

Using Sentiment from Twitter optimized by Genetic Algorithms to Predict the Stock Market

Carlos Simões
Instituto de Telecomunicações
Instituto Superior Técnico
Lisboa, Portugal
carlosvsimoes@tecnico.ulisboa.pt

Rui Neves
Instituto de Telecomunicações
Instituto Superior Técnico
Lisboa, Portugal
rui.neves@ist.utl.pt

Nuno Horta
Instituto de Telecomunicações
Instituto Superior Técnico
Lisboa, Portugal
nuno.horta@lx.it.pt

Abstract— In this work we propose to use Twitter to find companies with a good growth potential that could be good investment options. In order to achieve this we built a sentiment model using the text content of tweets. We make use of hashtags to collect Twitter posts from a broad range of emotions so that our sentiment model can reliably distinguish tweets containing different sentiment expressions. To guarantee that no human sentiment is left behind we adopted emotions from the Circumplex Model of Affect and their synonyms and used them as search terms on the Twitter API. Afterwards, we use those tweets with a support vector machine (SVM) classifier to build a sentiment model. This model was used to classify company related tweets in order to predict the predominant sentiment in them. With the sentiment measures of tweets from different companies we created a trading rule that was optimized by a Genetic Algorithm (GA) so that we can maximize profit. Our simulations show that using the rules we implemented it is possible to build a profitable strategy for trading in the stock market using Twitter with the rules we implemented. During our testing period (November 7, 2016 to December 16, 2016) we achieved a 11% return, outperforming the S&P 500, NASDAQ 100 and DJIA composites.

Key words: sentiment analysis; twitter; opinion mining; stock market.

I. INTRODUCTION

The stock market is the congregation of buyers and sellers of stocks. Shares or stocks represent ownership in a company so, if we own 10% of the shares of a company it means we own 10 % of the company. Of course, this brings opportunities to the public. If one buys shares of a company and, in turn, that company is able to grow their sales and turnover, the shares the investor initially bought are going to be worth more because the underlying company is also worth more. What all this participants have in common is that they are constantly looking for an edge and that is where Twitter can come in. Nowadays almost every news organization and every journalist has live twitter feeds, not only that but most companies have an extensive social media presence including their top managers and CEOs. Some large hedge funds are spending millions on building computer software able to generate trade signals based upon data from social media [1]. Indeed, if useful economic data is available on twitter, then using sentiment analysis or opinion mining, it can be possible to merge that functionality with the already existing automated trading algorithms. Most of these algorithms use real-time data that comes from the stock market exchanges, but if they were also able to access and interpret all the information available, including news feeds and public

conversations on social media, then they would be able to react instantly and make sound rational trading decisions.

In this work we propose to use tweets posted on the social Network Twitter, to predict the future price movements of individual stocks listed on the S&P 500, NASDAQ 100 and DJIA. Therefore in this work tweets with explicit sentiment were used to classify company related tweets. Tweets with explicit sentiment are identified by their hashtags and, afterwards, are used to build a sentiment model that classifies company's tweets. Then, with the sentiment of the tweets of each company we use a GA to create a trading rule that maximizes profit. The profit of our simulations will be compared to a buy and hold strategy. The main contributions of this work are:

- A set of procedures that filter out tweets without genuine emotive content;
- The use of the Circumplex Model of Affect to help map a broad range of human emotions;
- A sentiment model built with tweets collected using emotion words, in hashtag form, as search term;
- A set of rules that uses sentiment to choose which companies to buy.

In this paper besides the present first section, Introduction, we have section II - Background and State-of-the-Art, that contains theory and key concepts necessary to understand the procedures proposed in this work. Section III – Architecture of the proposed solution, that presents all the procedures performed to build the proposed solution. Section IV – Results, showing a few practical applications of the designed system on the stock market and describing their performance and, finally, section V – Conclusions and future work, in which we summarize the effectiveness of the results obtained and refer the limitations of the proposed solution, indicating possible improvements in our system.

II. BACKGROUND AND STATE-OF-THE-ART

In this section we discuss the role Twitter can have on the stock market analysis and provide some context, a few basic concepts and useful tools that are essential to understand sentiment analysis. Additionally we review related work regarding the use of Twitter in market analysis.

A. *The Power of Twitter over the stock market*

The power Twitter can have over the stock

market has been shown in the past. On March 30, 2015, Elon Musk, CEO of Tesla, tweeted “Major new Tesla product line -- not a car -- will be unveiled at our Hawthorne Design Studio on Thurs 8pm, April 30”. This soared the company value on the stock market by approximately one billion US\$, in just a few minutes. On a similar note, investor Carl Icahn, on August 13, 2013, tweeted “We currently have a large position in APPLE. We believe the company to be extremely undervalued. Spoke to Tim Cook today. More to come”. This boosted capitalization by more than ten billion US\$. On April 23, 2013, a twitter account of the Association Press, an American news organization, got hacked and hackers posted “Breaking: Two Explosions in the White House and Barack Obama is injured”. In the next three minutes the S&P index fell nearly 1%, that means that investors took more than 130 billion US\$ from the market in that time period [2].

The importance of Twitter is also demonstrated by the fact that companies, themselves, are starting to publish share price altering information, like earnings and reports, on twitter, even though, that information might be available elsewhere. Consequently, investors are already using data from Twitter to help them pick better trades. StockTwits [3] is a pioneer example of this. StockTwits “listens” to the stock market related chatter and gauges if the market is bearish or bullish.

B. Sentiment Analysis

Opinion mining or sentiment analysis is the computational interpretation of opinions, sentiments, affects, subjectivity, views or emotions, present in texts. The subject of sentiment analysis encompasses many different tasks, such as, opinion mining, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, review mining and others. However, now, they can be referred to as sentiment analysis or opinion mining [4].

Common Terms used in Sentiment Analysis

Before digging deeper into sentiment analysis related texts, there are a few basic concepts one needs to be familiarized with to be able to understand the procedures performed on the related research.

- **Emoticons** are used to express a feeling or a person’s mood and consist of a text representation of facial expressions, or feelings, with punctuation, numbers and letters. Example emoticons are “:-)”, “:-(”, “D:<”, “;)”.
- **Token** is a component of a text. It is going to be used as feature for the classifier. It can be a single word or two or even a question mark or an exclamation point. For example, in the tweet “This bread is stale :(“ the single word tokens are “This”, “bread”, “is”, and “stale”.
- **Tokenization** is the process of splitting a document into tokens. There is no single right way to do tokenization. The right technique is very application dependent. Sentences can be broken up on white spaces or a set of special characters or symbols or punctuation among others. An example of a tokenization strategy is the Whitespace tokenizer.
- **N-grams** is the number of words to include in each token. Typically, if we want to split a sentence into n-

grams, we take the first n words as a token, move one word forward and take that set of n words as another token. Finally, repeat these steps until the last word of the sentence is reached.

- **Document** in sentiment analysis is a single entity in a corpus. It is where the text to be processed and classified is. It is a collection of terms that belong to a single class. After the training process the classifier computes the probability of any given document belonging to a certain class. A document can be, for example, a tweet, a review, a blog post, an e-mail, or an article.
- **Corpus** is the collection of documents to be used in the sentiment analysis process. It can be a set of tweets, or articles, or reviews, or forum posts.
- **Document-term matrix** is merely a table that tells us how frequently each term, or token, appears in a corpus. It is a very common way of representing text for further computation [5], [6]. In a document-term matrix one has documents as rows, for example, tweets, and term frequencies as columns which are merely the number of times a term appears in that particular tweet. A document-term matrix for 2 example tweets can be seen in Table 1.

Table 1 - Example of a document-term matrix for the tweets “This cat hates me” and “This cat woke me up”.

	“cat”	“hates”	“me”	“This”	“up”	“woke”
Tweet 1	1	1	1	1	0	0
Tweet 2	1	0	1	1	1	1

- **Stemming**, in linguistics, is the process of reducing a word to its stem or root. That stem or root doesn't need to be a valid word, as defined in the English dictionary, it just has to map or be a representation of a collection of related words. For example, a stemming algorithm should identify the words “fisher”, “fishing”, “fished”, “fisherman” and replace them with the word “fish”. Stemming the words “argue”, “argued”, “argues”, “arguing”, and “argus” results in the stem “argu”, which is not a valid word but is a valid stem. This is a common technique that reduces complexity, without any significant loss in entropy, because it performs a dimension reduction. One of the best known stemming algorithms is the Porter's stemmer [7].
- **Stop words** are used to refer the most commonly used words in a language. Because they are so common, it is assumed they have zero emotional weight and hence are removed. Eliminating words such as “and”, “the”, “this”, “to”, “a”, “that”, “too”, “so” and “or” can bring a significant increase in precision in that, more information can be gained considering just the remaining words in the sentence rather than considering all the original words. The adequate stop words list for removal can be very application dependent. There are lists with just a few words and others with hundreds of words.

C. Related Work

Vu *et al.* [8] captured public mood from

Twitter to forecast the up and down stock price movement of four tech companies. They achieved an accuracy of 82.93%, 80.49%, 75.61%, and 75.00% for Apple, Google, Microsoft and Amazon, respectively, for a period of 41 days. Another case study intended to investigate if there is a type of company whose stock price is more easily predictable analyzing public sentiment using Twitter messages [9]. Hu *et al.* [10] proposed to study the correlation between one important user's tweets, the CEO of Tesla – a famous automobile company – Elon Musk and the price behavior of that company's stock using a Support Vector Machine algorithm. Xu and Keelj [11] tested whether social media sentiment is useful in a stock market forecasting. They proposed a two-stage method for sentiment analysis and stock market price movement prediction. They were able to predict up and down stock price movements with an accuracy of 58.9%. Bollen *et al.* [12] attempted to predict the price movement of the DJIA using daily Twitter feeds. They claimed an 87.6% accuracy predicting the closing price movement direction, although their test period starts at December 1, 2008 and ends on December 19, 2008. Another example evaluates the correlation between Twitter posts and stock market composites such as the Dow Jones, NASDAQ and S&P 500 [13]. In this work it was observed that the percentage of tweets with emotional content was negatively correlated with the three stock market indicators mentioned above, but positively correlated with the VIX (Chicago Board Options Exchange Volatility Index). Mittal and Goel [14] tried to find a correlation between public sentiment and market sentiment particularly they use Twitter to predict the DJIA price movements. Si *et al.* [15] proposed a new technique that uses a continuous Dirichlet Process Mixture (DPM) model to estimate the number of topics on Twitter streaming messages. With the application of this model, a sentiment score was calculated for each topic and a sentiment time series was built. The sentiment time series together with the stock market price series S&P100 (top 100 constituents of the Standard & Poor's index) were regressed to predict future price movements. Ranco *et al.* [16] investigated the relation between the volume and sentiment of tweets and the stock market performance of the 30 companies on the Dow Jones Industrial Average over a period of 15 months. During this time, the correlation and Granger causality between sentiment and stock price was found to be very low, except during peaks of Twitter activity. In fact, when Twitter activity surged a significant dependence between sentiment and stock returns was found. Porshnev *et al.* [17] proposed a model that used a lexicon-based approach to identify eight basic emotional states of Twitter users and improve accuracy of stock market predictions using Support Vector Machines and Neural Network algorithms on DJIA and S&P500 composites.

III. PROPOSED SYSTEM ARCHITECTURE

A. System's Overview

The rationale of our work was to use Twitter together with a genetic algorithm optimization to predict future price movements of companies quoted on

the stock market. More concretely, we intended to build a system that is able to automatically interpret the emotional content expressed on tweets in order to deduce what would be the future price movements of individual stocks on the financial market. A practical example of the proposed solution is stated below.

We started by searching and downloading tweets with emotion words. Here, emotion words and what we use as search terms on Twitter are for example the hashtags “#happy” and “#sad” and, for explanation purposes, we will stick with these two for now. Resuming, we search and download tweets that contain the term “#happy” or “#sad”. These tweets have nothing to do with the stock market or, if they do, it is purely coincidental. In fact, the contexts in which the term “#happy” appears can be very different from one tweet to another. The same applies to the term “#sad”. Examples of tweets downloaded using the term “#happy” are “#Happy #Friday! We hope you all have a beautiful day and a lovely weekend! #DavidJones #SayCheese” and “#popcorn makes me happy lol #movietime #happy”. Examples of tweets downloaded using the term “#sad” are “I love being #sad” and “December 2nd and it won't stop raining... #NoSnow #sad #december #foggy”.

Once we have these tweets and their corresponding labels we can build a sentiment model, using a classifier that performs statistical analysis. Here the labels could be simply happy or sad but, in our work, we grouped different emotion words into the same class, as explained in the next sections.

The objective is to differentiate the characteristics of sad tweets from the characteristics of happy tweets, or, in other words, what words are present in the happy tweets that are not present in the sad tweets and *vice versa*. After this, we take tweets mentioning a company, e.g. Google, and use the built model to tell us if the Google related tweets are more similar to the happy tweets class or the sad tweets class. We then do the same for a group of companies and assess their performance on the stock market in order to understand if the companies whose tweets were classified as happy perform better or worse than the companies whose tweets were classified as sad.

B. Architecture

In order to perform the different tasks required we created modules to handle each step. A diagram of the proposed solution and the inputs and outputs of each module are outlined in Figure 1.

The first module downloads tweets, using the Twitter API, and stores them on files. Basically, there were two types of tweets that were downloaded and stored. These types can be differentiated according to the type of search term used to query the tweet which was either a sentiment word or a publicly traded company's ticker.

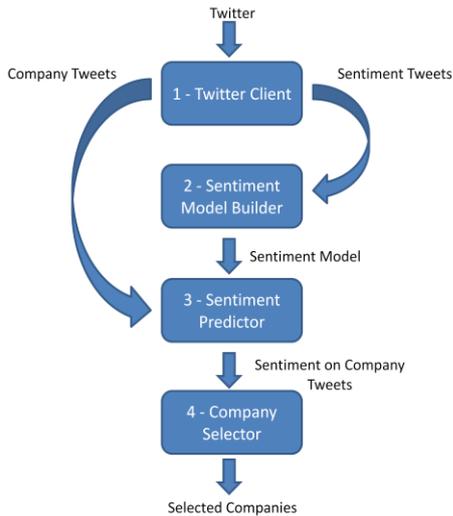
The second module reads collected tweets using the sentiment words from the corresponding files, preprocesses them and learns a sentiment model from them.

The third module reads tweets collected using company's tickers, from the corresponding files, preprocesses and uses the model, built in the second

module, to measure the sentiment in them. The companies whose ticker we chose to follow on Twitter are the ones that belong to the S&P 500, NASDAQ 100 and Dow Jones 30.

The fourth and final module takes the sentiment from the tweets of every company in the list and, using custom made trading rule together with a genetic algorithm, outputs which are the best to buy.

Figure 1 – Proposed system architecture



1st Module: Query and Store Tweets

This module is required for downloading tweets using the Twitter Search API. The goal here was to download tweets with a single specific emotion so that the classifier could correlate words with emotion. For this we made use of the Twitter hashtags. Hashtags are words preceded by the character “#”. They are very common in Twitter messages. Hashtags are a useful feature for sentiment and emotion classification because they can help associate messages and emotions as just the mere presence of a hashtag in a message classifies it emotionally. For example, the tweet “Thankful for unexpected day off #happy” ends with the hashtag #happy that perfectly describes how the user is feeling.

i. Circumplex Model of Affect

In order to map a broad range of the human sentiment we picked 80 emotion words that were used in hashtag form as search terms on Twitter. Essentially, they have the same purpose as the terms “#happy” and “#sad” referred before. These words were chosen based both on the Circumplex Model of Affect [18] and on EMOTEX [19].

The Circumplex Model of Affect, Figure 2 [20] (Used with copyright clearance), is an integrative approach to affective neuroscience, cognitive development, and psychopathology. This model proposes that all affective states can be expressed as a combination of two fundamental neurophysiological systems, pleasure and activity or arousal, so one can map all emotions in a two-dimensional plot where each human emotion is a linear combination of different degrees of both dimensions.

Each of the 80 chosen words belongs to one of four classes: Activity+ / Pleasure+; Activity- / Pleasure+; Activity+ / Pleasure- and; Activity- /

Pleasure-. The hastags for each class are listed below.

Class 1 - Activity+ / Pleasure+ → #elated, #overjoyed, #enjoy, #excited, #proud, #joyful, #feelhappy, #sohappy, #veryhappy, #happy, #superhappy, #happytweet, #feelblessed, #blessed, #amazing, #wonderful, #excelent, #delighted, #enthusiastic.

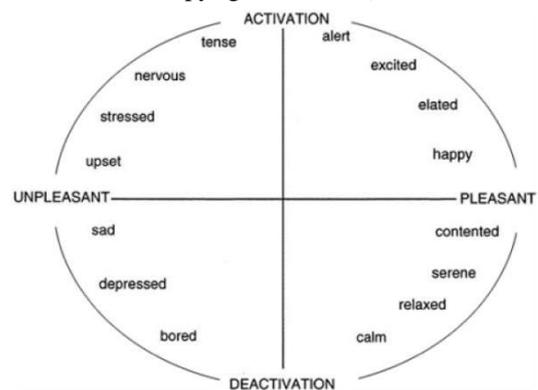
Class 2 - Activity- / Pleasure+ → #calm, #calming, #peaceful, #quiet, #silent, #serene, #convinced, #consent, #contented, #contentment, #satisfied, #relax, #relaxed, #relaxing, #sleepy, #sleepyhead, #asleep, #resting, #restful, #placid.

Class 3 - Activity+ / Pleasure- → #nervous, #anxious, #tension, #afraid, #fearful, #angry, #annoyed, #annoying, #stress, #distressed, #distress, #stressful, #stressed, #worried, #tense, #bothered, #disturbed, #irritated, #mad, #furious.

Class 4 - Activity- / Pleasure- → #sad, #ifeelsad, #feelsad, #sosad, #verysad, #sorrow, #disappointed, #supersad, #miserable, #hopeless, #depress, #depressed, #depression, #fatigued, #gloomy, #nothappy, #unhappy, #suicidal, #downhearted, #hapless, #dispirited.

In essence, each of these four classes represents a quadrant as shown in the Figure 2.

Figure 2 - Circumplex Model of Affect (Used with copyright clearance).



ii. Twitter Client

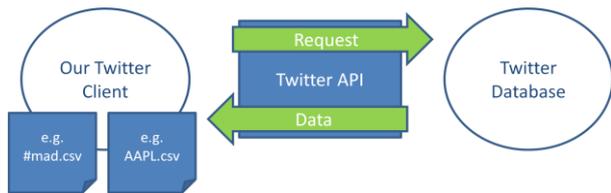
As for sentiment tweets, the same procedure is applied for company tweets, as can be seen in Figure 3. The data flow is quite intuitive. Essentially, our Twitter client makes requests to the Twitter API in order to access tweets with the required search terms, which, in turn, are stored in the appropriate files. The search term can be either a company or one of the 80 sentiment words.

We collected tweets from a group of companies using their tickers on the stock market prefixed by a dollar sign, “\$”. The company’s ticker is its symbol, usually three or four capital letters long, on the stock market. For instance, the ticker of the Apple Inc. is AAPL and to download tweets which subject is the Apple company, we used the search term “\$AAPL”. On Twitter this is called a cashtag.

We opted to download tweets related to all companies present one the S&P 500, NASDAQ 100, and Dow Jones Industrial Average 30. As most of the companies present in the NASDAQ 100 and the Dow Jones 30 are also present the S&P 500, the list contains

504 unique names (on 2016-05-18).

Figure 3 - Twitter data flow.



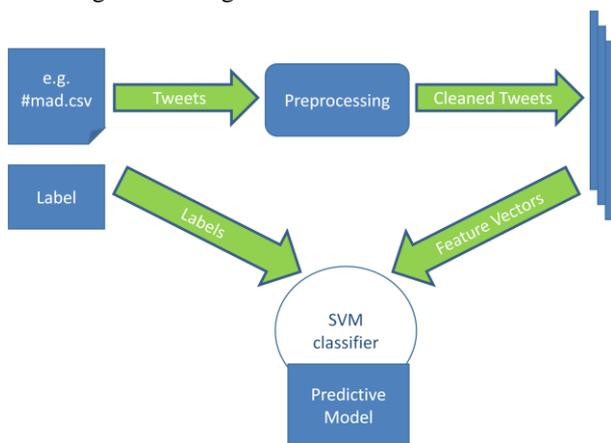
2nd Module: The sentiment model

To test the influence of each technique over the classification results, we measure the increase or decrease, in accuracy on predicting to which of the four classes our test data tweets belong to. In other words, suppose that, after preprocessing, we have 30 000 tweets. Each of these tweets belongs to one of four classes mentioned before. We use 80% of these tweets as training data, to train an SVM classifier and test its prediction accuracy on the remaining 20%, the testing data. Therefore, the trained model is going to make an educated guess in order to predict the class of each tweet on the test sample and it is this accuracy, or correct predictions, on the test sample, that we try to increase with the techniques described below.

i. Setup

Figure 4 presents the diagram of the module. The learning algorithm, in this case an SVM classifier, takes feature vectors, the document-term matrix, built from preprocessed tweets from all emotion words with the corresponding labels. The label given to the tweets of each emotion follows class enumeration mentioned before. After the learning process we end up with a model, the Predictive Model, capable of making predictions on unseen tweets.

Figure 4 - Diagram of the Sentiment Model.



ii. Preprocessing Tasks

Because the tweets' text content is highly unstructured we need to process it before it is turned into feature vectors. The following paragraphs describe all the preprocessing techniques we performed.

Harmonization of User Influence

On the downloaded tweets we observed that many users posted countless tweets that have very little difference between them. For instance, one user posted

dozens of tweets similar to "I am intelligent. #I_AM #positive #affirmation #happy" and "I am thankful for being able to choose. #thankful #positive #happy". Variations of the former tweet can be replicated by changing the word "intelligent" by a number of other adjectives, such as, "harmonious", "thankful", "genuine", "generous", "ok", "independent", "imaginative" and others. Variations of the latter tweet imply replacing the words "being able to choose" by "air fresheners", "vacation days", "being able to see", "coffee", "electricity", "fall leaves" and others. The point here is that if we keep these tweets and include them in our classifier, the results of the classification are going to be severely biased by the words used by such users.

In order to fix this we decided to discard tweets posted by users from whom we collected an unusual number of tweets and tweets contain many common words. Accuracies were higher when we discarded both tweets posted by users from whom we collected more than 3 posts and tweets with more than 50% of equal words.

We found that this method helped preventing a single user, from whom we collected a large number of tweets, having too much influence on the classification results and if he or she still had an above average number of tweets on our dataset, they must have a somewhat diverse content.

Usernames, Exclamation Points and Question Marks

On Twitter one can identify another person using the character '@' immediately before the username, e.g. @James. All the words prefixed by the char '@' were replaced by "UsernameHere". This was necessary to avoid a specific username from being, itself, a predictor of sentiment. Likewise, all exclamation points and question marks were replaced by the tags "ExclamationPointHere", "QuestionMarkHere" respectively.

Hashtags at the End of the Tweet

Hashtags that are at the end of a tweet and are not part on the sentence were striped. So, for example, in the tweet "Wonderful place for an early evening dog walk #peaceful #unspoil", the hashtags "#peaceful" and "#unspoil" were removed but in the tweet "#Saakalya wishes you all a #peaceful and #happy long weekend :)" the hashtags "#peaceful" and "#happy" were kept. This helps the classifier not to put too much weight on the hashtags because wanted the classifying algorithm to be able to correctly identify sentiment on company related tweets even if they don't have hashtags such as "#happy" or "#peaceful". In other words, we wanted to improve accuracy and generalization. If the hashtags are in the middle of the tweet, they are part of the sentence and essential to understand its meaning, hence, they were not removed.

Links

Many tweets contain links. We considered, like we did with usernames, replacing link occurrences by "urlHere" but this did not improve accuracy, thus, we simply remove them.

Repeated Letters

It is common on Twitter to repeat one letter in a word to accentuate it's meaning, e.g. "joyyyyy". In this work a letter repeated more than two times was replaced by three occurrences so "joyyyyy" becomes "joyyy". We chose to replace multiple letter repetitions by three occurrences, instead of just one, so we could differentiate a word's normal usage from its accentuated usage, hence "joyyyyy" becomes "joyyy" and not "joy".

Emoticons

Table 2 lists emoticons used and the corresponding categories. All the emoticons listed were replaced by "HappyEmoticon", "SadEmoticon" and "HeartEmoticon", according to their category.

Non-alpha-numeric Characters

To conclude the pre-processing steps, all non-alpha-numeric characters, sequences of two or more white spaces and white spaces at the end or beginning of the tweet were trimmed.

Lastly, in order to limit the number of tweets included in the model we favored those with the highest number of retweets plus favorites. This was necessary because we had a limited amount of processing power and including all the millions of tweets we collected in our classifier would require a much larger amount of time to train and test different models.

Table 2 - List of emoticons used for each category in this work.

Category	Emoticons
Happy	:)) :) :) :D :o) :] :3 :c) :> =] 8) =) :}
Sad	>:[:-(:(:-c :c :-< < :-[:{
Heart	<3

iii. Feature Vectors

Before we fed Twitter data into the learning algorithm we performed a couple of extra tasks.

The first task was to build the feature vectors, in this case, a document-term matrix, as shown in Table 1. The elements on the matrix are numeric counters of terms in documents. When this is the case, it is said we have a term frequency weighing. However, in this work, the best results were achieved using Term Frequency - Inverse Document Frequency weighing (TF-IDF) which, for explanation purposes, can be broken up in two parts as seen in the formula

$$TF - IDF = f_{t,d} \times \log \frac{N}{n_t}, \quad (1)$$

where term frequency (TF) $f_{t,d}$ is merely a measure of how frequently a term, or feature, occurs in a tweet or document. The inverse document frequency (IDF) $\log \frac{N}{n_t}$ measures how important a term is across multiple tweets. It is calculated as the logarithm of the total number of tweets (N) divided by the number of tweets containing the term (n_t). Thus if a term occurs many times in a tweet, it gets a high TF score, but, if that term appears in most tweets, it gets a low IDF score. In the specific case where a term appears in every tweet, it's IDF score is zero. The TF-IDF is the multiplication of TF by IDF therefore, even if a term has a high TF

weight, it can become less discriminative after the multiplication by the IDF weight.

iv. Training The Sentiment Model

After all the previous steps were taken building and training the sentiment model is the simplest part, specifically, we used an SVM algorithm implemented on the R package *e1071* [22] with a radial kernel, gamma equal to 1 and a cost of 100. The feature vectors or, the document-term matrix was built with the *RtextTools* package [21].

The goal was, after training the classifier with the first set of tweets (sentiment tweets), to use it on tweets from a specific company, to assert which of those four sentiment classes prevails or assumes a dominant presence. The same was done for all companies and their performance on the stock market was compared.

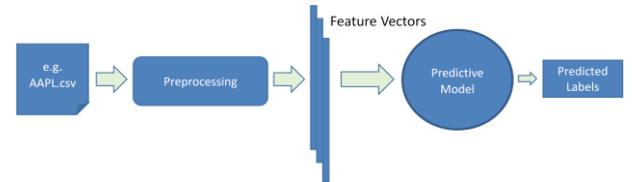
3rd Module: Company Tweets Classification

This module uses the model built and learned, from the tweets collected using the 80 sentiment words, to classify tweets from each company.

i. Preprocessing

As shown in Figure 5, company's tweets were read from the corresponding files and preprocessed. These preprocessing steps were the same as those applied on sentiment tweets that are performed by our custom Java function.

Figure 5 - Sentiment prediction on company's tweets.



ii. Tweet Selection

To predict the price movement on any business day, we gauged the sentiment from tweets posted in the previous non trading hours. Suppose we want to predict the price movement (up or down) of the Google stock or, put simply, we want to know if the closing price at 16:00 is bigger or smaller than the open price at 9:30 (New York Time). For that, we measure the sentiment on Google tweets posted between the previous market close time, i.e. 16:00 on the previous day, and the current day market open time, i.e. 9:30. This means that only tweets posted outside trading period were used for prediction, those others posted during trading hours were excluded. Furthermore, in order to predict price movement on a Monday the same method is applied, that is, we used tweets posted between 16:00 of the previous Sunday and 09:30 Monday.

Although the NYSE and NASDAQ exchanges close at 16:00, we do not include tweets posted during the hour immediately after that because our simulations show that excluding them leads to better results. Also, the exchanges open at 09:30, but we stop at 09:27 to allow 3 minutes time for processing and submitting the eventual purchasing orders to the stock market.

iii. Output

With the selected tweets we built a document-term matrix which was fed into the classifier model we built. For each tweet fed into the sentiment model we got two output values. One is the class attributed by the classifier to the tweet and the second is a number between zero and one that measures the certitude of the classifier on the first value. This implies that if the second value is 1 the classifier is 100% sure the tweet belongs to the class specified by the first value. The first value was named SVM_CLASS and the second SVM_PROB.

4th Module: Stock Selection

So far we have a collection of companies, their related tweets and corresponding predictions, thus a method is necessary to transform this into buy signals for the stock market.

Table 3 - Typical data table produced in module 3 in the used architecture.

Tweet	SVM_CLASS	SVM_PROB
Tweet 1	1	0.37
Tweet 2	4	0.66
Tweet 3	3	0.91
Tweet 4	3	0.44
Tweet 5	1	0.86
Tweet 6	3	0.52

i. Remove Neutral Tweets

Table 3 is an example of how the data was available to us from module 3. Notice that, at this stage, we only have tweets that were posted during off trading hours. Also, an important point is that there will be tweets whose sentiment is hard to discern, even if they were labeled by a human. However, independently of how hard it may be to understand the sentiment shown on a tweet, our classifier will most certainly have a favorite class to attribute to any given tweet. For example, given a tweet, the classifier can decide that there is 25.1%, 25.0%, 25.0%, 24.9% chance it belongs to classes 1, 2, 3 and 4 respectively, so, the tweet will be given the values 1 and 0,251 to SVM_CLASS and SVM_PROB respectively, when, in fact, for practical reasons, this situation should be called a draw. Moreover, there are, indeed, tweets whose sentiment is neutral and a rule needs to be created to account for and exclude tweets whose sentiment is hardly obvious. To achieve this goal, tweets whose SVM_PROB is less than a specific threshold are excluded. That threshold is going to be mentioned again later but, for now, suppose that it is equal to 0.5. This means that a tweet's sentiment is only going to be used if the classifier is at least 50% sure about its sentiment class, so, on the example in Table 3, tweet 1 and tweet 4 would be eliminated.

ii. Daily Data Congregation

Each day, when we want to pick the best companies to buy, we take all tweets posted on the previous off trading hours and summarize them in 13 values: 4 for the number of tweets belonging to each

SVM_CLASS; 1 for the total number of tweets; 4 for the average SVM_PROB values for each class and; 4 for the percentage of tweets from each SVM_CLASS. Those 13 values refer to a single day period and were calculated for each company in our set. Using Table 3 as an example, we constructed Table 4 with the 13 variables.

Table 4 - Daily Summary Table of the companies.

Day, e.g., 2016-06-23, Company, e.g., Apple Inc.	
Number of tweets with SVM_CLASS 1	1
Number of tweets with SVM_CLASS 2	0
Number of tweets with SVM_CLASS 3	2
Number of tweets with SVM_CLASS 4	1
Number of total tweets	4
Average SVM_PROB of tweets with SVM_CLASS 1	0.86
Average SVM_PROB of tweets with SVM_CLASS 2	0.0
Average SVM_PROB of tweets with SVM_CLASS 3	$(0.52+0.91)/2 = 0.715$
Average SVM_PROB of tweets with SVM_CLASS 4	0.66
Percentage SVM_CLASS 1 tweets over total number of tweets	$\frac{1}{4} = 0.25$
Percentage SVM_CLASS 2 tweets over total number of tweets	0.0
Percentage SVM_CLASS 3 tweets over total number of tweets	$\frac{2}{4} = 0.5$
Percentage SVM_CLASS 4 tweets over total number of tweets	$\frac{1}{4} = 0.25$

The values given to each variable in Table 4 were calculated based on the values on Table 3. The proposed setup is to have a table, just like Table 4, for each and every company we are looking into. The decision of which ones to buy is going to be made according to the values on those tables. The same procedure is taken every day we want to choose what the best companies to buy are.

iii. The Trading Rule optimized by a Genetic Algorithm

We arrive here with a summary table (see Table 4) for each of the 504 companies. Now we need a way to choose which of those companies to buy and which to leave well alone on any one day. Keep in mind that, through all the steps described above, such as limiting the number of hashtags per tweet or limiting the number of tweets per user or removing tweets with a low SVM_PROB or a low retweet count, amongst others, we have seriously reduced our tweets sample size. This is not that surprising if we consider that Twitter data can be very noisy, there is little information in it or, more precisely, there is a lot of contradictory information from a large number of sources. The point is that the summary table can be quite different from company to company and from one day to the next, so, we need a flexible rule that is able to pick the better companies to buy using very divergent tables. To design a trading rule we created 5 variables:

- *Minimum_Number_of_Tweets*;
- *Minimum_Number_of_Companies*;
- *Number_of_Companies_to_Buy*;
- *Variable_on_Summary_Table*;
- *Minimum_Probability*.

Minimum_Number_of_Tweets and *Minimum_Number_of_Companies* are integer numbers representing a number of tweets and a number of companies respectively. *Number_of_Companies_to_Buy* and *Variable_on_Summary_Table* are also integer numbers that refer to the number of companies to buy and one of the 13 variables in the summary table respectively. Lastly, *Minimum_Probability* is a decimal number that represents a probability. Table 5 lists all values these variables can take. The values of these variables are essential to the functionality of the system. The trading idea is explained below.

Suppose that on a particular day we want to decide which companies to buy and what we have from each of them is the summary table. Then, we take the tables of all the companies and subject them to a set of criterions:

- **Criterion 1** - All tweets with an SVM_PROB that is less than *Minimum_Probability* are excluded and only the ones that remain enter in the following calculations.
- **Criterion 2** - If a company has on its summary table *Minimum_Number_of_Tweets* or less tweets, the company is excluded.
- **Criterion 3** - If, for a given day, the total number of companies that passed **Criterion 2** is less than or equal to *Minimum_Number_of_Companies* no company is selected, that is, no buying signal is produced for the day.
- **Criterion 4** - From the list of companies that passed **Criterion 1**, **Criterion 2** and **Criterion 3** we select the *Number_of_Companies_to_Buy* with the highest values of *Variable_on_Summary_Table*.

Table 5 - Domains of the customized variables used with the *genoud* function.

Variable	Domain
<i>Minimum_Number_of_Tweets</i>	[0 - 20]
<i>Minimum_Number_of_Companies</i>	[1 - 20]
<i>Number_of_Companies_to_Buy</i>	[1 - 10]
<i>Variable_on_Summary_Table</i>	[1 - 13]
<i>Minimum_Probability</i>	[0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]

To calculate the best values for our 5 custom variables we used a genetic algorithm implementation. The chromosome contained the values for each of the 5 variables that are to be optimized. In order to implement a genetic algorithm optimization we used the *rgenoud* R package [23] with a fitness function that accounts for the total profit/loss for the period. The built in function in the *rgenoud* R package takes in over 40 arguments and the package authors provide very good documentation. Apart from variable domains and the

fitness function we used the default values for the other arguments such as 1000 for the population size and 100 for the maximum number of generations.

From Table 5 we can see that the search space or, the number of possible solutions is $21 \times 20 \times 10 \times 13 \times 7 = 382200$. We understand the search space is not massive but, as it is, our framework is prepared to handle a further increase in complexity if future work is pursued.

IV. RESULTS

In this section we present the results that we produced using the proposed system for trading on the stock market. We present the data of two different case studies.

Stock prices were obtained from Google Finance and our training period starts at May 18 and ends at September 2, 2016. We used tweets posted during this period to learn the best parameter combinations and to create 2 strategies, case study 1 and 2. The case studies strategies were tested on the period from November 7, 2016 to December 16, 2016.

Every stock we acquire was bought at market open (time and price) and every position we close was done at market close (time and price). Thus, if we buy a stock at open and sell it at close on the same day our holding period is one day.

A. Case Study 1

In this case study the trading rule is to buy 2 companies with the higher values of the variable 6 on the summary table provided that there are at least 20 companies with more than 11 tweets on their summary table with an SVM_PROB greater than 30%. Table 6 lists key simulation parameters.

Table 6 - Simulation parameters for case study 1.

Parameter	Value
Trading frequency	Daily
Holding Period (Days)	1
<i>Minimum_Number_of_Tweets</i>	11
<i>Minimum_Number_of_Companies</i>	20
<i>Number_of_Companies_to_Buy</i>	2
<i>Variable_on_Summary_Table</i>	6
<i>Minimum_Probability</i>	0.3

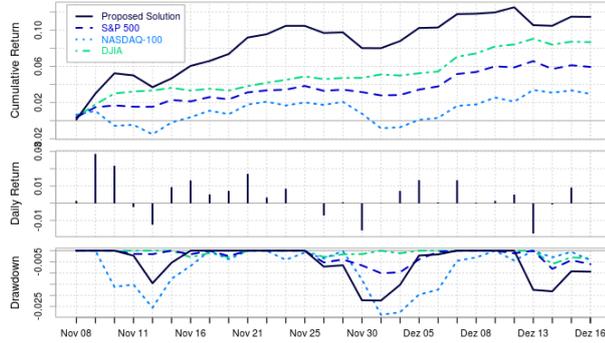
The return on investment (ROI) and drawdown achieved using the parameters of this case study are presented in Figure 6 alongside the cumulative returns of the S&P 500, NASDAQ 100 and DJIA. In December 16 the ROI was 11.47%, 5.94%, 2.96% and 8.67% for the proposed solution, S&P 500, NASDAQ 100 and DJIA, respectively.

The peak drawdown for the proposed solution is -2.08%, which is better than the NASDAQ 100 drawdown at -2.91% but worse than the drawdown of the S&P 500 and DJIA at -0.98% and -0.55% respectively.

The trading strategy described in this case study entered the market in 27 days, each time buying two different stocks. This corresponds to 54 trades of

which 35 had a positive return and 19 a negative return. The hit ratio is then $35/54 = 64.81\%$ and of the 27 days 70.37% ended with a profit. The same figure for the daily returns of the S&P 500, NASDAQ 100 and DJIA was 64.29% , 64.29% and 78.57% , respectively.

Figure 6 - Performance of case study 1 strategy.



B. Case Study 2

In this case study the trading rule is to buy 1 company with the highest value of the variable 11 on the summary table provided that there are at least 15 companies with more than 4 tweets on their summary table with an SVM_PROB greater than 50%. Table 7 lists key simulation parameters.

The ROI and drawdown achieved using the parameters of this case study are presented in . alongside the cumulative returns of the S&P 500, NASDAQ 100 and DJIA. In December 16 the ROI was 10.53% , 5.94% , 2.96% and 8.67% for the proposed solution, S&P 500, NASDAQ 100 and DJIA, respectively.

The peak drawdown for the proposed solution is -3.76% , which is better than the NASDAQ 100 drawdown at -2.91% but worse than the drawdown of the S&P 500 and DJIA at -0.98% and -0.55% respectively.

Table 7 - Simulation parameters for case study 2.

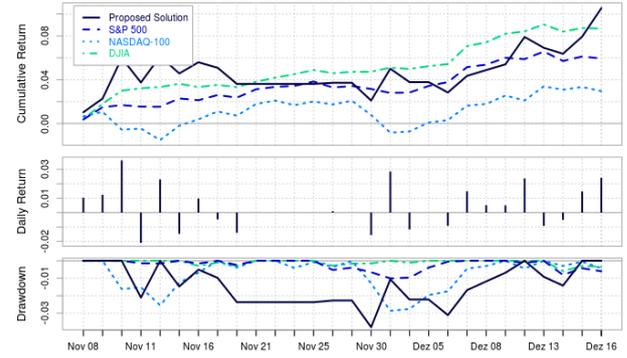
Parameter	Value
Trading frequency	Daily
Holding Period (Days)	1
<i>Minimum Number of Tweets</i>	4
<i>Minimum Number of Companies</i>	15
<i>Number of Companies to Buy</i>	1
<i>Variable on Summary Table</i>	11
<i>Minimum Probability</i>	0.5

The trading strategy described in this case study entered the market in 22 days, each time buying one company stock. This corresponds to 22 trades of which 13 had a positive return and 9 a negative return. The hit ratio is then $13/22 = 59.09\%$ which is also the percentage of days ending with a profit.

Case study 2 is our best solution as given by the genetic algorithm (in the training period) but it is also the riskiest because the whole investment is concentrated in only one company's stock. Case study 1 is also the best solution provided that we distribute the

investment in two different stocks.

Figure 7 – Performance of case study 2 strategy.



V. CONCLUSIONS AND FUTURE WORK

In this work we implemented a system that predicts stock market fluctuations using public mood collected from Twitter. To achieve that, we built a sentiment model that detects the sentiment present on tweets from different companies. Using this sentiment we built a trading rule that was optimized by a genetic algorithm in order to achieve the maximum profit.

The two case studies performed show how robust is the model when tested for different metrics such as return on investment, drawdown and risk. Based on these evidences we can state that it is possible to implement a profitable trading strategy using Twitter.

Future work should be devoted to adding more parameters to the GA optimization and to the development of stronger calibration in order to achieve better performances.

VI. ACKNOWLEDGEMENTS

This work was supported in part by Fundação para a Ciência e a Tecnologia (Project UID/EEA/50008/2013).

VII. REFERENCES

- [1] The New York Times - Computers That Trade on the News. Retrieved from http://www.nytimes.com/2010/12/23/business/23trading.html?_r=0.
- [2] The Telegraph - 'Bogus' AP tweet about explosion at the White House wipes billions off US markets. Retrieved from <http://www.telegraph.co.uk/finance/markets/10013768/Bogus-AP-tweet-about-explosion-at-the-White-House-wipes-billions-off-US-markets.html>.
- [3] Stocktwits - <http://stocktwits.com/>.
- [4] Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1), 1-167.
- [5] Shawe-Taylor, J., & Cristianini, N. (2004). Kernel methods for pattern analysis. Cambridge university press.
- [6] Berry, M. W. (2004). Survey of text mining. Computing Reviews, 45(9), 548.
- [7] Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130-137.
- [8] Vu, T. T., Chang, S., Ha, Q. T., & Collier, N. (2012). An experiment in integrating sentiment features for tech stock prediction in twitter.

- [9] Bing, L., Chan, K. C., & Ou, C. (2014, November). Public sentiment analysis in Twitter data for prediction of a company's stock price movements. In e-Business Engineering (ICEBE), 2014 IEEE 11th International Conference on (pp. 232-239). IEEE.
- [10] Hu, Z., Jiao, J., & Zhu, J. (2014). Using Tweets to Predict the Stock Market.
- [11] Xu, F., & Keelj, V. (2014, July). Collective Sentiment Mining of Microblogs in 24-Hour Stock Price Movement Prediction. In 2014 IEEE 16th Conference on Business Informatics (Vol. 2, pp. 60-67). IEEE.
- [12] Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1), 1-8.
- [13] Zhang, X., Fuehres, H., & Gloor, P. A. (2011). Predicting stock market indicators through twitter "I hope it is not as bad as I fear". *Procedia-Social and Behavioral Sciences*, 26, 55-62.
- [14] Mittal, A., & Goel, A. (2012). Stock prediction using twitter sentiment analysis. Stanford University, CS229 (2011 [http://cs229.stanford.edu/proj2011/GoelMittal-Stock Market Prediction Using Twitter Sentiment Analysis. pdf](http://cs229.stanford.edu/proj2011/GoelMittal-Stock%20Market%20Prediction%20Using%20Twitter%20Sentiment%20Analysis.pdf)).
- [15] Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., & Deng, X. (2013). Exploiting Topic based Twitter Sentiment for Stock Prediction. *ACL* (2), 2013, 24-29.
- [16] Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., & Mozetič, I. (2015). The effects of Twitter sentiment on stock price returns. *PloS one*, 10(9), e0138441.
- [17] Porshnev, A., Redkin, I., & Shevchenko, A. (2013, December). Machine learning in prediction of stock market indicators based on historical data and data from twitter sentiment analysis. In 2013 IEEE 13th International Conference on Data Mining Workshops (pp. 440-444). IEEE.
- [18] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, pp. 1161– 1178, 1980.
- [19] Hasan, M., Rundensteiner, E., & Agu, E. (2014). Emotex: Detecting emotions in twitter messages.
- [20] Posner, J., Russell, J. A., & Peterson, B. S. (2005). The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, 17(03), 715-734.
- [21] Timothy P. Jurka, Loren Collingwood, Amber E. Boydston, Emiliano Grossman and Wouter van Atteveldt (2014). RTextTools: Automatic Text Classification via Supervised Learning. R package version 1.4.2. <https://CRAN.R-project.org/package=RTextTools>
- [22] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2015). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6-7. <https://CRAN.Rproject.org/package=e1071>
- [23] Walter R. Mebane, Jr., Jasjeet S. Sekhon (2011). Genetic Optimization Using Derivatives: The rgenoud Package for R. *Journal of Statistical Software*, 42(11), 1-26. URL <http://www.jstatsoft.org/v42/i11/>.