Ethan Ma

CS 410: Text Information Systems

Prof. Chengxiang Zhai

6 November 2022

NLTK As Seen in Michael Reeves' WSB Sentiment Analysis Bot

Introduction

Programmatically printing money has long been the pipe dream of a wide swath of the first-world populace, from "finance bros" to the "Average Joe." To this end, a significant amount of software development in the financial sector has been geared towards profiting from the stock market. Algorithmic stock trading is a widely used, but seldom understood buzzword, incessantly thrown around between retail ("Average Joe") traders on stock-oriented internet forums (e.g., Reddit's infamous r/wallstreetbets and r/Superstonk). In these forums, large companies and hedge funds are often vilified for utilizing these algorithms to manipulate market prices. Though there are indeed questionable legal implications for doing so, the debate on the truth of these accusations is somewhat dubious, ongoing, and beyond the scope of this paper.

It would, however, be prudent to give a brief overview of what these algorithms aim to do. The Wall Street Journal summarizes it best as: "When Wall Street firms use algorithms, they are simply encoding a logic into the computer. A trading algorithm can be fundamentally driven--meaning it is based on old-fashioned company metrics--or based on quantitative signals such as a sweep of buying interest known as momentum or technical factors like a particular stock breaking through a 30-day average price. Or, it can be all three." (Hope, 2017) As we can see from this brief overview, these types of algorithms largely focus on quantitative, mathematical information and patterns, along with implementing strategies that these large companies find useful or profitable.

However, when a collection of people becomes large enough, or if there is enough moving capital (or both), many generalized versions of these algorithms are rendered useless. Queue r/wallstreetbets with its 13 million members. With a community this large, especially with its high-capital members termed "whales," the general sentiment on the forum begins to materially affect the prices of the stocks that they discuss. In much the same way that shady traders illegally "pump and dump" their stocks, tickers can be pumped or dumped based solely on the whims of the faceless Reddit trolls upvoting and downvoting posts on r/wallstreetbets. With the trademark caveat of "this is not financial advice, I'm just an "ape" that likes the stock," traders here skirt around legal gray areas, sometimes posting ridiculous 1000%+ gains, others posting complete losses of their portfolios ("guh").

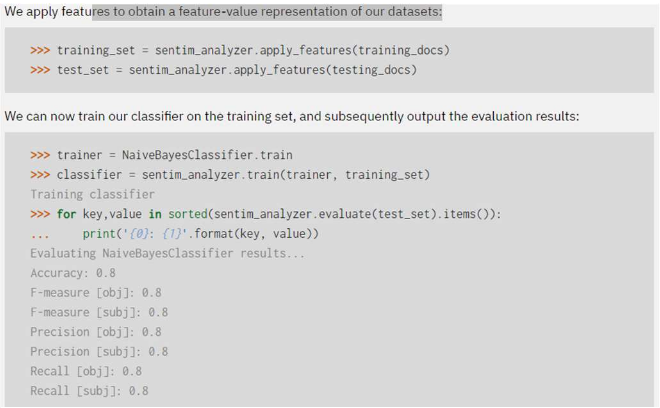
While this whole situation might appear chaotic to a layperson, a programmer with a discerning eye will realize that the text of these forums is a treasure trove for mining and sentiment analysis. One such programmer is famous YouTuber, Michael Reeves. As such, in this paper, we will follow Reeves on his journey through the sentiment analysis in his YouTube video "I Gave My Goldfish $50,000 to Trade Stocks," which just so happens to use our beloved r/wallstreetbets as a dataset.

Body

In his video, Reeves wanted to see whether his fish Frederick or the giant Subreddit r/wallstreetbets would make more money if they were given $50,000 to trade in the stock market. To accomplish this, he would have to create a sentiment analysis bot that scraped through the top posts of r/wallstreetbets each day that the competition was active, tagging both the relevant tickers, and whether sentiment towards that particular on the forum was positive or negative for that day. At first glance, to a somewhat experienced data/computer scientist that was unfamiliar with the subtleties of r/wallstreetbets, it might appear that this would be a classic sentiment analysis task, with plenty of sample data from existing Wall Street stock literature. However, for Reeves, this was not entirely the case.

You might have noticed the somewhat strange vocabulary that has been prevalent throughout the introduction and context of this paper. Terms such as “whale, YOLO, guh, ape, etc.” are quite uncommon in normal news and literature, but they are the lifeblood of the Reddit “apes” frequenting r/wallstreetbets. Indeed, each of these terms has its own meaning specific to the forum, despite having regular English counterparts. For example, the term “whale” in normal English is generally referring to an extremely large aquatic mammal. On Reddit, “whale” translates to “extremely wealthy person with money to gamble on meme stocks.” Obviously, this rather unique vocabulary poses problems to sentiment analysis models trained on more traditional texts.

This brings us to Reeves’ technology of choice to address this problem, NLTK. NLTK, according to their website “provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.” (nltk.org) From NLTK’s sentiment analysis example shown in figure 1, we can see that it requires both a testing and training set, with



We apply features to obtain a feature-value representation of our datasets:

```
>>> training_set = sentim_analyzer.apply_features(training_docs)
>>> test_set = sentim_analyzer.apply_features(testing_docs)
```

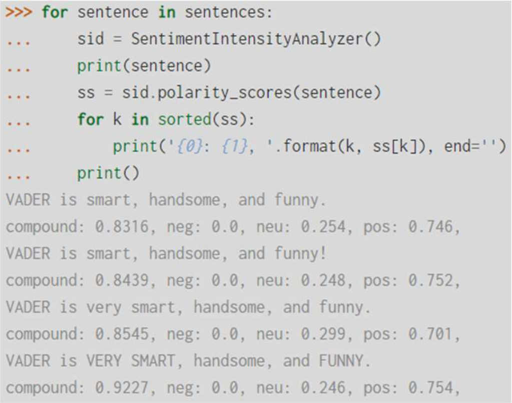
We can now train our classifier on the training set, and subsequently output the evaluation results:

```
>>> trainer = NaiveBayesClassifier.train
>>> classifier = sentim_analyzer.train(trainer, training_set)
Training classifier
>>> for key,value in sorted(sentim_analyzer.evaluate(test_set).items()):
...     print('{0}: {1}'.format(key, value))
Evaluating NaiveBayesClassifier results...
Accuracy: 0.8
F-measure [obj]: 0.8
F-measure [subj]: 0.8
Precision [obj]: 0.8
Precision [subj]: 0.8
Recall [obj]: 0.8
Recall [subj]: 0.8
```

Figure 1

labelled data, so that it can train its classifier. The classifier uses Naïve Bayes as its classifying algorithm of choice, which we discussed in class. Since Reddit’s vocabulary is so unique, he had to ask some of r/wallstreetbets’ top contributors to assist him in labelling the data for an entirely new training set based solely on r/wallstreetbets data, about 10,000 posts in total.

With this problem solved, Reeves could then have a scraper scrape the top posts on r/wallstreetbets each day, and use the output of the polarity_scores() function like in figure 2 in order to determine the general sentiment of the forum towards a particular ticker, and then used the Alpaca stock trading API to execute the trades. He then pitted these trades against the output from his fish Frederick, who essentially picked between two randomly selected stocks by “deciding” to spend more time on one side of his fish tank than the other. The fish won. While this outcome was somewhat unexpected considering the storied history of r/wallstreetbets, the procedures of the experiment were, of course, far from scientific, due to the comedic intent for the video. Of course, just because a video is comedic does not mean that the ideas used in and behind the video are of no value. In fact, NLTK, according to their Github, has recently been developing Twitter processing, to a similar effect as what Reeves was doing in his video. With even more advanced features like Part of Speech Tagging, Named Entity Recognition, irony/sarcasm detection, etc., which could be used to further augment similar programs in the future.



```
>>> for sentence in sentences:
...     sid = SentimentIntensityAnalyzer()
...     print(sentence)
...     ss = sid.polarity_scores(sentence)
...     for k in sorted(ss):
...         print('{0}: {1}, '.format(k, ss[k]), end='')
...     print()
VADER is smart, handsome, and funny.
compound: 0.8316, neg: 0.0, neu: 0.254, pos: 0.746,
VADER is smart, handsome, and funny!
compound: 0.8439, neg: 0.0, neu: 0.248, pos: 0.752,
VADER is very smart, handsome, and funny.
compound: 0.8545, neg: 0.0, neu: 0.299, pos: 0.701,
VADER is VERY SMART, handsome, and FUNNY.
compound: 0.9227, neg: 0.0, neu: 0.246, pos: 0.754,
```

Figure 2

Conclusion

All in all, Reeves' video provides an excellent comedic lens into some of the vast possibilities that can be explored when using sentiment analysis, specifically using NLTK. While the ending stock performance of Reeves' NLTK implementation of sentiment analysis on r/wallstreetbets was less than stellar, it proves a point that with some further research and development, a sentiment analysis tool like this could end up actually helping people in making their stock decisions on a daily basis. After all, sentiment and perception are extremely important factors of stock performance, perhaps even more so than traditional quantitative and technical factors used in current algorithmic trading.

References

Graphics, W. S. J. (2017, May 22). *Decoded: Breaking down how an actual trading algorithm works. want to impress your friends? learn how trading algos work*. The Wall Street Journal. Retrieved November 6, 2022, from https://www.wsj.com/graphics/journey-inside-a-real-life-trading-algorithm/

NLTK. (n.d.). Retrieved November 6, 2022, from https://www.nltk.org/

Nltk. (n.d.). *Twitter processing · NLTK/NLTK wiki*. GitHub. Retrieved November 6, 2022, from https://github.com/nltk/nltk/wiki/Twitter-Processing

R/wallstreetbets. reddit. (n.d.). Retrieved November 6, 2022, from https://www.reddit.com/r/wallstreetbets/

Reeves, M. (2022, March 31). *I gave my goldfish $50,000 to trade stocks*. YouTube. Retrieved November 6, 2022, from https://www.youtube.com/watch?v=USKD3vPD6ZA

Sentiment Analysis Example. NLTK. (n.d.). Retrieved November 6, 2022, from https://www.nltk.org/howto/sentiment.html