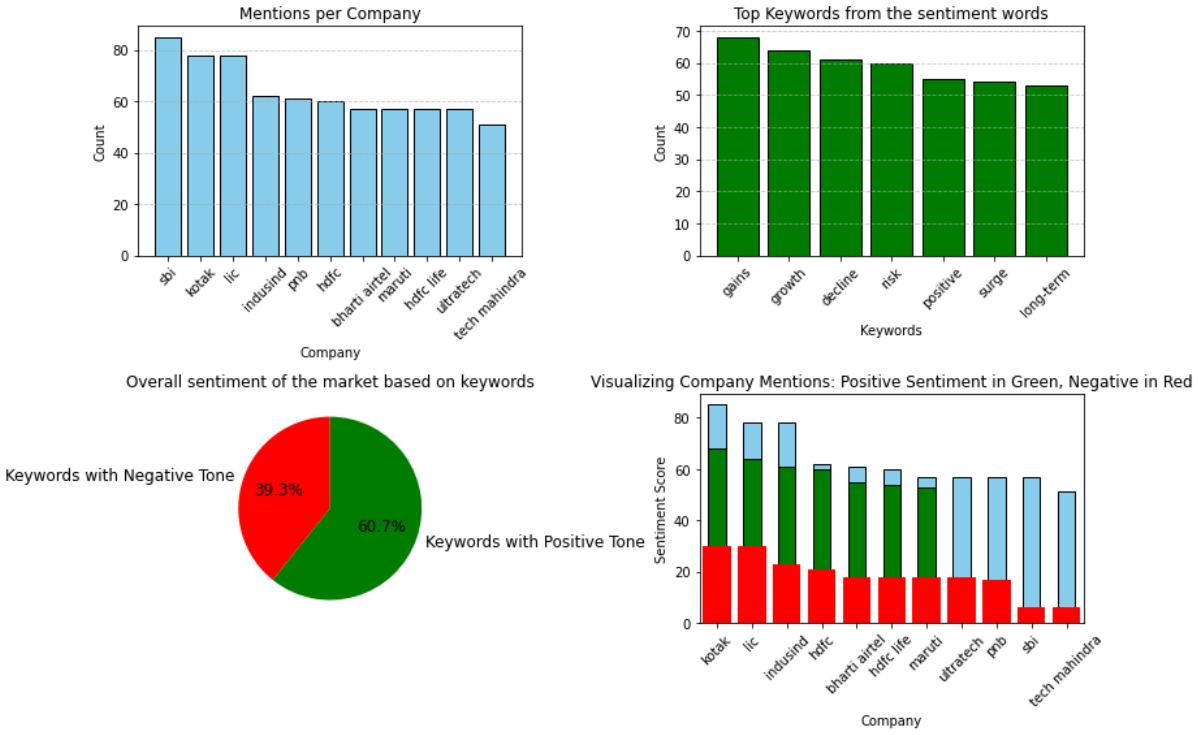
Author : Deepak Dalal
Date : 4 May, 2025
Topic : News Sentiment Analysis Dashboard: Making Market News Actionable

News Sentiment Analysis Dashboard: Making Market News Actionable

This dashboard performs **sentiment analysis on financial news headlines or content** by extracting keywords associated with either **positive** or **negative market sentiment**. It then visualizes the results using clean, interpretable plots.



1. Mentions per Company (Top-Left Chart)

This bar graph shows how often each company appears in the news dataset. A high number of mentions may indicate strong market attention—good or bad.

Use : Quickly identify which companies are trending in the news.

2. Top Keywords Driving Sentiment (Top-Right Chart)

Displays the most common **sentiment-driven keywords** (e.g., "gains", "growth", "bullish") and their frequencies.

Use : Understand which specific words are influencing sentiment and potentially driving stock movement.

3. Overall Market Sentiment (Bottom-Left Pie Chart)

Summarizes all keywords into **positive** and **negative sentiment proportions**. This gives a holistic view of market mood at a glance.

Use : Gauge if market tone is bullish or bearish based on news coverage.

4. Company-wise Sentiment Scores (Bottom-Right Chart)

This stacked bar chart shows the **positive (green)** and **negative (red)** keyword mentions for each company.

Use Case:

- Spot companies getting more praise (buy signals)
- Flag those under negative sentiment (risk alert)

The news content was extracted from **10 different financial websites**, providing a rich and diverse set of headlines related to listed companies.

💡 Future Improvements:

Improve **NER (Named Entity Recognition)** to identify companies more precisely.

Python code

For more such codes and projects, check out the [GitHub profile [deepakdalal081]]

```python
import pandas as pd
import numpy as np
import requests
from bs4 import BeautifulSoup
import re
import matplotlib as plt
import matplotlib.pyplot as plt

website = ["https://www.moneycontrol.com/"  , "https://www.ft.com/", "https://economictimes.indiatimes.com/",


def website_content(website) :

 try :
    request = requests.get(website, timeout = 10)

    soup = BeautifulSoup(request.text, "html.parser")
    #print(text)


    paragraph = soup.find_all([
    'p','h1', 'h2', 'h3', 'h4', 'h5', 'h6','section' , 'li', 'strong', 'em', 'span', 'article', 'div',
])

    #print(paragraph)

    sentence = []
    for para in paragraph :
        text = para.get_text(strip = True)

        if text :
            sentence.append(text)
    return sentence

 except:
  print("Site not fetched properly")
  return []

for url in website :
    content = website_content(url)
   # print(content)


sentiment_keywords = {
    # Strongly Positive (score: +2)
    "record high": 2, "breakthrough": 2, "beat estimates": 2, "beats estimates": 2,
    "significant growth": 2, "massive rally": 2, "record profit": 2, "strong buy": 2,
    "major contract win": 2, "blockbuster": 2, "multibagger": 2, "exceptional": 2,
    "best-ever": 2, "robust": 2, "surpassed": 2, "stellar": 2, "historic": 2
```

```
        # Moderately Positive (score: +1)
        "upgrade": 1, "bullish": 1, "outperform": 1, "positive": 1, "expansion": 1,
        "growth": 1, "profit rise": 1, "uptrend": 1, "recovery": 1, "gains": 1,
        "resilient": 1, "stable": 1, "consistent": 1, "long-term": 1, "favorable": 1,
        "acquisition": 1, "increased": 1, "momentum": 1, "fundamentals": 1, "strength": 1,
        "promising": 1, "improving": 1, "upside": 1, "solid results": 1, "dividend increase": 1,
        "strategic partnership": 1, "market share gain": 1, "cost-effective": 1,
        "innovation": 1, "competitive advantage": 1,

        # Moderately Negative (score: -1)
        "downgrade": -1, "bearish": -1, "underperform": -1, "weak": -1, "decline": -1,
        "drop": -1, "slump": -1, "underweight": -1, "reduce": -1, "sell": -1,
        "deceleration": -1, "soft": -1, "missed": -1, "margin pressure": -1, "risk": -1,
        "subdued": -1, "contraction": -1, "tepid": -1, "lagging": -1, "below average": -1,
        "uncertainty": -1, "disappointing": -1, "slowdown": -1, "cautious outlook": -1,
        "headwinds": -1, "challenging": -1, "cost pressure": -1, "restructuring": -1,
        "inventory issues": -1, "falling demand": -1,

        # Strongly Negative (score: -2)
        "missed estimates": -2, "profit warning": -2, "significant decline": -2, "fraud": -2,
        "investigation": -2, "default": -2, "bankruptcy": -2, "major loss": -2,
        "strong sell": -2, "crash": -2, "scandal": -2, "governance": -2, "terminated": -2,
        "breach": -2, "resignation": -2, "collapse": -2, "penalty": -2,
        "downgrade rating": -2, "shutdown": -2, "whistleblower": -2,
        "plummeted": -2, "catastrophic": -2, "dire situation": -2, "severe downturn": -2,
        "legal troubles": -2, "mass layoffs": -2, "financial irregularities": -2,
        "critical failure": -2, "regulatory crackdown": -2, "disastrous results": -2
    }

    def calulate_sentiment (sentence,sentiment_keywords):
        results = []
        for sentences in sentence :
            for keywords, score in sentiment_keywords.items() :
                if keywords in sentences.lower():
                    if re.search(rf'\b{re.escape(keywords)}\b', sentences):
                      result = {"sentence" : sentences, "Keywords" : keywords, "Score" : score }
                      results.append(result)

        return results

    companies = {
        "reliance": ["reliance industries", "ril", "reliance jio", "reliance retail"],
        "tata": ["tata motors", "tata steel", "tata consultancy services", "tcs", "tata power", "tata consumer products"],
        "hdfc": ["hdfc bank", "housing development finance corporation", "hdfc life", "hdfc ltd"],
        "infosys": ["infy", "infosys technologies", "infosys ltd"],
        "sbi": ["state bank of india", "sbi life", "sbi cards", "sbi"],
        "icici": ["icici bank", "icici prudential", "icici lombard", "icici securities"],
        "bajaj": ["bajaj auto", "bajaj finance", "bajaj finserv", "bajaj holdings"],
        "adani": ["adani enterprises", "adani ports", "adani power", "adani green", "adani transmission", "adani total gas",
```

```
}



def analyze_companies (sentence, sentiment_keywords,companies):
    print("analyzing len of", len(sentence))

    overall_results = []
    for sentences in sentence :

        for company_name, others_name in companies.items():
            for others_names in others_name :
                if re.search(rf'\b{re.escape(others_names.lower())}\b', sentences.lower()):

                    for keywords, score in sentiment_keywords.items() :
                        if re.search(rf'\b{re.escape(keywords)}\b', sentences.lower()):
                          overall_result = {
                                          "Sentence" : sentences,
                                          "Keywords" : keywords,
                                          "Company": company_name,
                                          "Score": score
                                          }
                        #print(overall_result)
                          overall_results.append(overall_result)
    return overall_results




for url in website :
  content = website_content(url)

  if content :
      sentiment = calulate_sentiment(content, sentiment_keywords)
      company_sentiment = analyze_companies(content, sentiment_keywords, companies)


      company_sentiment_df = pd.DataFrame(company_sentiment)

      company_counts = company_sentiment_df["Company"].value_counts()

      keywords_counts = company_sentiment_df["Keywords"].value_counts()

      top_keywords = keywords_counts.head(7)


      # for the 3rd quadrant
      overall_score = company_sentiment_df["Score"].value_counts()
```

```
224
225         # for the 3rd quadrant
226         overall_score = company_sentiment_df["Score"].value_counts()
227
228         negative = (company_sentiment_df["Score"] < 0 ).sum()
229         positive = (company_sentiment_df["Score"] > 0).sum()
230
231         labels = ['Keywords with Negative Tone', 'Keywords with Positive Tone']
232         sizes = [negative, positive]
233         colors = ['red', 'green']
234
235         # for the 4th quadrant
236
237         #positive_keywords = company_sentiment_df[""] > 0
238         company_scores = company_sentiment_df.groupby("Company")["Score"].sum().sort_values(ascending=False)
239
240
241
242         #print(company_sentiment_df[["Keywords","Score"]])
243         fig, axs = plt.subplots(2, 2, figsize=(12, 8))
244
245         company_counts.plot(kind='bar', color='skyblue', edgecolor='black')
246
247         axs[0, 0].bar(company_counts.index, company_counts.values, color='skyblue', edgecolor='black')
248         axs[0, 0].set_title("Mentions per Company")
249         axs[0, 0].set_xlabel("Company")
250         axs[0, 0].set_ylabel("Count")
251         axs[0, 0].tick_params(axis='x', rotation=45)
252         axs[0, 0].grid(axis='y', linestyle='--', alpha=0.7)
253
254         top_keywords.plot(kind='bar', color='green', edgecolor='black')
255
256         axs[0, 1].bar(top_keywords.index, top_keywords.values, color='green', edgecolor='black')
257         axs[0, 1].set_title("Top Keywords from the sentiment words")
258         axs[0, 1].set_xlabel("Keywords")
259         axs[0, 1].set_ylabel("Count")
260         axs[0, 1].tick_params(axis='x', rotation=45)
261         axs[0, 1].grid(axis='y', linestyle='--', alpha=0.7)
262
263          #Plot the sentences in bottom-right (3rd quadrant)
264
265        # For the 3rd quadrant (bottom left): Company sentiment summary
266         axs[1, 0].pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90, textprops={'fontsize': 12})
267         axs[1, 0].set_title("Overall sentiment of the market based on keywords")
268
269
270
271         company_scores.plot(kind='bar', color=['red' if x > 0 else 'black' for x in company_scores])
```

```
255
256         axs[0, 1].bar(top_keywords.index, top_keywords.values, color='green', edgecolor='black')
257         axs[0, 1].set_title("Top Keywords from the sentiment words")
258         axs[0, 1].set_xlabel("Keywords")
259         axs[0, 1].set_ylabel("Count")
260         axs[0, 1].tick_params(axis='x', rotation=45)
261         axs[0, 1].grid(axis='y', linestyle='--', alpha=0.7)
262
263          #Plot the sentences in bottom-right (3rd quadrant)
264
265        # For the 3rd quadrant (bottom left): Company sentiment summary
266         axs[1, 0].pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90, textprops={'fontsize': 12})
267         axs[1, 0].set_title("Overall sentiment of the market based on keywords")
268
269
270
271         company_scores.plot(kind='bar', color=['red' if x > 0 else 'black' for x in company_scores])
272
273
274         ax = axs[1, 1]
275         colors = ['red' if x > 0 else 'black' for x in company_scores]
276
277         ax.bar(company_scores.index, company_scores.values, color=colors)
278         ax.set_title("Visualizing Company Mentions: Positive Sentiment in Green, Negative in Red")
279         ax.set_xlabel("Company")
280         ax.set_ylabel("Sentiment Score")
281         ax.tick_params(axis='x', rotation=45)
282
283         plt.tight_layout(rect=[0, 0, 1, 0.95])
284
285         plt.tight_layout()
286         plt.show()
```