Stock Twin Visualizer

Team: David Francisco and James Scharf
Section: 601.466
Emails: dfranc25@jhu.edu and jscharf8@jhu.edu

Summary: Given an inputted stock and a sample size number, our web interface creates a model that calculates which of the randomly selected stocks are the most similar to the original by looking at news, Wikipedia summaries, references from Wikipedia, and the links from Wikipedia.

To run the web interface locally, clone our repository and run:

```
$ python Server.py
```

Then this will then initiate a locally hosted server at port 8080. The URL is also available in the terminal window. Copy and paste this into a web browser. The user should now see a webpage resembling **Figure 1.1**. Now, follow the instructions on the website to execute a query with the appropriate format. The user should be aware that increasing the size of the random sample will slow the program's execution. To improve speed in a production environment, we would delegate web crawling, information processing and internet work to several machines.

Once the user executes her query, a tabbed web interface with several dashboards will appear, as **Figure 2.1** shows. Each dashboard focuses on a different aspect of information about a stock. In the first, the user sees an overview of the relevances of each stock's corpora. In the second, she sees an analysis of the polarity of each stock over time. In the third, she can analyze the trustworthiness of news sources. The most similar stocks are portrayed first. We found it particularly interesting to utilize this tool in the aftermath of Donald Trump's tweets about trade because there were clear trends in sentimentality that paired with the market's abysmal performance.

Our methods and logic were as follows:

1. For newspaper extraction, our point of origin is Yahoo News. `Scraper.py` reads a stock's unique page to collect links to external news sources. Here, the program is scraping a web scraper, making it a partial metascraper. Next, it scrapes each of the sources to produce a corpus for this stock. This required us to analyze article pages of several typical sources to build a comprehensive function for extracting article texts from widely varying sources without compromising speed.

2. For general knowledge extraction, our point of origin is Wikipedia. `Scraper.py` utilizes a library called `wikipedia` that parses summaries, references and links. As the first dashboard demonstrates, we compare the relevance of each Wikipedia summary. We also scrape several references to build another corpus. However, our most useful method for understanding stock similarity is the comparison of links. Stocks that link to similar pages are, by their nature, similar.

So, we calculate the degree of overlap.

3. For corpus weighting, `compare.py` uses tf-idf. Its inverse document frequency component minimizes the effects of financial jargon that appear in many articles, thus characterizing each document by its most unique elements.

4. The normalized Wikipedia link similarity is calculated to show how close they are in Wikipedia's theoretical knowledge graph. This allows us to understand the relative position of a stock's page in comparison to another, but without transforming Wikipedia into a network/graph and conducting searches. Links are the hyperlink text in a Wikipedia article.

5. We process and compare the references of a Wikipedia page because it shows a) who these stocks associate with and b) what sort of websites are talking about this stock. If a magazine associated with one industry writes about both stock A and stock B, then they are likely very similar.

6. For polarity and subjectivity scores, we utilize Textblob. Although we could have utilized training data ourselves, Textblob is a commonly used library and is therefore thoroughly tested. We also included Textblob in case we extended our application to utilize cross-language IR, as it has an automatic translator.

7. Our visualization tool of choice was Bokeh. It allows the user to save graphs and drag images to zoom in on specific points in time. Unlike matploblib, Bokeh visualizations are interactive and draw in the user. Static images would discourage the user to carefully inspect the results and learn from the tool.

8. Our user interface relies upon Google's Material Design Lite package, which is speedy and user-friendly.

9. We use CherryPy for our server and implement a basic type of templating to embed our graphs and the webpages.

Extensions:

1. We would use a PCA to fully represent all of our metrics cleanly and in one graph.

2. We could predict future changes in stock using Doppelganger Discovery (as used in Baseball) via news polarity and subjectivity. We would scrape news articles from discrete intervals of time throughout a stock's past and create polarity graphs which can be used to predict how well a stock will perform via Doppelganger Discovery of the most similar stocks calculated form our metric.

3. In the event that we could not find a Wikipedia page, we would utilize cross-language IR to find it in another language version.

Limitations:

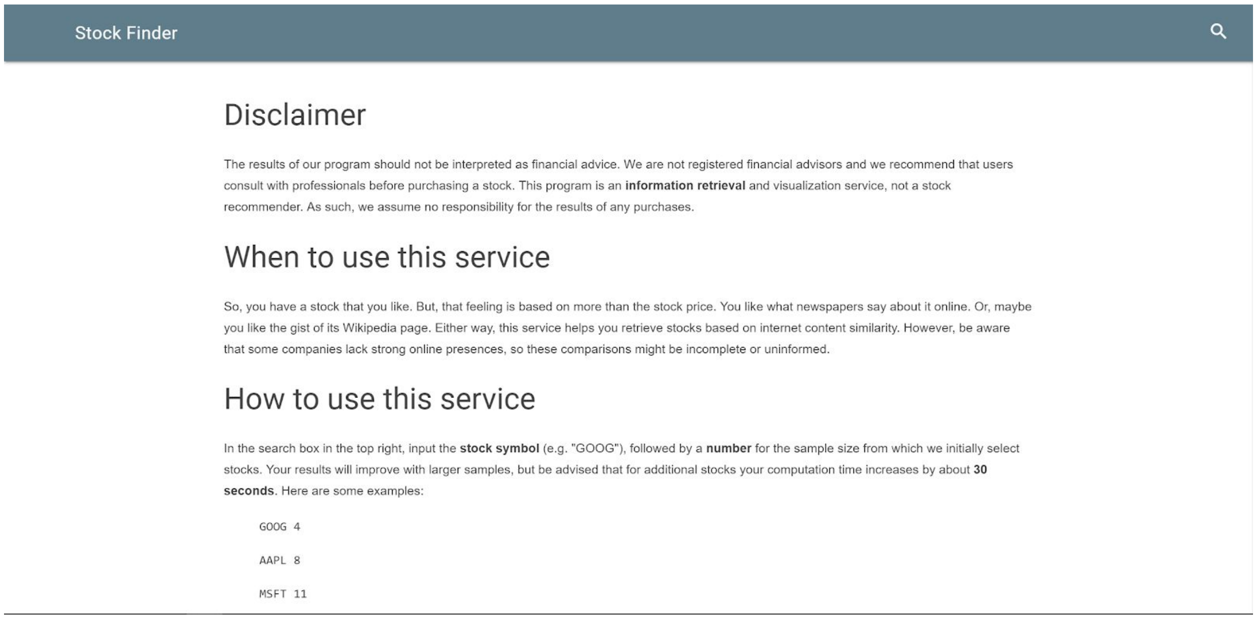
1. The program currently runs locally because of security restrictions. However, this could be easily modified.

2. Poor performance due to a large number of computations.

Empirical Evaluation:

Our focus was visualization, so an empirical analysis would not make sense. So, we would prefer that a grading metric focused on the usefulness of our dashboard for visualization. However, we were able to see that our similarity metrics showed quite a high similarity, around 70-80% when stocks are within the same category (noted in the company list doc), showing that the empirical work in the background is accurate.

1. **Figure 1.1**



Stock Finder

Disclaimer

The results of our program should not be interpreted as financial advice. We are not registered financial advisors and we recommend that users consult with professionals before purchasing a stock. This program is an **information retrieval** and visualization service, not a stock recommender. As such, we assume no responsibility for the results of any purchases.

When to use this service

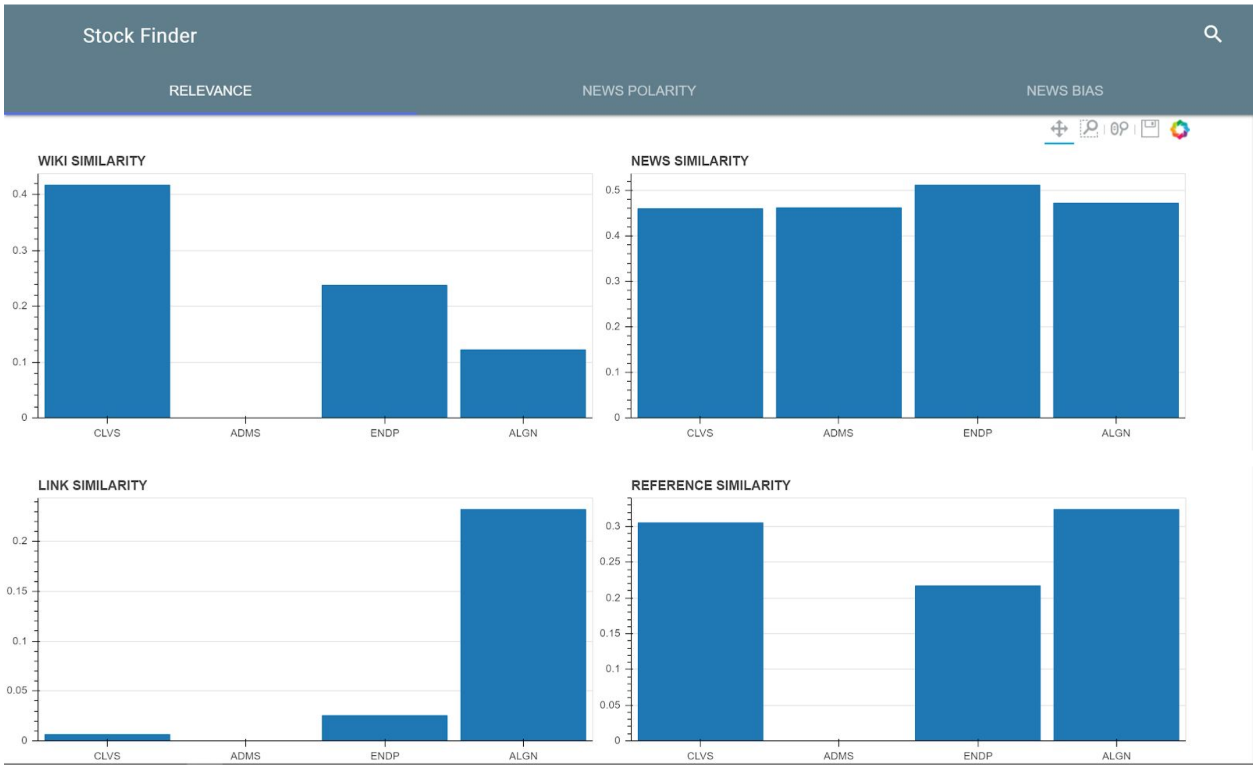
So, you have a stock that you like. But, that feeling is based on more than the stock price. You like what newspapers say about it online. Or, maybe you like the gist of its Wikipedia page. Either way, this service helps you retrieve stocks based on internet content similarity. However, be aware that some companies lack strong online presences, so these comparisons might be incomplete or uninformed.

How to use this service

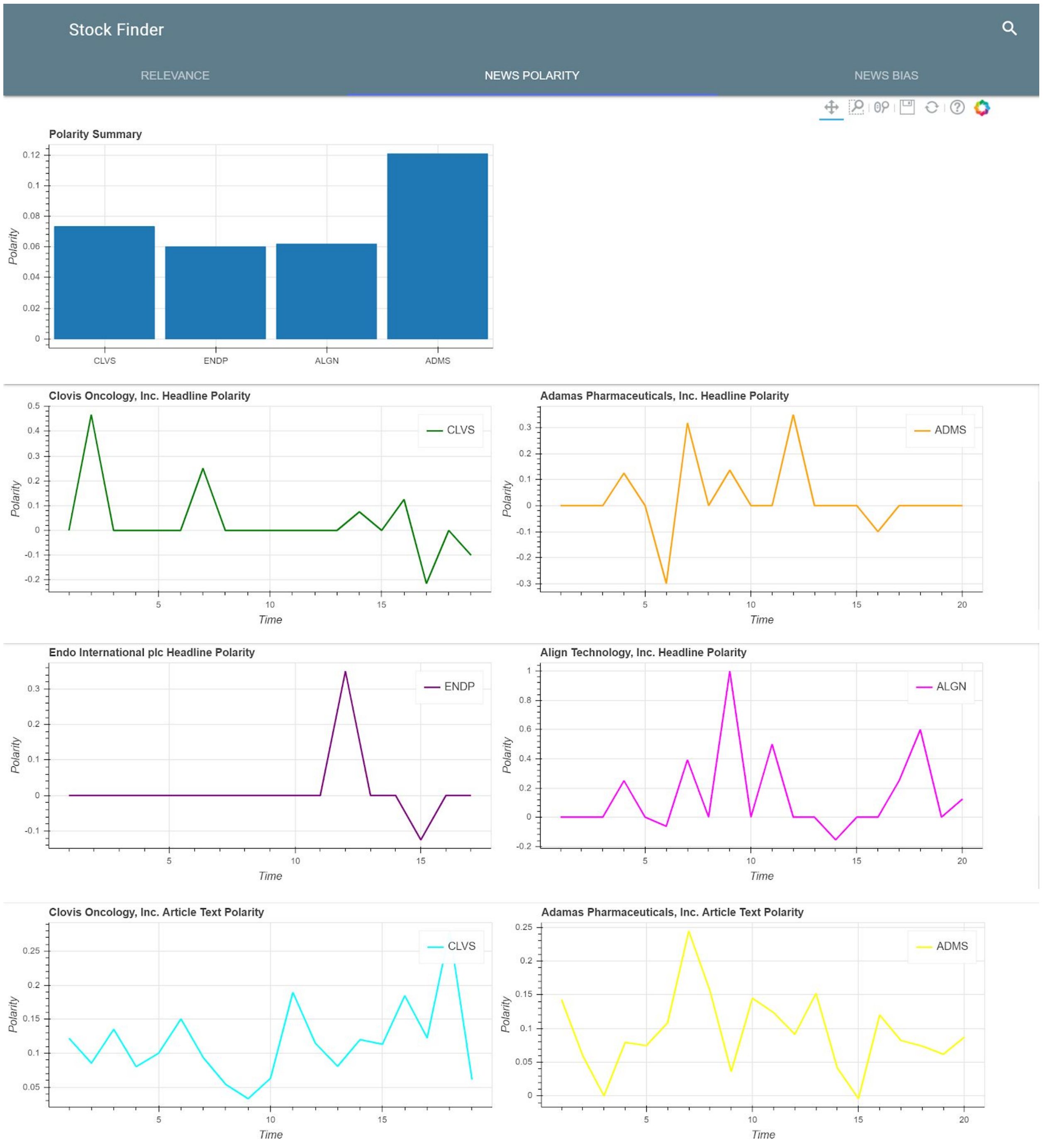
In the search box in the top right, input the **stock symbol** (e.g. "GOOG"), followed by a **number** for the sample size from which we initially select stocks. Your results will improve with larger samples, but be advised that for additional stocks your computation time increases by about **30 seconds**. Here are some examples:

```
GOOG 4

AAPL 8

MSFT 11
```

2. **Figure 2.1**



3. Figure 2.2


Stock Finder
RELEVANCE
NEWS POLARITY
NEWS BIAS
Polarity Summary
Polarity
CLVS
ENDP
ALGN
ADMS
Clovis Oncology, Inc. Headline Polarity
Adamas Pharmaceuticals, Inc. Headline Polarity
Endo International plc Headline Polarity
Align Technology, Inc. Headline Polarity
Clovis Oncology, Inc. Article Text Polarity
Adamas Pharmaceuticals, Inc. Article Text Polarity
Time