ScrapingGenAI

CS4624 Multimedia, Hypertext, and Information Access
Virginia Tech, Blacksburg, VA 24061
May 10, 2024

Group: James Do, Heewoon Bae, Julius Colby

Client: Dr. Florian Zach

Instructor: Dr. Edward A. Fox
# Table of Contents

Table of Figures .................................................................................................................. 3  
Table of Tables .................................................................................................................... 4  
Abstract .................................................................................................................................. 5  
Introduction (Problem, Motivation, General Approach) ....................................................... 7  
Requirements ......................................................................................................................... 8  
Design ..................................................................................................................................... 9  
Implementation ..................................................................................................................... 12  
  Web scraping ......................................................................................................................... 12  
  Parsing ................................................................................................................................... 13  
  Post Processing ..................................................................................................................... 14  
  Visualizations ....................................................................................................................... 15  
Testing/Evaluation/Assessment .............................................................................................. 18  
Users’ Manual ....................................................................................................................... 24  
  Reddit Web Scraper ............................................................................................................... 24  
  MarketBeat ........................................................................................................................... 27  
Developer’s Manual ............................................................................................................... 29  
  Project file and requirements ............................................................................................... 29  
  Prerequisites and Installation ............................................................................................... 29  
  Running the scripts ............................................................................................................... 30  
Lessons Learned .................................................................................................................. 31  
  Timeline ............................................................................................................................... 31  
  Problems .............................................................................................................................. 32  
  Solutions ............................................................................................................................... 33  
  Future work .......................................................................................................................... 33  
Acknowledgements ............................................................................................................... 34  
References ............................................................................................................................. 35  
Appendix ............................................................................................................................... Error! Bookmark not defined.
Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Portion of the Reddit scraper</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>Portion of the MarketBeat parser script</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Visualization produced by marketbeatVisualization.py</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>Links produced by script</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>User inputs for user_friendly.py</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>Reddit Visualization from user_friendly.py</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>Logfile</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>CSV file output of MarketBeat Parser</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>JSON file output of MarketBeat Parser</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>CSV file output of MarketBeat Analyzer</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>JSON file output of MarketBeat Analyzer</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>Visualization created by MarketBeat Visualizer</td>
<td>21</td>
</tr>
<tr>
<td>13</td>
<td>Test inputs for the Reddit web scraper</td>
<td>21</td>
</tr>
<tr>
<td>14</td>
<td>Test outputs for the Reddit web scraper</td>
<td>22</td>
</tr>
<tr>
<td>15</td>
<td>Manual check with the test outputs</td>
<td>23</td>
</tr>
<tr>
<td>16</td>
<td>File Directory</td>
<td>25</td>
</tr>
<tr>
<td>17</td>
<td>Running Reddit scraper script</td>
<td>26</td>
</tr>
<tr>
<td>18</td>
<td>Keyword input for Reddit scraper</td>
<td>26</td>
</tr>
<tr>
<td>19</td>
<td>Reddit JSON output naming</td>
<td>26</td>
</tr>
<tr>
<td>20</td>
<td>Reddit JSON file example</td>
<td>27</td>
</tr>
<tr>
<td>21</td>
<td>Keyword input example</td>
<td>28</td>
</tr>
<tr>
<td>22</td>
<td>Visualizer input example</td>
<td>28</td>
</tr>
<tr>
<td>23</td>
<td>Generated Marketbeat files</td>
<td>30</td>
</tr>
</tbody>
</table>
Table of Tables

Table 1: Output of MarketBeat Parser .................................................................13
Table 2: Output for Reddit Parser ......................................................................14
Table 3: Output of marketbeatNLTK.py .............................................................15
Abstract

AI has been widely used for many years and has been a constant front-page news topic. The recent but fast development of generative AI inspired many conversations, from concerns to aspirations. Understanding how the topic develops and when people become more supportive of generative AI is critical for social scientists to pinpoint which developments inspire public discussions. The use of generative AI is relatively new. The data and insight gathered could be used to determine if use in a commercial setting (like in Travel/Hospitality) is viable and what the potential feedback from the public might look like.

We developed two specialized web scrapers. The first targets specific keywords within Reddit subreddits to gauge public opinion, and the second extracts discussions from corporate earnings calls to capture the business perspective. The collected data were then processed and analyzed using Python libraries, with visualizations created in Matplotlib, Pandas, and Tkinter to depict trends through line charts, pie charts, and bar charts. We limited our analysis period from August 2022 to March 2024, which is significant as ChatGPT was released in November 2022, allowing us to observe notable changes. These tools not only show changes in public interest and sentiment but also provide a graphical representation of temporal shifts in the perception of AI technologies over time.

The final product is designed for anyone interested in company transcripts and in comparing them to the public perspective. The product offers users access to detailed data representations, including numerical trends and visual summaries to further understand the correlation between the company and the public. This comprehensive overview assists in understanding how public and corporate sentiments towards AI have shifted during a recent 20-month period.

A significant hurdle was using the PRAW API for Reddit data scraping. Through review of documentation, tutorials, and additional support from a teaching assistant, we successfully
implemented the functionality needed to extract and process the data from subreddits effectively.

To make our findings more accessible and engaging, future additional work transforming this product into a fully functional website would be beneficial. This platform would make the insights more readily available to a wider audience, including the general public and industry stakeholders. Doing so could enhance the impact and usefulness of our project.
Introduction (Problem, Motivation, General Approach)

Generative Artificial Intelligence (AI) is a new technology that has been developed in the form of ChatGPT and Google Gemini, just to name a few [10]. There has been a lot of coverage of generative AI. Due to this, our client, Florian Zach, an assistant professor in Hospitality and Tourism, is interested in gauging the potential for this innovative technology in future applications for the hospitality and tourism industry. An older but still massive innovation, the internet, changed hospitality and tourism entirely. People, when scheduling vacations and trips, use websites like Tripadvisor[8] and Google Maps[9] to check the location of their trip before even making reservations. There is potential for generative AI to impact the world in the same way. The goal of the project is to gather insights on this potential by combing through online sources such as Reddit[1] and MarketBeat. The general approach to do this fall into 4 phases.

1. Scrape the root of online sources to extract relevant links.
2. Comb through each link to extract the text of the page.
3. Post-process the text to highlight key sections.
4. Create visualizations.

Doing this compiles information around the web which is then analyzed to find and showcase certain patterns that emerge. This will allow for a more educated decision to be made when it comes to deciding whether this innovative technology should or shouldn’t be focused on in the future.
Requirements

To accomplish the goal of having better insights regarding how to incorporate generative AI into the tourism/hospitality industry, we needed to first understand generative AI itself and how it applies to other aspects. To do so, we went and developed our own web scraper to sift through Reddit and pull multiple certain posts by searching for key words in different subreddits about generative AI. This web scraper was possible due to importing the PRAW[2] library into the base code and utilizing the numerous functions that allow the web scraper to be able to do what it is supposed to do[4].

We also developed a different web scraper to scrape earning calls of companies in the tourism/hospitality industry. By utilizing the BeautifulSoup[11] import, we were able to scrape websites and have access to a company’s earning calls. The scripts had to be able to parse through each associated webpage and extract the text. When doing so, it should organize the text into a database. The database contains relevant metadata about each entry, including which company the transcript is from, their associated stock ticker, and the date of the transcript itself. Once the database is created, natural language processing is applied to the data. The algorithm does a keyword search on the data, returning to the user only the sections of the transcript that include a keyword. The section is also analyzed for its sentiment to see if the section talks positively or negatively.

To effectively visualize the processed data, appropriate libraries are necessary to adequately display trends and numerical data. We chose Matplotlib and Pandas because these libraries are well-suited for creating a wide range of static and interactive visualizations, which qualifies for detailed and dynamic data representation.
Design

The Reddit web scraper was developed using the PRAW library to be able to have access to certain functions that allow the scraper to look through Reddit and pull data. Certain data that were pulled using the web scraper were posts from a multitude of subreddits that fit in line with our project goals. Subreddits such as r/ Artificial and r/ Generative were scraped of posts and comments from different users to view different perspectives on certain discussions about generative AI. By using the Reddit developer mode and starting an application, we are then able to link the app with the web scraper code to use an existing account to go into Reddit and scrape all the data that fit the parameters set. Figure 1 shows a base code for the Reddit web scraper with the client id, client secret, and user agent keys to link to the Reddit developer app[3] and be able to scrape Reddit.

```
1 import praw
2
3 def fetch_posts_and_comments(subreddit, keywords, limit=10):
4     for keyword in keywords:
5         print(f"Searching for '{keyword}' in {subreddit.display_name}..."
6         # Search for the keyword in the subreddit
7         for post in subreddit.search(keyword, limit=limit):
8             print(f"Initial: {post.title.encode('ascii', 'ignore')}"
9                 print(f"URL: {post.url}"
10                print("Top comments:"))
11
12            # Limit comments to the top 5 comments; adjust as needed
13            post.comments.replace_more(limit=0) # Show top 5 comments; adjust as needed
14
15            for comment in post.comments.list():
16                print(f"- {comment.body.encode('ascii', 'ignore')[:108]}..." # Showing first 100 chars of each comment
17
18    # Initialize a Reddit instance
19    reddit = praw.Reddit(
20        client_id='UX8rCvbwEw1t498',
21        client_secret='TaMr9oE0QXePDx0gC0wYvL9yvA',
22        user_agent='ActionMaster"
23    )
24    # Choose a subreddit
25    subreddit = reddit.subreddit('Artificial')
26
27 # List of keywords to search for
28    keywords = ['travel', 'tourism', 'hotel', 'hospitality', 'vacation', 'trip']
29
30 # Fetch posts and comments for the keywords
31    fetch_posts_and_comments(subreddit, keywords, limit=10)
```

**Figure 1: Portion of the Reddit scraper**

For the MarketBeat[7] web scraper, the BeautifulSoup Python library was used. The web scraper utilized BeautifulSoup's functions to traverse a root website containing the earning calls
of different companies. It scraped the root website to gather and pass on links to individual earning calls to another script that parses the pages.

```python
import requests
from bs4 import BeautifulSoup

URL = "https://www.marketbeat.com/earnings/transcripts/"
page = requests.get(URL)
soup = BeautifulSoup(page.content, "html.parser")
results = soup.find(id="phPrimaryContent_pnlFilterTable")
body_results = results.find("tbody")
job_elements = body_results.find_all("td");

for element in job_elements:
    #print(element.prettify())

    link_el = element.find_all("td")
    if link_el:
        link_url = link_el[2]["data-clean"]
        file.write(link_url + '\n')
        print(link_url)
```

**Figure 2: Portion of the MarketBeat parser script**

The parser is also made using BeautifulSoup. Figure 2 shows the base code for the MarketBeat parser that would then be improved upon later. The parser populates a CSV database named after the date (ex. (March 20, 2024) marketbeatTranscript.csv) based on the information found within each individual earning call page. The file contains the stock ticker of the company, the company name, the quarter and year of the earning call, and the sections of the transcript. Each individual earning call is separated into an individual row. Each row is then fed into a natural language processing model.

The model uses the default training data provided by the NLTK[14] Python library. The model uses a list of keywords to determine if a section of the text is relevant or not. If the section contains a keyword, then it is relevant. If relevant, the model analyzes the section that is relevant for positive or negative sentiment. This is recorded in another CSV file (ex. (March 20, 2024) marketbeatNLTK.csv) that can be used later for visualization.
Visualizations are created using the Matplotlib, Pandas, and Tkinter Python libraries.

The MarketBeat Visualizer repeats data collected by the analyzer but in a more useful way. It shows which companies had hits for the imputed keywords. It also shows if the companies had positive or negative things to say. The graph is further above zero if more of the sections were positive, and the graph is below zero if more of the sections were negative. Figure 3 shows the visualization of the positivity scores of companies for AI from the marketbeatVisualization.py script.

*Figure 3: Visualization produced by marketbeatVisualization.py*
Implementation

There are 4 sections to the project.

1. Web scraping for links
2. Parsing the links to pull relevant text/content
3. Post-processing
4. Creating visualizations

Web scraping

For web scraping, multiple Python libraries were used to create scripts capable of gathering links from multiple online databases (MarketBeat, Reddit). The libraries used were BeautifulSoup and PRAW. BeautifulSoup is a web scraping library with the ability to iterate through static HTML pages and extract relevant information. BeautifulSoup was used to scrape MarketBeat. Using HTML tags, links that direct to the individual earning call transcripts from each company are passed to the parser. The Reddit web scraper used the PRAW library to successfully scrape Reddit of its posts and comments in certain subreddits of the user's choosing. All the corresponding links will be gathered in an external database (.xml file) and have corresponding information detailing when the links were gathered. This allows future users to see past links, while also allowing for more up to date links to be collected. Figure 4 shows the links for earning calls of each company.

Figure 4: Links produced by script
Parsing

Parsing occurs by taking in each link and extracting all relevant text from it. The parser populates a row in the resulting CSV file for each company. The parser populates the row with information pertaining to the company. The information stored includes stock ticker, company name, quarter and year of the call, and the sections of the text. The parser is called by the scraper, so no user input is required. The output file is a CSV file that is named after the date (ex. (March 20, 2024)marketbeatTranscript.csv). Table 1 shows the output table when running the MarketBeat parser script.

<table>
<thead>
<tr>
<th>Ticker</th>
<th>Company</th>
<th>Date of Transcript</th>
<th>Transcript</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAYX</td>
<td>Paychex Q3 2024</td>
<td>04/02/24 9:30 AM</td>
<td>Good day, everyone. And it is now my plea: Thank you, Mike. Th. This morning, before I'm going to start the well, We delivered solid</td>
</tr>
<tr>
<td>PVH</td>
<td>PVH Q4 2023 Earnings 04/02/24 9:00 AM</td>
<td>Good morning, every. Thank you, operator: These statements are PVH does not undert. Thank you, Sheryl. We continued to do</td>
<td></td>
</tr>
<tr>
<td>WBA</td>
<td>Walgreens Boots All 03/29/24 8:30 AM Good day and thank</td>
<td>Good morning. Th. As always, during the You can find our pre. I will now turn the ci</td>
<td></td>
</tr>
<tr>
<td>CTAS</td>
<td>Cortx Q3 2024 Earnings 03/30/24 10:00 AM Good day everyone</td>
<td>Thank you for joining. This conference call I will now turn the call. Thank you. We are listening now</td>
<td></td>
</tr>
<tr>
<td>CCL</td>
<td>Carnival Co. &amp; Q1 21 03/30/24 10:00 AM Good morning. This</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NKC</td>
<td>McCormick &amp; Co. 03/30/24 8:00 AM Good morning. This</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOX</td>
<td>FedEx Q3 2024 Earnings 03/31/24 5:30 PM Good day, and well</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NKE</td>
<td>Nike Q3 2024 Earnings 03/31/24 5:00 PM Good afternoon. We leading today's call</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRI</td>
<td>Darden Restaurants 03/31/24 8:30 AM Hello, we welcome</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACN</td>
<td>Accenture Q2 24 03/31/24 8:00 AM Good morning. Th. I would now like to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU</td>
<td>Micron Technology 03/31/24 4:30 PM Thank you for standing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GIS</td>
<td>General Mills Q3 20 03/31/24 9:00 AM Good morning. My n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADBE</td>
<td>Adobe Q1 24 03/14/24 5:00 PM Good day, and well</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ULTA</td>
<td>Ulta Beauty Q4 20 03/14/24 4:30 PM Good afternoon. Th is now my pleas</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEE</td>
<td>Lennar Q1 24 03/14/24 11:00 AM Welcome to Lennar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DGG</td>
<td>Dollar General Q1 20 03/14/24 10:00 AM Good morning. My n. No, I'd like turn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Output of MarketBeat Parser

The Reddit parser takes the user inputs and creates a CSV file containing the post and comments that were scraped. Additionally, it also outputs the keywords that were found in each post and the content relating to the posts. Table 2 shows the output table for the Reddit parser with all the posts that contained a keyword in the subreddit that was scraped.
Post-processing

Post-processing is done by NLTK, a Python library. NLTK supports Natural Language Processing (NLP) and Machine Learning. This allows for keyword matching and sentiment analysis. Each section of a transcript is run through the model and checked to see if it contains a keyword. If so, then the section is returned, and sentiment analysis is run on it. This is recorded in another file that can be saved for later. The script takes in a CSV file as input and returns a CSV file as output. The user inputs the name of the CSV file produced by the scraper and parser. Table 3 shows the output table of the marketBeatNLTK.py script with the positivity scores being displayed for every company.
Table 3: Output of marketbeatNLTK.py

The Reddit web scraper’s post-processing step is incorporated into the web scraping step. Essentially, the script takes the properties of the main Reddit web scraper and uses them along with our other libraries like the NLTK library to output data such as the positivity score for both the comments and the posts.

Visualizations

The MarketBeat visualization script utilizes Matplotlib[12], Pandas[13], and Mplcursors[15]. Matplotlib is used to create a bar graph. Each bar of the graph is a company that has at least one section of their earning call transcript contain a keyword that was inputted by the user. The measure of each bar is the sum of the number of positive sections present minus the number of negative sections present. The bar for each company shows green if above zero and red if below zero. The title of the graph is “Positivity of companies towards AI.” The scale of the graph automatically is sized to fit all the data provided by the input file. The
input file is a JSON output from the main analyzer script. The visualization script can be run by itself by inputting the name of the JSON file into the terminal after running the visualization script.

The Reddit visualization script, user_friendly.py, also utilizes Matplotlib and Pandas to create four graphs based on the processed data collected[16]. It uses the same prompt messages from the original Reddit web scraper script to create the visualizations. It creates a pie graph that contains the number of comments that were made for different months. It also creates three line graphs that compare the sentiment score of post titles and their comments, shows the change in the number of posts made containing a keyword, and shows the change in the average positivity score. Figure 5 shows the input box for users to type their variables.

![Reddit Data Fetcher](image)

**Figure 5: User inputs for ‘user_friendly.py’**
Figure 6 shows the visuals for the user_friendly.py after user inputs have been made in figure 5.

**Figure 6: Reddit Visualization from user_friendly.py**
Testing/Evaluation/Assessment

The MarketBeat scraper was tested by running all portions at once and seeing the recorded data match the expected output. The keywords used to test the project were “generative” and “ai”. All parts of the script were ran using MarketBeat.py. The script creates five files.

1. (April 26, 2024) marketbeatLogfile.txt
2. (April 26, 2024) marketbeatNLTK.csv
3. (April 26, 2024) marketbeatNLTK.json
4. (April 26, 2024) marketbeatTranscripts.csv
5. (April 26, 2024) marketbeatTranscripts.json

The logfiles contain the correct data about which keywords were used. The CSV and JSON files were all formatted according to the implementation and were up to date when compared to the MarketBeat website. The date of the test and the recording of the date were also correct. The visualization created by the resulting information was also checked and corrected. Figure 7 shows the logfile of the keywords used for the MarketBeat Scraper.

![Figure 7: Logfile](image)

Figure 8 shows the CSV output for the MarketBeat parser with it being the transcripts from earning calls.
Figure 8: CSV file output of MarketBeat Parser

Figure 9 shows the JSON output for the MarketBeat parser with the file showing the Company name, year of the transcript, and quarter of the year it was done.

Figure 9: JSON file output of MarketBeat Parser

Figure 10 shows the CSV output for the MarketBeat analyzer which analyzes the transcript from the earning calls of the companies and returns the positivity score for each.

Figure 10: CSV file output of MarketBeat Analyzer
Figure 10: CSV file output of MarketBeat Analyzer

Figure 11 shows the JSON output for the MarketBeat analyzer with the same results as Figure 10.

Figure 11: JSON file output of MarketBeat Analyzer

Figure 12 shows the visualizations for the MarketBeat visualizer and represents the data from the MarketBeat web scraper.
The Reddit web scraper was tested manually by using a different subreddit and keywords to make sure the right posts and comments were scraped. Based on the JSON output file, we manually navigated Reddit and confirmed the existence of the post along with the comments and content. Figure 13 showed the test inputs that were used for testing the Reddit web scraper.

```
[jamesdc@birch CS4624]$ python redditScraper_json.py
Enter the subreddit name: news
Enter keywords separated by comma: Phones
Enter the start date in YYYY-MM-DD format (or leave blank for no start date): 2024-01-01
Enter the end date in YYYY-MM-DD format (or leave blank for no end date): 2024-03-31
Enter the file path to save the search results (e.g., output.json): output7.json
```

**Figure 13: Test inputs for the Reddit web scraper**

Figure 14 showed the JSON output for the test inputs.
Figure 14: Test outputs for the Reddit web scraper

Figure 15 shows the actual post on Reddit which is to confirm the test outputs for the Reddit web scraper works.
Figure 15: Manual check with the test outputs
Users’ Manual

Reddit Web Scraper

Subreddits: r/artificial[5], r/generative, r/ArtificialIntelligence
Keywords: Travel, Tourism, Hospitality, Hotel, Vacation, Trip

There are two Python scripts the user can run.

1. If the user wants to see the data visualized, they can run ‘user_friendly.py’. This file combines web scraping and data visualization code into a single script, eliminating the need for multiple steps to visualize the data.

2. If the user only wants to see the JSON output with post title, comments, positivity score, and the date they were posted, they can run “redditScraper_json.py”. This script focuses on seeing the actual scraped data in detail from subreddits.

Running user_friendly.py

After downloading ‘user_friendly.py’ from GitLab and opening it in an IDE (Visual Studio Code[17]), the user can run the file by clicking the ‘Run Python File’ button located at the top right (as seen in Figure 16) or by typing the command ‘python user_friendly.py’ in the terminal. Subsequently, the GUI for user input will appear, as shown in Figure 17.

![Figure 16: "Run Python Button" at the top right with a red circle](image-url)
After writing all the inputs for each row, the user can click the “Fetch Data” button. Then, the visualization of the data will pop up.

Running redditScraper_json.py

Users that want to access and utilize the Reddit scraper would first need to be in the right repository that has the file redditScraper_json.py. Figure 18 shows the file directory for all the python scripts.
Then run the command “python redditScraper_json.py” to the terminal. The user should then be given a prompt for which subreddit will be scraped as shown in Figure 19.

![Figure 19: Running Reddit scraper script](image1.png)

Then, the user will be given a prompt for keywords to specifically filter the posts and comments that will be scraped as shown in figure 20.

![Figure 20: Keyword input for Reddit scraper](image2.png)

Afterwards, the user will be given a prompt to customize the name of the JSON file as shown in Figure 21.

![Figure 21: Reddit JSON output naming](image3.png)

Once the user types in their keywords, the code will then output a JSON file with all the posts and comments along with the keywords to filter them as shown in Figure 22.
MarketBeat

The user should download any and all files with MarketBeat from the repository to ensure all the proper files are accounted for. These files include:

1. MarketBeat.py
2. marketbeatLogfile.py
3. marketbeatNLTK.py
4. marketbeatParser.py
5. marketbeatScraperParser.py
6. marketbeatToJson.py
7. marketbeatVisualizer.py

To run all the scripts for MarketBeat, the user must install Python, BeautifulSoup, NLTK, Pandas, Matplotlib, and Mplcursors. To create all the files, the user should make sure to install all dependencies, by running the command, pip install <Library Name> (ex. pip install BeautifulSoup).
beautifulsoup). The user should also make sure to install all NLTK prerequisites by running `nltk.download('all')`. The user then can run `MarketBeat.py` to generate all the files. Five files will be made. The user will be prompted to provide keywords to feed into the analyzer. The user should input the keywords they want to search for in a comma separated list as shown in Figure 23.

```
Enter keywords (comma separated): generative, ai
```

**Figure 23: Keyword input example**

The user can see the figure separately from generating files as well by running `marketbeatVisualizer.py` and feeding the name of the post processed JSON file as shown in Figure 24.

```
Enter the JSON filename: (May 01, 2024)marketbeatNLTK.json
```

**Figure 24: Visualizer input example**

Running `MarketBeat.py` will generate five files that allow the user to keep a log of data. The files follow the naming convention,

1. `(Month Day, Year)marketbeatLogfile.txt`
2. `(Month Day, Year)marketbeatNLTK.csv`
3. `(Month Day, Year)marketbeatNLTK.json`
4. `(Month Day, Year)marketbeatTranscripts.csv`
5. `(Month Day, Year)marketbeatTranscripts.json`

The log file contains information about what keywords were supplied to run the programs.
Developer’s Manual

Project file and requirements

The Git repository is found at: https://code.vt.edu/florian/gen_ai_voices.

Scripts for the project run in Python.

The required libraries for the project are:

1. BeautifulSoup for Python
2. NLTK for Python
3. PRAW for Python
4. Matplotlib for Python
5. Pandas for Python
6. Mplcursors for Python

Prerequisites and Installation

To start, ensure that Python is installed. This is a requirement for the project as the scripts are made in Python. The libraries are required to be installed to run the scripts in the project.

To install BeautifulSoup, run

“pip install beautifulsoup4”

To install PRAW, run

“pip install praw”

To install NLTK, run

“pip install --user -U nltk”

To install Matplotlib, run

“pip install matplotlib”

To install Pandas, run

“pip install pandas”
To install Mplcursors, run

"pip install mplcursors"

Clone the Git repository: run

git clone https://code.vt.edu/florian/gen_ai voices

Running the scripts

To generate the MarketBeat files and visualization, run

“python ./MarketBeat.py” as shown in Figure 25.

Then input the keywords as a comma separated list to generate the files.

To just see the figure without generating MarketBeat files, run

“python ./marketbeatVisualizer.py”

Then supply the analyzed JSON file (ex. (April 24, 2024)marketbeatNLTK.json).

To generate the Reddit links, run

“python ./redditScraper.py”

Figure 25: Generated Marketbeat files
Lessons Learned

Timeline

Project management and organization were utilized through Microsoft Teams, GitLab, Discord, and Zoom. We used Discord for daily communications and online meetings among teammates, Microsoft Teams for document sharing, GitLab for version control and code sharing, and Zoom for online meetings with the client. We conducted bi-weekly in-person meetings and occasional online meetings with our team and bi-weekly online meetings with the client to accurately reflect the client’s requirements and ensure consistent coordination.

<table>
<thead>
<tr>
<th>January 30</th>
<th>• Chose this project and met with the client to understand their needs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 13</td>
<td>• Implemented a simple web scraper for subreddits[6].</td>
</tr>
<tr>
<td></td>
<td>• Implemented a rough version of the MarketBeat web scraper</td>
</tr>
<tr>
<td>February 22</td>
<td>• Developed a functional web scraper for both Reddit and Seeking Alpha using PRAW and BeautifulSoup.</td>
</tr>
<tr>
<td></td>
<td>• Modified the output of the MarketBeat scraper</td>
</tr>
<tr>
<td>March 12</td>
<td>• Modified the web scraper to retrieve only the target data with JSON output.</td>
</tr>
<tr>
<td></td>
<td>• Adjusted the code to capture top comments, dates, titles, keywords, URLs, upvotes, and comments containing specific keywords.</td>
</tr>
<tr>
<td></td>
<td>• Started the marketbeatNLTK.py python script</td>
</tr>
<tr>
<td>March 26</td>
<td>• Completely scraped data from databases (Reddit and Seeking Alpha Earnings Calls)</td>
</tr>
</tbody>
</table>
- Started data processing using NLTK

<table>
<thead>
<tr>
<th>April 9</th>
</tr>
</thead>
</table>
| - Met with the client to discuss the required types of visualizations.  
  - Began implementing data visualizations using Matplotlib and Pandas.  
  - Uploaded marketbeatNLTK.py to git |

<table>
<thead>
<tr>
<th>April 19</th>
</tr>
</thead>
</table>
| - Complete data visualization using the processed data from subreddit and company earning call transcripts.  
  - Redid output of both marketbeatNLTK.py and marketbeatScraperParser.py with client feedback |

<table>
<thead>
<tr>
<th>April 25</th>
</tr>
</thead>
</table>
| - Final fix ups and cleaning up the project  
  - Checkups with client over the project |

<table>
<thead>
<tr>
<th>May 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Final Presentation, Final Report, and final meeting with the client delivering our product.</td>
</tr>
</tbody>
</table>

Problems

Earlier in the semester, there were some misunderstandings and communication issues, leading to one team member leaving. Consequently, we had to redistribute our work after writing the contract with our client and come up with a more efficient way of handling the project. There were some technical errors with the Reddit web scraper, with the Reddit API, and the limitations it had. The Reddit web scraper was not able to scrape from a large time frame given by the user input due to the request rate limit.
Solutions

After knowing that one member was leaving, two members of our current team directly spoke with the departing member to understand what led to his departure and to resolve any misunderstandings. With the team now reduced to three members, we redistributed the work, taking on the tasks of the member who left. Additionally, we made efforts to enhance our communication, specifically regarding our schedules, and engaged in more discussions to ensure everyone was on the same page and prevent any future misunderstandings. Regarding the Reddit API limitations, we had to shorten the time range of when the Reddit web scraper was scraping from to fit the limit. Additionally, we received assistance from GTA, Satvik Chekuri, during his office hours and obtained the help we needed.

Future work

For future developments, it would be beneficial to create a fully functioning website that displays data visualizations from both Reddit and MarketBeat to compare public and industry views on certain keywords. This would make the data accessible to users without any coding background. Furthermore, enhancing the logic for scraping posts based on specific keywords is anticipated. Additional improvements could include developing a MarketBeat scraper that targets individual sectors rather than all companies and implementing a history scraper capable of retrieving earnings call transcripts from a specific company’s historical records.
Acknowledgements

Our gratitude goes to our client, Dr. Florian Zach, for giving us the opportunity to work on this project as well as being accommodating despite multiple hardships. He was a great resource to us and gave us the necessary advice and guidance to develop and finalize our product. We would also like to extend our gratitude to Professor Fox for giving us the opportunity to work on this project and GTA, Satvik Chekuri, for his help in developing our web scrapers.
References


[accessed May 4, 2024]

https://pypi.org/project/beautifulsoup4/ [accessed May 4, 2024]

https://pandas.pydata.org/ [access May 4, 2024]

https://matplotlib.org/ [accessed May 4. 2024]

https://www.nltk.org/ [accessed May 4, 2024]

www.mplcursors.readthedocs.io

https://medium.com/@sujathamudadla1213/explain-pattern-python-library-5bb4c530956b
[accessed May 4, 2024]

[accessed May 3, 2024]