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.

`news_df` is not a suitable dataset for this investigation because it only contains the first 330 characters of each article, which may exclude mentions of "AI" that appear later in the text. This dataset also only includes articles from 2019 to 2024, lacking any data from the 1980s onward, making it unsuitable for analyzing trends. Furthermore, "AI" could be referenced as "Artificial Intelligence" or have other meanings, potentially leading to misinterpretations and undercounting in the analysis.

0.0.1 Question 2f

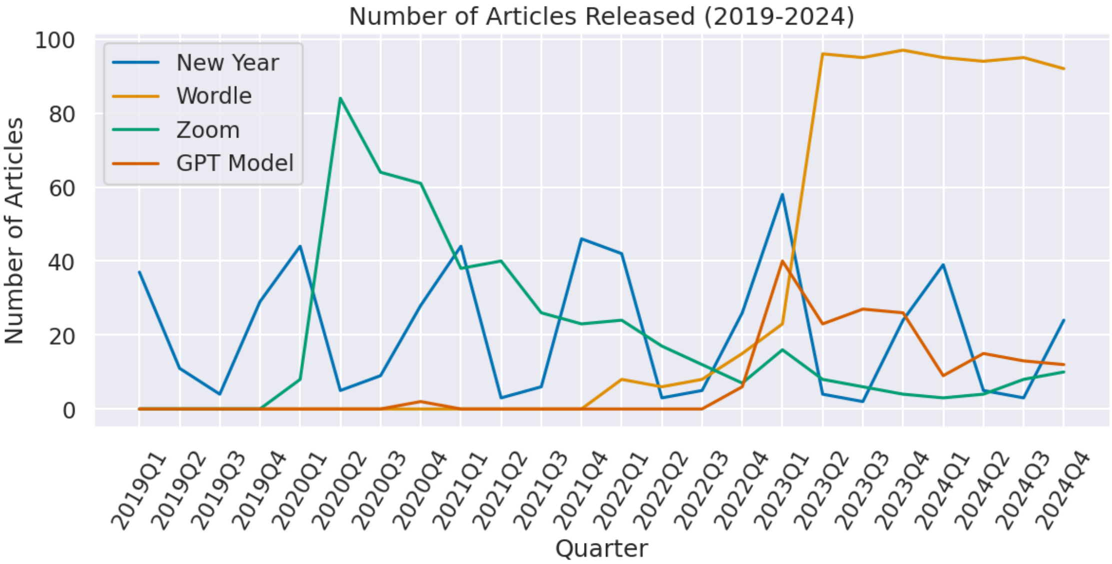
Let’s visualize the article counts for each topic by quarter from 2019 to 2024.

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

```
In [46]: plt.figure(figsize=(12, 5)) # DO NOT MODIFY

         for topic in topics:
             sns.lineplot(data=topic_mentions, x='Quarter', 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.

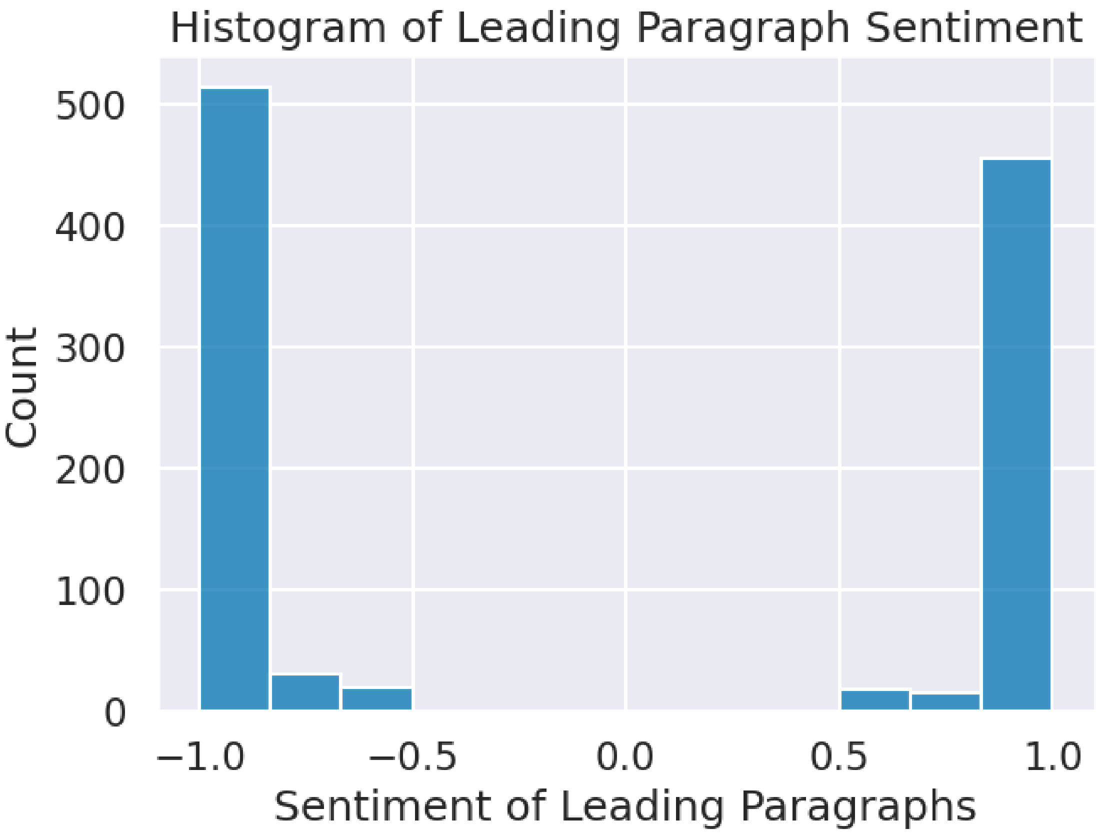
For each of the four topics, I noticed the peaks aligned with real-world events or trends. New Year consistently peaks in Q1 of most years, as it marks the start of a new calendar year, except for Q4 2021, which deviates from the usual pattern for Q1 2022. Wordle experienced its first peak in Q1 2022 when NYT purchased the game. It continued to grow and reached its highest peak in Q2 2023 and has been popular since, possibly due to viral trends and sustained interest in solving the daily puzzle. Zoom peaked in Q1 2020, coinciding with the start of the COVID-19 pandemic and the sudden shift to online communication. Finally, GPT Model peaked in Q1 2023, following the widespread adoption of ChatGPT, which was released in November 2022 but gained popularity in the following months.

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 [75]: 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**.

I am not surprised by the distribution of sentiment in the graph above. I notice that it is very difficult to create sentences that have a sentiment analysis score between -0.5 and 0.5. Even in my answer to 3a, I wrote a random sentence that contained both positive and negative elements, yet it still received a sentiment analysis score of -0.57. When I experimented with other sentences, most scored between 0.97 and 0.83 (positive and negative), reinforcing the idea that sentiments tend to be strongly positive or negative emotional tone rather than neutral. This aligns with the histogram, which peaks at the extreme ends of the sentiment scale, suggesting that leading paragraphs in the NYT articles often convey an emotional tone as opposed to a neutral tone.

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 do not agree with the current sentiment-based ordering of news articles and would rearrange the ordering. When reviewing the top positive results, the second-highest scored article appears random in its sentiment. While it discusses Twyla Tharp's work, it does not convey strong positive emotions, making its high score questionable. Similarly, when examining the top negative results, some classifications seem misleading. For instance, the second-most negative article includes the phrase "saw its stock hit a new high," which could be interpreted as positive for those invested in the company. This suggests that the model may struggle to accurately differentiate between positive, negative, and neutral sentiments, especially when context is required. I don't believe DistilBERT is a good model for our task of analyzing sentiment in news articles. It seems to rely heavily on certain words rather than understanding their contextual meaning.

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 `DataFrames` 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.

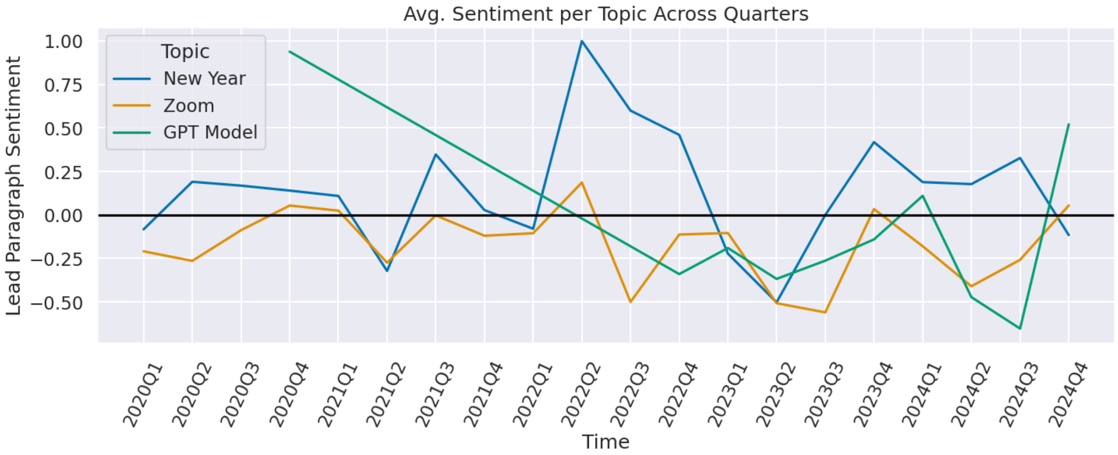
```
In [80]: fig, ax = plt.subplots(figsize=(15, 5))
         dfs_per_topic = []

         for topic in topics:
             df_of_current_topic = news_df_sentiment[news_df_sentiment[topic] == 1].groupby('Quarter')[
             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="Topic

         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 abo
         plt.axhline(0, color='black')
         plt.xticks(rotation=65);
```


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