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.

No, news_df is not suitable for investigating trends in how often the word "AI" is mentioned in NYT articles since the 1980s. According to the assignment, news_df only contains filtered NYT articles from 2019 to 2024, meaning it does not include data from the 1980s. Since trend analysis requires a dataset covering multiple decades, a more comprehensive dataset from the NYT Archive API with articles from the 1980s onward would be needed.

Additionally, news_df appears to contain only a subset of articles rather than a complete collection of all NYT articles, which may introduce selection bias. A more comprehensive dataset, such as one covering full NYT archives, would be more appropriate for this investigation.

0.0.1 Question 2f

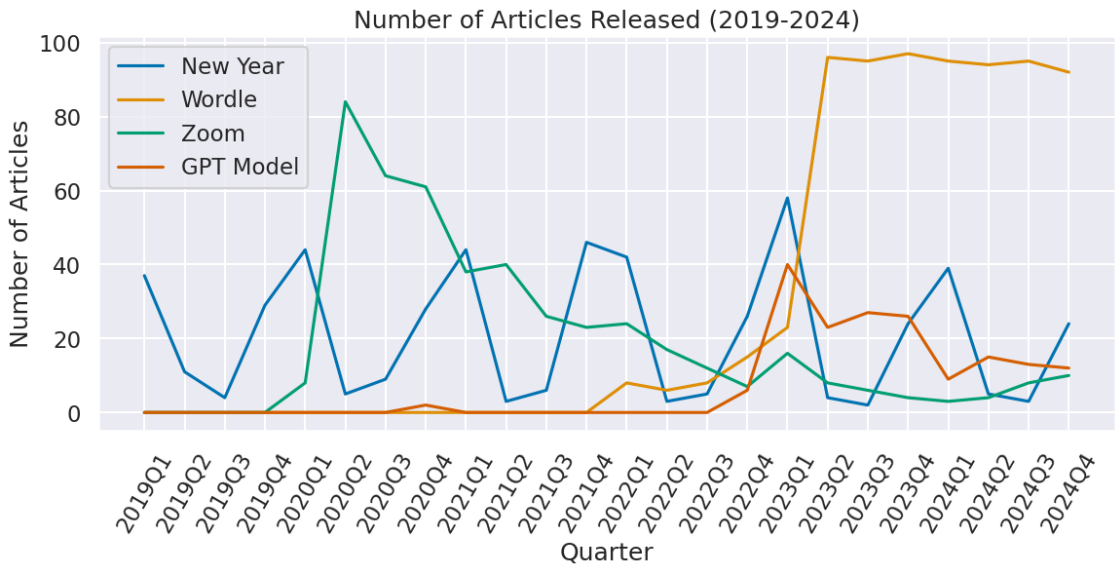
Let's visualize the article counts for each topic by quarter from 2019 to 2024.

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

```
In [554]: 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.

New Year: The mentions of "New Year" show a recurring spike in the first quarter of each year, which makes sense given that news coverage around New Year's Eve, celebrations, and resolutions tends to peak in January. After that, mentions drop off as other topics become more relevant.

Wordle: There is a sharp increase in early 2022, which corresponds with when Wordle became widely popular. The game went viral on social media, and major news outlets covered it extensively, leading to a surge in mentions. After that, the mentions stabilize at a lower level, likely as its novelty wore off.

Zoom: The mentions of "Zoom" peak dramatically around 2020Q2, which aligns with the onset of the COVID-19 pandemic. As workplaces, schools, and events transitioned to remote settings, discussions around Zoom skyrocketed. After the initial surge, the mentions gradually declined, likely due to hybrid work becoming more common and the initial novelty of video conferencing wearing off.

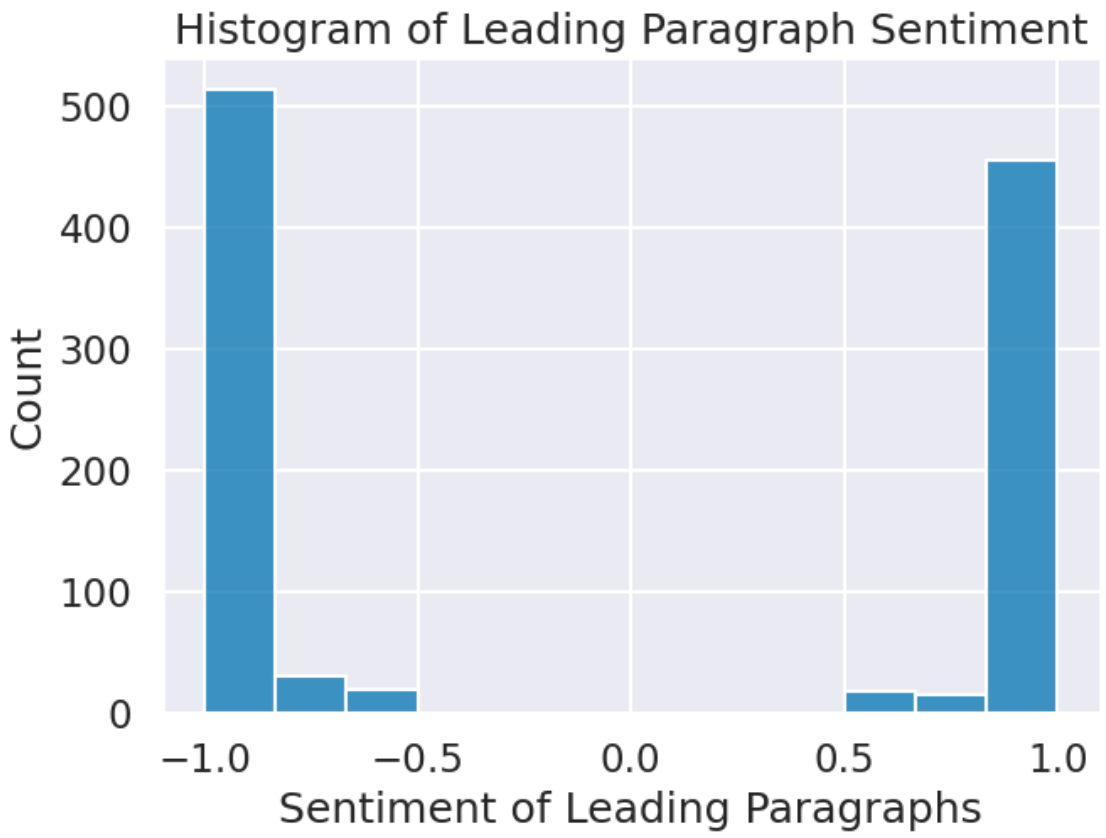
GPT Model: There's a clear increase in mentions starting around 2023, which likely reflects the rise of AI models like ChatGPT and other advancements in generative AI. The steady level of mentions after the spike suggests that AI remained a major topic in tech and business discussions.

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 [571]: 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**.

The distribution of sentiment in the histogram is not surprising. The graph shows that most articles have sentiment scores near -1 or 1, with fewer articles around 0. This suggests that the model tends to classify articles as either strongly positive or strongly negative, rather than neutral. This pattern aligns with Part 3a, where we examined how sentiment is derived from the text. Since the sentiment scores are based on a model that assigns polarity, the presence of strong sentiment peaks is expected

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?

The sentiment-based ordering largely makes sense, with positive articles featuring celebratory language and negative ones discussing layoffs and market declines. However, DistilBERT may overstate sentiment in news writing, which often aims for neutrality. For example, factual reports on stock declines might be classified as highly negative due to word associations. While the model performs well at a broad level, fine-tuning on journalistic data or incorporating context-aware scoring could improve accuracy.

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.

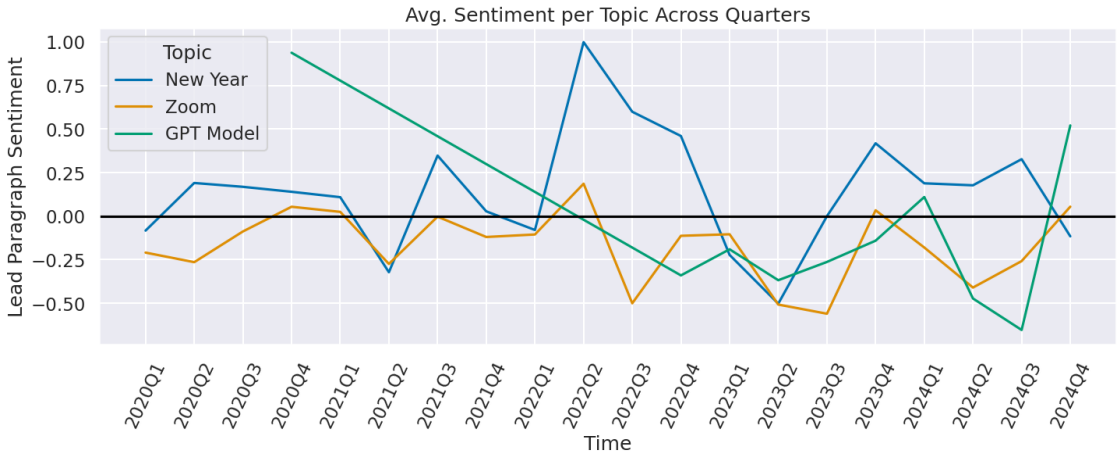
```
In [580]: fig, ax = plt.subplots(figsize=(15, 5))
          dfs_per_topic = []


          for topic in topics:
              df_of_current_topic = (
                  news_df_sentiment[news_df_sentiment[topic] > 0]
                  .groupby("Quarter")["article_sentiment"]
                  .mean()
                  .reset_index())
              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 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