Doug Dellolio

Component Design

Component 1 – Reading the data and displaying on chart

Yahoo Finance API

The most important component for my project is reading the data because nothing can be analyzed if there is no data. I will use the Yahoo Finance API to read market data for the general information section and also a list of the headlines. All of the volume data will also be read in using the Yahoo API. The first component will include a chart of the past volume for a year. I get todays date and then calculate a year prior. The API will provide a CSV with all of the data requested by using tags. The "v" at the end refers to volume.

An example request that returns csv file:
http://ichart.finance.yahoo.com/table.csv?s=AAPL&f=v

This csv file is then parsed and plotted on a chart using the JFreeChart library. Volume is then stored into an *ArrayList*.

Furthermore, all of the headlines are read from Yahoo Finance and the dates are recorded using string manipulation on the returned values. A *HashMap* is then used where the key is the headline and the volume is the value. The dates are then matched to the dates in the chart and an arrow is produced using *XYPointerAnnotation* provided by JFreeChart. More on how it is plotted below.

Boilerpipe

This library will help me actually read the press releases from the website. Given a website URL, it will be able to distinguish what is part of a news article and what is not. An example call is shown below:

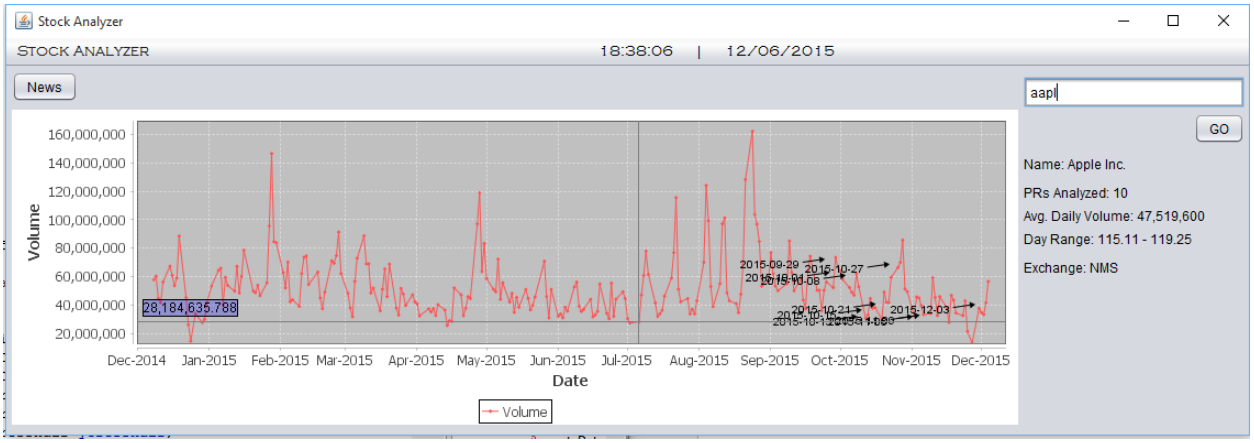
URL url = new URL("url here");

String text = ArticleExtractor.INSTANCE.getText(url);

Since this process can take a while if there are multiple press releases, I have set up a cache folder that will take *text* above and write it to a file. Every stock symbol entered will cache all of the press releases. When a symbol is typed in, it will match the number of files in the directory to the number of headlines read (from component 1). If it is different, it knows that it is missing files so it will reread the data from the source. However, if it is equal, then no data is changed and the press release content will be read from the directory only. This allows for a very quick response time.

JFreeChart

In order to plot this data I will use the JFreeChart Library. For this particular chart I will use *XYPlot* and have the volume as the Y axis and the date on the X axis. They will be read from the *HashMap* created in part 1 of this section. This contains the date as the key and the volume as the value. Some settings I can add include the crosshair and displaying the volume in blue. I run through the *HashMap* and feed it into the *XYPlot* which will then display it as a nice line graph.



<u>Component 2 – Topic Identification</u>

Lingpipe

The main library that will help me with identifying the topics of all of the press releases is Lingpipe. Lingpipe will allow me to create a trained model of all of the data for every sector. The sectors include: consumer goods, technology, utilities, services, health care, financials and more. In each folder, a number of press releases from stocks in those sectors will be added so that the model can be trained on that data. Each press release is then compared to the model and is evaluated. Whichever it is closely related to, the chart will highlight it. An example of this is provided below. In this case, the press release titled “Enhanced Editions of Harry Potter Series Now Available Exclusively on iBook’s for iPhone, iPad & iPod touch” is considered to be under the topic of consumer goods. This is a good thing because Apple Inc. is in the consumer goods sector.

JFreeChart

Similar to component 1, I will be used the JFreeChart library to create a 3D pie chart of the different sectors and how it scores in each. I will use a *PiePlot3D* and a *DefaultPiedataset* to add the data returned by Lingpipe. The chart below is the data obtained by Yahoo Finance and then evaluated by Lingpipe. The sector that the press release most aligns with will be highlighted in yellow. This will be used later on when predicting stock price movement based on terms.

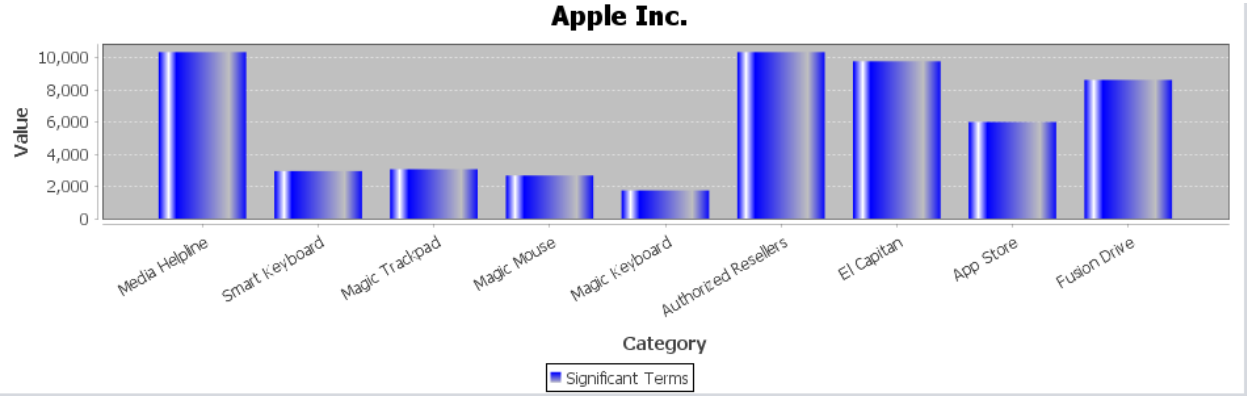
Component 3 – Significant Phrases

Lingpipe

The main library that will help me with the significant phrases is Lingpipe as well. Similar to component 2, this library will allow me to create a trained model. It then compares each press release to that model and finds the most significant phrases. It takes things such as frequency into account to calculate this. Below is an example of the chart taken from Apple. The most significant terms include things like “El Capitan” and “Fusion Drive”. Both are products that Apple have recently introduced. El Capitan being the latest edition to OSX and Fusion Drive being Apple’s name for its implementation of a state of the art hybrid drive.

JFreeChart

Similar to the last 2 components, I will be using JFreeChart to display this data neatly. I decided on a bar chart because I think it can easily show the different significant terms and its “value” ranking determined by Lingpipe. I will be using a regular *DefaultCategoryDataset* provided by JFreeChart to create the dataset necessary for this chart. All of the significant terms and its values are then returned by Lingpipe and I read it into the graph.



Component 4 – Sentiment Analysis

Lingpipe

The last component will be a sentiment analysis on the data. Like the other components, I will use Lingpipe to help me achieve this. As a result, I want to be able to take the press releases by companies and categorize them into positive, negative, and neutral categories. This way, I will be able to get a sense as to what future press releases may do. This works by first training a model. In order to do this, I had to hand select positive and negative press releases. The library than creates a model and will compare each press release to these models to determine the best fit. There is no GUI component yet for this, but you will notice that at the end of each release is either "pos" or "neg". This is an example from Apple (AAPL). The training model has to be tweaked a bit to get a more accurate reading. This is currently being worked on.

```
Data Directory=C:\git\StockMarketProgram\trainers\sentiment.analysis

Training.
  # Training Cases=16
  # Training Chars=49372

Evaluating.
Apple Reports Record Fourth Quarter Results.txtneg
Apple Announces New ResearchKit Studies for Autism, Epilepsy & Melanoma.txtpos
Apple Brings Apple Music, iTunes Movies & iBooks to Customers in China Starting Today.txtpos
Apple Launches New Clean Energy Programs in China To Promote Low-Carbon Manufacturing and Green Growth.txtpos
Apple Updates iMac Family with Stunning New Retina Displays.txtpos
Enhanced Editions of Harry Potter Series Now Available Exclusively on iBooks for iPhone, iPad & iPod touch.txtpos
Epic 12.9-inch iPad Pro Available to Order Online Wednesday & Arrives in Stores Later This Week.txtpos
James Bell Joins Apple's Board of Directors.txtpos
OS X El Capitan Available as a Free Update Tomorrow.txtpos
Record Debut Weekend Sales of New Products - Report on Apple.txtpos
  # Test Cases=20
  # Correct=20
  % Correct=1.0
```

JFreeChart

Lastly, I will be using the JFreeChart library again to construct a graph that will easily illustrate these results. That is the next part that I will be working on since it has not been completed yet. This components results as well as the other components results will all come together to help me predict movement of a stock price based on certain given words and phrases.