

Analyse des Sentiments der Nachrichtenartikeln

```
In [2]: #Import Bibliothek
import pandas as pd
from textblob import TextBlob
import matplotlib.pyplot as plt
```

```
C:\Users\MarkE\AppData\Roaming\Python\Python310\site-packages\pandas\core\arrays\masked.py:60: UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

```
In [3]: #Abruf Daten
file_path = r'C:\Users\MarkE\OneDrive\_Career\2021-2024_Master Big Data\5_Semester\_Th
df = pd.read_csv(file_path)

#Datum in das Format Jahr.Monat umwandeln
df['Jahr.Monat'] = pd.to_datetime(df['Datum']).dt.to_period('M')

#Kategorisierung nach Zeitraum
def categorize_period(date):
    if date <= pd.Period('2022-11', 'M'):
        return '0. vor ChatGPT'
    elif pd.Period('2022-12', 'M') <= date <= pd.Period('2023-11', 'M'):
        return '1. ChatGPT-Jahr'
    elif pd.Period('2023-12', 'M') <= date <= pd.Period('2024-11', 'M'):
        return '2. ChatGPT-Jahr'
    return 'Andere'

df['Zeitraum'] = df['Jahr.Monat'].apply(categorize_period)

print(df.head())
```

	Quelle	Datum	Link \
0	FAZ	11/30/2023	https://www.faz.net/aktuell/wirtschaft/unterne...
1	FAZ	11/30/2023	https://www.faz.net/aktuell/wirtschaft/kuenstl...
2	FAZ	11/29/2023	https://www.faz.net/aktuell/feuilleton/medien/...
3	FAZ	11/28/2023	https://www.faz.net/pro/d-economy/kuenstliche-...
4	FAZ	11/28/2023	https://www.faz.net/pro/d-economy/prompt-der-w...

	Titel \
0	KI: Warum wir nicht mit Roboter-Autos vollauto...
1	Microsoft: Sind Jahrzehnte von einer künstlich...
2	Künstliche Intelligenz: Europa muss von neuen ...
3	Was die Superintelligenz-KI anrichten könnte, ...
4	Künstliche Intelligenz: Wie man sich seine Pro...

	Text	Anzahl Woerter	Text \
0	Roboterautos faszinieren viele - die Augen der...	1963	
1	Der Krieg von Mensch gegen Maschine verschiebt...	310	
2	Durch den Streik gegen die Hollywoodbosse habe...	1285	
3	ChatGPT macht Spaß, aber was ist, wenn es erns...	459	
4	Häufig gibt es wiederkehrende Anweisungen an d...	968	

	Text_bereinigt \
0	many fascinating robot car eye world rightly a...
1	war person machine shift accord Microsoft unli...
2	strike Hollywood boss creative United States a...
3	chatgpt fun get serious expert expect superint...
4	often recur instruction artificial intelligenc...

	Titel_bereinigt	KI Anteil \
0	NaN	0.042115
1	Microsoft decade remove artificial superintell...	0.058442
2	artificial intelligence Europe learn new rule ...	0.031593
3	superintelligence AI could today	0.063241
4	artificial intelligence organize prompt	0.046512

	Anzahl KI Wörter	Einmalige KI Wörter \
0	47	6
1	9	3
2	23	3
3	16	4
4	26	5

	KI Wörter	Jahr.Monat \
0	ai (33), artificial intelligence (4), robotics...	2023-11
1	ai (5), artificial intelligence (3), chatgpt (1)	2023-11
2	ai (17), artificial intelligence (2), chatgpt (4)	2023-11
3	ai (10), artificial intelligence (3), chatgpt ...	2023-11
4	ai (14), artificial intelligence (2), chatgpt ...	2023-11

	Zeitraum
0	1. ChatGPT-Jahr
1	1. ChatGPT-Jahr
2	1. ChatGPT-Jahr
3	1. ChatGPT-Jahr
4	1. ChatGPT-Jahr

```
In [13]: #Sentimentanalyse der Artikelinhalte mit Textblob
def analyse_sentiment(text):
    #Stimmungsanalyse für den Text durch
    try:
```

```

    if text: #Sicherstellen, dass der Text nicht None ist
        blob = TextBlob(text)
        return blob.sentiment.polarity, blob.sentiment.subjectivity
    else:
        return None, None #None-Werte zurückgeben, falls kein Text vorhanden ist
except Exception as e:
    print(f"Fehler bei der Stimmungsanalyse: {e}")
    return None, None

#Stimmungsanalyse auf den gesamten DataFrame anwenden
df[['Polaritaet Textblob [Text]', 'Subjektivitaet Textblob [Text]']] = df['Text_bereinigt'].apply(
    lambda text: pd.Series(analyse_sentiment(text)) if text else pd.Series([None, None])
)

#Berechne den Durchschnitt der Sentiment Analyse
durchschnitt_polaritaet = round(df['Polaritaet Textblob [Text]'].mean(), 3)
durchschnitt_subjektivitaet = round(df['Subjektivitaet Textblob [Text]'].mean(), 3)

print("Durchschnitt der Polaritaet:", durchschnitt_polaritaet)
print("Durchschnitt der Subjektivitaet:", durchschnitt_subjektivitaet)

#Anzeige des aktualisierten DataFrame, um Änderungen anzuzeigen
print(df[['Text_bereinigt', 'Polaritaet Textblob [Text]', 'Subjektivitaet Textblob [Text]']])

```

Durchschnitt der Polaritaet: 0.045

Durchschnitt der Subjektivitaet: 0.464

	Text_bereinigt \	
0	many fascinating robot car eye world rightly a...	
1	war person machine shift accord Microsoft unli...	
2	strike Hollywood boss creative United States a...	
3	chatgpt fun get serious expert expect superint...	
4	often recur instruction artificial intelligenc...	
...
2044	race artificial intelligence Google want take ...	
2045	Minister Transport also digital minister talk ...	
2046	since attack Hamas Israel israeli army Israel ...	
2047	Tina Kretschmer professor educational sciences...	
2048	Hamburg new police chief Falk Schnabel also re...	
	Polaritaet Textblob [Text]	Subjektivitaet Textblob [Text]
0	0.044258	0.416579
1	-0.119641	0.476712
2	0.035922	0.400387
3	0.015293	0.667680
4	0.065850	0.491788
...
2044	0.047496	0.475041
2045	-0.024912	0.480864
2046	-0.045683	0.453117
2047	0.041504	0.524860
2048	0.083374	0.544897

[2049 rows x 3 columns]

```

In [12]: #Sentimentanalyse der Ueberschriften mit Textblob
def analyse_sentiment(text):
    #Stimmungsanalyse für den Text
    try:
        if text is not None: #Sicherstellen, dass der Text nicht None ist
            text_str = str(text) #Konvertiere den Text zu einem String, wenn notwendig

```

```

        blob = TextBlob(text_str)
        return blob.sentiment.polarity, blob.sentiment.subjectivity
    else:
        return None, None #None-Werte zurückgeben, falls kein Text vorhanden ist
except Exception as e:
    print(f"Fehler bei der Stimmungsanalyse: {e}")
    return None, None

#Stimmungsanalyse auf den gesamten DataFrame anwenden
df[['Polaritaet Textblob [Titel]', 'Subjektivitaet Textblob [Titel]']] = df['Titel_bereinigt'].apply(
    lambda text: pd.Series(analyse_sentiment(text)) if text else pd.Series([None, None])
)

#Berechne den Durchschnitt der Sentiment Analyse
durchschnitt_polaritaet = round(df['Polaritaet Textblob [Titel]'].mean(),3)
durchschnitt_subjektivitaet = round(df['Subjektivitaet Textblob [Titel]'].mean(), 3)

print("Durchschnitt der Polaritaet:", durchschnitt_polaritaet)
print("Durchschnitt der Subjektivitaet:", durchschnitt_subjektivitaet)

#Anzeige des aktualisierten DataFrame, um Änderungen anzuzeigen
print(df[['Text_bereinigt', 'Polaritaet Textblob [Titel]', 'Subjektivitaet Textblob [Titel]']])

```

Durchschnitt der Polaritaet: -0.062
 Durchschnitt der Subjektivitaet: 0.284

	Text_bereinigt \		
0	many fascinating robot car eye world rightly a...		
1	war person machine shift accord Microsoft unli...		
2	strike Hollywood boss creative United States a...		
3	chatgpt fun get serious expert expect superint...		
4	often recur instruction artificial intelligenc...		
...	...		
2044	race artificial intelligence Google want take ...		
2045	Minister Transport also digital minister talk ...		
2046	since attack Hamas Israel israeli army Israel ...		
2047	Tina Kretschmer professor educational sciences...		
2048	Hamburg new police chief Falk Schnabel also re...		

	Polaritaet Textblob [Titel]	Subjektivitaet Textblob [Titel]	\
0	0.000000	0.000000	
1	-0.600000	1.000000	
2	-0.231818	0.727273	
3	0.000000	0.000000	
4	-0.600000	1.000000	
...	
2044	0.136364	0.454545	
2045	0.000000	0.000000	
2046	0.000000	0.000000	
2047	-0.600000	1.000000	
2048	0.136364	0.454545	

	Polaritaet Textblob [Text]	Subjektivitaet Textblob [Text]
0	0.044258	0.416579
1	-0.119641	0.476712
2	0.035922	0.400387
3	0.015293	0.667680
4	0.065850	0.491788
...
2044	0.047496	0.475041
2045	-0.024912	0.480864
2046	-0.045683	0.453117
2047	0.041504	0.524860
2048	0.083374	0.544897

[2049 rows x 5 columns]

```
In [6]: #Speichern des geänderten DataFrame in einer neuen CSV-Datei
df.to_csv('2_Daten_nach_Sentiment.csv', index=False)
```

```
In [7]: #Sentiment vor ChatGPT und nach ChatGPT
#Gruppieren nach 'Zeitraum' und Berechnen der erforderlichen Statistiken
result = df.groupby('Zeitraum').agg(
    Anzahl_Artikel=('Quelle', 'size'), # Zählen der Datensätze
    Polaritaet_text=('Polaritaet Textblob [Text]', 'mean'), # Durchschnitt der "Polaritaet"
    Subjektivitaet_text=('Subjektivitaet Textblob [Text]', 'mean'), # Durchschnitt der "Subjektivitaet"
    Polaritaet_titel=('Polaritaet Textblob [Titel]', 'mean'), # Durchschnitt der "Polaritaet"
    Subjektivitaet_titel=('Subjektivitaet Textblob [Titel]', 'mean') # Durchschnitt der "Subjektivitaet"
).reset_index()

#Runden der Ergebnisse auf drei Dezimalstellen
result[['Polaritaet_text', 'Subjektivitaet_text', 'Polaritaet_titel', 'Subjektivitaet_titel']]
```

```
print("Zusammenfassung der Daten nach Zeitraum:")
print(result)
```

Zusammenfassung der Daten nach Zeitraum:

	Zeitraum	Anzahl_Artikel	Polaritaet_text	Subjektivitaet_text \
0	0. vor ChatGPT	155	0.057	0.469
1	1. ChatGPT-Jahr	1178	0.041	0.469
2	2. ChatGPT-Jahr	716	0.049	0.456

	Polaritaet_titel	Subjektivitaet_titel
0	-0.083	0.314
1	-0.065	0.271
2	-0.051	0.300

```
In [8]: #Sentiment nach Monaten
#Gruppieren nach 'Jahr.Monat' und Berechnen der erforderlichen Statistiken
result = df.groupby('Jahr.Monat').agg(
    Anzahl_Artikel=('Quelle', 'size'), # Zählen der Datensätze
    Polaritaet_text=('Polaritaet Textblob [Text]', 'mean'), # Durchschnitt der "Polar
    Subjektivitaet_text=('Subjektivitaet Textblob [Text]', 'mean'), # Durchschnitt de
    Polaritaet_titel=('Polaritaet Textblob [Titel]', 'mean'), # Durchschnitt der "Pol
    Subjektivitaet_titel=('Subjektivitaet Textblob [Titel]', 'mean') # Durchschnitt c
).reset_index()

#Runden der Ergebnisse auf drei Dezimalstellen
result[['Polaritaet_text', 'Subjektivitaet_text', 'Polaritaet_titel', 'Subjektivitaet_

print("Zusammenfassung der Daten nach Zeitraum:")
print(result)

#Konvertiere den Index in Strings
result.index = result.index.astype(str)

#Erstelle das erste Liniendiagramm für die Polaritätswerte
plt.figure(figsize=(10, 5))
plt.plot(result.index, result['Polaritaet_text'], label='Text', marker='o', color='blue')
plt.plot(result.index, result['Polaritaet_titel'], label='Titel', marker='^', color='green')
plt.title('Durchschnittliche Polarität über Zeit')
plt.xlabel('Jahr.Monat')
plt.ylabel('Durchschnittliche Polarität')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()

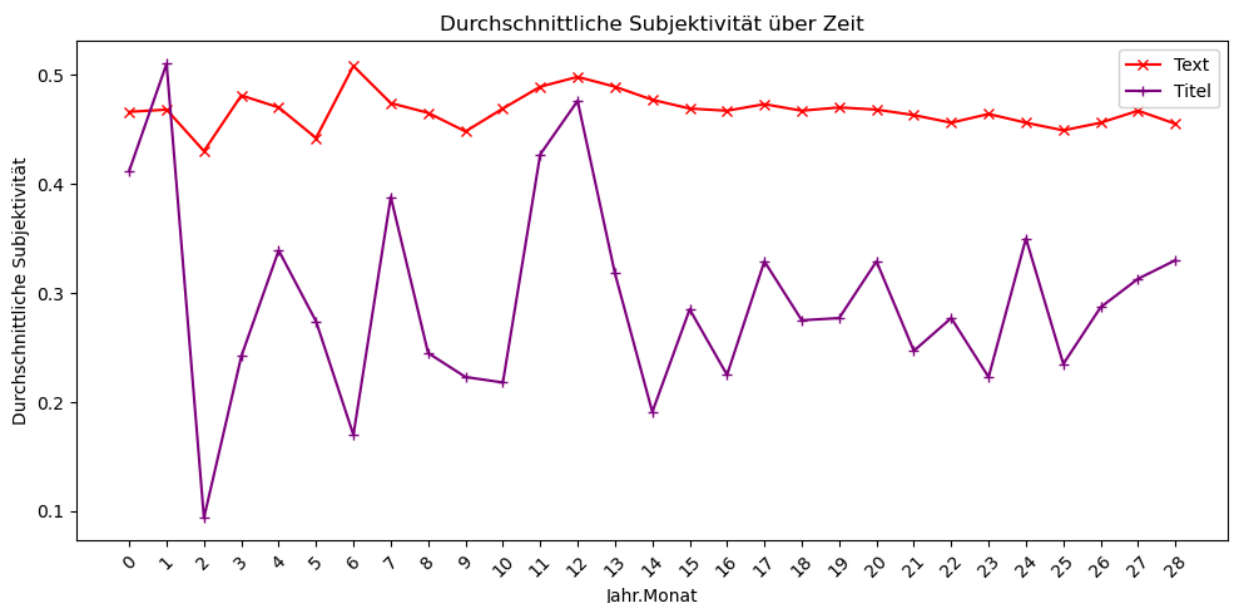
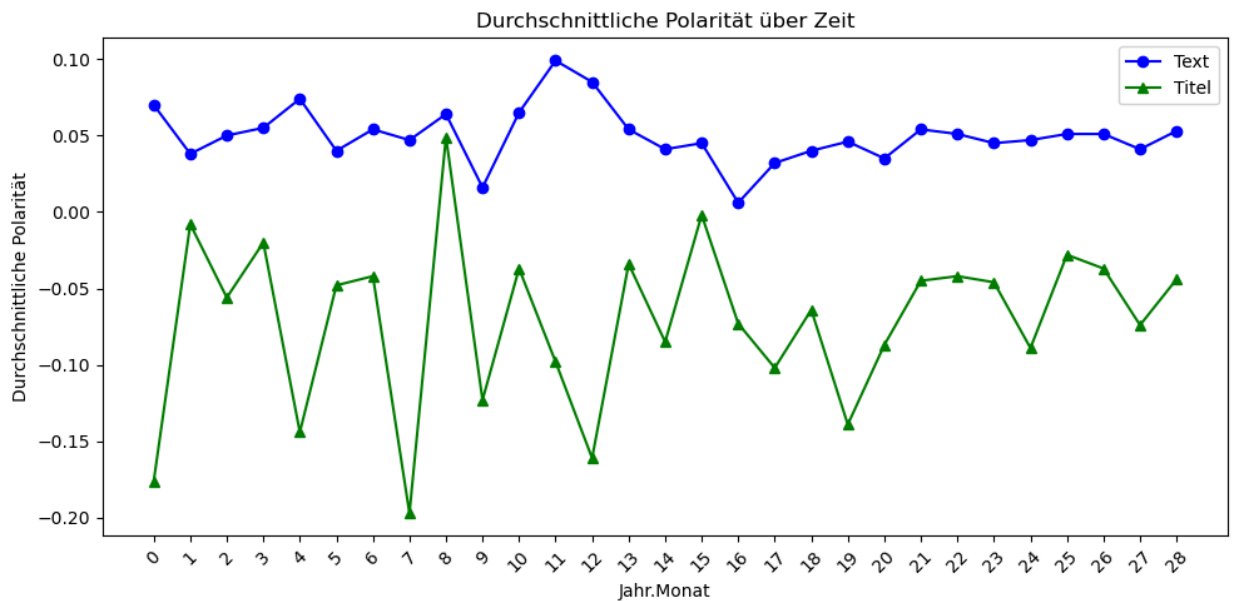
#Erstelle das zweite Liniendiagramm für die Subjektivitätswerte
plt.figure(figsize=(10, 5))
plt.plot(result.index, result['Subjektivitaet_text'], label='Text', marker='x', color='blue')
plt.plot(result.index, result['Subjektivitaet_titel'], label='Titel', marker='+', color='green')
plt.title('Durchschnittliche Subjektivität über Zeit')
plt.xlabel('Jahr.Monat')
plt.ylabel('Durchschnittliche Subjektivität')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```

Zusammenfassung der Daten nach Zeitraum:

	Jahr.Monat	Anzahl_Artikel	Polaritaet_text	Subjektivitaet_text	\
0	2021-12	19	0.070	0.466	
1	2022-01	16	0.038	0.468	
2	2022-02	6	0.050	0.430	
3	2022-03	10	0.055	0.481	
4	2022-04	9	0.074	0.470	
5	2022-05	11	0.040	0.442	
6	2022-06	12	0.054	0.508	
7	2022-07	16	0.047	0.474	
8	2022-08	7	0.064	0.465	
9	2022-09	12	0.016	0.448	
10	2022-10	23	0.065	0.469	
11	2022-11	14	0.099	0.489	
12	2022-12	25	0.085	0.498	
13	2023-01	53	0.054	0.489	
14	2023-02	70	0.041	0.477	
15	2023-03	123	0.045	0.469	
16	2023-04	111	0.006	0.467	
17	2023-05	132	0.032	0.473	
18	2023-06	91	0.040	0.467	
19	2023-07	96	0.046	0.470	
20	2023-08	65	0.035	0.468	
21	2023-09	105	0.054	0.463	
22	2023-10	109	0.051	0.456	
23	2023-11	198	0.045	0.464	
24	2023-12	103	0.047	0.456	
25	2024-01	154	0.051	0.449	
26	2024-02	161	0.051	0.456	
27	2024-03	134	0.041	0.467	
28	2024-04	164	0.053	0.455	

	Polaritaet_titel	Subjektivitaet_titel
0	-0.176	0.412
1	-0.008	0.510
2	-0.056	0.094
3	-0.020	0.242
4	-0.144	0.339
5	-0.048	0.274
6	-0.042	0.170
7	-0.197	0.388
8	0.049	0.245
9	-0.123	0.223
10	-0.037	0.218
11	-0.098	0.427
12	-0.161	0.476
13	-0.034	0.318
14	-0.085	0.191
15	-0.002	0.285
16	-0.073	0.225
17	-0.102	0.329
18	-0.064	0.275
19	-0.139	0.277
20	-0.087	0.329
21	-0.045	0.247
22	-0.042	0.277
23	-0.046	0.223
24	-0.089	0.350
25	-0.028	0.235
26	-0.037	0.287

27 -0.074 0.313
 28 -0.044 0.330



```
In [9]: #Sentiment je Nachrichtenportal
#Gruppieren nach 'Quelle' und Berechnen der erforderlichen Statistiken
result = df.groupby(['Quelle']).agg(
    Number_of_Records=('Quelle', 'size'), # Zählen der Datensätze
    Polaritaet_text=('Polaritaet Textblob [Text]', 'mean'), # Durchschnitt der "Polar
    Subjektivitaet_text=('Subjektivitaet Textblob [Text]', 'mean'), # Durchschnitt de
    Polaritaet_titel=('Polaritaet Textblob [Titel]', 'mean'), # Durchschnitt der "Pol
    Subjektivitaet_titel=('Subjektivitaet Textblob [Titel]', 'mean') # Durchschnitt d
).reset_index()

#Runden der Ergebnisse auf drei Dezimalstellen
result[['Polaritaet_text', 'Subjektivitaet_text', 'Polaritaet_titel', 'Subjektivitaet_

#Ergebnis nach 'Polaritaet_text' sortieren
result_sorted = result.sort_values(by='Polaritaet_text', ascending=False)

#Ausgabe des sortierten und gerundeten Ergebnisses
print(result_sorted)
```


	Quelle	Number_of_Records	Polaritaet_text	Subjektivitaet_text	\
6	Watson	58	0.065	0.474	
1	Focus	192	0.055	0.456	
0	FAZ	590	0.052	0.462	
4	Sueddeutsche	235	0.045	0.474	
7	Zeit	326	0.043	0.469	
2	RND	263	0.041	0.461	
5	Tagesspiegel	135	0.033	0.463	
3	Spiegel	250	0.032	0.464	

	Polaritaet_titel	Subjektivitaet_titel
6	-0.031	0.326
1	0.004	0.233
0	-0.056	0.297
4	-0.060	0.244
7	-0.109	0.318
2	-0.040	0.295
5	-0.076	0.223
3	-0.087	0.300

```
In [10]: #Sentiment je Nachrichtenportal vor ChatGPT und nach ChatGPT

#Gruppieren zuerst nach 'Quelle' und dann nach 'Jahr.Monat'
grouped = df.groupby(['Quelle', 'Zeitraum']).agg(
    Anzahl_Artikel=('Quelle', 'size'), # Zählen der Datensätze
    Polaritaet_text=('Polaritaet Textblob [Text]', 'mean'), # Durchschnitt der "Polar
    Subjektivitaet_text=('Subjektivitaet Textblob [Text]', 'mean'), # Durchschnitt de
    Polaritaet_titel=('Polaritaet Textblob [Titel]', 'mean'), # Durchschnitt der "Pol
    Subjektivitaet_titel=('Subjektivitaet Textblob [Titel]', 'mean') # Durchschnitt d
).reset_index()

#Runde alle numerischen Werte auf drei Dezimalstellen
grouped[['Polaritaet_text', 'Subjektivitaet_text', 'Polaritaet_titel', 'Subjektivitaet

#Durchlaufe alle eindeutigen Quellen
for quelle in grouped['Quelle'].unique():
    source_data = grouped[grouped['Quelle'] == quelle]
    source_data.set_index('Zeitraum', inplace=True)

    print(f"Tabelle für Quelle: {quelle}")
    print(source_data[['Anzahl_Artikel', 'Polaritaet_text', 'Subjektivitaet_text', 'Po
```

Tabelle für Quelle: FAZ

	Anzahl_Artikel	Polaritaet_text	Subjektivitaet_text \
Zeitraum			
0. vor ChatGPT	29	0.065	0.473
1. ChatGPT-Jahr	313	0.048	0.467
2. ChatGPT-Jahr	248	0.056	0.454

	Polaritaet_titel	Subjektivitaet_titel
Zeitraum		
0. vor ChatGPT	-0.172	0.445
1. ChatGPT-Jahr	-0.050	0.290
2. ChatGPT-Jahr	-0.051	0.288

Tabelle für Quelle: Focus

	Anzahl_Artikel	Polaritaet_text	Subjektivitaet_text \
Zeitraum			
0. vor ChatGPT	13	0.066	0.428
1. ChatGPT-Jahr	89	0.058	0.469
2. ChatGPT-Jahr	90	0.051	0.448

	Polaritaet_titel	Subjektivitaet_titel
Zeitraum		
0. vor ChatGPT	0.011	0.199
1. ChatGPT-Jahr	-0.035	0.241
2. ChatGPT-Jahr	0.043	0.231

Tabelle für Quelle: RND

	Anzahl_Artikel	Polaritaet_text	Subjektivitaet_text \
Zeitraum			
0. vor ChatGPT	21	0.046	0.449
1. ChatGPT-Jahr	157	0.035	0.465
2. ChatGPT-Jahr	85	0.049	0.455

	Polaritaet_titel	Subjektivitaet_titel
Zeitraum		
0. vor ChatGPT	-0.026	0.365
1. ChatGPT-Jahr	-0.063	0.297
2. ChatGPT-Jahr	0.000	0.274

Tabelle für Quelle: Spiegel

	Anzahl_Artikel	Polaritaet_text	Subjektivitaet_text \
Zeitraum			
0. vor ChatGPT	22	0.055	0.484
1. ChatGPT-Jahr	176	0.029	0.461
2. ChatGPT-Jahr	52	0.035	0.467

	Polaritaet_titel	Subjektivitaet_titel
Zeitraum		
0. vor ChatGPT	0.004	0.076
1. ChatGPT-Jahr	-0.102	0.317
2. ChatGPT-Jahr	-0.076	0.337

Tabelle für Quelle: Sueddeutsche

	Anzahl_Artikel	Polaritaet_text	Subjektivitaet_text \
Zeitraum			
0. vor ChatGPT	31	0.070	0.486
1. ChatGPT-Jahr	123	0.039	0.472
2. ChatGPT-Jahr	81	0.045	0.471

	Polaritaet_titel	Subjektivitaet_titel
Zeitraum		
0. vor ChatGPT	-0.089	0.308
1. ChatGPT-Jahr	-0.028	0.202
2. ChatGPT-Jahr	-0.098	0.283

Tabelle für Quelle: Tagesspiegel

	Anzahl_Artikel	Polaritaet_text	Subjektivitaet_text \
Zeitraum			
0. vor ChatGPT	12	0.041	0.438
1. ChatGPT-Jahr	92	0.033	0.468
2. ChatGPT-Jahr	31	0.029	0.456

	Polaritaet_titel	Subjektivitaet_titel
Zeitraum		
0. vor ChatGPT	-0.114	0.340
1. ChatGPT-Jahr	-0.056	0.153
2. ChatGPT-Jahr	-0.120	0.383

Tabelle für Quelle: Watson

	Anzahl_Artikel	Polaritaet_text	Subjektivitaet_text \
Zeitraum			
0. vor ChatGPT	5	0.082	0.440
1. ChatGPT-Jahr	30	0.050	0.499
2. ChatGPT-Jahr	23	0.080	0.449

	Polaritaet_titel	Subjektivitaet_titel
Zeitraum		
0. vor ChatGPT	0.080	0.160
1. ChatGPT-Jahr	-0.009	0.324
2. ChatGPT-Jahr	-0.085	0.365

Tabelle für Quelle: Zeit

	Anzahl_Artikel	Polaritaet_text	Subjektivitaet_text \
Zeitraum			
0. vor ChatGPT	22	0.039	0.494
1. ChatGPT-Jahr	198	0.044	0.473
2. ChatGPT-Jahr	106	0.040	0.456

	Polaritaet_titel	Subjektivitaet_titel
Zeitraum		
0. vor ChatGPT	-0.172	0.428
1. ChatGPT-Jahr	-0.108	0.282
2. ChatGPT-Jahr	-0.098	0.363

In []: