Correlation between public sentiment and financial market fluctuations

CHUKWUKA AJEH
SWANSEA UNIVERSITY
991129

2020
OCTOBER

Abstract

For the past few decades, financial institutions have increasingly begun to make use of algorithms and mathematical models to automate trading in the stock market. With the recent rise in popularity of research in Deep Learning and Big Data, along with increased processing speed, financial firms are now capable developing highly sophisticated systems, that can learn market trends in real-time using a diverse array of sources. It has not been without difficulty, however. Stock markets are notoriously noisy due to their volatility and, any sudden news can break any naturally occurring trend. Nevertheless, with a proven ability to take in a vast amount of data, learn from it and then apply these learned techniques to new situations, using machine learning to find correlations and predict future price movements will only continue to increase.

2020

Initial Document

PROJECT TITLE: CORRELATION BETWEEN PUBLIC SENTIMENT AND MARKET FLUCTUATIONS

CHUKS AJEH

Contents

1 Introduction ... 3

1.1 Motivation ... 4

1.2 Project Aims ... 5

2 Background Research ... 6

2.1 A brief Overview of Financial Markets ... 6

2.2 An introduction to Natural Language Processing (NLP) ... 7

2.2.1 Overview ... 7

2.2.2 discussion into implementations of NLP approaches and conclusion ... 9

2.3 Case Study: Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network ... 13

3. Project Management: ... 16

3.1 Methodology & Work Schedule ... 16

3.2 Risk Analysis ... 17

3.3 Project Timeline ... 19

4 Conclusion ... 20

5 Glossary ... 21

Bibliography ... 22

1 Introduction

Technology has fundamentally changed and re-shaped society. All industries in some form have changed to accommodate some form of automation, digitisation or made use of the processing power computers to relay information and perform complex computations. The adoption of technology has been notably faster in the Financial Industry; specifically, in financial markets (for example, trading). Before this, during the 19th and early 20th-century financial market trading was primarily done over the phone, or in person. As computers became more commercially available, finance companies adopted them and, the automation of markets began. Since then, we have increasingly gone from active human traders to high frequency and automated trading.

Companies, such as UBS, are creating and using algorithms using Big Data and Machine Learning (BD&ML) techniques & Artificial Intelligence (AI). These algorithms are taking in vast amounts of data from many factors all over the world, before then making millions of price action decisions, in milliseconds. Due to the vast amounts of data being collected and used by these algorithms, market flash crashes have become more common. One key factor that these algorithms make use of is trader & public sentiment. How well do these algorithms analyse and respond to such an ambiguous and complex type of data?

What is the correlation between the public sentiment of financial markets and market fluctuations? Given the wealth of publicly accessible sentiment data, I want to understand how public sentiment from various sources affects the financial market; how the markets fluctuate in the face of such complex data. Where given an event, I plan to look for clear indicators that the public sentiment is the driving force for a market flash crash or plunging the Dow Jones 500 points. Alternatively, on the flip side where news such as the covid-19 stimulus cheque in the US sent markets higher (BBC News, 2020).

That is what this project will cover; I will construct a Natural Language Processing (NLP) model that will capture and analyse public sentiment data. Using the results of this, I will then correlate it to historical stock market price movements. This model will be developed using the machine learning framework TensorFlow and API Keras and will be implemented in Python. Both Keras and TensorFlow are open-source resources used in the development of neural networks. I will be teaching a Neural Network on how to embed and understand the public sentiment data. Should the embedding be successful, I will move on to mapping the relevant data points to match specific points in the historical market data. To finish, I will visualise this mapping as evidence of the accuracy. To extend the project further, I will look at its use within stock market price prediction.

1.1 Motivation

There is already precedent for investigating said correlations. Several related works have investigated the relationship between public sentiment and market fluctuations. Frequently highlighted points of motivation are improving the accuracy of predicting stock price movements (Xin Du, 2020). The immediate benefits to a solution for this are, of course, higher accuracy for predicting stock price movements. However, by finding new ways to apply these deep learning techniques for NLP, we can extract the new techniques and apply them elsewhere. The idea of broader application leads on to the next point of motivation; enabling companies to make informed decisions. My hope is, whilst implementing my model; I will gain a deeper understanding as to how this field of research is being conducted all the while, producing clean, professional code that follows Software Engineering principles.

Governments and central banks frequently release press releases and statements about governmental and economic policies. Central banks can release statements that can be interpreted as 'dovish' or hawkish. By making use of NLP techniques companies can analyse the sentiment of the newly released statement and based on related historical data correlating to either a historical price drop or price rise, the companies and firms can then make informed decisions about certain investments in economic areas or other companies.

Another point of motivation is the adoption of NLP in the broader industry. As I mentioned above, companies are making use of automated trading through complex algorithms geared towards trading. Improved accuracy and a higher proven correlation will improve how these algorithms respond to a quick turnaround of highly impactful news.

The rest of this document will cover the following: my project aims as well as outcomes. I will then highlight my background research, covering a brief overview of financial markets, Natural Language Processing (NLP), neural networks and their implementations, before ending with a case study of related work. The next section will cover my programming language of choice (Python) as well as the chosen libraries and APIs. I will include my detailed plan of research as well as provide a risk analysis and GANTT chart to breakdown tasks and expected milestones. The final sections detail my conclusion, an appendix containing further information and a bibliography citing my references.

1.2 Project Aims

The aims of the project are as follows:

- I will use NLP to capture and analyse sentiment from historical public sentiment data. I intend to use the Python language and the TensorFlow library and Keras API to implement the model.
- I will then correlate the results against historical market data. I will visualise it in a graph to make it clear for all readers to interpret and understand.
- Look into how strong the correlation is. I will showcase key examples as evidence and explain using the broader context of the given the situation of said example.
- Extend the project and discuss possible applications as well apply the model to more recent data to see if it can be applied to stock prediction.

The primary outcome is that I will be able to implement a model that is capable of capturing and successfully interpreting the public sentiment data before then mapping it across to create a visual graph that highlights closely correlated points on the market prices.

Should this be successful, I will then attempt to implement my final project aim as an extra to take the project further.

2 Background Research

2.1 A brief Overview of Financial Markets

Financial markets are marketplaces that provide an avenue for the purchasing and selling of stocks, bonds, foreign exchange (FX) and derivatives (Corporate Financial Institute , 2015). It allows businesses and investors a way to make more money or raise capital to grow their business.

There are different kinds of markets, as mentioned above, and there are markets for stocks, bonds, FX, and derivatives. I will briefly outline each one, but the focus for this initial document will be on stock price movement.

Stock markets deal in the trading of shares of a company. All shares are priced, and this price will rise and fall depending on how said company performs. Events like mergers and acquisitions can cause prices to rise and fall, and significantly negative news, such as a scandal or bankruptcy can cause a stock price to nosedive. Usually, investors can either invest in a company directly, such as with Tesla and will have a stock price symbol accordingly. Tesla's is TSLA. Alternatively, one can also invest in indices, these are collections of leading companies for a given country and are used to also represent country market health, due to the fact the leading companies span all industries. A notable example is the FTSE 100 in the United Kingdom (UK).

The bond market's focus is for governments and companies to be able to raise money to fund projects or investments (Corporate Financial Institute , 2015). Investors can buy and sell debt securities from a company or government for a fixed period before then paying it back along with interest.

The foreign exchange (FX) market is where the trading of currencies takes place. It is merely traders exchanging money at specific rates and then converting back when those rates are better. Unlike the other marketplaces, there is no centralised place for FX. Instead, it is done over the counter electronically (James Chen, 2020).

Finally, we have the derivatives market. As the name suggests, this is the buying and selling of derivatives; contracts whose value is based on the value of an asset that is being traded (Corporate Financial Institute , 2015). Commodity Futures is an example of this.

For the project, the focus will be on stock market predictions as prices are purely down to company performance, perceived or otherwise.

2.2 An introduction to Natural Language Processing (NLP)

2.2.1 Overview

Natural Language Processing (NLP) is the study of programs being able to take in and process human language. More specifically, to understand what the text is trying to say. NLP applies algorithms that work to extract natural language rules such that the data can be converted to a form that the computer understands. (Garbade, 2018). The computer will then work to extract meanings from all the sentence given to it as input. This is an essential field as with today's computers and their processing power, the NLP algorithms can analyse more data than we humans could ever achieve, without rest, and at a consistently high level (SAS, n.d). The problem with the data humans produce is that it is immense, complex, diverse, and most importantly, unstructured. A lot of human languages are ambiguous and are always changing and evolving (Goldberg, 2017). As such, we must find ways to structure the data in a way that allows the computers to understand.

At present, the best-known methods for dealing with the challenge of human language is supervised machine-learning algorithms. They attempt to find patterns and regularities that provide a way of defining rules to the pre-annotated data (Goldberg, 2017). By feeding the algorithm pre-annotated data, we can train it to come up with patterns that allow it to categorise the language.

Another problem for NLP is that the data, in this case, the human text is discrete (Goldberg, 2017). Whilst we as a human can understand the relation of terms and specific words, besides its external meaning. Furthermore, as words come together to form sentences, the overall meaning of the sentence changes, these properties mean that the formed meanings of the words can be infinite (Goldberg, 2017).

Neural Networks (NN) provide a way to combat this challenge. A significant component of NNs is that they map discrete symbols to continuous vectors in low dimensional space (Goldberg, 2017). This is known as an Embedding layer. By doing this, we allow them to become mathematical objects for which we can then calculate the distance between these vectors and by extension, the words of the sentence themselves. As the supervised learning progresses, the network learns to combine the words in a way that then makes them useful. In the case of my project, it will be to find positive and negative meaning within the text to then further map to price movements on the stock market.

Before diving into the discussion of network implementation, I will first provide examples of techniques that NLPs are used to accomplish. Two main techniques used to complete NLP tasks: **syntactic analysis** and **semantic analysis** (Garbade, 2018). Syntactic analysis is used to check how well a given piece of text aligns with a given set of grammatical rules. A few notable examples of techniques used to achieve this are lemmatisation, word segmentation and parsing. Lemmatisation is the process of reducing inflected forms of a word into a single form for easy analysis. Word segmentation, as the term suggests, is the breaking up of large texts into smaller distinct units. Lastly, parsing. Parsing involves a high degree of processing, due to it using the grammatical analysis to understand the given text.

Semantic analysis works to understand the meaning behind the text. This is the tricky part of NLP as we must apply algorithms to understand the interpretation of what the text is trying to convey. Techniques within semantic analysis include; Natural language generation, word sense disambiguation and Named Entity Recognition (NER) (Garbade, 2018). Natural language generation is the use of databases to try to find the semantics and translate this to human language. Word sense works to understand the context of the text, to provide meaning to the words within the body of the text. Finally, named entity recognition, categorises test into pre-set groups (Garbade, 2018).

2.2.2 discussion into implementations of NLP approaches and conclusion

Natural Language Processing is done mainly via the use of a neural network. Loosely modelled on neurons in the brain, a neural network is an algorithm that has been fed hundreds of data points for the algorithm to analyse and gradually learn from. The data is usually labelled in advance, and the hope is that when run on new data, the algorithm can then decide on the new data. During the training process, the network utilises a loss function which drives the learning by comparing the level of inaccuracy of the predicted result with the actual result. To then minimise the inaccuracy, the error is then backpropagated along the network, updating the weights as they go along (Haghian, 2019). The most basic form is a perceptron shown in the figure below.

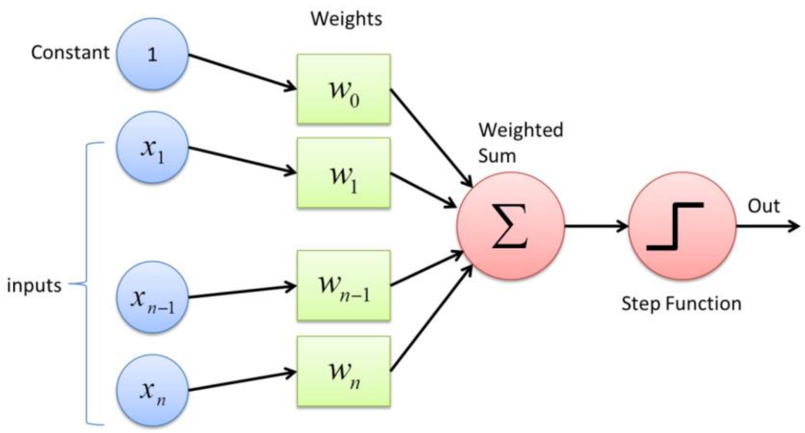

Figure 1: Perceptron example (Sharma, 2017)

As the figure shows, each neuron in the perceptron is assigned a weight. Then all the weighted neurons are then summed together to get the weighted sum. This value is very arbitrary. What would be better is to reduce the value to a range that is useful to us. This is where a sigmoid function comes in.

A (logistic) sigmoid function is one that converts an arbitrary value into one that can be interpreted as a probability. Thus, the value is usually between 0 and 1. In the context of Neural Networks, they act as an activation function. This means if they exceed a given threshold, the output value will be large. It can be considered as the neuron" firing' (Wood, Deep AI: Activation Function , n.d) Otherwise; it will be small value and effectively does nothing when the algorithm is learning. For each neuron, there will be a value of bias that acts as an area of inactivity. This acts as the threshold before neuron fires. Furthermore, this will be repeated for every neuron, and then every layer of the neural network. A network can have multiple layers before we get to the output layer. It is the job of the engineer to tweak and modify the biases and weights until we get an optimal solution. Figures 2 and 3 show a sigmoid function that captures and converts the weighted sum into a probability as well as a complete illustration of a neural network.

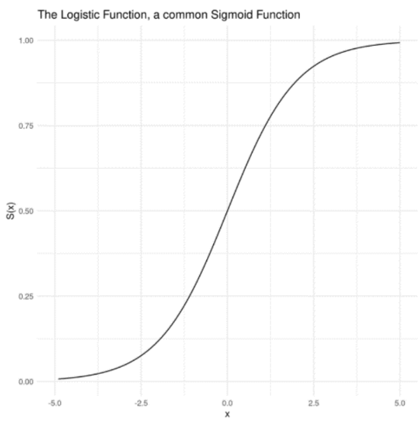

Figure 2: Common Logistic Sigmoid Function (Wood, Deep AI: Sigmoid Function , n.d)

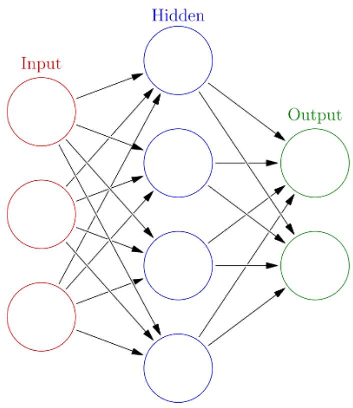

Figure 3: Illustration of a neural network (Wikipedia, n.d)

One thing to note is that there are different variants of a sigmoid function; the simplest is that of a logistic sigmoid function, which is shown in the figure above. Mathematically we can define a (Logistic) sigmoid function as follows:

$$\begin{aligned} S(x) &= \frac{1}{1+e^{-x}} \\ &= \frac{e^{x}}{e^{x}+1} \end{aligned}$$

Figure 4: Mathematical Definition for a Sigmoid Function (Wood, Deep AI: Sigmoid Function , n.d)

The function takes in any real-valued input and returns a value between 0 and 1. It is in this way we can turn an input value into the probability. The equation below shows how the sigmoid function is applied to a neuron:

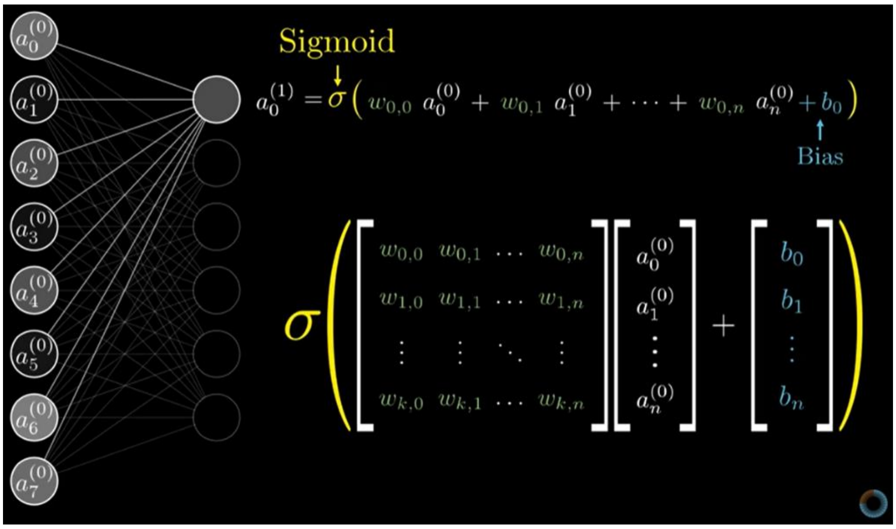

Figure 5: Application of sigmoid to a neuron (3Blue1Brown, 2017)

What we see here is the summation of all activations $a^{(n)}{}_n$ with a vector product with a matrix of all the weights. This vector represents all the weights of one layer that have a connection with a particular neuron in the next layer (3Blue1Brown, 2017). The bias for each neuron can also be organised into a vector and then added. We can then apply the sigmoid function to the entirety of this, as shown above, to get the activation for that neuron in the next layer. Every component inside the vector will be subject to the sigmoid function. By doing this, we can generalise the above equation to:

$$a^{(1)} = \sigma(Wa^{(0)} + b)$$

Where $a^{(1)}$ is the activation of the neuron in the new layer. We would then repeat this on every neuron and every layer until we get to the output layer.

There are many ways to implement a neural network for the task of NLP. The first form is making use of Feed-Forward Networks (FFN). My explanation above is the basis for this type of network. The connections between the neurons do not produce any cycles (Deep AI, n.d). Information is processed in one direction; each neuron in a layer perform their computations and then output it to the next layer. Each neuron from the input layer to the output layer is the result of the weighted sum of the previous layer passed through a sigmoid function (Haghian, 2019). A notable example of a Feed-Forward Network is a Convolutional Neural Network (CNN). The figure below has been provided as an example of the above explanation:

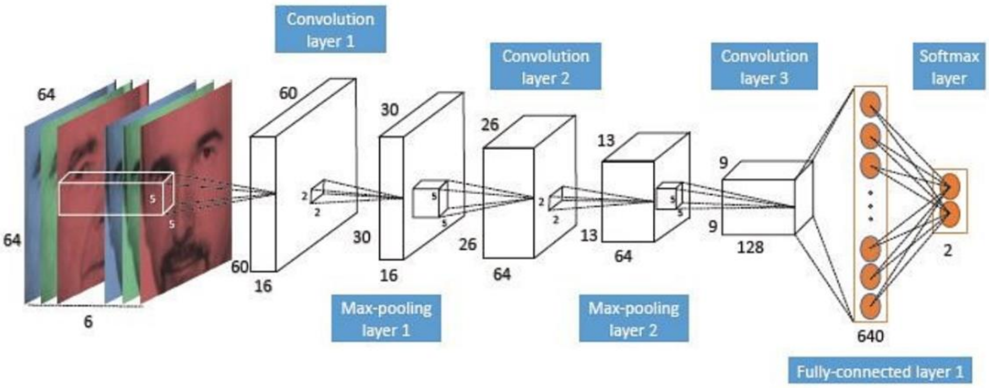

Figure 6: A Convolutional Neural Network

Recurrent Neural Networks (RNN) are another type of neural network that allows for information to be stored. This is because they contain loops within the network. This allows them to build upon experience to learn by creating a kind of memory. (Deep AI , n.d) (Haghian, 2019). RNNs are particularly useful for NLP as they are specialised for sequential data. Input is a sequence of items that produce an output vector of fixed size that 'summarises the 'sentence' (Goldberg, 2017). A key benefit of RNNs is that they are trainable components that can be fed into other networks. An example of this could be feeding RNN output data into an FFN network. A diagram of the network is shown below:

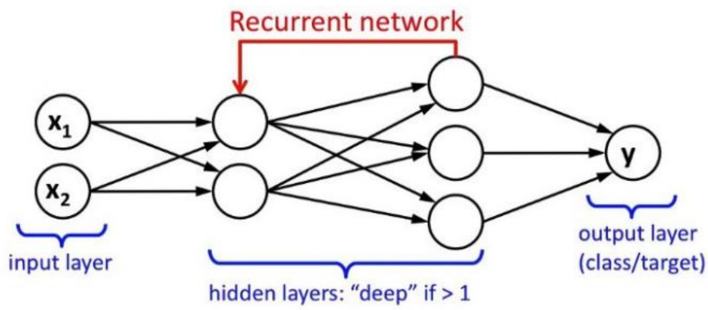

Figure 7: A diagram of a Recurrent Neural Network provided by Leonardo Araujo dos Santo's Artificial Intelligence

The recurrence relation is defined as (Valkov, 2017):

$$S_t = f(S_{(t-1)} * W_{rec} + X_t * W_x)$$

Where:

- S_t – the state at time step t
- X_t – External input at time t
- $S_{(t-1)}$ – the previous state
- W_{rec} – weights parameter
- W_x - weights parameter

Because of the feedback loops, RNNs can compute the current state from the previous state, as well as predict the next state from the current state. An example of this is a Long Short-Term Memory network (LSTM). The composition is outlined below:

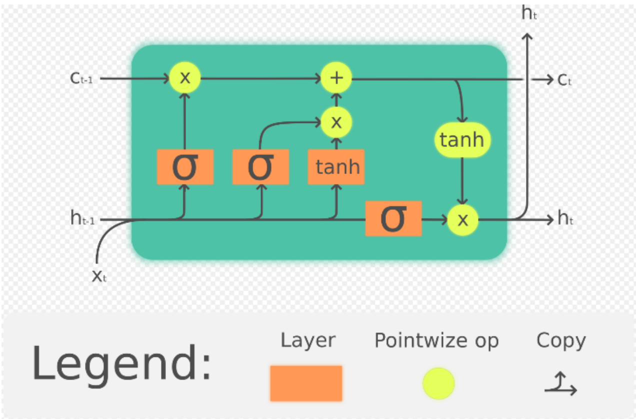

Figure 8: A LSTM cell (Wikipedia , n.d)

An LSTM is composed of a cell unit, an input gate, an output gate, and a forget gate. For a specific time period, the LSTM can retain information about the previous state (Wikipedia , n.d).

Of the two options to creating a neural network, I intend to choose the RNN approach due to its ability to retain a memory of the previous state. I believe this is crucial in being able to construct sentiment from a sentence given the compositional nature of human text data (Goldberg, 2017). There is the problem of vanishing and exploding gradients (Valkov, 2017), and this will be addressed by using the Long Short Term Memory (LSTM) architecture. However, as RNNs are generally a component rather than a standalone network (Goldberg, 2017), I plan to embed the RNN within a Feed-Forward network to make use of its classification properties.

2.3 Case Study: Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network

Huicheng Liu's project makes use of attention-based LSTM to make predictions on the S&P 500 market. This project is very similar to mine as my project will make use of a generic LTSM model to capture and analyse financial news and then correlate it against a chosen financial network.

In this project, an RNN is used to specifically encode news text and context information. What differentiates his approach from my approach is that he used a self-attention mechanism to distribute the attention received by words across the text, whereas I will be omitting this process in my project. Financial Data was acquired from Reuters and Bloomberg, which is also where I will be collecting my Financial data.

Liu's model was split into four stages: Input and word embedding layer, news-level Bi-LSTM and attention layer, day-level – Bi-LSTM and self-attention layer, and finally, the output layer (bibtex, 2018). The model is shown below:

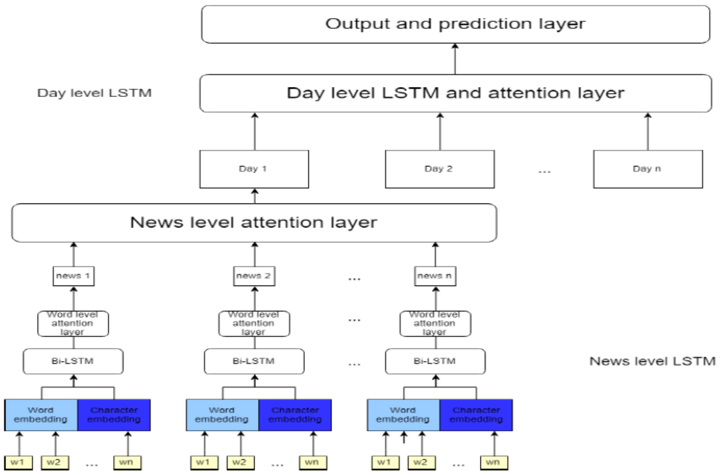

Figure 9 Model for the Attention Based RNN

From the paper, there are several key takeaways:

- Liu used a variant of an RNN known as a Bidirectional LSTM (Bi-LSTM). Unlike the generic LSTM I had planned to use, a Bi-LSTM is better at understanding the context of a given text as it is bidirectional.
- Previous papers had reduced the dimensionality of the data, but this resulted in a loss of information. To combat this, the researcher removed punctuation and encoded the entire text. This method tackles the sparsity problem (Goldberg, 2017) (bibtex, 2018).
- They made use of a self-attention mechanism. This distributes the attention given to embedded words when taken as input.
- For the data, it was collected from Reuters and Bloomberg during 2006 and 2013. Reuters was used to collect data for 473 companies listed on the S&P 500 from a period of 2013 to 2018. Next historical price data between the period of 2006-2018 was collected from Yahoo Finance. The data was then partitioned according to the purpose.

Data for S&P 500 index prediction			
Data set	Training	Development	Testing
Time interval	20/10/2006-27/06/2012	28/06/2012-13/03/2013	14/03/2013-20/11/2013
News	445,262	55,658	55,658

One thing that was significant was that it was not the full content of the news but rather the headlines that were used.

After looking through this case study, it has informed me of areas of the project that need adjusting to align with the goals of my project—the dataset. The final data set was rather a collection of data sets instead of just the one. For a greater breadth in data, I will incorporate this into my project. Some data will also be marked for training, development and testing as was done in the case study. Also, as I mentioned earlier in this document, RNNs work better as a component rather than standalone networks. The kind of RNN model I implement will be vital to gaining better accuracy when training the network.

3. Project Management:

3.1 Methodology & Work Schedule

Including this document and the dissertation, the project will have four phases; Initial Document, Implement RNN, Find the correlation with market data and visualise and the dissertation. Due to the nature of stages two and three, there will be continuous testing throughout the development. I will use a SCRUM methodology, splitting the work into several stories for a sprint cycle. Each sprint cycle will last two weeks, with each block of work assigned a story number, as well as a status. As I am the only one doing the project, I will be both SCRUM master and programmer making the sprint. The timescale for this project is seven months, from the original setting of the initial document to the dissertation.

Below are the technologies and languages I will be using for my project:

- **Python:** Python is a high-level, interpreted programming language. It is the ease of use, and low learning curve makes it highly accessible, and it has several libraries dedicated to machine learning and Data Science, that make it perfect for use in my project.
- **PyCharm:** Developed by JetBrains, it is an IDE made specifically for the Python programming language. It has several features, such as code suggestion and completion, highlighting of errors, as well as an extensive library of debugging features. It has support for external plugins as well as native support for version control using Git. This will be my IDE of choice when implementing the software aspect of the project. These IDE features will help to speed up development when testing and to look for errors. The inbuilt version control will allow another way to protect the integrity of the project when coding.
- **GitHub:** Owned by Microsoft, it provides hosting for source code as well as version control via Git. The features of Git, especially branches, make this my primary choice for hosting my source code for my project. It will allow me to back up, roll back to previous versions and implement correct software development principles by having a master branch and incrementally integrating new stories.
- **TensorFlow:** TensorFlow is a free and open-source library for Machine Learning (ML). Developed by Google Brain Team, it has a focus towards training and development of Neural Networks. It is a library within Python so I will be making use of it in the development of my project.
- **Keras:** I will also be making extensive use of Keras. Keras is an API [6] that acts as an interface for TensorFlow but speeds up development by simplifying the process. There is not a need to understand Tensors, operations and sessions (Heller, 2019) when working with Keras. Keras is designed to be user-friendly which, coupled with the documentation, will significantly speed up development, reducing the learning curve.
- **Google Drive:** Google Drive is cloud storage hosted by Google. I can use it store both my Initial Document and my dissertation as well as the source code for my project. Google Drive also allows me to roll back to previous versions. At present, the storage space is approximately 100GB, which will make it suitable for storing the above.

3.2Risk Analysis

Below I have outlined both the risks to myself and the project.

Risk Analysis			
Risk	**Impact**	**Probability**	**Description**
Technical Risk			
Dataset issue – lack of access.	Medium	Medium	Some datasets are locked behind paywalls or are not subject to use by others. To mitigate the risk of lack of dataset access, the focus will be looking at places that hold open-source resources for datasets.
Quality of Data	Medium	Medium	The quality of the data may not meet the requirements of the project. This will lead to issues when training the network. To combat this, I will find high-quality data that meets the specifications of the project. I may also need to change the dimensionality of the data to suit my needs.
Performance	High	Medium	The model may not perform well for my project. The kind of adjustments made must be made continuously and for justified reasons to prevent this. I will need to increase my background knowledge as a result.
Loss of project	High	Low	Loss of my project will include the loss of the trained model and the dataset. This will cause me to fail the module. I will back up the project in multiple places. I will make use of cloud storages like OneDrive and Google Drive. I will also use version

			control software Git.to keep a consistent backup.
Loss of dataset	Medium	Low	Not as catastrophic as losing a project. Depending on dataset availability and size, recovering from project delay may range from moderate to severe. It might even result in having to choose a new dataset. To avoid this, I will keep backups of all datasets used.
Personal Risk			
Severe Illness (COVID-19, Ebola, Malaria)	High	Low	Depending on my immune response to the illness, an extreme response may result in being bedridden for a few weeks. This will significantly stall progress on the project. To mitigate this, I will adhere to lockdown measures to prevent exposure to potential carriers.

3.3 Project Timeline

The Gantt Chart Below outlines the plans for the project going forward for the next seven months:

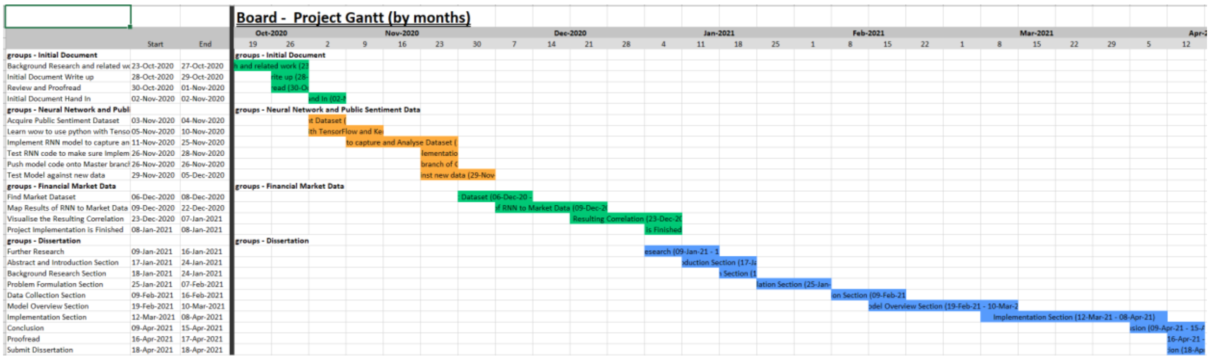

Figure 10: Gantt Chart displaying Project timeline in months

The Gantt Chart has been created on a project management software called Monday.com. Should deliverables be met early or delayed, the Gantt chart can be updated automatically, and a new excel spreadsheet released.

4 Conclusion

The aim of this project, as outlined above, is to explore the correlation between two seemingly very different types of data using deep learning techniques. By using powerful machine libraries to leverage neural network variants, the goal is that at the end of the project I will have a clean, professional solution that visually shows the correlation and expected conclusion, but also adheres to professional software engineering principles. The base of the model will employ a Recurrent Neural Network before using a general feed-forward network to classify whether the sentiment from the data is negative or positive. The results will then be mapped to observe correlation. Based on my project timeline outlined in Figure 10, the project should be completed by April 2021.

5 Glossary

- **UBS Orca** – An algorithmic Trading system by UBS. It utilises Machine Learning to learn in real-time using historical price data, allowing it to provide liquidity in volatile market conditions.
- **Machine Learning** – A sub-field of Artificial Intelligence, it is a discipline that focuses on algorithms that learn through experience.
- Artificial Intelligence
- **Big Data** – A field which aims to deal with extracting information from data sets that are too large or complex to be extracted from using traditional methods.
- **Dovish** – Associated with statements and press releases from financial institutions. Generally, statements of this nature mean a firm will be taking soft or passive decisions. Usually when in negative economic situations statements are of a 'dovish' nature
- **Hawkish -** Associated with statements and press releases from financial institutions. Generally, statements of this nature mean a firm will strong or aggressive decisions. Usually, when in favourable economic situations, statements are 'hawkish'.
- **Neural Network** – System inspired mainly by biological neural networks. Based on the concept of neurons and are used to train a system to learn a particular task.
- **Deep Learning** – A sub-field within Machine Learning, focuses on using algorithms to construct artificial neural networks.
- **Natural Language Processing** – Using machine learning techniques to analyse text data and derive sentiment and context from the inputted text.
- **Python** – Python is a high-level, interpreted language, that emphasises user-friendly code. It has an extensive collection of machine learning libraries that have made it the go-to language for machine learning.
- **TensorFlow** – TensorFlow is an open-source machine learning library created by Google Brain Team. Their focus is mostly on training deep neural networks.
- **Keras** – Keras is an open-source library that acts as an interface for TensorFlow when making artificial networks.
- **API** – Application Program Interface is an interface that facilitates interactions between multiple pieces of software—usually done via API calls.
- **Long Short-Term Memory (LSTM)** – It is an Architecture for Recurrent Neural Networks, a variant of neural networks. LSTM aims to solve the problem of gradient vanishing and exploding.

Bibliography

3Blue1Brown. (2017, October 5). *Youtube: But What is a Neural Network? [Video]*. Retrieved from Youtube: https://youtu.be/aircAruvnKk

BBC News. (2020, March 25). *US markets close higher after emergency virus deal*. Retrieved from BBC News: https://www.bbc.co.uk/news/business-52030034

bibtex, a. [.-f. (2018). Leveraging Financial News for Stock Trend Prediction. *arXiv*.

Corporate Financial Institute . (2015). *Financial Markets* . Retrieved from Corporate Financial Institute: https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/financial-markets/#:~:text=Financial%20markets%2C%20from%20the%20name,of%20assets%20such%20as%20bonds&text=Simply%20put%2C%20businesses%20and%20investors,to%20make%20more%20money%

Deep AI . (n.d, n.d n.d). *Deep AI: What is a Recurrent Neural Network* . Retrieved from Deep AI : https://deepai.org/machine-learning-glossary-and-terms/recurrent-neural-network

Deep AI. (n.d, n.d n.d). *Deep AI: What is a Feed Forward Network*. Retrieved from Deep AI: https://deepai.org/machine-learning-glossary-and-terms/feed-forward-neural-network

Garbade, D. M. (2018, October 15). *A simple introduction to Natural Language Processing*. Retrieved from Becoming AI : https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32

Goldberg, Y. (2017). *Neural Networks for Natural Language Processing* . California: Morgan & Claypool Publishers.

Haghian, P. Z. (2019). Deep Representation Learning and Prediction for Forest Wildfires. *n/a*, n/a.

Heller, M. (2019, January 28). *InfoWorld: What is Keras? The deep neural network API explained*. Retrieved from InfoWorld: https://www.infoworld.com/article/3336192/what-is-keras-the-deep-neural-network-api-explained.html

James Chen, G. S. (2020, September 12). *Forex Trading: A beginner's guide Investopedia* . Retrieved from Investopedia : https://www.investopedia.com/articles/forex/11/why-trade-forex.asp#what-is-the-forex-market

SAS. (n.d, n.d n.d). *SAS Insights: Analytics and Data Science Insights: Natural Language Processing (NLP) what it is and why it matters*. Retrieved from SAS: https://www.sas.com/en_gb/insights/analytics/what-is-natural-language-processing-nlp.html#howitworks

Sharma, S. (2017, September 9). *Towards Data Science: What the Hell is a Perceptron?* Retrieved from Towards Data Science : https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53

Valkov, V. (2017, May 25). *Medium: Making a Predictive Keyboard using Recurrent Neural Networks — TensorFlow for Hackers (Part V)*. Retrieved from Medium: https://medium.com/@curiousily/making-a-predictive-keyboard-using-recurrent-neural-networks-tensorflow-for-hackers-part-v-3f238d824218

Wikipedia . (n.d, n.d n.d). *Wikipedia: Long Short Term Memory*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Long_short-term_memory

Wikipedia. (n.d, n.d n.d). *Wikipedia: Artificial Neural Networks* . Retrieved from Wikipedia : https://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Colored_neural_network.svg

Wood, T. (n.d, n.d n.d). *Deep AI: Activation Function* . Retrieved from Deep AI : https://deepai.org/machine-learning-glossary-and-terms/activation-function

Wood, T. (n.d, n.d n.d). *Deep AI: Sigmoid Function* . Retrieved from Deep AI: https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function

Xin Du, K. T.-I. (2020). Stock Embeddings Acquired from News Articles and Price History, and. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3353–3363.