Predicting Financial Market Trends through Multi-Agent Simulation using Social Data Mining

Project supervisor: Sebastian Stein
Second examiner: Jeff Reeve

A project report submitted for the award of MEng Computer Science
Abstract

The purpose of this project is to model the short term rise and fall of the foreign exchange market (the forex) using a multi-agent based simulation to create a virtual model of the market and to influence this model externally using sentiment collected from corresponding tweets on the social media platform Twitter. This research aims to see if there is a correlation between Twitter and future market price movement of specific currency pairs when modelled in an agent-based environment. This external influence from Twitter is the first of its kind used to predict and model the specific activity in the foreign exchange market. The model allows predictions of market prices for specific currency pairs along variable time intervals, therefore enabling the user to visualise predicted market movements.

The research done in this project has successfully modelled a virtual forex market and has found a surprisingly high correlation between modelled simulations using Twitter sentiment and the real world trends of the same time period. This simulation framework would hence prove a valuable asset in the toolbox of professional forex traders who wish to have another angle of prediction on the short term future trend of the market.
Acknowledgments and Statement of Originality

I hereby state that I am the sole author of this paper and that the work done in this project is entirely my own unless stated otherwise.

I certify that, to the best of my knowledge, this paper does not infringe upon anyone’s copyright and that any ideas, techniques, quotations, or any other material from the work of other persons have been fully acknowledged in accordance to standard referencing practices.

I would also like to acknowledge and thank this projects supervisor Sebastian Stein for the hard work, valuable input and guidance throughout this project.
Contents List

1 Project Description........................................................................................................page 5

2 Background Research and Literature......................................................................page 7
  2.1 The Foreign Exchange Market........................................................................page 7
  2.2 Twitter..............................................................................................................page 8
  2.3 Multi-Agent Simulations....................................................................................page 8

3 Approach & Implementation.................................................................................page 9
  3.1 Stage 1 – Twitter Mining..................................................................................page 9
  3.2 Stage 2 – Sentiment Analysis..........................................................................page 10
  3.3 Stage 3 – Multi-Agent Simulation....................................................................page 11
    3.3.1 Random Traders........................................................................................page 13
    3.3.2 Momentum Traders....................................................................................page 13
    3.3.3 Inverse Momentum Traders.................................................................page 14
    3.3.4 Fundamentalist Traders............................................................................page 14

4 User Interface Design...............................................................................................page 15

5 Testing..................................................................................................................page 17
  5.1 Unit & Integration Testing.............................................................................page 17
  5.2 Complete System Testing.............................................................................page 18

6 Management..........................................................................................................page 20
  6.1 Risk Assessment.............................................................................................page 21

7 Results..................................................................................................................page 22
  7.1 Analysis of simulations vs. real-world data.....................................................page 23
  7.2 Simulation Patterns & Behaviours.................................................................page 26

8 Conclusion..............................................................................................................page 28

References................................................................................................................page 30

Appendix..................................................................................................................page 32
1 Project Description

The foreign exchange market is a dynamically changing continuous double auction of currency pairs (e.g. Euro/Dollar), the price movement of which is determined by a huge variety of factors from supply and demand to the economic state and GDP of a country. This project’s research is based on the hypothesis that all the factors causing price movements can be summed up into a general sentiment of a specific currency within a currency pair which will ultimately determine the subsequent short term rise or fall of the specified currency pair. This project believes that Twitter is a key source to gather this sentiment information as it provides up to date and generally un-biased views of the great majority which is what this research relies on.

The software framework which this project has developed is aimed at all types of traders, from personal traders who are making investments from their personal assets for their own gain, to large scale traders working for international banks and hedge funds trading currencies on the global scale in the Interbank foreign exchange market.

Currently used conventional modelling methods adopt the use of one or more static algorithms that take past data of the exchange pairs and calculate a general trend from that\textsuperscript{1}, which simulates the behaviour of an individual trader. This project’s approach is to use an agent-based simulation, which simulates the whole market of traders interacting with each other. The reason for using a multi-agent simulation is that it can model complex interactions much more accurately and realistically compared to any static algorithm. Various static algorithms have been common in market prediction for years and some do generally well, these algorithms commonly work by taking a set of historical prices from the market and analysing them by using different mathematic equations (running averages, high-low points, candlestick calculations) to produce a very short term prediction on the general rise or fall of the market. But the limitation of these algorithms is that they usually only predict a very short term rise or fall (the next couple of market iterations, be that next few minutes, hours or days in that specific granularity), but don’t predict actual market behaviour over a longer period (over more than a few market iterations). Multi-Agent simulation uses the concept of many agents in a system to model various objects that contain interactions (e.g. a financial market\textsuperscript{2}). On their own these agents are relatively un-intelligent, governed only by a set of rules of interaction. But with the introduction of many agents into the system, the rules of interaction form much more complex behaviours and patterns that are difficult and in many cases impossible to model using conventional algorithms, as these rules of interaction for each agent are a subset of the algorithms used for static prediction, as mentioned earlier. So what is actually observed in a multi-agent simulation is the result of the interaction of all the various algorithms working together and against each other.

The forex is then, by definition, a multi-agent system with agents being traders (either electronic or actual people), each governed by their own set of rules (but rules in general being similar) of how to trade to make a profit. These traders combined make transactions on the forex and cause a change in the price of various currency pairs. This is what this project is exploiting; by making a virtual multi-agent system of the specific section of the forex that corresponds to specified currency pairs this project will model the movement of the currency pairs ahead of the forex, giving users of this software an advantage in investing on those currency pairs.
An integral feature of this project is that, rather than it being a closed multi-agent system, the agent traders’ “mood” towards the specific currencies in the currency pairs is influenced by mining social media platforms. This project has chosen to use the Twitter platform, as it is one of the world’s largest social media platforms, generating up to 200 million tweets per day[3]. Many financial traders, news agencies and automated trading tools also post tweets about the forex on Twitter and Twitter has in the past been linked to general stock market prediction [4][20] due to its ability to capture the majority sentiment, which in turn drives the mood of the real world traders, causing a market price shift. This project attempts to capture this mood of the majority and inject it into its artificial traders so they act as similarly as possible to their real world counterparts. This will allow the project to simulate the forex more accurately, giving a greater prediction accuracy and hence advantage to traders using this framework.

The framework of this project consist of 3 main stages:

1. Twitter data mining – This gathers the tweets from Twitter on the specified search query(s) and filters them to within a specified date range.
2. Sentiment analysis – This takes the gathered tweets and analyses them on their sentiment for each specified currency.
3. Multi-agent simulation – This is the heart of the project framework and models the specific currency pair section of the market using agents acting as traders in the market with trading decisions influenced by the Twitter sentiment.

Below is a diagram of the system stages that the framework will execute and data passed between each stage[fig. 1.1].

![System stages and links between each stage](image)

fig. 1.1 – System stages and links between each stage

The rest of this report is structured as follows: Section 2 looks at the background research of the three major parts that this project covers. Section 3 gives a technical and detailed overview of the system and modelling approach and implementation. Section 4
covers the visual design aspect of the system as intended for the target audience. Section 5 talks about the testing done on the framework during the development of the project. Section 6 gives an overview of the management of this project and the identified risks and subsequent risk management strategies. Section 7 analyses the results of the simulation outputs and their correlation to the real world markets as well as the behaviour patterns of the simulation. Section 8 concludes by talking about the meaning behind the results analysed on Section 7, the envisaged distribution and use of this framework as well as potential future developments.

2 Background Research & Literature

Background research was a key component in the development of this project and provided a solid base on which to build the framework and make important design decisions. The following section provides an overview of the background research which allows one to understand the complexities and functions of this project.

2.1 The Foreign Exchange Market

This project focuses on predicting the foreign exchange market. Whilst general financial markets underpin the world’s economy, the foreign exchange markets “underpin all of the other financial markets”\(^5\) as they determine how much one country’s currency is worth relative to another country’s. The forex is primarily used by investors for profit. Traders invest in certain currencies with the prediction that they will rise and they will be able to sell them at a profit. For this reason this project provides a valuable tool in trading, as knowing whether a currency in a currency pair will rise or fall against the other currency, and what rise-fall patterns will occur allows traders to make decisions on whether to buy or sell that currency to maximise profits.

The forex employs its own terminology for trading between currency pairs. This projects application adopts the following terminology for its target audience: A currency pair is split into two elements. The “Base” which is the first currency of the pair and the “Quote” which is the second \(^6\). The exchange price between the pair symbolises how many units of the quote price can one unit of the base price buy.

![fig. 2.1 – The Euro vs. US Dollar, with the Euro as Base and US Dollar as Quote]

This project primarily focuses on the Euro vs. US Dollar (EUR/USD) currency pair in its models, as that is the largest and most traded currency pair globally, the most diversely traded (traded by every type of trader, from small to large) and it also has the biggest online social coverage which helps the Twitter data mining aspect of the program.
The forex market in this framework works on time iterations, much the same as the real market (although the real market is a lot more granulated in terms of time iterations as it has a much higher number of traders and volume of transactions). On this market traders place limit orders for each time iteration. A limit order is an order to buy or sell a specific currency at a set price or better (e.g. A buy limit order is executed if the market price is at the limit price or lower). If the traders limit order is executed, that trader will have an open transaction on the market which corresponds to the trader owning a specific volume of a certain currency on the market at the price that the limit order was executed. The trader can choose to close a trade at the current market price to withdraw their transaction from the market and take the profit or loss made from that trade. The market mechanics of this framework are covered in more depth in section 3.3.

2.2 Twitter

Twitter is a social media platform developed first in 2006 by Jack Dorsey. It is known as a micro-messaging service and is powered by tweets which are messages limited to 140 characters[7]. Users on Twitter range from normal people to celebrities, news agencies and businesses. According to a study of Twitter by Pear Analytics in 2009 tweets range from conversations between Twitter users and “pointless babble” which takes up approximately 78% of tweets, to spam, self-promotion and news[8]. The news tweets are what this project is focusing on and as they take up approximately 3.6% of all tweets[8] and there are approximately 200million tweets per day[3] that gives about 7.2million tweets that are news related. Out of those a significant majority are financial tweets and out of the financial tweets many are based around certain currency pairs and the forex. These tweets are mainly produced by 3 types of sources: financial news outlets who echo the current status of currency pairs on the forex every set time interval or on new developments, automatic tweets generated by financial platforms and market brokers, and tweets by individual traders. This project looks at the above three sources of tweets and analyses the sentiment on each currency pair of the tweet to determine whether the mood for a specific currency compared to its pair is positive, negative or neutral. This sentiment information is passed to the agents in the simulation to allow them to calculate the trend in the direction of the real-world market.

2.3 Multi-Agent Simulations

“Agent-based modelling and simulation (ABMS) is a new approach to modelling systems comprised of interacting autonomous agents”[9]. These systems can range from modelling epidemics and social interactions, to modelling complex trader behaviour in financial markets. This project uses multi-agent based modelling for the latter reason to simulate the forex and traders within it. The core to multi-agent systems are the agents themselves. An agent, in the context of this project, is an instance of a set of rules, algorithms and behaviours to respond to changes of the exchange price between the currency pair being traded in the model and also to act according to sentiment information, mined from Twitter, on a given currency.

In a paper by Casti[10] he argues that agents should contain two sets of rules: a base-level set, for behaviour. And a high-level set of “rules to change the rules.”. The base-level set provides responses to the environment while the “rules to change the rules” provide adaptation. This principle by Casti is used in this projects simulation; the agents have a base-level set of rules which model basic trader behaviour such as ‘if base is rising and
quote is falling: buy’, ‘if base is falling and quote is rising: sell’ and obey rules like ‘entry orders’, ‘stop-loss orders’ and ‘profit limits’ which are all basic trading mechanics\[6\]. The high-level set of rules include more complex trader behaviour such as specific strategies, the use of running averages, double bluff and fundamental prices\[6\]. These change certain base-level rules such as going against the trend of ‘if base is rising and quote is falling: buy’ by having more complex strategies for profit. These high-level rules are a lot more agent specific, with every agent using different pre-determined high-level rules to mimic real traders who also use different strategies between them.

The idea of using multi-agent simulations to model a financial market is relatively new, with very few commercial real-world applications of the models. The approach of using social media trends as an input to a multi-agent simulation has only been attempted once\[2\] and that model used information from the news releases of one financial company. Using Twitter as an input to a multi agent simulation has, to this papers knowledge, never been attempted and this paper believes that using Twitter to take the sentiment from an un-biased majority will be a greater advantage then using biased news sources written by a small minority.

3 Approach & Implementation

The project is composed of a stand-alone Java application that completes all stages of the project: Twitter Mining, Sentiment Analysis, Multi-Agent Simulation and Visual Output of the Simulation. The Java programming language was used as it is very versatile so can handle all the stages of the project as well as being cross compatible on various operating systems, making the software available to all traders on all operating platforms that can run the Java virtual machine.

3.1 Stage 1 – Twitter Mining

For the Twitter mining stage the project uses the 3rd party Java library ‘Twitter4J’\[11\] which provides simple Twitter functionality for Java in the form of classes and functions. Due to the large scope of this project and the limited timeframe to complete it, using Twitter4J was a viable alternative to coding directly with the Twitter API. The application gathers tweets with the search query(s) specified in the main GUI. These queries work best when in the form “Euro vs. US Dollar” and “EUR/USD” as these make the query’s direct and in the form to help with the sentiment analysis stage. A date filter is performed (if enabled) on the gathered tweets, and all the tweets are then placed in a single arraylist.

The Twitter API does pose some limitations on the framework. The first being a minor one of speed; in the test runs of this project the API took ~5 seconds per query to respond. This isn’t too much of a limitation though as the models predicted by this framework are for the next 24hours of market movement. The second limitation imposed by the Twitter API is that historical tweets only go back 7 days. This in itself doesn’t pose a limitation when running the framework to predict future prices (as this would only take the past day of tweets), but when attempting to analyse the effectiveness of the
simulations over past market movements it does pose as a severe restriction to only be able to analyse the past 7 days.

### 3.2 Stage 2 – Sentiment Analysis

The arraylist of tweets is passed onto the second stage; Sentiment Analysis. This stage takes the tweets and individually analyses them for sentiment information contained within the specific currencies. Through research on the sorts of financial tweets the search queries return, it was found that the majority of useful tweets are structured in the manner: `[some text][base/quote][some text][sentiment keyword][splitter][some text][quote/base][some text][fig. 3.1]`. From this information the technique this project uses is that it splits the tweet into two at the splitter keyword (vs, against or versus) and then looks for either a base or quote keyword on the former string. If it finds either base or quote, it notes which one it finds and then splits the former string into two on that keyword. It then takes the latter split of the former string and looks for a sentiment keyword (rise, fall, up, down, high, low, etc...). If it finds a sentiment keyword, it then looks in the latter split of the original tweet for the opposite keyword to what it found in the former split (if former = base, latter looks for quote. Visa versa). If that pattern matches, it takes the sentiment keyword found, analyses if it has come from the list of positive or negative keywords and then increments an appropriate hashmap for the base or quote that corresponds to the positive or negative statement[fig. 3.1].

![CANDIDATE TWEET](image)

Dragni opinions over rate cut were divided on timing, not substance. **EUR keeps sinking vs USD, GBP, CHF.**

**OUTPUT: NEGATIVE SENTIMENT FOR EUR**

**fig. 3.1 – A visual representation of the structure of the majority of financial tweets**

Another method of sentiment analysis that was discussed with a sentiment researcher from the University of Southampton, would have been to dynamically gather sentiment by analysing both the market prices and the tweets generated between periods of time. As an example, a timeframe could be taken of 1 hour where the market price movement is in a steady decline. The same 1 hour period of tweets with the same base/quote keywords as the market pairs could be analysed and the resulting grammar and words...
used in those tweets saved. This would then be repeated multiple times for 1 hour periods where the market is either rising, falling or neutral which would eventually build up a database with patterns of words that are linked to each type of market price movement. This method of sentiment analysis, although very good in the sense that it responds continuously to dynamic market movement and it’s corresponding Twitter sentiment, would not be ideal for this framework for two reasons; the first is that the timeframe’s needed to capture periods of a steady rise or fall in the market are usually a few minutes, with a maximum of 1 hour. Unfortunately, those are not timeframes on which many tweets of the financial nature are released, which would mean that the few tweets gathered in those timeframes would not be enough to give an accurate depiction of the word patterns corresponding to certain market behaviours. The second reason why this sentiment analysis technique would not be ideal for this framework is due to the fact of not being able to separate the individual sentiment for the base and quote. This is important in this framework as it allows the traders to base their market decisions on the sentiment of the individual currencies.

So this framework will be employing the use of the initially mentioned method of sentiment analysis as it seems the best for this project as normal sentiment analysis methods, which analyse the sentiment of the tweet as a whole, would not work due to having two currencies in the tweets that this project mines, of which the sentiment of either would contradict the other causing either a neutrality on the tweets sentiment overall or a false output.

3.3 Stage 3 – Multi-Agent Simulation

The third stage of the project is the multi-agent simulation. This models the market segment of the specific currency pairs by creating a virtual market. This virtual market works in the same way as the real foreign exchange market for a currency pair. It is initialised by creating a specified number of agents which act as the traders in this virtual market and assigning each agent a trading strategy as well as providing an initial comma separated value (CSV) file containing the past 200 prices of the real world market to act as an initial guide for the traders. These traders place buy/sell limit orders according to their strategies and the current market price (traders have a user-defined variable probability of placing a trade on a trading iteration. The default property is set at 0.1). After each trading period the market takes all the submitted buy and sell orders submitted in that trading period and clears the market by calculating a market clearing price, executing all buy/sell limit orders that are below/above the new market price respectively and discarding all other orders. The market then enters into the next trading period.

The market clearing price is calculated by finding the intersection of the supply and demand curves of submitted limit orders. The algorithm used is described by Raberto et al. (2001) and is the most accurate representation of real-world market clearing which also works on supply and demand curve intersection, but on a continuous basis. The algorithm consists of separating the buy and sell limit orders with their associated quantity amount, summing each type of order and the amounts up to the highest/lowest supply/demand order price submitted. Then progressively subtracting the amount of next largest/lowest associated supply/demand price order off the current supply/demand sum. This, when plotted on a graph, forms two curves; A supply and a demand curve, both of which start the opposite ends of the graph and gradually converge.
on each other. It is where these two curves intersect that the new market clearing price is calculated, as it is where the supply meets the demand[fig. 3.3]. This approach can be demonstrated mathematically by the following equations for each curve where \( u \) are the number of buy orders issued and \( v \) are the number of sell orders issued at time \( t \). \( a \) is the quantity amount of each order of associated limit price \( b_u \) \( v \) [fig. 3.2]:

\[
\text{(Demand Curve)} \quad f_t(p) = \sum_{u \mid b_u(t) \geq p} a^b_u(t)
\]

\[
\text{(Supply Curve)} \quad g_t(p) = \sum_{v \mid s_v(t) \leq p} a^s_v(t)
\]

fig. 3.2 – Equations for the supply and demand curves of the market at each time step

One must note that in order for the price to clear the supply/demand curves must intersect which requires a wide spread of both buy and sell orders that cross each other, as well as a proportionate quantity order amount on both curves. For this reason there is probability that the supply and demand curves might not intersect and the market might not clear. This probability is reduced in an inversely exponential manner with the amount of agents running in the simulation.

fig. 3.3 – An example of a supply/demand graph with a cleared price
The simulation used for this project’s application contains 4 different types of traders with specified trading rules. The 4 trader types were chosen as they have been demonstrated to work well in modelling an accurate multi-agent market by several research projects\(^[13][14][15]\). The 4 trader types stem from a single trader parent class which offers the functions in common with every trader (making orders, closing orders, as well as holding values such as historical windows for separate traders). The 4 types of traders work on the following principles:

3.3.1 Random Traders

Random traders attempt a trade on each trading iteration by taking the last market price, calculating a price variance by taking a random number between 0 and 10 and dividing it by 100,000 to give a number in the 5th decimal place range which is equivalent to one decimal place under a pip (percentage in point)\(^[6]\). By making this variance 1 point under a pip ensures that the random traders don’t make the market too volatile beyond the limits expected of the real market but still adds slight random volatility to the market. This price variance is then added or subtracted from the latest market price and a buy or sell order is issued respectively at the new price with a random probability of either buying or selling of 0.5.

The amount to trade is a random amount between 1 and 100. This gives a good variance between amounts that traders trade and in the small scale of this framework’s virtual market it equates to amounts that influential traders trade in the real market (small scale traders in the real world markets don’t trade enough quantity to make any difference to the market clearing price, so can be neglected here).

3.3.2 Momentum Traders

Momentum traders attempt to follow the trend of the market to make profit as their principal belief is that if a market price is rising/falling it will continue to rise/fall\(^[6][13][15]\). The traders calculate a trend of the recent market prices by taking the current price and taking it away from the price a set historical distance away \(n\) in a moving historical window then dividing the result by the number of prices (steps) in the historical window\(^[fig. 3.5]\).
\[
\text{trend} = \frac{p_t - p^{(t-n)}}{n-1}
\]

fig. 3.5 – The trend equation for momentum traders

This trend corresponds to the rise/fall of the price from the start of the historical window to the end (positive trend relates to rising and negative relates to falling) and also corresponds to the volatility of the price in relation to the historical window size. So a large change over a short window would produce a larger trend hence would be considered more volatile than a small change over a large window.

If the trader has open transactions of either buy or sell, and the moving trend changes direction to the type of transaction order placed, the trader immediately closes the transaction.

The buy/sell price for a trade is calculated by multiplying the trend with the size of the historical window and then adding that to the current market price. If the trend is greater than 0 a buy order is executed with the calculated price. If the trend is less than 0 a sell order is executed instead. This way the momentum traders are constantly following the trend of the market as relayed to them by their historical window.

As with random traders, amount to trade is a random amount between 1 and 100.

3.3.3 Inverse Momentum Traders

These are identical to momentum traders, apart from they take the trend and execute a market order against it rather than with it. These traders work on the belief principle that a rising/falling price will shortly stop rising/falling and start falling/rising respectively\[15\].

3.3.4 Fundamentalist Traders

These traders work on the belief that the market price will ultimately settle to a fundamental price, so attempt to trade towards this fundamental price\[15][13]\.

The fundamental price is calculated for each trader by taking the mean average price across a moving historical window[fig. 3.6]. If the current market price is higher/lower than the fundamental price the traders sell/buy respectively. Once a trader has an open transaction and the market price gets to equal or drops below (if an open sell transaction)/rises above (if an open buy transaction) the transaction price, the trader closes the transaction.

\[
p^{\text{fundamental}} = \frac{\sum_{i=1}^{h} p_i}{h}
\]

fig. 3.6 – The fundamental price equation

As with the other types of traders, the amount to trade is a random amount between 1 and 100.
The different trader types are by default allocated dynamically into the market on each trading iteration, as this models the constantly changing traders and strategies used on the real market and was found to give the best results in market correlation compared to having a set ratio of trader types throughout the simulation.

If Twitter influence is enabled for the simulation, then each trader’s limit order is influenced by the sentiment gathered from the Twitter mining. This influence comes in two parts;

- The Effect - The strength of the overall sentiment which is calculated by the amount of a certain type of sentiment for a specific currency in the pair [fig. 3.7].
- The Influence – A user set percentage to indicate how much the effect should affect each limit order.

\[
\text{var} = \frac{\left( \frac{S_{b+} + S_{q-}}{S_{b+} + S_{q+}} \right) - \left( \frac{S_{b-} + S_{q+}}{S_{b+} + S_{q-} + S_{b-} + S_{q+}} \right)}{100}
\]

fig. 3.7 – The effect calculation. \( S \) refers to the number of tweets of a particular sentiment and \( b^{+/\text{-}} \) & \( q^{+/\text{-}} \) represent base positive/negative & quote positive/negative sentiments respectively.

The Twitter overall influence on a limit order is the effect multiplied by the influence which is then subtracted from/added to a buy/sell limit order respectively.

4 User Interface Design

The design of the application was a key aspect of this project as it is more than a pure research project and is intended for real world distribution. The design had to withstand real-world application and use whilst making it simple, intuitive and practical to use for the intended clientele (both professional and new traders). It was decided that to make the overview as simple and versatile as possible, the application would be a series of windows [fig. 4.1], with the main application window being the centre point for the application's control. The other windows would allow various forms of viewing and controlling information generated by the application and were designed to be fully resizable so monitor resolution wouldn't be an issue in use. The main application window was made to be 1024x600 pixels in size to allow it to fit onto a portable netbook computer, increasing the applications versatility to be used on the go.
The colour scheme picked for the application is orange, white and dark grey which is also an important factor to allow the application to be both visually pleasing to the user as well as providing a continuous theme which allows the application to look professional. As well as that it has the added effect of providing a clear contrast for users with sight impairments. The layouts and interactive elements of the interface adhere to good practice and clean design standards and make use of natural mappings to provide a more flowing and natural interface feel[16].

The separate windows of the application are described as follows:

- **The main application window**: This is the core of the application, it has been split into 2 main panels; top and bottom with the top being the main control section and bottom being an output graph. The top control panel is further split into 3 sections that represent the 3 main stages of the simulation:
  - 1. The Twitter section - This allows the user to enter relevant search queries to mine Twitter and also allows those resulting tweets to be filtered down to a specific date range.
  - 2. The sentiment analysis section - This section shows the information gathered from the sentiment analysis stage of the simulation. It also allows the user to specify multiple base and quote keywords to filter the sentiment from.
  - 3. The multi-agent section - This allows the user to set the initial and key variables for the actual simulation part; number of agents, time to run, average number of runs and the initial CSV file of past historical prices to initialise the simulation (a link to a financial website that provides the relevant CSV files is provided, as well as a pop-up help window which specifies how CSV files should be constructed for users wishing to provide their own CSV files).

The bottom panel shows a standard Price vs. Time line graph as well as the current price (for each time step in the simulation) allowing the output to be quickly assessed.

- **The graph display windows**: These windows can be opened from the main application and show the currently simulated Time vs. Price graphs (if the simulation is running, these graphs update in real-time). The windows are fully resizable and allow the user to zoom in and out of the graphs for more detailed analysis of the simulation. The application currently provides two types of graph
to be displayed: A standard line graph and an area graph. This caters for the various ways that users prefer to visualise graphs. The graphs themselves are generated by the 3rd party library JFreeChart[17].

- The sentiment pie chart window[apx. 3] - As with the graph windows this is fully resizable and displays a colour coded pie-chart of the sentiment information gathered from the sentiment analysis part of the simulation. This provides an easy to analyse visual method of viewing the specific sentiment information compared to reading the individual sentiment figures.

- The tweets window[apx. 4] - This is another resizable window allowing the raw tweets gathered in the Twitter mining section to be viewed.

- The advanced settings window[apx. 5] - This window provides access to some more advanced settings for the multi-agent simulation. This allows to user to tweak and experiment with their own methods of running the simulation as it provides access to most of the variables used in the simulation:
  - Agent distribution ratios (as well as an option for the use of continuous dynamic allocation)
  - Agent trade probability per iteration
  - Number of allowed open trades per agent
  - The probability of Twitter sentiment influence on an agents trade per iteration
  - The strength of the Twitter sentiment influence on a trade

5 Testing

Testing the framework and system as a whole was a critical stage to this project as the system is aimed for the real-word trading professionals so needs to be robust against everyday use by a variety of clients. The testing process of the framework can be split into two sections; unit and integration testing during development and complete system testing after completion of the framework.

5.1 Unit & Integration Testing

To be able to proceed with the building of the project with as few problems as possible, each stage class and method of the framework was tested individually using custom build test cases to check it was functioning correctly before proceeding. After each number of classes were joined together their integrity was again tested to check that they interacted and functioned together correctly. These test stages were performed up on each class join up until the whole system was joined together. The main tests carried out when joining together classes consisted of the following:

- Confirm that the former class is calling the latter at the correct time in the program run.
• Validate that each method and function can be called and works as it should and if expecting a return value, that the correct value is returned depending on the input. These were tested with dummy data to simulate every data type that might be inputted into the application.
• Verify the correct throughput flow of the data was happening when the classes were joined together (so that data would go as it should into the first class, which would then use the data in its methods, create an instance (or instances) of the second class, input and retrieve the data from those instances, before finally passing it off to the rest of the program).

Below is a hierarchical class diagram of the system which shows how the classes of the framework are joined together [fig. 5.1].

![Hierarchical Class Diagram of System](image)

**fig. 5.1 - Hierarchical class diagram of overall system**

The unit and integration testing wasn’t without its problems, with several functions and classes failing initial tests and then classes failing further testing when integrated with other classes. These problems were relatively minor and only took a maximum of 5 hours per problem to fix. Had unit and integration testing not been performed on these components then these problems would have evolved and escalated into something much more serious and complex to fix, putting the project behind schedule.

### 5.2 Complete System Testing

A complete system test is used to test the robustness and integrity of the system once it is fully completed to prepare it for use in the real world. For this, a series of test cases were constructed [fig. 5.2] and acted out on the running system. Each test case had an expected result which was compared to the actual result of the test case and a note was made whether the test case passed or failed and if it failed the system would be fixed until that test case passed and a note was made on the change required for the fix [fig. 5.2].

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Pass/Fail?</th>
<th>Fix</th>
</tr>
</thead>
<tbody>
<tr>
<td>When no search phrases are entered, give the user a warning and stop simulation.</td>
<td>Pass</td>
<td>-</td>
</tr>
<tr>
<td>If no Base or Quote keywords are specified, warn the user and stop simulation.</td>
<td>Pass</td>
<td>-</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>If no CSV file is specified, warn the user and stop simulation.</td>
<td>Fail</td>
<td>User experienced an error message, but the simulation was attempting to run regardless and critically failing. Fixed by adding market abort code in the catch of the CSV method exception.</td>
</tr>
<tr>
<td>Whilst simulation is running, no variable boxes in the main window should be changeable.</td>
<td>Pass</td>
<td>-</td>
</tr>
<tr>
<td>The graph buttons at the bottom of the main window should respectively; clear all graphs, display the line graph window, display the area graph window and display the sentiment analysis window.</td>
<td>Fail</td>
<td>The clear button was clearing the main windows graph but not graphs in separate window. Fixed by adapting the clear graphs method so it extended to any open graph windows.</td>
</tr>
<tr>
<td>All user set variables can be set and changed whilst simulation is not running and successfully get implemented in the simulation once started.</td>
<td>Pass</td>
<td>-</td>
</tr>
<tr>
<td>The “Download CSV” opens up the users native web browser and navigates to the pre-defined website.</td>
<td>Pass</td>
<td>-</td>
</tr>
<tr>
<td>Multiple runs of a separate simulation are ‘overlaid’ on top of the previous simulations graph if it has not been cleared.</td>
<td>Pass</td>
<td>-</td>
</tr>
<tr>
<td>The simulation should not run if invalid user data is entered into the textboxes (i.e. non-numeric data where numeric is expected)</td>
<td>Fail</td>
<td>Fixed by creating a parse numeric input method which gets run before the main simulation starts, this parses the user input and if any validation fails, the simulation does not run and an error is outputted.</td>
</tr>
</tbody>
</table>

**fig. 5.2 – A list of main test cases that the system needs to be able to perform to be considered robust enough for real world use.**

These tests where constructed to make the system user friendly in a manner that means the system never results in terminating with a fatal error, and all errors are displayed by a user friendly pop up box informing the user of the error in a descriptive manner.

As part of the complete system testing, a user test was also carried to makes sure the system could be intuitively operated by 3rd parties. In ideal circumstances the users
performing the tests would have been professional foreign exchange traders. Unfortunately with the limitation of time and professional traders being hard to involve in the tests as their time constraints and office hours meant non were available to evaluate the system, this system was only tested on a couple of people from non-financial backgrounds. Both of the test subjects were explained the use of this system and the used terminology prior to the test. Both subjects were then asked to perform a full range of available tasks on the framework (run a simulation with & without twitter input, filtered from a range of dates and change the settings of variables before re-running simulations). The subjects were asked to specify if they had any problems performing any of these tasks to which both replied that they did not. This has satisfied a very simple 3rd party test to show users are able to use the system intuitively. Before being deployed into the wider world, this system would require a much more extensive user test stage with test subjects being the target audience of foreign exchange traders who could evaluate the system as needed for its intended use.

6 Management

Project management has been critical for the organization and handling of the project build and research, as without a properly structured time management plan the project would have inevitably fallen into difficulties due to the large amount of work to do on the various project sections.

To manage the time for the project’s development a Gantt chart was made [fig. 6.1] at the very beginning of the project’s development and the development of the project was split into clearly defined sections which were to act as milestones. Time for each milestone was estimated with a conservative approximation to give a buffer zone for any unforeseen hold-ups in the development stage. These buffer zone’s proved to be a good decision as, at times, they were required due to unforeseen circumstances in the delay of the development. After each stage on the Gantt chart was completed, the project progress and work to do was re-evaluated and checked against the Gantt chart to make sure enough time was left for the project completion.

As well as the use of the Gantt chart for time management and planning, weekly supervisor meetings were scheduled to provide a third party view into the project development and timekeeping as well as provide valuable input into the project’s current state and expansion(s) where necessary.
6.1 Risk Assessment

Completing a risk assessment was a key part of the initial project management and working out milestones on the Gantt chart, as risks can happen during any stage of the project so it is necessary to plan for them and to have response actions planned to minimise the effects of any risks on the overall project.

Below is the table of the identified risks in the project and their possible elimination/reduction solutions [fig. 6.2]. These risks covered all scenarios that could have been encountered in the project, and no further risks were identified during the development of the project.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Severity</th>
<th>Likelihood</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not being able to work on project due to unexpected illness/injury.</td>
<td>Low – medium</td>
<td>low - medium</td>
<td>Depending on severity of illness/injury work could be delayed from several days to several weeks. For minor illness/injury good time planning will allow for a buffer zone to be implemented in which any recently missed work can be caught up. For a more serious illness/injury, an extension on the project deadline may be sought.</td>
</tr>
<tr>
<td>Loss of project documents and source files work due to computer failure.</td>
<td>High</td>
<td>low</td>
<td>Use the online backup system Dropbox to store all project files immediately as they are created or modified. This also synchronises the files on several linked computers, so chance of hardware failure on Dropbox and two computers at the same time is extremely minimal.</td>
</tr>
<tr>
<td>Project getting stuck on a specific stage and no progress can be made for some time.</td>
<td>Low</td>
<td>medium</td>
<td>If the project starts to lag behind and gets too far off the project Gantt chart, talk to supervisor to seek help in progressing (maybe look at alternative ways of tackling the problem).</td>
</tr>
<tr>
<td>Simulation showing no correspondence to real world market trend.</td>
<td>Low</td>
<td>medium – high</td>
<td>Analyse and compare simulation output to the forex to spot of there are any kinds of similarities. Attempt to change the variables of the simulation to match the real-world market more closely.</td>
</tr>
</tbody>
</table>
7 Results

This section analyses the output of the framework’s simulation compared to the real world price graph for the simulated period. The results obtained from the simulation under different variables are crucial, as they dictate the accuracy of using an agent based simulation approach with Twitter input and prove that this framework and approach are a viable way to predict the trend of the foreign exchange market. They also allow users of the software to see to what extent the models in this simulation are accurate hence allowing the users to make informed trading decisions based on the obtained simulation.

In real-world tests made with the framework, this paper has found some very surprising connections between real-world and simulated trends on the forex. The relations between the simulated and the real world market was analysed by overlaying the line-graph generated by the simulation for a timeframe with a line of the real-world price vs. time graph. The real-world graph data used was from the website XE.com which is the world’s most used online currency site, and was chosen as it provided a free scalable graph of the current and historical prices of currency pairs which could easily be taken and overlaid over this framework’s simulation graph.

The tests performed on the simulation have consisted mainly of 24hour simulated runs. This translates to 1440 iterations on the simulation using an initial CSV file of 1 minute intervals (hence each time iteration simulating a 1 minute interval). The 24hour simulation was chosen as the main simulation type of this framework as it provides sufficient enough granularity to be able to perform a multi-agent simulation which corresponds to the fast changing supply and demand level of the market, but also is long enough to be able to use Twitter input as Twitter sentiment is not instantaneous and one wants to be able to gather the past 24hours of sentiment to be able to influence the next 24 hours.

To keep the various simulation runs even, all simulations were run with data in the initial CSV file ending at midnight hence making the simulation timeframe last 24 hours (1440 minutes) from midnight to midnight. This also allowed the tweets to be gathered for the active 24 hours before the simulated run, giving an overall sentiment of the previous day which influenced the agents trading the day after.

To finally smooth over any anomalies in individual simulations, the outputs tested here were run with 10-average runs. Setting this on the framework runs the simulation 10
times from start to finish, then takes the mean average price for each time iteration from all of the runs and only plots the average price vs. time graph of the 10 runs. This made sure that the simulation output was a true representation of the multi-agent simulation and had not been affected by any randomly occurring irregularities. A price average of 10 runs was chosen as it was found during testing that 10 runs or more produced almost identical outputs as within 10 runs any major anomalies are eliminated.

7.1 Analysis of simulations vs. real-world data

Normally run simulations where performed for the 3rd, 4th, 5th, 9th, 10th and 25th of April as this was a scheduled testing time for the project (the 25th was 2 weeks after the last test to prove the other tests done in the one week period were not time period dependant on that week). For those runs the overlays show a surprising pattern correlation for the simulated graph vs. the real world graph. This is particularly noticeable in the simulated run from April 9th – April 10th where both the simulation and the real-world graphs show the same spike rise in the same 3 hours.

![Simulation prediction vs Real-world overlay for April 9th - 10th](image)

This spike also has surrounding trends that match the real-world graph (initial dip before the spike, and a dip after the top of the spike) which leads to believe that these are agent/trader specific behaviours that occur when certain market patterns are demonstrated. This would also correspond to the trading strategies and theories mentioned in the Essentials of Forex Trading which talks about breakout trading in chapter 5 and describes the patterns seen in the simulations. The actual corresponding spike rise in the simulation corresponds to the Twitter sentiment. As can be seen from the sentiment pie chart in the above simulation, there is an overwhelming amount of positive sentiment towards the Euro which would correspond to the Euro becoming stronger on the dollar and hence the exchange rate rising (as 1 Euro would buy more of the dollar). This is indeed demonstrated by the spike rise, but as to why the rise happens
as suddenly as it does and in the timeframe that it does is something that can only be explained by the complex agent behaviour interactions that happen within the simulation. One thing that should be noted about the simulation runs is that although the output graph shows the same patterns as the real world market, the range of the simulation output is usually not consistent with the range of the real-world graph. This is most probably due to the fact that the real-world market has a much more diverse range of traders with a much higher difference in order quantities which result in a more granulated graph and much tighter market prices. This has been attempted to be simulated in this framework, but it was quickly found that it was very hard to do as both the real market statistics were unavailable to be analysed to base variables on, the real-market most likely has many more complex traders which cannot all be modelled in the time constraints of this project, and it has also been found that increasing the agents on this framework’s simulation beyond a certain level started causing a pattern in behaviour of each simulation that didn’t in any way correspond to the real world market (this is detailed more in section 7.2). For this reason the overlay of the real-world graphs to the simulation outputs as demonstrated in this paper have been scaled to cover the simulation graph range, hence making the overlay accurate to patterns but disregarding price range. So this framework is indicative of the market pattern that will occur but not necessarily of the market prices.

Another simulation run with a very strong correlation of the real world market is the simulation from April 3rd – April 4th [fig. 7.2]. This simulation run predicted a sharp downward spike of the market 800 minutes from midnight (simulating a drop at an approximated 1:30pm GMT). The actual market drop happened at 1100 minutes from midnight (equating to approximately 6:30pm GMT) but although the actual drop was later than predicted, it displayed the exact patterns of the simulated drop (in fact this was one of the few predictions where the simulated price range perfectly matched up to the real world, so on the output [fig. 7.2] the two lines are in the same price range).

![fig. 7.2 - Simulation prediction](image-url)

fig. 7.2 - Simulation prediction [Orange Line] for April 3rd – April 4th. Yellow Line is the Real-World overlay.
As was to be expected though not all simulation runs showed as much direct correlation with the market as the ones mentioned above. Simulation runs for April 5th and 6th [apx. 7,9] were slightly less accurate and in some cases failed to show any substantial patterns. Although, compared to simulations run on the same timeframe with the Twitter input turned off (effectively making it a pure multi-agent simulation with the only external influence being the initial CSV file) the simulations with Twitter input turned on still trended much more towards the way the real-world market trended. This seems to point to the evidence that Twitter sentimentality of the specific currency pairs for the previous day does indeed have a profound effect on the market movement the following day.

As can be seen on one of the tests between Twitter input turned on [fig. 7.3] and Twitter input turned off [fig. 7.4] the simulation run with Twitter sentimental influence turned on mimics the real world market line much more closely than the simulation run with Twitter influence turned off which is a lot more volatile in its price movements and doesn’t represent the real world line anywhere near as accurately.

![fig. 7.3 – Simulation run for April 20th – April 21st (with Twitter influence ON)](image-url)
A secondary series of tests were run on the market simulations where the prediction timeframe was 24 hours from times other than midnight – midnight. These runs proved to be a lot less accurate than the former runs, with many not following the market trend at all. This is speculated to be due to one of two factors;

1. The fact that not enough tweets about the market currencies had been posted at the time of the simulation run to be effective in giving a solid sentiment of the currency pairs.
2. The tweets gathered were for the previous day and, as the simulation was then started a significant way into the next trading day, so the tweets from the previous day were already redundant at the point from which the simulation was started.

So from the simulation runs performed the optimal timeframe to run a prediction from was 0000 GMT – 0000 GMT (midnight to midnight).

### 7.2 Simulation Patterns & Behaviours

During the testing of the simulation runs, it was found that the simulations displayed certain behaviour trends during the runs and when presented with different variables.

In most simulation runs one can see that the market run has, at times, significant upwards and downwards spikes and crashes that occur very quickly and in most cases recover within the next iteration. These market spikes and crashes don’t happen at the minute time level in the real world market, but recent research has shown that in the real market under the 950ms level, which is a level only used by computer traders as it’s faster than any human trading, instantaneous market spikes and crashes are very common! These financial ‘black swans’ as they have been named have been found to have occurred more than 18,520 times over the past few years in this sub-human market level. In light of this research, it is thought that due to this framework consisting of automated traders entirely, that these phenomenon occur in normal trading of the simulations, hence the results on the simulation outputs. The majority of these spikes on the output runs can then be ignored; as they happen quickly enough and recover so do not affect the overall trend of the simulation.
It was also found that there is an optimal range of the number of agents for this simulation to correspond to the real-world trend, contrary to the idea that the more agents the more accurate the market behaviour. The simulations run for the optimal results, as discussed in this paper, were all run with 1,000 traders. The simulations were tested with more and less traders and it was found that if more than 1,500 traders were in a simulation run, the behaviour started to resemble the real-world market trend less and less and instead the behaviour always turned into a stair-step type upwards trend[fig. 7.3]. This trend is thought to be due to the market being too overfilled with traders whose behaviours start to synchronise with each other at high numbers, hence creating a stair-step effect of trading. On the opposite scale, having fewer than 1,000 traders started causing rapid and severe volatility on the simulation and with less than 300 traders on a run, the probability that the market would not be able to clear on an iteration due to the supply and demand lines not intersecting got significant enough to cause the market to not clear on most simulation runs.

![fig. 7.3 – The stair step effect pattern experienced when the model uses too many traders](image)

Finally the framework allows for the user to turn off dynamic agent allocation (which makes a random distribution of trader types on each iteration) and choose a ratio of fixed agent types for the whole simulation. This feature is purely for experimenting with trader strategies as it was found that there is no optimal agent ratio. Each ratio produced different results, each of which displayed reoccurring market patterns which were due to the agents collating together to display patterns in trading behaviour which didn’t correlate with the real-world market in any way.
8 Conclusion

The aim of this framework was to research into the effectiveness of using an agent-based simulation to model the short term trends in the foreign exchange market and using Twitter sentimentality as an influence into the simulation. This was achieved by developing a full simulation framework in Java which has been designed and developed to be distributed and used as a practical application for foreign exchange traders as a tool to increase confidence in the trend and price movement of a currency pair.

The developed framework has shown some interesting and in some cases surprisingly accurate correlations between the simulated outputs and the real world market for a currency pair over a specific timeframe. These correlations are not always 100% accurate (especially when the Twitter sentimentality is verging on neutral for a currency pair) but in most of the simulations run at the optimal time of 0000GMT for the timeframe between 0000GMT and 24hrs+0000GMT with the tweets from the previous day, have produced output graphs which almost always have shown a trending price movement or trend pattern that has then occurred as predicted in the correct direction. This leads this paper to believe that Twitter sentimentality not only reflects the current mood of a currency, but that this mood leads on to influence the price movement in the next trading day. This price movement can then be modelled using an agent-based simulation with an initial CSV file of past prices up to the last price one would want to start the model from. The agent-based model developed by this framework seems to portray the market of a currency pair to a good degree by modelling the 4 main types of trader behaviour in a market. There have been cases where the simulated outputs have produced some market pattern which has then not been seen in the real market. This of course can be expected though because as long as the real world market has a human element of trading within it, it will also have human logic and reasoning behind it, which no simulation could model.

Although the simulations of the tests run so far have shown good potential, prospective users should note that this framework would require months of much more extensive testing before trusting its simulation outputs. Users should also note that this paper has discussed the simulations and models produced by the framework almost only with the Euro vs. US Dollar currency pair. No extensive tests have been carried out thus far on any other currency pairs and the correlation between the simulated output on them (apart from one minor test on GBP/USD[^12] to determine the framework works on other currency pairs. This test proved to work, and with a good correlation, but cannot be taken as conclusive evidence without further tests).

This framework and the research within it was made in very strict time constraints, so might not have as extensive functionality as an experienced trader might be used to in a financial platform. As a future development some changes/additions to the framework could be implemented. These additions include:

- Fetching and loading the historical data automatically, rather than directing users to download a historical price CSV file.
- Allowing for a more diverse range of visual outputs (possibilities include candlestick graphs, Bollinger bands, a volume chart and a moving averages graph)
- The possibility to analyse several currency pairs concurrently
• The possibility to expand away from the foreign exchange market and simulate the market trends of other assets in various markets.

What this project has created then, is what it envisages as a bespoke tool in a foreign exchange traders toolbox. This tool, although bespoke in its modelling methods to the foreign exchange market will be distributed as open source to allow it to be used across all types of traders, from professional, large hedge fund traders with an extensive arsenal of bespoke tools at their disposal, to first time, small-scale traders who cannot afford to purchase or fund any expensive trading prediction algorithms. This framework and the research it represents then hopes to be valuable to professional traders as well as to open the world of foreign exchange trading up to new traders.

Although this framework is aimed solely at the foreign exchange market, the research developed in this project could have the potential to be expanded onto other markets and commodities. Alongside this, other social data sources (e.g. online news and media and possibly Facebook statuses) could potentially be used to aid social sentiment even more for a more socially weighted prediction influence. These are all possible future expansions of the base framework developed in this project.

Finally one must note that this project believes there is no such thing as a perfect market prediction algorithm or strategy, as the discovery of being able to model markets to a good degree would counter act the use of them consequently collapsing them, much in the same way that the Black-Scholes formula (developed in the 1970s) has recently been linked with financial crashes\cite{19}. The framework developed here might help with some prediction of price movement, but overuse and widespread use of these methods by the great majority of traders would ultimately render this method of prediction unusable.
References


Appendix

apx. 1 – The main application control window

apx. 2 – The graph display window with an area graph displaying
apx. 3 – The sentiment pie chart window

apx. 4 – The tweets window
apx. 5 – The advanced settings window (with default settings)

apx. 6 – Test run for April 4th – April 5th
apx. 7 – Test run for April 5th – April 6th

apx. 8 – Test run for April 10th – April 11th
apx. 9 – Test run for April 6th – April 7th

apx. 10 – Test run for April 18th – April 19th
apx. 11 – Test run for April 25th – April 26th

apx. 12 – Test run for GBP vs. USD for April 25th – April 26th
FiMASS

Financial Multi-Agent Simulation Software

SIMULATING THE FOREIGN EXCHANGE MARKET USING AGENT BASED SIMULATION METHODS WITH INFLUENCE DRAWN FROM TWITTER SENTIMENTALITY AS WELL AS HISTORICAL ANALYSIS

* THIS FRAMEWORK IS DISTRIBUTED AS ONGOING RESEARCH AND USERS SHOULD FULLY UNDERSTAND THE RISKS OF FOREIGN EXCHANGE TRADING AND THAT BY USING THIS FRAMEWORK TO BASE THEIR TRADING DECISIONS THEY DO SO AT THEIR OWN RISK

apx. 13 – An advertisement describing and showing the framework, as might be used to promote it to the target audience
Project Brief

The problem; many small-term traders who wish to make investments on the financial markets do not have the money to buy expensive prediction algorithms or the relevant experience to make efficient, consistent predictions of future share prices.

The aim of this project will be to develop an application that predicts the rise/fall of the market share price using mass social information gathered by data-mining social media, mainly focusing on the service ‘Twitter’. The application will gather information on a certain market asset by looking for their names and certain key words within Twitter.

For the scope of this project I will specifically focus on the FOREX Market and the trading of the US Dollar vs. the Euro. I am using this as it's the largest trading on the FOREX market with lots of social coverage.

I will start by mining the data from Twitter. Once the data is mined and returned to the application, the application will interpret the stories and articles with keywords such as ‘high’, ‘low’, ‘rise’, ‘fall’ etc. that will determine whether the story is good or bad. This data will be classed into several groups of which each will have an overall weighting depending on how many similar news stories make up the group and how ‘positive’ or ‘negative’ the news is. The weightings will be fed into a multi-agent simulation of a financial market. I am making a multi-agent simulation step, rather than just predicting trends through the gathered data as multi-agent simulation will allow me to model behaviors which will happen in a real world market and so with that give more accurate predictions and also more long-term prediction trends. The agents in the simulation will act as traders in real markets which will be governed by simple rules such as; buy/sell more when there is a lot of ‘positive' news from the input data, and keep hold of shares/try to sell quickly when the news is 'negative'. The agents will also implement rules such as each one having a certain stop-loss order and each one ‘following the trend’ to a different extent. As well as being governed by these rules, I will introduce slight random elements into my agents to model a non-perfect system that is a lot closer to real-life trading, and also to help prevent agents getting locked into loops which might have been a high probability if they were to be all identical.

The results of my simulation will be outputted in a graphical format mapping the simulated ask price against time. This graph will then be compared by to the real-world market share price over time and prediction successfulness will be calculated.

What I hope to see is my simulation predicting the real-world market price quite successfully at $t_0$ but then as time progresses the successful prediction probability decreasing. Depending on the successfulness of my model my successful prediction probability will either be decreasing exponentially (not very successful long-term prediction) or linearly (reasonably good longer-term prediction).

The main project scope, in short, will be to:

1. Data-mine Twitter for stories about the market price and performance of the Dollar vs. the Euro.
2. Analyse the information and separate the 'positive' and 'negative' stories into weighted groups.
3. Make a multi-agent simulation of stock market trading and use the weighted groups to influence the trading styles of the agents.
4. Produce a time vs. ask price graph of the simulation and compare it to the real-world time vs. ask price graphs to produce a time vs. application success rate percentage graph.

If time allows, my project scope will be extended to:

- Use market assets other than the dollar vs. euro (i.e. other currencies, companies etc.)
- Use social/non-social feeds other than Twitter (i.e. BBC News, Financial Websites)
- Improve the grouping and weighting of the information groups by weighing sources by their 'trustworthiness' which will be determined by past performance and successfulness of the source.
- Improve the modeling of the multi-agent simulation by using feedback, so the agents will be aware of past performance and successfulness.

apx. 14 – The original project brief (dated October 13th 2011)