.009*"GIFs" + 0.009*"Button" + 0.009*"hängen"'), (11, '0.007*"App" + 0.006*"Story" + 0.006*"hängen" + 0.006*"Update" + 0.006*"Instagram" + 0.006*"mehr" + 0.006*"Reels" + 0.005*"immer" + 0.005*"Button" + 0.005*"Kommentaren"'), (1
2, '0.011*"App" + 0.008*"Reels" + 0.008*"mehr" + 0.007*"Instagram" + 0.007*"hängen" + 0.007*"immer" + 0.007*"Button" + 0.007*"Update" + 0.007*"GIFs" + 0.007*"GIF"'), (13, '0.013*"App" + 0.010*"Update" + 0.009*"GIF" + 0.008*"Reel
s" + 0.008*"hängen" + 0.007*"mehr" + 0.007*"Kommentaren" + 0.007*"Story" + 0.007*"immer" + 0.007*"GIFs"'), (14, '0.008*"App" + 0.007*"immer" + 0.007*"Update" + 0.006*"Reels" + 0.006*"Instagram" + 0.006*"hängen" + 0.006*"GIF" + 0.
006*"Kommentaren" + 0.006*"Button" + 0.006*"Probleme"'), (15, '0.013*"App" + 0.011*"Update" + 0.008*"Kommentaren" + 0.008*"mehr" + 0.008*"Story" + 0.008*"hängen" + 0.008*"GIF" + 0.008*"Button" + 0.008*"Instagram" + 0.007*"Reel
s"'), (16, '0.015*"App" + 0.013*"Update" + 0.012*"Instagram" + 0.012*"mehr" + 0.012*"Button" + 0.011*"GIF" + 0.011*"hängen" + 0.011*"Kommentaren" + 0.010*"Reels" + 0.009*"immer"'), (17, '0.011*"App" + 0.010*"Update" + 0.008*"GIF
s" + 0.008*"Button" + 0.008*"Instagram" + 0.008*"hängen" + 0.008*"Kommentaren" + 0.007*"Reels" + 0.007*"Story" + 0.007*"GIF"'), (18, '0.012*"App" + 0.008*"immer" + 0.008*"Reels" + 0.008*"Instagram" + 0.008*"GIF" + 0.008*"Update"
+ 0.008*"Button" + 0.007*"GIFs" + 0.007*"hängen" + 0.007*"schlecht"'), (19, '0.021*"App" + 0.014*"GIFs" + 0.013*"Update" + 0.011*"immer" + 0.010*"Reels" + 0.010*"GIF" + 0.010*"Story" + 0.009*"Kommentaren" + 0.009*"Button" + 0.009
*"mehr"')]
```

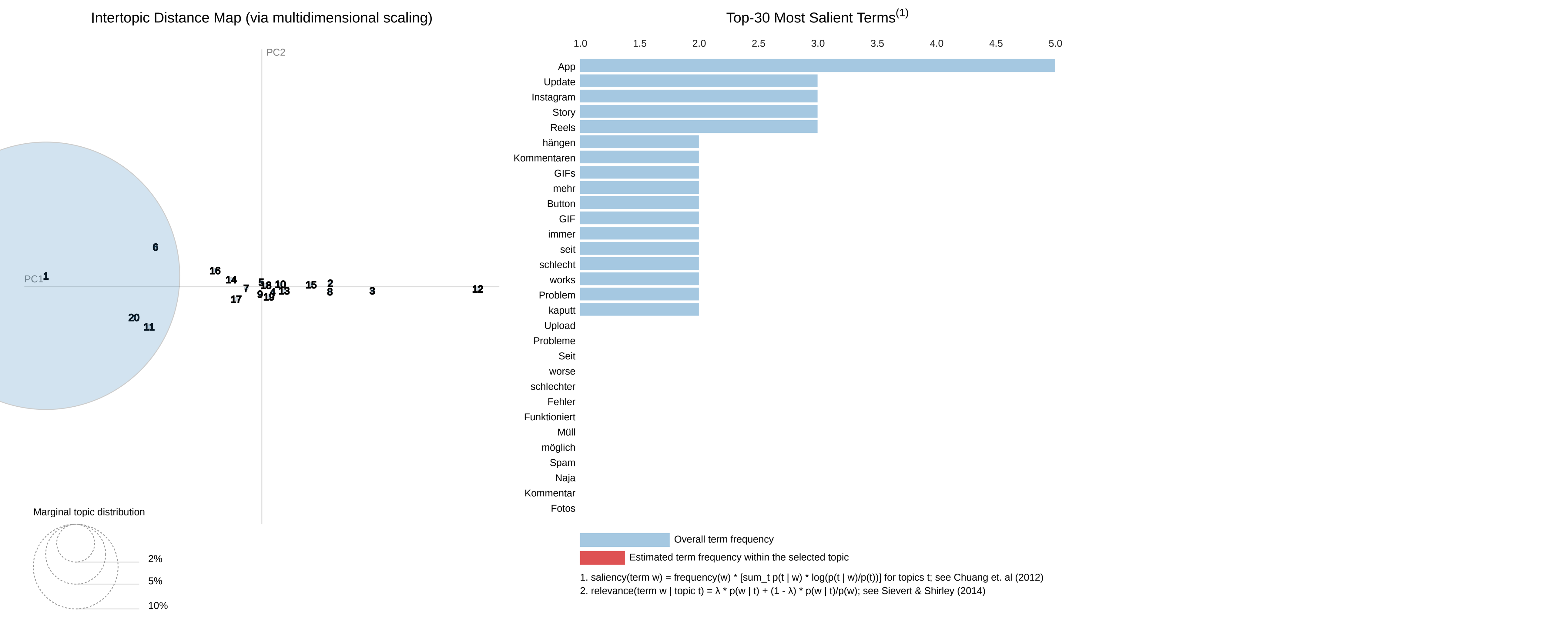
In [57]:
```python
import pyLDAvis
import pyLDAvis.gensim_models as gensimvis

prepared_vis = gensimvis.prepare(lda_model, corpus, dictionary)
pyLDAvis.display(prepared_vis)
```

Out[57]:

Selected Topic: 0 | Previous Topic | Next Topic | Clear Topic

Slide to adjust relevance metric:(2) λ = 1 — 0.0 0.2 0.4 0.6 0.8 1.0



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

In [62]:
```python
# we notice that some issues seem to come from with the gif bottun, story, reels and comments. We must criticizes these results as
# there are drawn from a small number of comments (one hundred) and from the titles only. We could extend this investigation to other country and base
# the study on the comment rather than titles
```

In [ ]: