Question 2b, Part iii Suppose we wanted to investigate trends in how often the word `"AI"` is mentioned in NYT articles since the 1980s.

Is `news_df` a suitable dataset for this investigation? Explain your reasoning.

I dont think so. It is not suitable dataset for that because it only include articles between 2019 and 2024. I believe the technology at around 1980s was not that good to investigate trends.

0.0.1 Question 2f

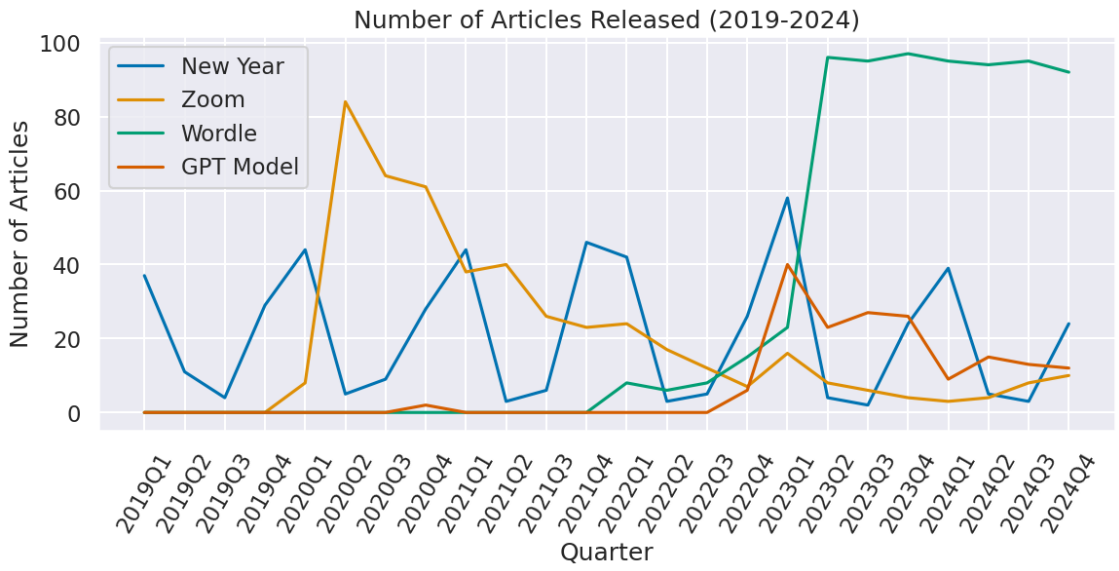
Let's visualize the article counts for each topic by quarter from 2019 to 2024.

Question 2f, Part i Using `sns.lineplot` (documentation) and `topic_mentions`, visualize the topic trends across quarters. Your plot should look like this:

In [388]:
```
plt.figure(figsize=(12, 5)) # DO NOT MODIFY

for topic in topics:
    sns.lineplot(data=topic_mentions, x=topic_mentions.index, y=topic, label=topic)

# DO NOT MODIFY THE CODE BELOW
# If your solution above is correct, running this cell should produce the plot above.
plt.xticks(rotation=60)
plt.yticks()
plt.ylabel("Number of Articles")
plt.xlabel("Quarter")
plt.title("Number of Articles Released (2019-2024)")
plt.gcf().set_facecolor('white')
plt.show()
```



Question 2f, Part ii For each of the four topics, identify one interesting pattern in the visualization and provide a tentative explanation of why you think the pattern exists.

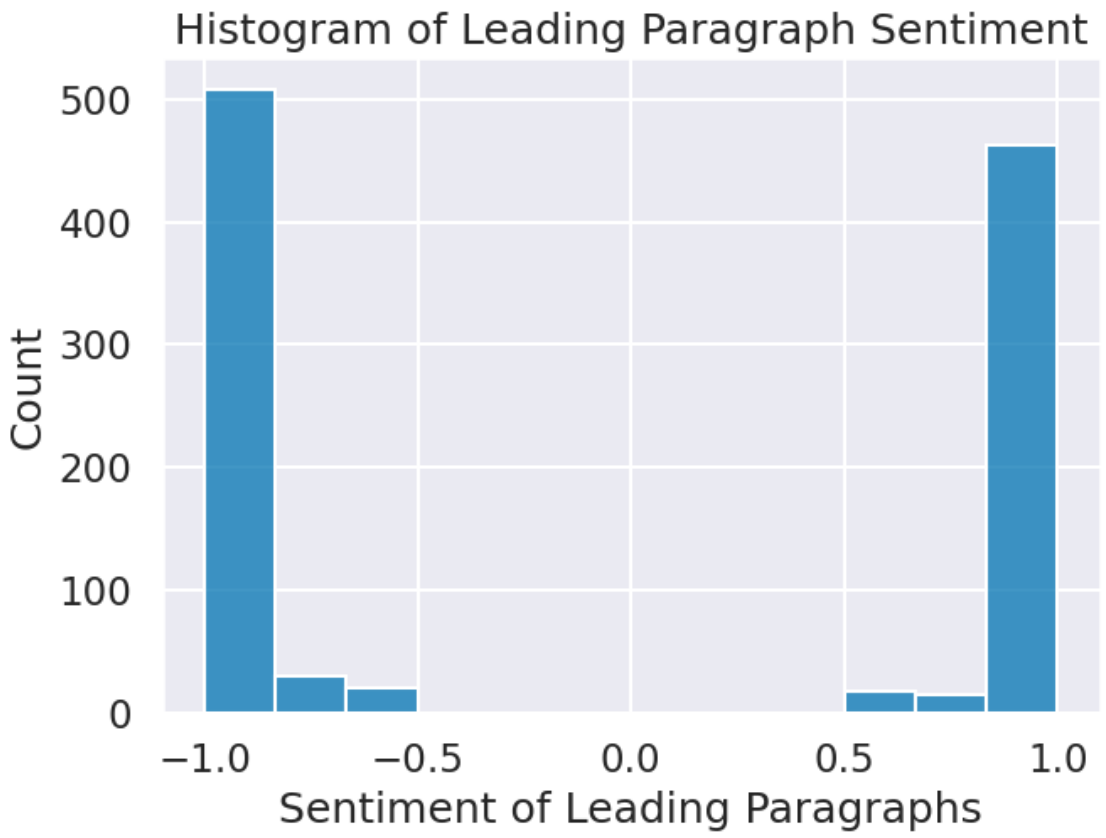
1. New Year: Pattern I see that it spike in Q4 and more in Q1 and down in Q2 and Q3. It might be because New Year happen in January.
2. Wordle: It spike starting from around 2022Q1 through 2023Q4. It was during the pandemic period and this corresponds with the viral rise of the wordle game.
3. Zoom: I saw that there is a sharp increase starting from 2020Q1. It was also during the pandemic period and everyone move to "work from home" and "Online classes". However, it become a gradual decline. It might be because less frequent as remote work and school became back to in-person.
4. GPT Model: I can see that it is significantly increased starting from 2022Q4 and remain the same high until now. I can say from that is the power of AI model became mainstream in daily life.

0.0.2 Question 3c

Let's now visualize the distribution of article sentiment.

Using `seaborn`, we created a histogram to visualize the distribution of `article_sentiment`. Run the cell below to display the plot.

```
In [399]: sns.histplot(data=news_df_sentiment, x='article_sentiment')
          plt.xlabel('Sentiment of Leading Paragraphs')
          plt.title('Histogram of Leading Paragraph Sentiment')
          plt.plot();
```



Are you at all surprised by the distribution of sentiment in the graph above? Describe what you notice

about the graph and how it relates to what you learned in part **3a**.

Yes, I am surprised that most of the article sentiments are either strongly positive or strongly negative, with very few neutral sentiments. This suggests that NYT articles tend to express strong opinions rather than neutral reporting. What I found in 3a is that sentences with negative aspects such as not," "reject," or other negative connotations will have a really high negative score, and vice versa for the positive. Additionally, sentences with a balanced mix of negative and positive words tend to receive neutral scores.

Question 3d, Part ii Do you agree with the current sentiment-based ordering of news articles, or would you rearrange the ordering? Do you feel that the DistilBERT model is a good model for our task of analyzing sentiment in news articles?

I agree with that because it is resonable but some articles may be misclassified due to the context misinterpretation. I think DistilBERT model is a good model for our task of analyzing sentiment.

0.0.3 Question 3e

Let's visualize our data more effectively. We will still use `sns.lineplot`, but instead of plotting every observation, we will first aggregate our data, and then plot the aggregated values.

We will also compare sentiment scores across three topics: `New Year`, `Zoom`, and `GPT`.

We will use the `DataFrame` `news_df_sentiment` in this question.

1. For each topic, generate a `DataFrame` that shows the average article sentiment for each quarter. In each `DataFrame`, be sure to include a column called `Topic` that has the same string value in every row (either `New Year`, `Zoom` or `GPT`).
2. Concatenate the `DataFrame`s obtained from step (1) using `pd.concat` (documentation). Assign this to `all_topic_qtr_avg_sentiments`.
3. Finally, we have provided the code to plot each topic's average article sentiment in each quarter using `all_topic_qrt_avg_sentiments`.

Your graph should have a similar title, axis labels, markers, and x-axis tick label ordering as the one below.

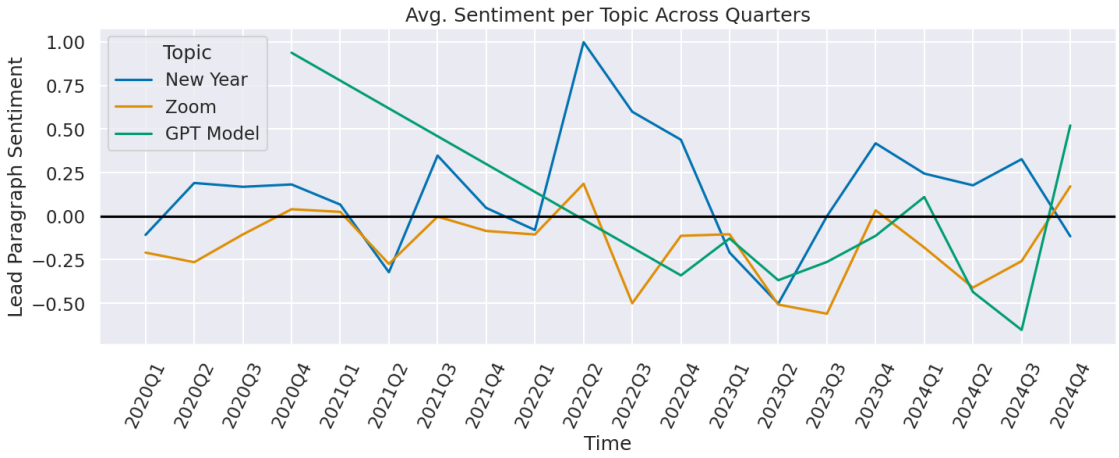
```
In [404]: fig, ax = plt.subplots(figsize=(15, 5))
          dfs_per_topic = []

          for topic in topics:
              df_of_current_topic = news_df_sentiment[news_df_sentiment[topic] > 0].groupby("Quarter", a
              df_of_current_topic["Topic"] = topic
              dfs_per_topic.append(df_of_current_topic)

          all_topic_qtr_avg_sentiments = pd.concat(dfs_per_topic)
          sns.lineplot(data=all_topic_qtr_avg_sentiments, x="Quarter", y="article_sentiment", hue="Topi

          plt.title('Avg. Sentiment per Topic Across Quarters')
          plt.xlabel('Time')
          plt.ylabel('Lead Paragraph Sentiment')

          # If the above are implemented correctly, running this cell should produce the graph shown ab
          plt.axhline(0, color='black')
          plt.xticks(rotation=65);
```


Avg. Sentiment per Topic Across Quarters
Topic
New Year
Zoom
GPT Model
Lead Paragraph Sentiment
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
2020Q1
2020Q2
2020Q3
2020Q4
2021Q1
2021Q2
2021Q3
2021Q4
2022Q1
2022Q2
2022Q3
2022Q4
2023Q1
2023Q2
2023Q3
2023Q4
2024Q1
2024Q2
2024Q3
2024Q4
Time