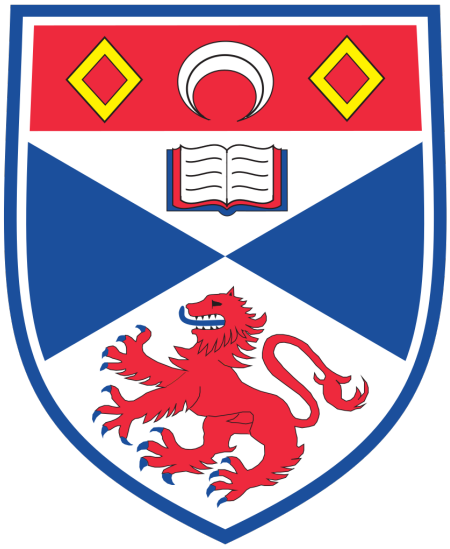
Using NLP and Geopolitical Events Presented in the Global Database for Events, Language, and Tone to Perform Predictive Modelling on Stock Market Data

Jalaj Khandelwal
Student ID: 160001602



University of St Andrews

School of Mathematics and Statistics

Supervisor: Dr Carl Donovan

Module: MT5099 Masters Dissertation in Partial Fulfillment of the MSc. in Data Intensive Analysis

Date: August 17, 2020

Declaration

I hereby certify that this dissertation, which is approximately 14,500 words in length, has been composed by me, that it is the record of work carried out by me and that it has not been submitted in any previous application for a degree. This project was conducted by me at University of St Andrews from June 2020 to August 2020 towards fulfilment of the requirements of the University of St Andrews for the degree of MSc. in Data Intensive Analysis under the supervision of Dr Carl Donovan.

17th August 2020

Jalaj Khandelwal

Abstract

This Masters Dissertation was focused on trying to use geopolitical events to predict changes in the stock market. This aim was to first use the Global Database for Language and Tone (GDELT) to build topic models, which organise the geopolitical news into topics which could be used as a filter for the daily news cycle. Then, using the Tone and Geopolitical conflict scores provided by GDELT, models would be made made to predict changes in the stock market. There were two main topic modelling approaches, firstly using Latent Dirichlet Allocation, and secondly using K-Means clustering with TF-IDF. Both of these approaches either were incapable, or struggled to filter unseen news effectively, which means that they would not be effective in picking up new topics. However, many different stock predictive models were tried, and it was found that the accuracy of models using information provided by GDELT was higher in predicting binary stock shift, compared to a model which just only used the previous day's stock shift as a predictor. The best of the models also were able to demonstrate adding value to a simple portfolio, which suggests that using GDELT is a viable metric by which to predict the stock market.

Impact of COVID-19

The impacts of COVID-19 was minimal on this dissertation. The only issue was the amount of data used being restricted due to the machine constraints. Without Covid-19 data covering a longer time period from GDELT would have been used.

Contents

List of Tables

List of Figures

1 Introduction

There are many geopolitical events occurring worldwide, which may have a knock on affect on stock markets. To be able to find and predict the impact such events would have on markets would be of great interest to investors. If, by mining the geopolitical news these events are reported in, mathematical predictions are possible and accurate with regards to the direction the stock market will go, it would mean investors will be able to not only not *lose* money, e.g. by selling in a bear market before the price crashes, but in certain circumstances, be able to make money, e.g. buying in a bull market before the price increases.

This dissertation was focused around using the Global Database of Events, Language, and Tone (GDELT) to build topic models which could be used with models which predict the changes in the stock market. The topic models and the GDELT data used was related to geopolitical information. The aim was to use GDELT data to build a topic model which could be used to filter news information for specifically geopolitical news. Then the geopolitical measure of the conflict cooperation Goldstein Scale for geopolitical events provided by GDELT would be used to try and predict information on the stock market.

The methodology of curating topic models from geopolitical news involved taking news headlines from GDELT data over a defined period of time. Then the underlying topics present in the news were teased out by grouping the words in the headlines together using different techniques. The aim for the underlying topics which emerge from word groupings was that they would be focused on and related to geopolitical topics. The next step was to use the curated topic model to filter other news to end up with headlines specifically related to that geopolitical topic. The other news would be 'new' news which the model had not seen before, which would in the real world represent the daily news cycle.

The initial approach to building the topic models was to use existing topic modelling algorithms. The first topic model used was Latent Dirichlet Allocation. This was run on data from the GDELT events table, which was specific to events which concerned the USA or China. The aim for this data would be to sort all of the words present into a predefined number of topics, henceforth referred to as K, this is selected by the user before building a model. USA and China specific data was chosen for two reasons. Firstly it was to narrow down the dataset, as GDELT is a very large resource, and it would be more feasible initially to run on a smaller dataset. The second reason for this is using geopolitical news would have to be targeted considerably to certain events in specific countries and particular markets. Over the time period from March to April, there were significant events which took place specifically between those two countries, and it was thought this would provide the best opportunity to find signal within the data.

For a topic model to filter other news, there would be a binary or quantitative measure produced when a specific item of news is tested in topic model. If binary, this would either say the news item is part of the topic or is not. A quantitative measure, referred to as the distance, would provide a numerical value which would represent how 'far' away a news item is from the topic in the topic model, with smaller values suggesting the news item is related to the topic in the topic model, and larger values suggesting the news item is less related to the topic in the topic model. The LDA model did not provide a binary metric, and was not able to provide a quantitative measure for news which included words not already present in the topic model. Thus, a different approach was used to represent topics, which would be able to provide quantitative measures.

The K-Means clustering algorithm was used alongside the Term Frequency Inverse Document Frequency (TF-IDF) to build clusters from the words in the headlines. This approach used TF-IDF to represent the words numerically, such that the K-Means algorithm could be used. This is explained more fully in chapter 4. This would mean that the clusters would be built based on words and phrases which occur frequently and were important in the text. It was hoped those clusters would represent the words and the phrases which are relevant to geopolitical information. Whilst this approach was able to provide a quantitative distance, the actual distance values produced were

unusable as they did not reliably filter topic relevant news from irrelevant news or noise.

The initial aim was to use a topic modelling algorithm trained on USA/China to filter all of the news provided by GDELT to news which was just relevant to the main topics present in the USA/China data. Neither of the topic modelling approaches could filter 'new' news headlines accurately. Thus the dataset used to build the stock prediction models was the same USA/China dataset used to build the topic models.

The models for stock prediction modelled the Dow Jones Industrial Average. This is a market index which represents the averaged stock performance of 30 large USA companies on the NASDAQ and the New York Stock Exchange stock markets. This was chosen again because geopolitical risk prediction would be done on specific markets, and it was thought that any events concerning the USA and China would be reflected into the Dow Jones. When making predictive models, stock shifts, meaning here a binary increase or decrease of the index on a daily basis, was used for predictions as opposed to predicting exact stock index value. It would also be assumed that the influence on the stock prices and thus the Dow Jones from a geopolitical event was considered to potentially extend over several days, and thus any models for prediction would have to take this into consideration.

GDELT provides two quantitative values in the data, the Goldstein Scale and the Average Tone. Both of these describe the news articles and the events that they are describing and are explained more fully in chapter 3. These two were the main features used in modelling the Dow Jones Average, alongside using the previous day's stock shift. Many different classification models were tried, including Random Forests Classifiers, Support Vector Machines (SVMs), Logistic Regression, and a more Bayesian approach in Naive Bayes. Furthermore a simple Multi Layer Perceptron Neural Network was also tried. Several different methods of introducing lag into the Goldstein Scale and Average Tone were also tried, these are explained further in chapter 6, and used alongside the models listed.

Since the prediction models were predicting a binary change in the stock price index, the best model was picked on the basis of the model with highest accuracy, which meant the model which predicted whether the price went up or down correctly for the highest number of days over the period of time the models were trained on. This was a Random Forests Classifier using exponential lagging. This model was further tested over a longer time interval by predicting the stock index shift in the interval of May and June 2020. To gauge whether the Average Tone and Goldstein Score values were useful in predicting the shift, a reference Random Forests model was trained which just used the previous day's stock change as a predictor. This reference model was also tested over the May-June interval, and it was found that the 'best' Random Forests model accrued a higher accuracy than the reference model over the prediction interval. Furthermore, a 'real life' test was also performed for the best model and the reference model, where they were used as predictors in managing a simple portfolio, and it was found that if the model was used, the portfolio did grow in value over the May-June interval.

There remains significant further modelling possible to explore both GDELT and topic modelling further, both in terms of the scale of the data being predicted, and different markets and modelling strategies.

This report will cover relevant background literature in chapters 2, 3, and 4; an overview of the experiments performed and results in chapters 6 and 7, and a discussion of the results in chapter 9.

2 Existing Work

There exist several approaches to model and predict stock prices. These approaches use many different sources of data and varying modelling styles. However, there remains limited prior work in modelling stock prices using calculated Geopolitical risk, though there are limited approaches to calculating geopolitical risk on its own.

2.1 Geopolitical risk

The majority of approaches in modelling Geopolitical risk work on using news information along with manually defined terms of geopolitical risk. One of the most recent approaches was done by Dario Caldara and Matteo Iacoviello, where they created a Geopolitical risk index from a series of news articles [11].

This risk index was calculated by manually creating search terms which define geopolitical risk in several categories, and then over time calculate the percentage of articles of selected number of news media in which the search terms appear. This is then used to model the stock prices.

There are other risk measures present, Black Rock also have a publicly viewable index called the BlackRock Geopolitical Risk Indicator (BGRI) [1], which works in a similar fashion where key words are selected which define geopolitical risk, followed by an application of a sentiment score from a list of predefined words, and then those two scores are combined to make the BGRI total score. Furthermore, there are also many other organisations which almost certainly will model Geopolitical Indices, such as Reuters, but will not disclose the exact modelling for proprietary purposes.

One of the major differences between the existing approaches in modelling geopolitical risk and the approach used in this dissertation, is here we incorporate the cooperation conflict scale provided in GDELT. GDELT also differs from the data sources used in existing modelling approaches. In existing approaches, only manually selected news media are used for the data, whereas GDELT does not restrict which news are stored and calculated, and thus has a much wider scope in terms of news and events stored. Furthermore, this approach attempts to use topic modelling to filter the news, as opposed to using manually defined search terms, in an attempt to make the prediction process more automated.

2.2 Modelling Stock Prices

There is a large vested interest for people to model the stock market effectively, as any investor with extra knowledge about where the market will go will be able to take a position most beneficial to them. Thus, since having accurate predictive models are so beneficial, there exist many models which aim to predict market changes.

The Efficient Markets Hypothesis asserts that current securities prices reflect all available information and expectations[15]. This is related to random walk theory, which in finance literature is used to describe a price series where all price changes are random departures from earlier prices, which means that any specific day's price change is not related to the previous day's price and only on the news that is received on that day. Whilst this theory has become more controversial [23], it is accepted that stock prices are related to new news. This means that Geopolitical events which generate news, and by extension an index of risk which track them, should have a measurable effect

[1] https://www.blackrock.com/corporate/insights/blackrock-investment-institute/interactive-charts/geopolitical-risk-dashboard

on stock prices. For example, the risk would increase if a 'bad' event occurs, such as an increase in international conflict, or a negative piece of news is published. Since the risk would be higher it could be assumed that stock prices would decrease as investors could be less likely to buy and more likely to sell which would cause the decrease in the price. If the reverse were to happen with the risk index, it could be assumed the opposite would happen in the market.

As stock market modelling has existed for many years there have been several esoteric factors used to model and predict stock prices aside from the news, such as the weather on Wall Street [32]. For most of the non quantitative data, sentiment analysis is commonly used to help predict stock prices, where natural language processing is used to gain a measure of tone in the data being used. The tone is a quantitative measure of how positive or negative the text in the data being used. This is then used to model the price changes. A number of studies shows that large amounts of data on social medias such as Facebook and Twitter can be useful for forecasting economic indicators such as the stock market [9] [5].

There are many different machine learning and statistical models used for modelling stock prices. One of the simplest approaches to stock price modelling is to perform autoregressive modelling on purely the time series data for stock prices. In this approach, the only thing which is used to predict the prices are the past values of the stock. One common approach is to use an autoregressive integrated moving average model (ARIMA) [6]. This is part of a set of models where the response variable, Y, is an ordered time series, where the Y for any given time is dependent on and calculated only from previous values in the time series.

When predicting stock prices from variables other than the previous prices, many different machine learning algorithms are used. Aside from classical logistic regression models, these include SVMs [12], Random Forests [18], and Neural Networks [14], along with Bayesian approaches such as Naive Bayes [20]. These models can be used to predict both the stock prices, with a numeric response variable, and binary stock shift [27], with a class response.

There is another concern with modelling on the stock market. When predicting any stock market, one of the major issues which arises is that in the long term the stock market always grows, for the S and P for example there is an annualised growth of on average 9.5% [3] without the affect of inflation factored in. Thus any investor investing over a sufficiently long period of time will nearly always make money. The aim then, is to build models which would give better results than the normal stock market rise, and thus any strategy or predictions by any model would have to prove and provide better returns than just investing in the long term growth of the stock market.

A mixture of these algorithms were used alongside the GDELT algorithm to try and predict stock shift.

3 GDELT

The Global Database for Events Language and Tone was the basis for this dissertation. This is a database which stores a record of every broadcast, print, and web news published since 1979, in over 100 languages from all parts of the world.

The records are stored across multiple database tables accessible either directly, or through SQL in Google Big Query. The main table which was the source of the data in this dissertation was the Events table. This table maintains the record of all events, along with cursory information about each event, such as which countries it happened in or between, the dates, and the URLs of the first news article this appeared in. The events table contains over a quarter of a billion georeferenced records each referring to an event which occurred since the 1st of January 1979.

For each event, there are three variables of note for this dissertation, the source URL, the Average Tone and the Goldstein Scale. The source URLs were the URLs of the first headline which covered the event. The topic models were built on these headlines of the articles extracted from the source URLs, where present. The Average Tone is a measure of the sentiment of all the news media which describes the specific event. It is on a scale of -100 (Extremely Negative) to +100 (Extremely Positive). The most common values observed are usually between -10 and 10, and 0 indicated a neutral tone.

The Goldstein Scale is the main metric by which geopolitical modelling occurs, and was initially defined as the Conflict-Cooperation Scale for Work Environment Impact Scale (WEIS) events data [17]. This is a scale between -10 and +10 which captures the potential impact an event will have on the stability of the entity in question (usually a country), with -10 being the worst impact an event can have and +10 being the most positive impact an event can have. It should be noted that these numbers are predefined and do not take into consideration the extent to which an event occurs, for example event type 071, Extending economic aid; give, buy, sell, or borrow, is given a score of +7.4, regardless of whether it is £1 of aid given or £1 million.

GDELT has been used for stock predictions in the past [25] [4], however, it is used much more commonly for predicting and analysing geopolitical events themselves [29] [37].

4 Topic Modelling

Topic Modelling is an unsupervised machine learning and statistical modelling approach which aims to discover the abstract 'topics' in a collection of documents, referred to as the corpus. This can be used to find hidden semantic structures in a document. This works by looking at how often words appear in the corpus and trying to create topics made of words which are related to each other. For example an article about dogs would be more likely to have words such as 'dog' and 'bone', and thus the topic model would most likely group those two words together into a topic.

There are many approaches to topic modelling, one of the foremost examples is Latent Dirichlet Allocation [8], which aims to reverse-engineer the topics by assuming the text was created from K topics and all words directly relate to one of the K topics. There are other forms of clustering algorithm which can be applied to words, such as the K-Means clustering algorithm.

4.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is an unsupervised learning model which aims to collate words into K topics. When performing LDA, the text to be analysed is split into a series of documents, each composed of a bag of words where the order of the words does not matter. Most of the time, this will require pre-processing to remove words which do not contribute to the topics present in the work such as 'the', 'is', and 'a'.

This algorithm assumes that the documents was made by first picking K topics, and any words present in the document belong to one of those K topics. The algorithm aims to reverse-engineer this process.

Initially if the corpus is composed of a set of documents, D, the algorithm would first perform tokenisation so that each individual word is treated as a unique identity. Then each of the tokens for each of the documents would be parsed to remove the stop words. At this point each document is a list of tokens representing words which are not stop words. From this a document term matrix is created. This matrix has the dimensions m x v, where M is the number of documents present in the corpus. V is the number of unique words present across all of the documents, or the vocabulary. This document term matrix records the term frequency of all of the words in the vocabulary for all of the documents present.

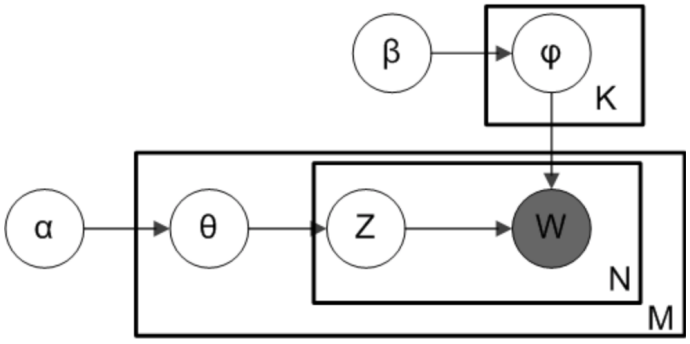


Figure 1: A Graphical Plate representation of (smoothed) LDA (image from [35])

The reverse-engineering process is shown graphically in Figure 1. M denotes the number of documents, N refers to an individual document, and W refers to a single word, and is the only observable variable in the system. The algorithm assumes several matrices. $\boldsymbol{\varphi}$ is defined as the word distribution across topics. In practice this is a matrix where $\boldsymbol{\varphi_k}$ is the probability distribution across the V

for topic K, such that $\boldsymbol{\varphi_{j,k}}$ represents the probability that the j^{th} word in the vocabulary belongs in topic K. $\boldsymbol{\theta}$ is defined as the topic distribution across documents, which means $\boldsymbol{\theta_i}$ represents the topic distribution for document i, and that $\boldsymbol{\theta_{i,k}}$ represents the probability of topic k being in document i. Z represents the matrix of documents and topics, where $\boldsymbol{Z_{i,j}}$ is the currently assigned topic for the j^{th} word in document i.

α and β are the external parameters which control the initial distributions. α is the parameter which initially sets the shape of the topic distribution across documents, $\boldsymbol{\theta}$, and β is the parameter which initially sets the word distribution across topics, $\boldsymbol{\varphi}$. The aim is to optimise parameters α and β to find the best distribution of word and document probabilities which have generated the corpus most accurately. In the original paper [8], both the topic-word distribution, β, and the topic-distributions, α can be modelled using a sparse Dirichlet prior, as it would be thought that the probability distribution of words in a topic and documents across topic would not necessarily be symmetric, and not all documents/words would contain all topics. Large values in α push the document topic distribution towards being more balanced between topics, and smaller alpha values push the document topic distribution probabilities towards being weighted more towards certain topics than being weighted evenly.

This can be represented as a posterior probability as below (Equation taken from [2]):

$$p(\varphi_{1:k}, \theta_{1:M}, z_{1:M}|D; \alpha_{1:M}, \beta_{1:K})$$

This can be calculated using variational inference, as the probability defined above is intractable. This entails calculating an approximation of the true posterior probability, and minimising the difference between the true posterior and the estimated posterior. In this case, the difference between the true and approximated posterior is the distance between them. To obtain the most accurate approximation, this distance has to be minimised which in this case is achieved by minimising the KL divergence [21] between the approximation and the true posterior probability. The optimisation is shown below (Equation taken from [2]):

$$\gamma^*, \phi^*, \lambda^* = argmin_{\gamma^*,\phi^*,\lambda^*} D(q(\varphi, \theta, z, |\gamma, \phi, \lambda)||p(\varphi, \theta, z|D; \alpha, \beta))$$

γ, ϕ, and λ are the free variational parameters used to approximate $\boldsymbol{\theta}$, z, and $\boldsymbol{\varphi}$ with. D(q||p) represents the KL divergence between q and p. Changing the γ, ϕ, and λ parameters changes the distance between the estimate, q, and the true posterior, p, and the aim is to find the values which minimises that distance.

In algorithmic terms this works on the basis of optimising one of $\boldsymbol{\varphi}$, $\boldsymbol{\theta}$, and $\mathbf{z}$ at a time. This is because these matrices are are intrinsically linked to one another. Pseudocode for the algorithm is shown below (Algorithm taken from [1]):

Initialise Topics based on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
repeat
    for each document do
        repeat
            Update the topic assignment Variational parameters ($\boldsymbol{\theta}$)
            Update the topic proportions Variational parameters ($\boldsymbol{\varphi}$)
        until document objective converges
    end for
    update topics from aggregated per-document parameters ($\mathbf{z}$)
until corpus objective converged

LDA has been used widely for topic modelling across many fields. This has also been used in stock market predictions by attempting to mine many different kinds of data to used as prediction. This includes seeing if anything from social media data [27], to topics in financial news [16] affect stock

prices. This is similar to the goal of this dissertation, and thus this was deemed a suitable metric to use for this purpose.

4.2 Term Frequency-Inverse Document Frequency

TF-IDF is a well known numeric statistic used to calculate the importance of a word to a document in a collection or corpus. It works based on of creating a weighting for each word, based on the product of the term frequency and the inverse document frequency. The term frequency is the count of how many times each word appeared in the document. The inverse document frequency aims to measure how much information a word provides to the document, i.e. if the word appears extremely often (e.g. the word 'the'), it would attain a lower IDF score, and vice versa. It is calculated by taking the logarithm of the inverse of the fraction of documents which contain that word. The final TF-IDF value is calculated by multiplying the term frequency and inverse document frequencies together. This is shown below:

$$TF(t,d) = f_{t,d}$$
$$IDF(t,D) = log\frac{N}{\{d \in D : t \in D\}}$$
$$TF - IDF(t,d,D) = TF(t,d) * IDF(t,D)$$

The term frequency, *tf*, for a term t, in a document d, is found by calculating the frequency of term t in document d. The inverse document frequency, for term t, in a set of documents, D, is the log of the number of documents in the corpus, N, divided by the number of documents, d, in the set of documents, where term t is in the document.

TF-IDF is used extensively in text mining, as it shows the most important words of a corpus, this is used extensively, from paper recommender systems [7], search engines [36], and digital libraries [28] alongside other uses. This has also been used extensively for predicting stock prices, as part of a wider prediction using sentiment analysis model.

4.3 K-Means Clustering

Another methodology of grouping objects together is to use a clustering technique such as the K-Means algorithm. This algorithm, given a matrix $\mathbf{X}$, of dimension n x p, where each row vector represents a point in p-dimensional space, places K candidate cluster centres randomly in p-dimensional space. Each of the points in p-dimensional space is allocated to the closest cluster. This is found for each point by calculating which cluster centre has the minimal distance to that point.

The distance metric used for calculating distances between points in p-dimensional space can vary but it is most often the Euclidean Distance. The algorithm is an optimisation problem which aims to minimise the Within Cluster Sum of Squares, which is also the cluster variance. After assigning the points to the clusters, the location of the cluster centres are shifted to the mean of all of the points assigned to that cluster. Then the distances to the cluster centres are recalculated for all of the points, and the points are reassigned clusters to the cluster who's centre is the closest. This process of moving the cluster centres and reassigning the points continues until the assignments of the points to the clusters do not change. It should be noted that this algorithm is heavily dependant on the starting positions of the cluster centres, and is not guaranteed to find the optimal solution. As such it remains a computationally NP hard problem [33].

The K-Means clustering algorithm only works with numerical data in n dimensions as it needs to quantify the distance between points. Thus if K-means were used to cluster words, the words

would have to be transformed into a numeric representation. A naive solution would be to just use the ASCII values of the words. However to perform clustering effectively based on the meaning of the words and relevance to the text as a whole, the numerical value would preserve any underlying relationship between the words and the document overall, which is not possible if ASCII was the one which was used. TF-IDF would be such a method, as it weights the importance of a word to a text, and the frequency of usage.

This combined method of using TF-IDF with K-Means is widely used. It has been used for summarisation of document spaces [19], for classification [10], and specifically for topic detection [38]. Thus, this was considered a suitable approach for topic classification.

4.4 Mahalanobis distance

Using the Euclidean distance for word clusters often presents a challenge, as the clusters may not end up being spherical in nature [30], thus a different metric can be used, the Mahalanobis distance. This can also be used to calculate the distance between 2 points. This is calculated by measuring the number of standard deviations between points a and b, and can generalise to higher dimensions via the variance/covariance matrix.

The Mahalanobis distance would mainly be used after the cluster has been fitted (since the final variance covariance matrix is required), to calculate the distance between a point and the centres of the clusters. The Mahalanobis distance calculation is shown below. $\mathbf{X}$ is a matrix of $\boldsymbol{n}$ vectors, such that $\boldsymbol{x_i}$ is a vector which represents a point in p-dimensional space. $\boldsymbol{X_c}$ represents the matrix where the points have been centred, and thus the variance covariance matrix can be calculated by matrix multiplying the transpose of the centred matrix and the centred matrix and dividing by the n -1, where n is the number of points. $\bar{\boldsymbol{x}}$ refers to a vector, in this case a centroid, which represents a point in p-dimensional space. Thus the Mahalanobis distance is the square root of the difference between the point $\boldsymbol{x_i}$ and the cluster centre $\bar{\boldsymbol{x}}$, multiplied by the inverse of the variance covariance matrix, multiplied by the transpose of the difference between $\boldsymbol{x_i}$ and the cluster centre.

$$\text{Covariance Matrix: } \boldsymbol{C_x} = \frac{\mathbf{1}}{\boldsymbol{n-1}}(\boldsymbol{X_c})^{\boldsymbol{T}}(\boldsymbol{X_c})$$

$$\text{Mahalanobis Distance: } MD_i = \sqrt{(\boldsymbol{x_i} - \bar{\boldsymbol{x}})\boldsymbol{C_x^{-1}}(\boldsymbol{x_i} - \bar{\boldsymbol{x}})^{\boldsymbol{T}}}$$

The Mahalanobis distance has been used with clustering algorithms such as K-Means [24] [13], and is frequently used in distributions of clusters which are either elliptical [26] or non normal distributions [34]. Since the distribution of points around the K-Means clusters is likely to be unknown if TF-IDF is used to numerically transform the words, this was seen as a suitable metric to use for evaluating the clusters the algorithm would create.

5 Scope

GDELT is a large resource which accumulates vast quantities of data each day, thus by necessity, the scope of the dissertation has to be restricted. There were several initial approaches, firstly looking at one day's data on GDELT, and one article to test out and try the topic modelling approaches. The headlines were used from the source URLs instead of the entire articles. This is due to the fact that it would be infeasible to retrieve each article from the source URL and process in an automated fashion, as each article and site would have a completely different HTML structure, and automating the parsing would be beyond the scope of this dissertation. Furthermore, some articles may be beyond paywalls, which would be difficult to reach with conventional web scraping methods.

The final data which was tested was the time period between the 1st of March 2020 and the 30th of April 2020 (inclusive). To try to curate the topic model, the GDELT filtering system was used, to only look at USA/China related media.

The stock market which was used was the Dow Jones Index. This was used as it was thought that this would give the best potential correlation between the GDELT data and stock change.

It should be noted that GDELT is a database which is updated on a daily basis with any missing backdated information being added. Thus the March and April data used in this dissertation was collected on the 11th of June 2020, and the May June data used was collated on the 24th of July 2020.

5.1 Code and Language

The results and code for this dissertation were all written in the python programming language. This was chosen as there were many existing libraries for the algorithms used in this dissertation, along with significant support for building classification models.

6 Experiments

The modelling and experiments of this dissertation were split into two parts. The first part explored different topic modelling approaches, in an aim to build and use a curated topic model. This topic model was trained on event data filtered in GDELT where the events were only those concerning the USA or China. This topic model could then be used on more data from GDELT, to filter out new events where the headlines from the URLs which match the topics used in the topic model. The same GDELT filtered USA/China dataset was used to model the binary stock shift with the Goldstein Scale and Average Tone values.

These two approaches were run concurrently, with the eventual aim being that they would be merged if the initial experiments were successful, and then tested on another dataset, to get a measure of how accurate and useful this method might be.

6.1 Data Acquisition

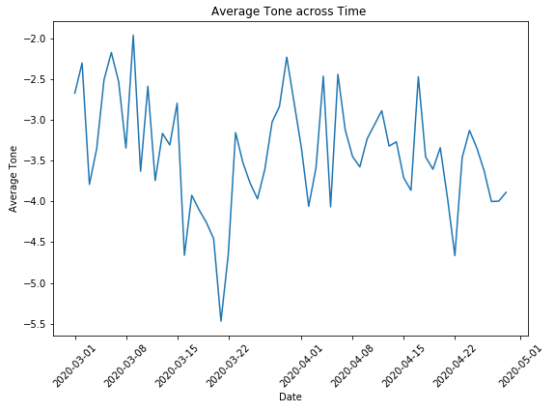
There were 2 main sources of data acquisition, firstly data was taken from the GDELT Events table. This data was for the use of building the topic model, and getting the average tone and Goldstein scale values for the events on specific days. The second dataset to be used was data for the Dow Jones Industrial average.

6.1.1 GDELT

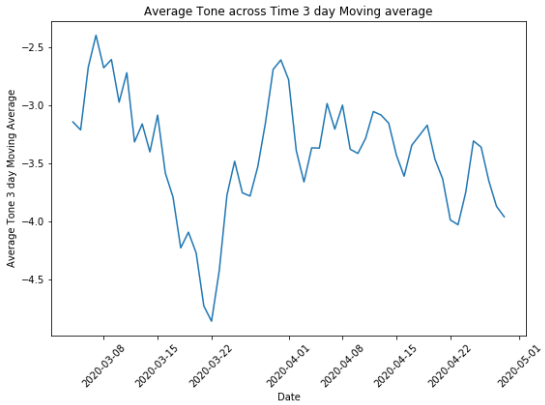
There were three sections of GDELT data. This was acquired using two different procedures, firstly a python package was used to retrieve the GDELT events table data for a single day. The day was chosen arbitrarily and was the of 1st of November 2019. The Google Cloud platform, and Google Big Query was used to collect the larger USA China Datasets. These datasets consisted of the events data for the months of March-April 2020, and May-June 2020 respectively. There were two main reasons for selecting this subset of data. Firstly, there was significant news in terms of amount and in terms of being of a geopolitical nature being generated between the USA and China during this period, and the stock market remained volatile, thus it seemed most promising to find a relationship, and build a curated topic model. Secondly, the events table data had to be restricted, as there would be too much data to process on the machine the models were run on.

After the data was collected, There was a significant chunk of processing required for the GDELT data. Firstly the URLs had to be parsed to extract headlines from the URL text. The parsing was performed in an automated fashion and assumed the headlines would be the last item present in a URL. Thus for each URL, the last item in the URL was retrieved, and split into the words. This is where the parsing was not entirely effective, as the last item may not be split-able, or the last item may contain words other than just the headline. Some preliminary structure based parsing was done to eliminate non-headline URLs where they could be stripped out in an automated fashion. However, there was no feasible method to fix this in an automated fashion due to the large size and varied style of URLs in the dataset within the scope, thus there remained some parsing errors in the dataset. After the headlines were parsed, the words in the headlines were stripped and made into a corpus of documents, to be used in the topic modelling.

The Goldstein scale and average tone values were aggregated across the dates for the March-April interval, and the results shown in Figures 2 and 3. Figure 2 shows the daily Average tone of the events across days. This data is fairly chaotic, and thus the moving average for 3 days was plotted and is shown in Figure 2b. It is immediately apparent that the Average Tone is always negative, which is perhaps to be expected given the frosty relationship between the USA and China during the months of March and April.



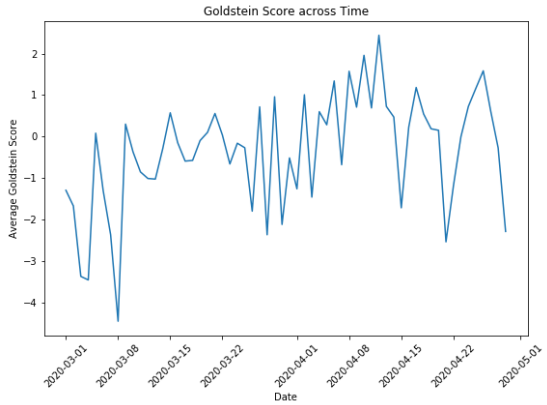
(a) GDELT Average Tone



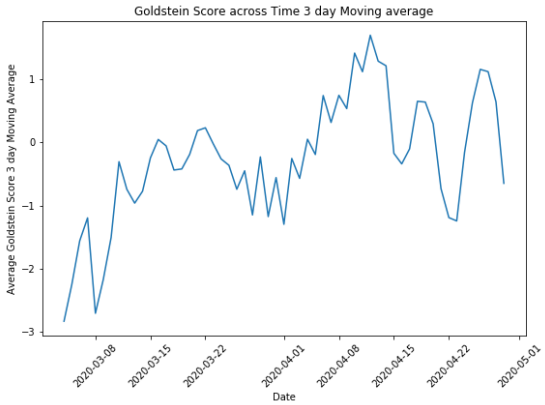
(b) GDELT Average Tone 3 Day Moving Average

Figure 2: GDELT Average Tone over time and Moving Average

Figure 3 shows the daily average of the Goldstein Scale of events across time, and as with the average tone plots, the 3 day moving average is shown in Figure 3b. The Goldstein Score, like the Average tone, appears to be chaotic, though in this case it shifts frequently from positive to negative. The moving average shows that there appears to be a peak halfway through the time period, followed by more fluctuation.



(a) GDELT Goldstein Scale



(b) GDELT Goldstein Scale 3 Day Moving Average

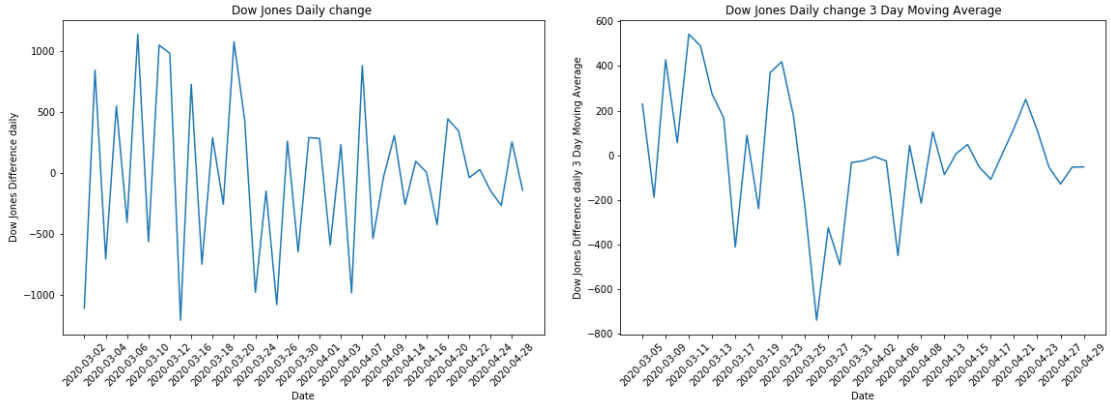
Figure 3: GDELT Goldstein Scale over time and Moving Average

6.1.2 Stock Market Data

Stock Market data was acquired from the yahoo! finance United Kingdom website. For the stock market data, the model would predict whether the stock would rise or fall over an entire day. Since there were potentially lots of events occurring on each day, the average was taken of the Goldstein scale and the Average Tone for all events on a specific day.

Figure 4 shows the Dow difference between open and close prices on a daily basis. This data is fairly chaotic, with the difference jumping between positive and negative frequently and without an apparent pattern, thus a moving average of 3 days was plotted, which also remains slightly chaotic, but more trends appear. The market difference appears to be getting negatively larger towards the middle of the time period before recovering towards 0. The stock prices appeared to take a dive throughout March, which can be explained by the USA China trade flareups, and more

pertinently, the impact of Covid-19. Examining the moving average, the daily difference in the Dow Jones follows a somewhat similar path as the average tone, with a, specifically with a large trough present in the time interval. This trough was present a few days after the average tone trough, as the average tone trough occurred on the 22nd of March, whereas the daily difference trough was on the 24th of the month.



(a) Dow Jones Daily Difference

(b) Dow Jone Moving Average

Figure 4: Dow Jones Daily Difference and Daily Difference Moving Average

6.2 Topic Modelling

There were several topic models run using several data sources. Firstly, an LDA model was run on a single day's GDELT event data. This was the whole day's worth of data with no country filter applied. This was to examine whether topic modelling was possible for the headlines data.

The LDA models were applied on the USA/China GDELT data within the interval of the 1st of March to the 30th of April.

After running the LDA models, the K-Means cluster models were run on the USA/China GDELT data with TF-IDF preprocessing.

6.2.1 Preprocessing

All of topic models worked on the basis of using a corpus for the data, This meant that the URLs were parsed, and then headlines taken from them. Each headline was used as a document, and split into the requisite words to be used. In this process the text was cleaned to remove punctuation and special characters, and formatted to build a corpus full of documents, with each document being a list of words from one URL.

6.2.2 TF-IDF

The TF-IDF vectoriser present in the sklearn package was used to calculate the vector encodings, and was used as the basis of the cluster analysis for K-Means clustering. It was run across the USA/China filtered data to get an initial understanding of which terms would be most important across the dataset. This was then used to numerically transform the headlines for the K-Means algorithm.

6.2.3 LDA

To run the LDA model, the gensim package in the python programming language was used [31]. Several LDA models were fitted, each with a different number of topics, to examine which would be the best number of topics for the data. After the data was fitted, several phrases were inserted in the LDA model to see the probability of those words being in that topic or not.

It was found that LDA models were only able to return a probability value for words which were already in the corpus. This meant that any news headlines which contained *any* word not originally in corpus would not be able to be accurately classified by an LDA model. This meant that this type of model could not be used to separate and filter irrelevant headlines from those headlines which were focused on the headlines in the model.

6.2.4 K-Means

K-means clustering was also fitted on the data. The package used was the K-Means clustering algorithm already implemented in the sklearn python package. Several cluster numbers were tried, from fitting with 2 centroids to fitting with 4 centroids. The Mahalanobis distance and the euclidean distance was calculated for each clusters' points. Furthermore, sample phrases were tested against K-Means models, to see if the cluster was meaningfully able to recognise the difference between words related to the clusters and noise in the algorithms. From an implementation perspective, a random seed was set for each of the clusters so the centres would start at the same point, and thus the results would be reproducible.

To test the efficacy of the models with regards to filtering data, the Mahalanobis distance was calculated for each of the clusters for a selection of phrases. These phrases were varied in length and content. Some of the phrases were relevant to the main topics/content of the topic models, namely related to the pandemic, or the USA or China, such as the phrase 'coronavirus hits remote utah'. Other phrases were completely unrelated to the topic models such as 'the nikkei closes 90 points down' and 'aboriginal peoples australia complain'. A full list of phrases is shown in appendix A. If the cluster models were accurate in filtering information, the phrases closer to the topics at hand would be physically closer to the clusters in p-dimensional space and thus have smaller Mahalanobis distances than phrases which weren't close to the topics in hand.

6.3 Stock Modelling

The main aim for this type of modelling was to predict whether the stock shifted up or down over the course of a day. Each day's difference was calculated between the opening and closing prices, and either a 1 or a 0 was used to represent the stock market going up and down. For the scope of the project, and similar to other modelling approaches, it was decided to only predict either the market going up or down. There were several models which were tried, detailed in 6.3.2. In all cases the response variable being predicted is the change in the stock price over a day. This takes a value of 1/0 for either a fall or rise. This is denoted Y at time t ($\boldsymbol{Y_t}$), where t is a day, and ordered 1 to T, for the time interval used for training the model (April 1st to May 31st 2020). The predictors used were the Average Tone and Goldstein scale values for each day, along with differing amounts of predictors for previous day's data.

All of these models were tested and ranked on their accuracy. This was a percentage of the number of days that the model predicted whether the stock went up or down correctly. The 'best' model was selected to be the one which had the highest accuracy values. The best model was then used to predict the stock shifts for the days between the 1st of May and the 30th of June, on the same set of data, i.e. the USA China GDELT filtered data.

Another measure of the validity/accuracy of the best model was comparing it to a reference model. This reference model would be of the same model type as the final model, but only use the previous day's stock shift, and *not* the Average Tone or the Goldstein Scale values for as a variable used for prediction. The aim with this reference model was to see if the Average Tone and/or the Goldstein Scale actually provided any extra information and insight into the stock market to a model compared to a model which just used the previous day's stock shift.

6.3.1 Preprocessing

There was substantial preprocessing required for the data, first of all the stock market does not open on weekends or other holidays, however news and events do. Thus, the news over weekends was collated and averaged into the Friday figures. This meant that the prediction data had to the shifted, to ensure that information from the future was not being used to predict the data.

The next issue to consider whilst preprocessing the data was the issue of lag modelling. It is reasonable to expect that if there is an underlying relationship between the Goldstein Score/Average Tone and the stock price, a specific day's stock price changes would not restricted to just the previous day's news, but instead could be impacted by news over the previous several days. Thus the average scores and the Goldstein scales would have to be smoothed using several moving window calculations.

6.3.2 Models

All of the models tried were classification models, as by reducing the stock price changes into up or down, it became binary data. There were 4 main models which were tried, a Naive Bayes model, a Random Forests Model, a Support Vector Machine, and a simple Logistic Regression. Furthermore, a simple multi layer neural network was also tried on the data to see if there was any underlying representation that could be learned. All of the models were trained using 3 predictors, the Goldstein Score and the Average Tone of each day's events, and the previous day's stock change.

Initially simple models were fitted with no averaging or smoothing of the predictors performed. However, it was assumed that an individual event could affect prices for several days, thus the Goldstein scale and Average tone predictors were smoothed using several different kinds of moving windows. There were 3 different moving window types tried, firstly a simple average was taken where each day was weighted evenly. The second method of moving window was an exponential decay window, and similarly a half Gaussian moving window. The moving windows were all spread over three days, as it was felt that this would provide the best balance at weighting past events evenly, without diluting the effect of the most recent news.

To test whether the previous N days' stock shift affected a specific day, three extra predictors were added, these predictors, along with the previous day's change, represented the stock shift for each of the previous four days. These were used alongside the unsmoothed Average Tone and Goldstein Scale values to predict the stock prices. These models are henceforth referred to as the Manually Lagged Models.

All of these models were trained and tested on the March-April data, and the results are reported in chapter 7.

To both show how this sort of predictive model would be used in the real world, and to test the 'best' model's predictive capacity in 'real' time, the 'best' model was used to build a simple portfolio. For time reasons this only consisted of a single stock, where shares were bought or sold on a daily basis based on the model's confidence in the following day's stock shift. This was to mimic the behaviour of being used in real time, where there would only be new news for the

current day, and based on the previous news and each day's news, the next day's prediction would be made. The stock chosen was Apple Inc. as this stock had the highest weighting in the Dow Jones index. The Apple historical data values were also procured from the finance Yahoo United Kingdom website.

This was a very simple portfolio test, where each day's prediction triggered buying and selling of the stocks, i.e. stock was only ever held when the confidence of market shift was not above a threshold, otherwise stock would be bought and sold daily. There were several hyperparameters involved, namely these were the initial capital, initial number of shares, the confidence threshold required to trigger buying or selling stocks. There were also two additional hyperparameters, these variables controlled the fraction of capital used to buy stock, or the buy-factor, and the fraction of stock to sell at any given point, sell-factor. These variables were constant for every run, so for one test on the May to June time period, every time the confidence in the prediction for the next day was greater than the confidence threshold, and the market was predicted to increase, the buy-factor amount of the capital at that point was used to 'purchase' stock. The reverse happened when the market was projected to decrease, with the sell-factor fraction of the stock being held being sold. Since geopolitical events can occur after the market close which are factored in by the predictive model, the stock close price was used as the reference price for buying and selling stock.

For the May to June time period, the number of shares, and the capital was tracked, along with the total asset value, which was the total capital plus the monetary value of those shares at the closing price each day. Three different strategies were used, one which would maximise the shares by using more capital to buy shares, one which would maximise capital by minimising the number of shares held at any point, and one which would be in the middle of those two strategies.

The 'best' predictive model's portfolio was also compared to a portfolio which used the reference model, and the results are shown in chapter 7.

7 Results

The results are included for the Topic Modelling experiments and the Stock Modelling experiments.

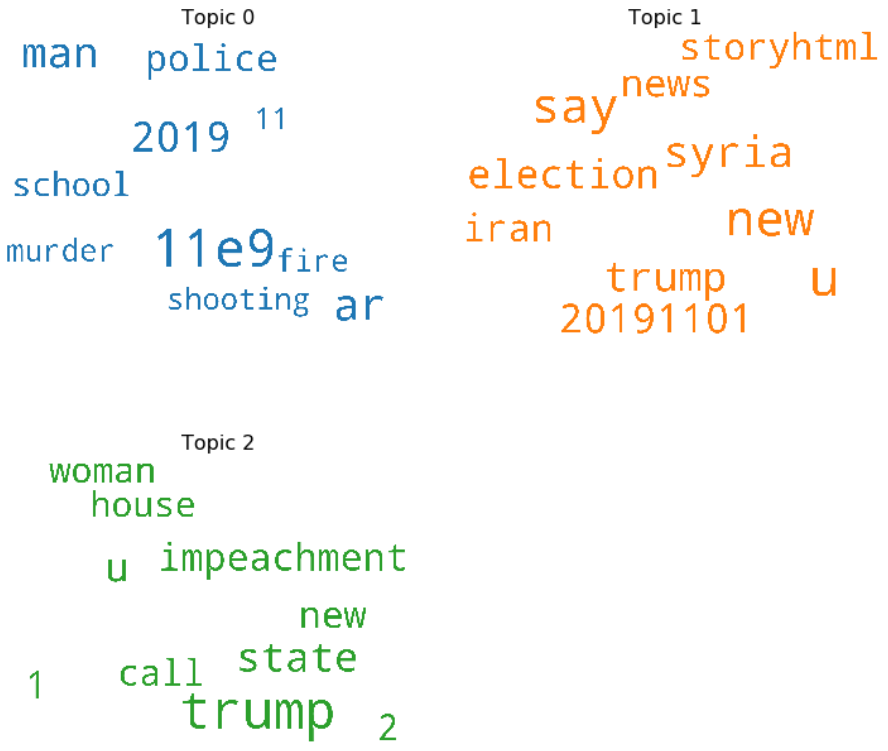
7.1 Topic Modelling

For the topic modelling, the results are included for both LDA examples, the initial example on one day's worth of GDELT events data and the USA/China March-April data. The results also include the TF-IDF top values for the USA/China data, and the K-Means results on the same dataset.

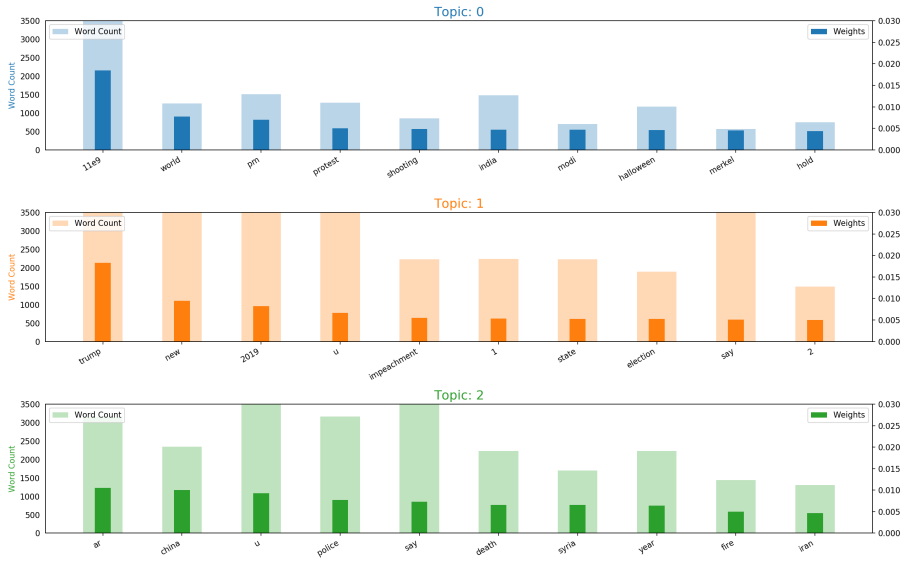
7.1.1 Initial LDA

The LDA models are included for running on the one day event data and the USA/China dataset. The day was the 1st of November 2019. This was done as a reference point before running the LDA models on the specific USA/China chosen data.

There were two different topic models tried. Firstly one model with 3 topics was tried and then one model with 5 topics. It was unknown how many topics would be present in the data, thus two different values were chosen, one with a smaller number of topics (3), and a slightly larger number of topics (5). The two values were tested to see if a smaller or a larger number of topics would better represent any topics present in the data. The top words from each model are shown in word cloud format in Figures 5a and 6a. Alongside the word cloud, for each topics, the weight and the word count of the top words was also calculated and plotted.



(a) Word Cloud 3 Topics

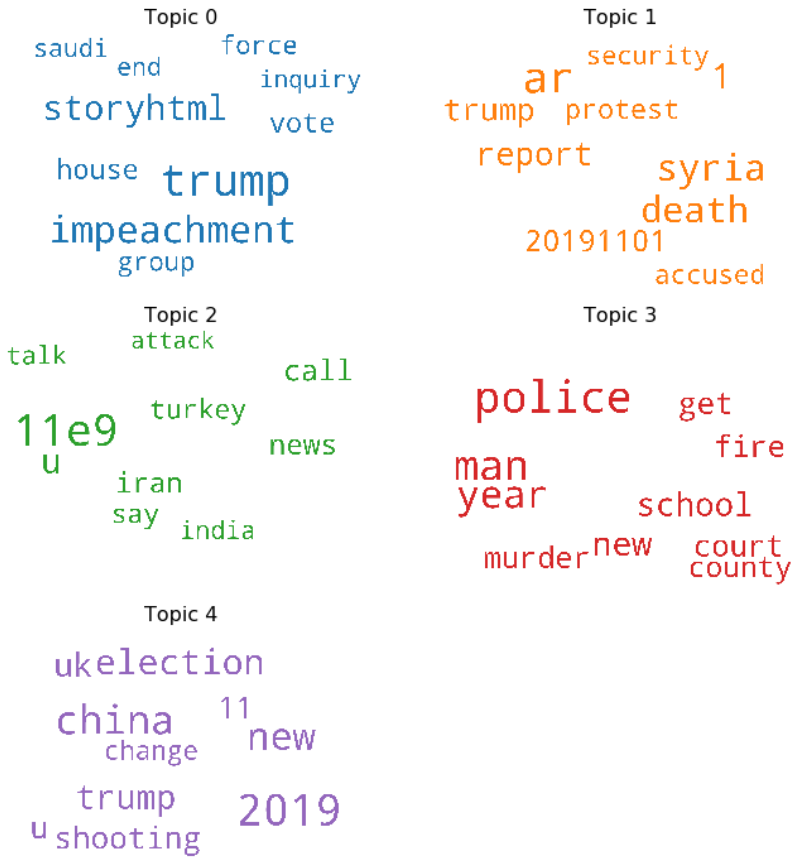


(b) Word Weights 3 topics

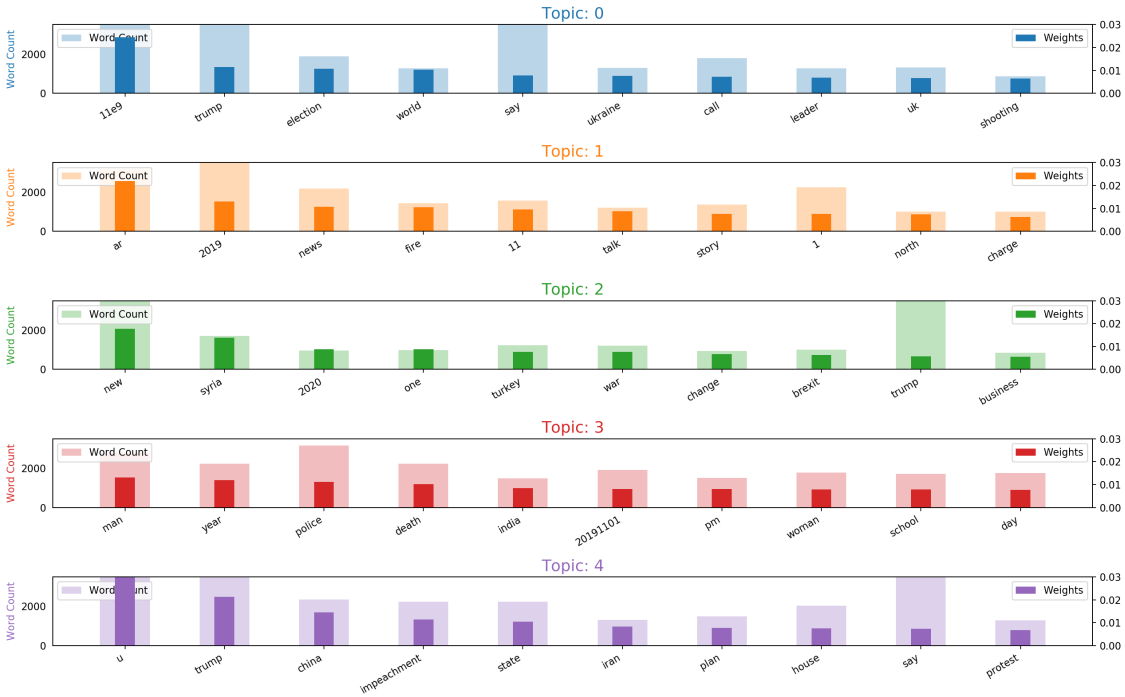
Figure 5: Single Day Word Clouds and Word Weights for 3 topics

Examining the word clouds for the model with three topics, there are not any clear topics which are apparent. Topic 2 could vaguely be about the impeachment process for Donald Trump, Topic 0 appears to be focused on police brutality as a topic, and Topic 1 could broadly be referred to in terms of international news. Examining the word importances, the main theme across topics is that the word count is not the same as the word importance, in that some words have much higher

occurrences, but lower weights and vice versa. This is perhaps to be expected, as the word count and importance do not have to be linked.



(a) Word Cloud 5 Topics



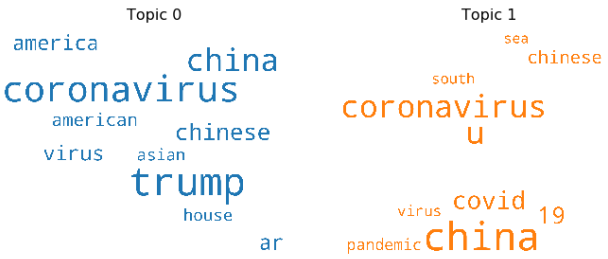
(b) Word Weights 5 topics

Figure 6: Single Day Word Clouds and Word Weights for 5 topics

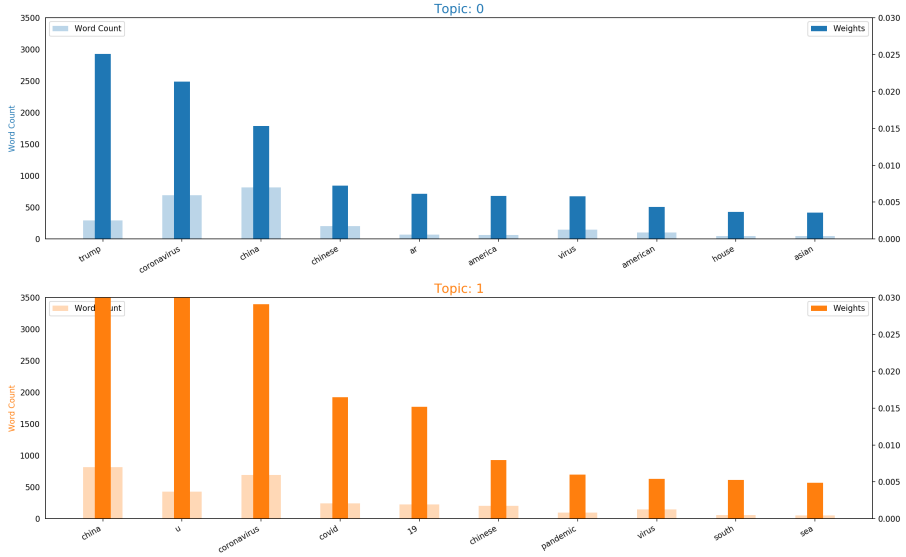
Examining the 5 topic model, the topics are slightly closer together, its very difficult to find a central topic for each topic, Topic 3 could potentially be about police and court information, but aside from that there does not appear to any coherency otherwise, with words like Trump being in multiple topics and international countries spread across topics. The word importance and weight plot also does not reveal anything new, like the previous model, the word's count is not related to the importance and perhaps expectedly, the words in the topics are not related to each other.

7.1.2 USA/China Data

A similar procedure was used for the USA/China data, but models with 2, 3, and 4 topics each were tried, as it was not completely clear from the initial LDA model whether a smaller or larger amount of topics would represent the data better. The word clouds of the results and the subsequent word importances to each topic are shown in Figures 7, 8, and 9.



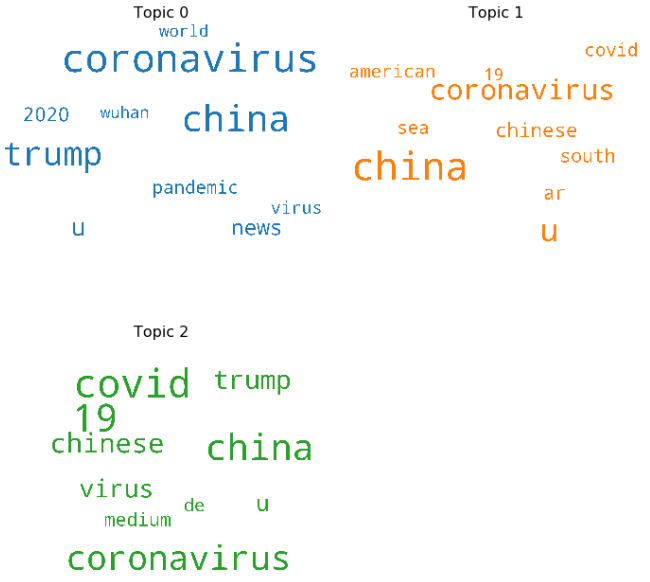
(a) USA/China Word Cloud 2 Topics



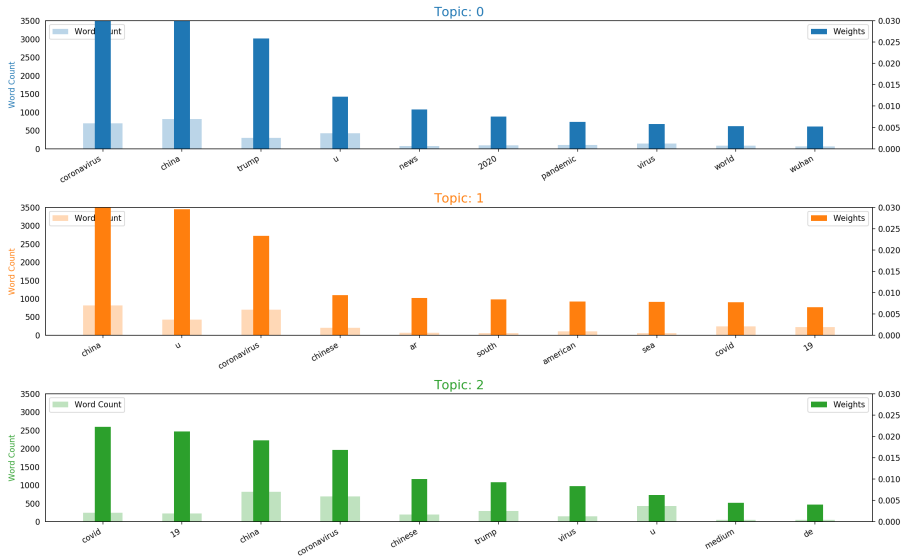
(b) USA/China Word Weights 2 topics

Figure 7: USA/China Word Clouds for 2 topics

Examining the first LDA model which had 2 topics on the USA China data, the themes are very similar. Words related to the pandemic, and words such as China and Trump appear in both topics, which suggest the model has not been effective in differentiating between the topics effectively. Looking at the word weights in Figure 7b, for all of the words, the weights are all higher than the word counts. The highest word weights by topic are Trump, Coronavirus, China for Topic 0 and China, Coronavirus, and 'U' for Topic 1 . 'U' appears to be an issue with the URL parsing.



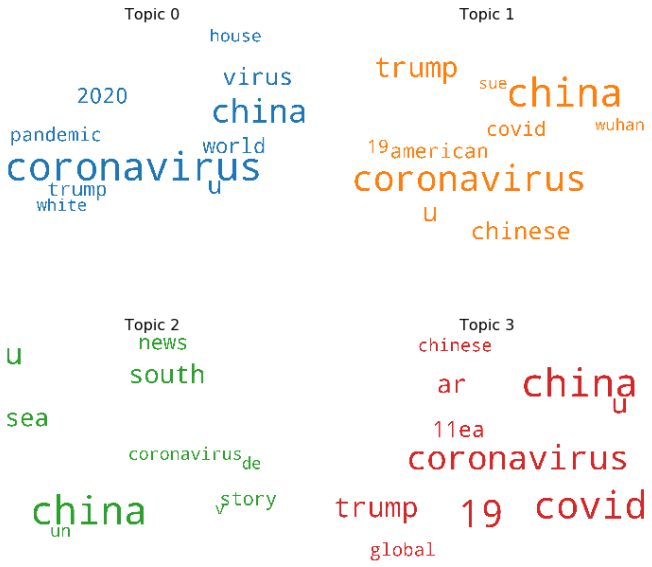
(a) USA/China Word Cloud 3 Topics



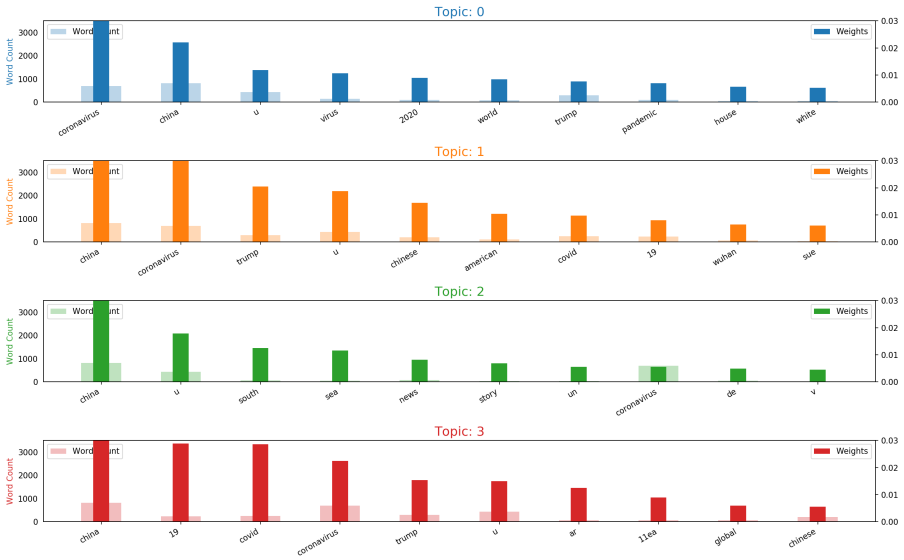
(b) USA/China Word Weights 3 topics

Figure 8: USA/China Word Clouds for 3 topics

Looking the the three topic model, the topics are even closer together than the two topic model. The main words as before appear in all of the topics, suggesting the topic model hasn't separated any topics well. The word weights are also similar to the two topic model. One of the differences between the two topic and the three topic model is the 3rd topic weights are noticeably smaller than the first and second topics.



(a) USA/China Word Cloud 4 Topics



(b) USA/China Word Weights 4 topics

Figure 9: USA/China Word Clouds for 4 topics

The 4 topic model behaves in a similar manner to the previous 2 topic models. the main words are split across the 4 topics with no real distinction between the topics present.

7.1.3 TF-IDF

The top TF-IDF terms are shown for the USA China data across the corpus in Figure 10. This data was achieved by taking the average of the TF-IDF values across the entire dataset. The average values are slightly lower across the corpus compared to individual documents, as there will be a number of documents where specific words have TF-IDF values of 0, as those words do not exist in those documents. These 0 TF-IDF values are included in the mean calculations as

the parsing process for URLs is not perfect in retrieving headlines. If the 0 TF-IDF values were excluded from the averaging calculations, the top words by TF-IDF would be non-representative of the whole corpus. This is due to some 'words' present only in a small number of documents, i.e. those 'words' which represent parsing errors. These words have high TF-IDF values in the documents they exist in. Thus if their 0 TF-IDF values from all other documents are not included in the averaging calculation, these words would have artificially high TF-IDF values across all of the documents and not be representative of the actual top words in the corpus.

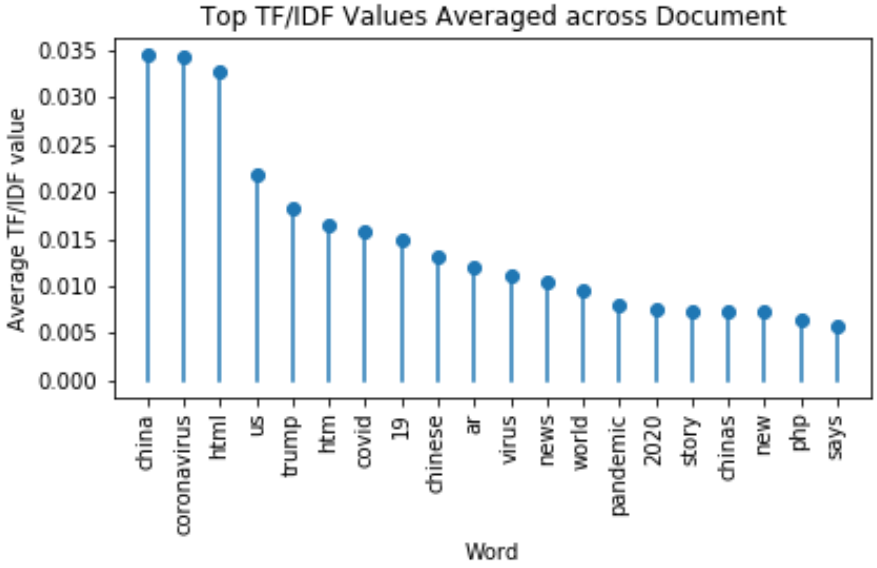


Figure 10: Plot of the top 15 words which used TF-IDF in the USA/China specific data

Perhaps as expected, the most important and often occurring words are China and coronavirus. Amongst the top words are also US, and Trump, along with variations of China and covid-19 and references to the pandemic. This is most interesting as a result, as something which no one had heard of prior to January/February dominated the news in March and April.

One of the other main words which pops up is html. This is most likely as a result of the fact that most of the URLs end with '.htm' or '.html', and during the parsing html gets treated as a commonly occurring word. It was not removed as there could be legitimate stories which have the word in them.

For the top values, the distribution of the TF-IDF values across documents was calculated, excluding the 0 values. This is shown in Figure 11. The distributions are different to each other, but both follow a similar pattern in having a centre of the distribution be around a TF-IDF value of around 0.25. One of the notable exceptions to this is the word 'ar', which is another error as a result of parsing. World and news both have a spike later on, but that is most likely due to the smaller sample size.

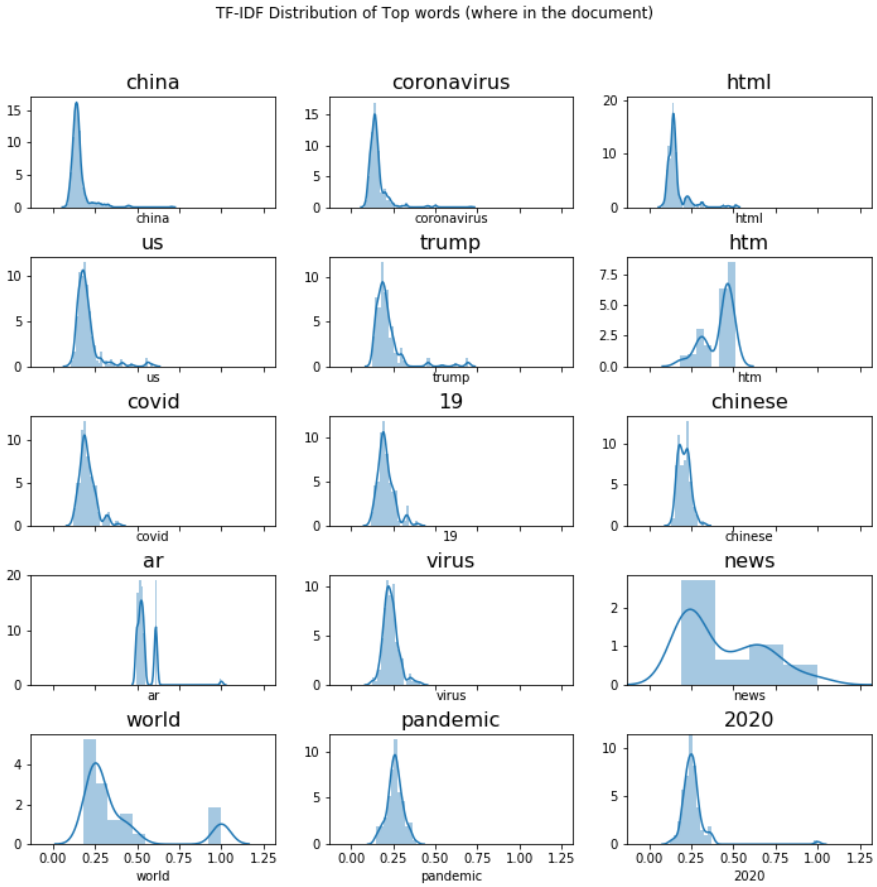


Figure 11: Distribution of the TF-IDF values across documents of the top 15 words (excluding documents where the TF-IDF value was 0)

7.1.4 K-Means

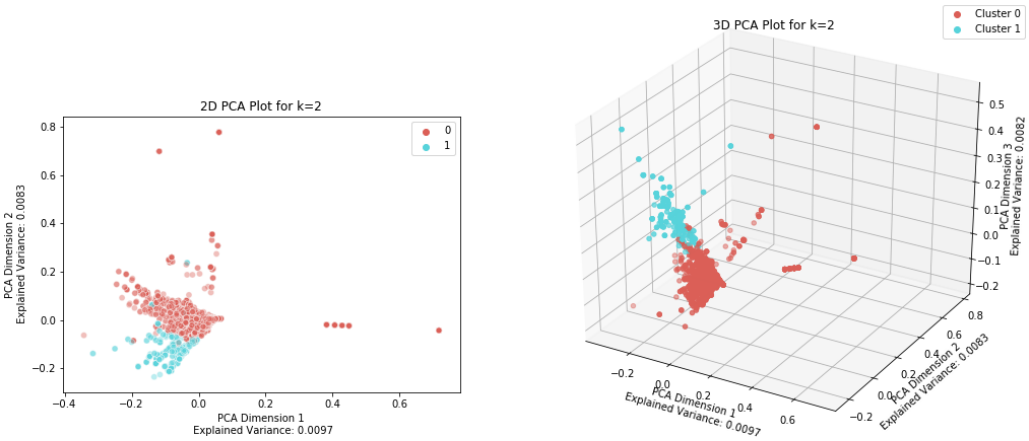
Similar to the LDA models, the K-means was tried with 2, 3, and 4 clusters. A range of cluster numbers were tried, as after running the initial LDA, and the USA/China LDA models, it was not evidently clear which 'K' would be most appropriate. The word cloud of the top words are shown in Figures 12, 22, and 26 respectively. Alongside the word clouds, the clusters were decomposed into 2 and 3 dimensions by both Principal Component Analysis (PCA) and T-distributed Stochastic Neighbour Embedding (T-SNE) [22]. This was to see whether the clusters had been successful in separating the data into well defined clusters at any level. This is shown for the different number of clusters in Figures 13, 23, and 27 respectively. Furthermore, the Mahalanobis and Euclidean distances were plotted for all of the points associated with a cluster for all of the clusters present. This is shown in Figures 14, 24, and 28 respectively.

To compare the efficacy of clusters in being able to filter information, the Mahalanobis distance from centre of clusters for variety of phrases was calculated, and plotted in Figures 15, 25, and 29, alongside the distribution of points for each cluster. The full phrase list is in Appendix A. For all of the Mahalanobis distances, the phrases were several orders of magnitude out from the distribution of points for each cluster, so the Mahalanobis distance results were logged before being plotted.

7.1.4.1 Clustering with k=2

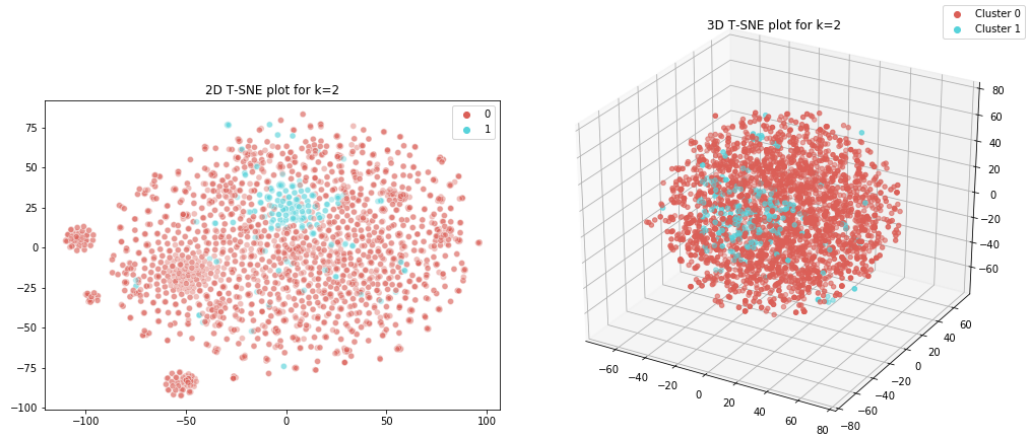
Figure 12: Word Cloud for k=2 clusters

Examining the word cloud created for a two cluster model, a similar picture appears as with the LDA models. There are several words which are close to both of the clusters, which, in a similar fashion to the LDA model, mean that any topics present are not being differentiated. This is evident in Figure 13, in which both T-SNE and PCA in both 2 and 3 dimensions show that the clusters are not distinct from each other. Interestingly, both the 2d and 3D PCA plots, Figures 13a and 13b, appear to show defined boundaries between clusters, however, the cluster definitions aren't what a human would draw. The main caveat with the PCA plots is that the proportion of explained variance in the dimensions selected is extremely poor, for both the 2 dimensional and 3 dimensional decomposition, each dimension captures less than 0.01 of the explained variance. This means that the clusters may still be accurately portraying the data in another dimension/combination of dimensions, but it is not visible in this dimension.



(a) 2D PCA k=2

(b) 3D PCA k=2

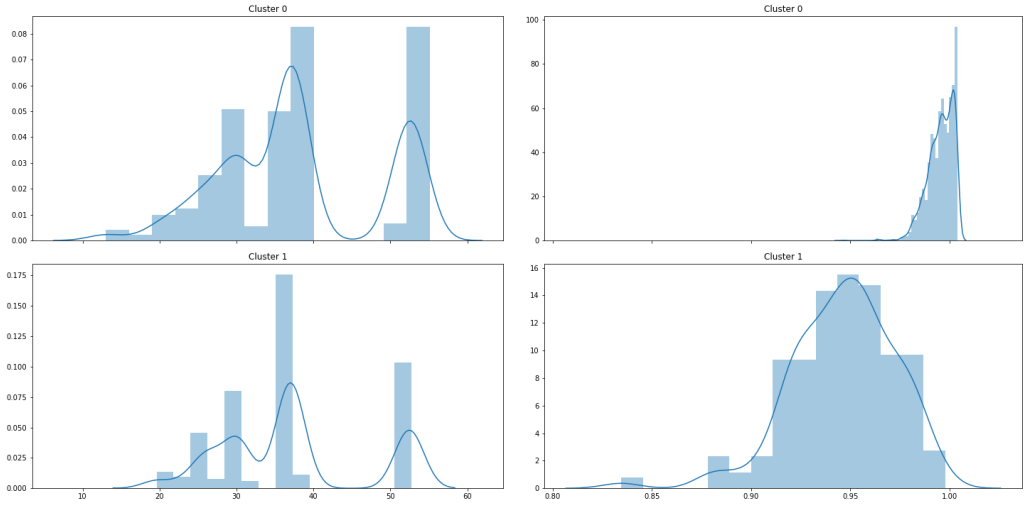


(c) 2D T-SNE k=2

(d) 3D T-SNE k=2

Figure 13: Decompositions of the clusters in 2 and 3 dimensions using PCA and T-SNE for k=2

Examining the Mahalanobis distances for the two clusters, there does not appear to be any noticeable pattern, the distributions are slightly bimodal, compared to the Euclidean distances, which appears to be slightly normally distributed shape wise, though normality would not be expected as the 0 distance would be the cluster centre, from which the distances are measured. The majority of points are associated with cluster 0, with the rest going to Cluster 1.



(a) Mahalanobis Distance (b) Euclidean Distances

Figure 14: Cluster Distances (Mahalanobis and Euclidean) for k=2 clusters

Comparing the results plotted on the log scale shown in Figure 15, the first thing which becomes apparent is that the distances for the words appears to be the same between clusters, i.e. all phrases tested are equally far from both cluster 0 and cluster 1, this is perhaps unsurprising as the distance from both clusters is extremely large. Comparing the words themselves, aside from the individual word ‘covid‘, the phrases are fairly close together, with what appears to be two sets of lines. The first lines are the ‘coronavirus hits remote utah‘, and the ‘aboriginal peoples australia complain‘ with the remainder of the headlines making up the remainder of the lines, nearly all of the results are between 27 and 28 in terms of the absolute logged Mahalanobis distance.

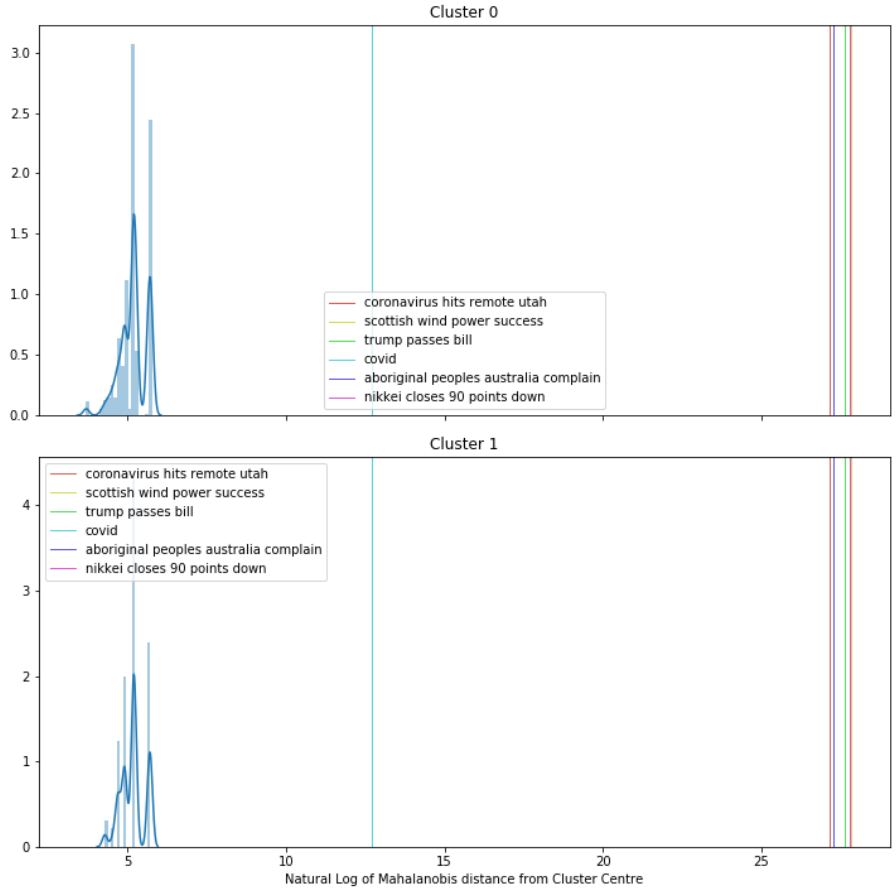


Figure 15: Log of Mahalanobis Distances for Clusters and Selected Phrases for k=2 clusters

7.1.4.2 Clustering when k=3 or k=4

When the number of clusters was increased to 3 or 4, it was found that there was no material change in either the topic word clouds or the decomposition plots. Thus, the word cloud, decomposition plots, and Mahalanobis distance plots for k=3 and k=4 are included in the Appendices B and C respectively. The word clouds for both k=3 and k=4 behaved the same as k=2, where the topics were all very similar, and the decomposition plots suggest the cluster definitions are definitively useful, and the boundaries are not as clear as they were for k=2. There are a few features of note however. For each cluster model tried, there were 2 sets of points away from the main cluster, but when examined, these represented the points associated with the parsing errors from the URLs. The cluster Mahalanobis distances appear to be bimodal with the same range of distances for the points. As expected, the Mahalanobis distance for the selected phrases were similar distances away, with the pattern of distances being similar.

7.2 Modelling Stock Prices

The table of prediction accuracies for the different models is shown in Table 1. This is the percentage of days over the prediction interval of roughly 8 days in March and April that each model was able to predict correctly. Roughly 8 days of the March April dataset were set aside to test the generalisation accuracy of the models on unseen data, and the remainder of the data was used to train the models. Several models had similar accuracy scores, though it should be noted that the predictions were on a very small dataset, so even one or two correct incorrect predictions could

sway the result considerably.

The manually lagged models appeared to be the least best, in these models 5 days worth of previous days were added as 5 features to the data, and no other averaging methodology applied. The Random Forests classifiers appeared to do well with whatever data smoothing style was used. Furthermore, models which applied moving window calculations did well, with on average the exponentially smoothed models doing the best.

With all of that taken into consideration, the best model was picked to be the one which performed exponential smoothing on the data and was a Random Forests Classifier. Exponential smoothing was chosen as all of the models with exponential smoothing generally did better than other types of data smoothing. Similarly, the Random Forests was good across the board, thus the best model picked was one a Random Forests model which performed exponential smoothing.

Thus the reference model was chosen to be a Random Forests Classifier model, which only used the previous day's binary stock shift value as a predictor.

Classifier	Accuracy
Reference Classifier with Previous Day	0.750
Naive Bayes	0.750
Logistic Regression	0.750
Averaged Random Forests	0.750
Gaussian Averaged Random Forests	0.750
Exponential Naive Bayes	0.750
Exponential Logistic Regression	0.750
Exponential Random Forests	0.750
Random Forests	0.625
Support Vector Machine	0.625
Averaged Naive Bayes	0.625
Averaged Logistic Regression	0.625
Averaged Support Vector Machine	0.625
Gaussian Averaged Naive Bayes	0.625
Gaussian Averaged Logistic Regression	0.625
Gaussian Averaged Support Vector Machine	0.625
Exponential Support Vector Machine	0.625
Manually 4 day lagged Random Forests	0.625
Neural Network	0.500
Manually 4 day lagged Naive Bayes	0.500
Manually 4 day lagged Logistic Regression	0.500
Manually 4 day lagged Support Vector Machine	0.375

Table 1: Table of Models and the Accuracy achieved during training

The Exponential Random Forests model's feature importances are shown in Figure 16. This shows how important the features are to in the model in predicting the Dow Jones Index. The Average Tone and Goldstein Scale are both similarly rated, however the previous day is rated at around half of the importance of the Goldstein Scale by the model.

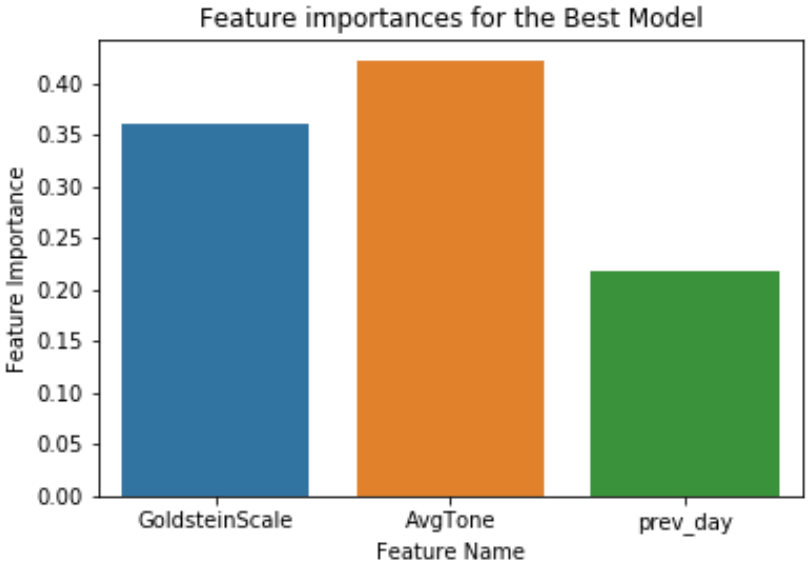


Figure 16: The Feature Importances of the Features in the best model

Comparing the reference model to the exponentially lagged Random Forests Model on May-June Dow Jones data, The reference model was 50% accurate, and the exponentially smoothed random forest achieved an accuracy of 62.5%, which would appear to suggest that the Average Tone and Goldstein Scale values do provide value in terms of predicting shift.

The predictions for the May-June data are shown in figure 17, where the magnitude of the differences has been plotted, alongside whether the best model was correct in predicting a positive or negative shift (blanks are weekends or days when the stock market did not operate). A green value of +1 represents the model predicting correctly on that day, and a red value of -1 represents the model incorrectly predicting the shift for that day. There does not appear to be a time based prediction component, the model is just as likely to predict correctly or incorrectly regardless of the time that it is predicting on. The model does not appear to be very good at predicting for large stretches correctly, it correctly predicts for a day or two, and then has incorrect predictions for a similar length of interval. The exception to this is when halfway through the prediction, the model correctly predicts the spike and the subsequent fall and recovery.

The distribution of magnitude of the positive and negative daily differences of the Stock Index across the May-June Interval is plotted in Figure 18. It is apparent that the model is better at predicting smaller differences in the stock market. The majority of incorrect predictions come from larger negative values closer to -500 points and slightly from the positive jumps. There are significantly fewer incorrect predictions when the market jumps between -250 and +250 points. Interestingly, there is a small bump at a large positive value of +1000, suggesting the model was good at predicting large positive values.

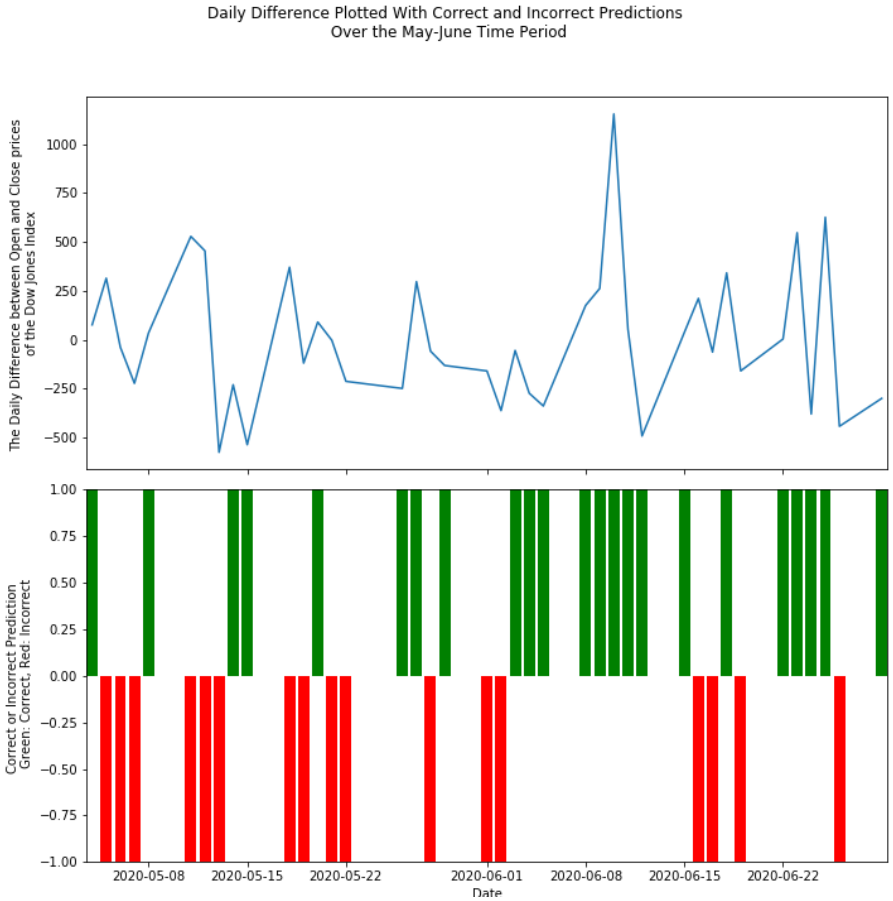


Figure 17: The Dow Jones daily changes in prediction dataset, and whether the model predicted the result correctly or not on a daily basis over the prediction interval of May and June 2020

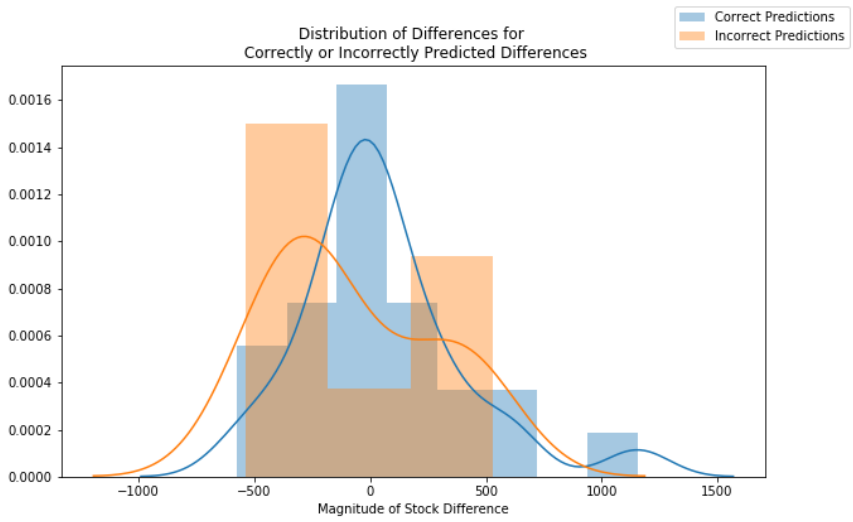


Figure 18: The distribution of the daily differences between open and close prices where the model predicted correctly vs predicted incorrectly

7.2.1 Portfolio Results

3 different portfolio styles were tried, firstly a balanced strategy, where half of the capital is used every time shares are bought, and half of the shares are sold whenever shares are sold. This is shown in Figure 19. The second strategy focused on maximising capital, so only $\frac{1}{5}$ of capital was used to buy shares, and $\frac{4}{5}$ of shares were sold whenever shares were sold. This is shown in Figure 20. The final strategy was a shares maximising strategy where $\frac{4}{5}$ of capital was used to buy shares any time shares were brought, and $\frac{1}{5}$ of shares were sold whenever shares were sold. This is shown in Figure 21. All of these strategies started off with no initial shares, and an initial capital of 2000, and used a prediction sensitivity threshold of 60%. A low threshold was used to maximise the number of decisions the predictive model was taking. Though, this would increase the risk as the model may not be as confident in the prediction.

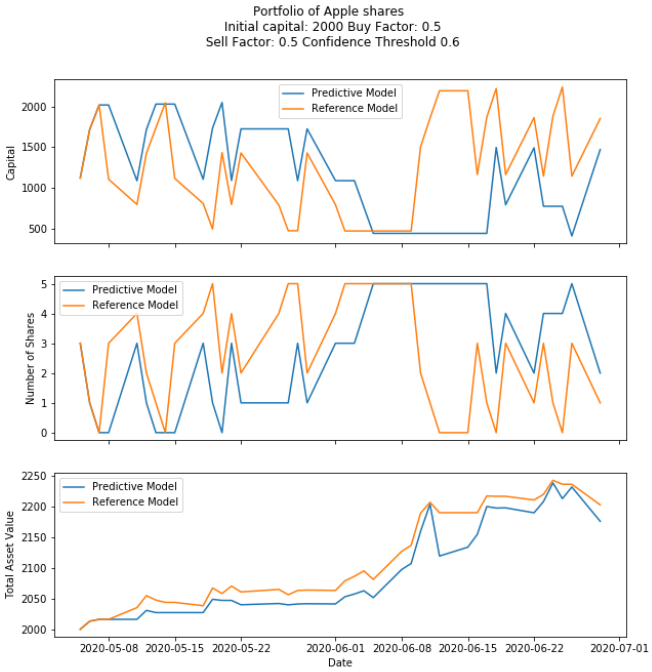


Figure 19: The amount of shares, capital, and Total Asset Value plotted through the time for the first portfolio strategy, the half and half balanced strategy

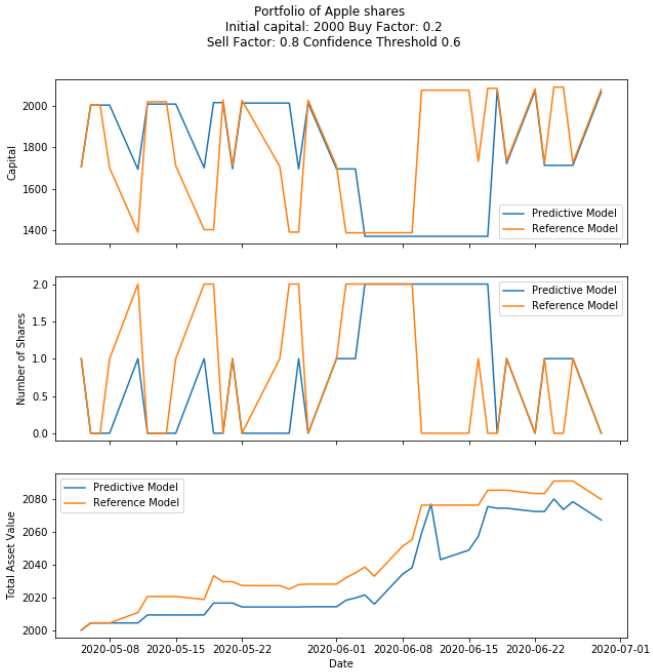


Figure 20: The amount of shares, capital, and Total Asset Value plotted through the time for the second portfolio strategy, the maximising capital strategy

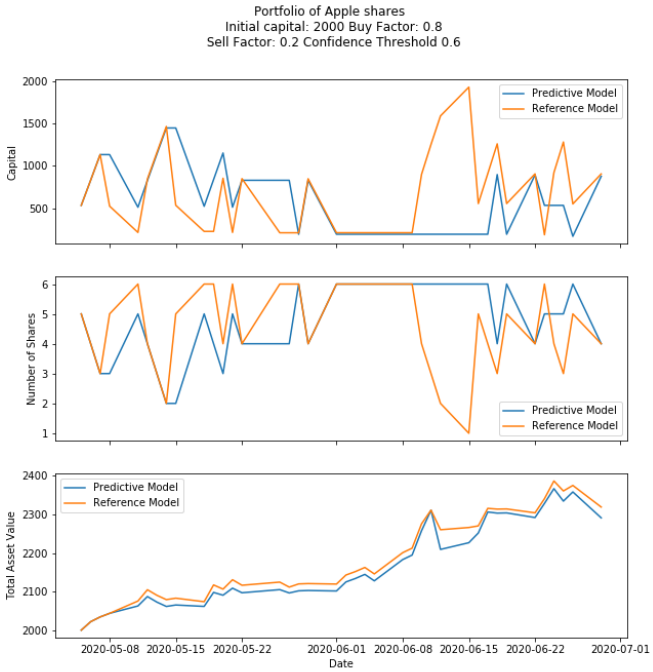


Figure 21: The amount of shares, capital, and Total Asset Value plotted through the time for the third portfolio strategy, the maximising shares strategy

The first thing to be noted in all of the strategies, is that in all cases, the predictive model was slightly worse over the time period compared to the reference model. This is mostly likely because

the predictive model in all cases held on to the stock around the 10th of June, after which the value of the stock decreased considerably and the model was not able to make up the difference within the remaining time period.

The second point to be made is that all of the strategies accrued financial gain, though the model to maximise shares had the highest total asset value at the end of the time period, followed by the half and half strategy, followed by the capital strategy. For the maximising shares strategy, the total asset value at the end was significantly beyond 2200, which represents more than a 10% gain in the market over the initial capital investment. However, by comparison the maximising capital strategy only achieved an asset value of 2060, which is a significantly lower growth. The half capital and half shares strategy was doing well, especially recovering after the drop in the middle of the month, but it fell slightly to record a value of nearly 2200, which still is nearly a 10% gain. However, it should be noted, that for every strategy bar the capital maximising strategy, the majority of the total asset value is represented in stock value and not capital.

8 Discussion

The broader purpose of topic modelling was to build a topic model which would be able to categorise new and upcoming geopolitical risks, and thus be able to filter through only the relevant information. In the context of using GDELT, it would mean filtering out the headlines to only use the tone and Goldstein scale values of events where the headlines were related to the topic doing the filtering. This would mean that the topic model would not only have to be able to curate existing risks adequately, but also be able to catch new emerging risks too.

After performing the cluster analysis, and topic modelling, there were several things which became apparent. Firstly, the topic models are good at finding the main topics. Both the clusters and the LDA topic models were able to find what would be considered the 'main' words in the topic, in that if a human were to look for the most important words, this is what they would be. Furthermore, if the TF-IDF results were examined, the results were surprisingly insightful into the data. The results were mostly coherent and representative of what a human would achieve for the time period of the training data.

However, one of the main concerns for both the LDA model and the clustering algorithm is the lack of separability. In both of the topic modelling attempts it appeared that there was only ever one topic being detected, in that there was no separation between the words such as 'china' and 'coronavirus'. This could perhaps be a result of there not being at least two topics present in the data. However, comparing the single day data and the USA China data, neither set of word clouds strongly show a prevailing topic, though both represent the data reasonably. This in itself is extremely promising, as the corpus and documents themselves were composed of URLs which were brute force parsed, which in turn meant that there was a lot of 'garbage' data in the corpus, the result of parsed URLs where the headline was not coherent or even not there.

The main concern with the topic models is the fact that existing topic models cannot be used effectively in terms of filtering words, phrases, or topics unknown to the historical topic development, and thus cannot detect new topics. One of the biggest issues with LDA models is that the documents themselves have been generated from a specific set of topics, which means that if the word being filtered did not exist in the corpus there is no probability/closeness metric that can generated for each topic. This means that one of the only ways a topic model could be used for new data is if a dictionary was used to generate the corpus along with the news media at hand. This in turn would create two other problems. Firstly, it would disrupt the meaning of the topics as there would be an excess of unrelated words to the documents, this however could theoretically be solved by using extremely large datasets, and significantly larger amounts of processing power. The second issue with this approach is more prohibitive. New words and topics appear constantly in the media. For example, a topic model in November would not have been able to predict the importance of topics associated with Coronavirus. This is especially problematic for geopolitical modelling, as often it would be events appearing out of the blue which cause the most amount of disruption. This problem is not fixable by any amount of data or processing power, as no dataset can ever be complete indefinitely, new words/topics will always appear in the data.

A similar problem exists for K-Means. The K-Means algorithm for words in this instance uses the TF-IDF vectorization to build clusters. This relies upon all of the words existing in corpus in a manner where the most common important words appear most frequently. This means inherently the weighting is designed for a fully complete corpus, and not one which is being used to analyse new information. One of the main differences between the K-Means algorithm and the LDA algorithm is that for K-Means it is possible to get a quantitative distance measure for a word not originally in the corpus to a particular cluster centre, as TF-IDF weightings can be calculated by inserting an offset for words which are not in the corpus. However, this means that the distances to the cluster centres for words not in the corpus are much less useful. There is no way for an existing cluster model to tell apart a new rising topic from information which is irrelevant or noise as both would have the same calculated TF-IDF value with regards to the corpus the TF-IDF is calculated from. For example a pre-covid trained cluster model, words related to coronavirus or covid would

be categorised as noise or irrelevant instead of an upcoming rising topic.

This is shown in Figures 15, 25, and 29. The results were plotted on a log scale, as distances between points for words and clusters can be significantly larger than the distances within clusters. Firstly, it becomes apparent that it does not matter which cluster is chosen in terms of calculating Mahalanobis distance for a new word or phrase, because all of the clusters are so close together, and the vectorised words are so far away. This means that when the log of the distance is taken, plotting the results mean that for each of the clusters the distance is exactly the same.

Furthermore, some phrases do appear closer to any and all of the clusters than other phrases. This seems promising, as specifically the 'coronavirus hits remote utah' is a relevant headline to the topic being detected, and thus it being closer suggests it might be useful. However, the difference on the log scale between that phrase and a totally irrelevant phrase (aboriginal peoples australia complain) is barely noticeable which furthers the idea that the TF-IDF is transforming relevant and 'garbage' information in the same way. The single word 'covid' is considerably closer, but a headline would rarely consist of only a single word, and more pertinently, in terms of relative distance for points within the cluster it is still orders of magnitude away. These plots also show that the low explained variance in the PCA decomposed dimensions isn't just a problem with the dimension picked, but rather is indicative of the clusters not being able to cluster the data effectively.

As such this was the main reason the topic model was not used for filtering for a secondary dataset and why this approach in this fashion appears to not be useful in fashion for the task being attempted. Moreover, one of the biggest challenges with this work is that the topic models only work on a single word basis. All of the clusters and LDA models are based on single words, and not phrases, or content. This means there would still have to be a large amount of manual work required when using this approach, to ensure that the algorithms do not end up working with incoherence. The principle of garbage in garbage out also applies here, there is a substantial amount of preprocessing required to ensure that you're feeding the algorithm useful information, and even after that it is not guaranteed if the content is capable of making clusters or not.

Looking at the results from the stock prediction models, it is difficult to say with any certainty what the best model is. The models were only trained on 2 months worth of data, which is not sufficient to conclusively identify which model is the 'best', or at least most accurate. However, there are certain features of the models which become apparent, generally weighted models deal well with the data, the manually lagged non-weighted model was certainly the worst off of all of the models which were tried. This does back up the theory that the Average Tone and the Goldstein Score may influence stock prices several days down the line, with the most important day being the last day in the moving window.

Looking at the model structures, the Random Forests classifiers did well with whatever style of smoothing performed on the data, suggesting that a Tree based approach may provide better results for this sort of problem. Examining the results of the 'best' random forests model, it appears the model finds success in predicting upwards, and predicting small changes in the market. However, the model struggles with predicting shock falls in the stock market, which most likely would not be captured in any information available to the model.

This shock also explains the difference in portfolios between the best Random Forests model and the reference model. The best Random Forests model around the middle of June completely missed the crash in the market, and held onto the stock which meant that in all of the strategies tried, the model wouldn't after that be able to catch up to the 'reference' model. Interestingly, the reference model *was* able to capture the fall in the middle of June, as in all three strategies, the model sold the stock, and thus didn't suffer a fall in total asset value. This reference model was only based on the previous day's prices, whereas the 'best' model used the Average Tone and Goldstein Scale values as well. However, the previous day's feature importance was much less for the best model, shown in Figure 16, so any internal pattern that the previous day was able to show to predict the fall, if present, may have been ignored as that feature was deemed less useful by the 'best' model.

One of the other issues with this prediction interval is that the data restricts what the model has learned on. The stock market went through remarkable changes over this period, and for large periods of time, there is no guarantee that this would be representative of regular behaviour on the stock market. Longer term more stable predictions would be more useful for investors looking to invest over the long term, as this strategy will over a longer period of time generally offer better results than short term investing.

It should also be noted that any analysis of potential monetary gain which might arise from using a model would have to be compared against stock market long term growth, as for the model to be fully useful, it would mean that it would be able to make gains *beyond* the normal growth of the market. Interestingly, all of the portfolio strategies were able to make money in the time period which extrapolated could be comparable to the 9.5% annual growth, and in one of the portfolio strategies, namely the stock maximising strategy, the total asset value at the end of the prediction period was in fact greater than the 9.5% annual growth in just a two month period. This suggests that these models are in fact useful in terms of predicting the market, and could provide genuine monetary gain.

9 Conclusions

There are many geopolitical events in the world, which affect markets. These are often represented and reported to people in various news sources. To be able to find and predict the impact such events would have on markets, these events and news sources would have to be treated mathematically. This would involve identifying the events determining the tone and impact of the news associated with these events. Then, to see how these events affect stock markets, build models on historical data linking markets to the event-tone and then automate the process such that it can be used as part of an advisory system for investors/shareholders in markets.

This was distilled down to two features, firstly to be able to capture and cluster these events occurring in the news via topic models, and then use classical machine learning algorithms to use the quantitative values of the Tone and Goldstein Scale and predict shifts in the stock market.

This dissertation aimed to use the GDELT database to build topic models from news headlines. If that were successful to aim was to use the topic model as a filter to collect relevant news which contained topics similar to the topic model. Then the eventual aim was to use the Conflict Cooperation Goldstein scale and the Average tone of the articles which covered these events as a geopolitical risk measure to predict changes in the Dow Jones index.

There were several approaches to calculating geopolitical risk in the past which work from manually calculating occurrences of key words in news articles. This dissertation used a pre-existing Conflict Cooperation scale from events as a measure of geopolitical risk, with the main filtering occurring by choosing which events to factor in based on the headlines of those events. The original attempt for topic modelling was to use a LDA model on restricted data to attempt to separate topics.

It was found that this type of model was unable to provide a quantitative value for unseen information, as all information which could be classified would have to exist in the corpus when the topic model was created, which would mean that it cannot detect new topics occurring. This meant that this approach would not be usable.

Thus to try to gain a more quantitative approach, the K-Means algorithm was used with TF-IDF on the words, which would firstly build clusters of words in an attempt to collate the individual topics present into those clusters. Then the objective was to use the Mahalanobis distance to filter out news and unseen headlines which are irrelevant to the topics in the cluster by calculating the distance to the cluster centres for the unseen headlines. The Mahalanobis distance was used as it provides a standardised measurement of the distance between, which isn't going to be affected by the shape of the cluster.

However the main issue with this was that the distance metric for unseen headlines and news was not useful. The TF-IDF method meant that the algorithm placed new information the same as noise and irrelevant data. Thus this too would not be able to be used as it would not be able to distinguish new topics, nor was it able to provide distance calculations accurately for filtering purposes.

Since neither of the topic modelling approaches appeared to work effectively, the modelling was done on the GDELT data filtered via Google Big Query in GDELT itself, as opposed to filtered through the topic models. The models used were classification models which aimed to predict stock shift up or down. These models appeared to be reasonably accurate, with the best of them being more accurate on a separate set of data as compared to a reference model which just used the previous day's data. The best of these models, when tested in a real world portfolio example, was even able to increase the portfolio value over the predictive period. Only a few model types and strategies were explored in this dissertation, there is definitely the potential in terms of further work to explore other different types of models.

Over the course of this dissertation, several topic modelling algorithms were built, but they ran into the problem of the topics and clusters being too close together, and thus neither the topics nor the clusters were distinct. More pertinently the issue of cluster and topic models being designed to work on a complete dataset, and not providing accurate results for unseen data was the main reason why this approach was thought to not work in this fashion. These algorithms might work if the data was more carefully selected with regards to TF-IDF values, but it does not appear to be useful in the current form.

With regards to the stock predictive models, there always remains more scope for trying different types of models which may present more accurate results, but the results of the experiments appear to suggest that the Goldstein score and the Average tone might both be useful in terms of predicting on the stock market. Across the initial simple portfolio, the investors would be making money on the stock from the predictive models created, which suggests that this approach is feasible and viable for use in the real world.

9.1 Further Work

This dissertation was an attempt into seeing if the Goldstein scale alongside the Average Tone from the GDELT database can be used as a measure of geopolitical risk and calculated against the curated parts of the Stock Market. There is a large amount of further work which can be explored from this, in all aspects, from the topic modelling, to the modelling of Average Tone against the Goldstein Scale.

One of the most apparent extensions of this work would be to try these methods on more data. GDELT is a large resource, and a vast amount of data is available to be explored and tried, from data from multiple other countries, to using more historical long term data.

Furthermore, from these experiments alone it appears unlikely that topic modelling can be used in the way that it was initially thought in this dissertation. Even if the distance metrics are relatively accurate for new words and phrases, the topic models by design will inherently not be able to detect the presence of new topics, which would be a key aspect of this.

Another extension which could be performed is to use a method which works on more than just one word at a time. Both the TF-IDF K-Means and LDA model approaches work on a single word basis, which may not be enough to sufficiently separate the topics, thus an approach which takes into consideration phrases or bigrams may allow for more context to be explored which in turn may lead to better separability within the topics.

One of the other main features which could be explored from a more modelling perspective could be an attempt to predict the stock shift in a more fine grained manner, this work only aimed to predict any changes in stock, and as a result this does not take magnitude of change into consideration, a large stock fall is the same as a small stock fall, but investor reactions may be different if it is only a minor fall in the market, thus being able to predict small, medium, or large shifts may become useful in the long term. Another factor which was not modelled was building autoregressive style lagging for the stock prices themselves, along with the lagged models for the Average Tone and the Goldstein Scale.

9.2 Final Remarks

I feel this dissertation was able to explore several topic modelling avenues with respect to geopolitical news, and was able to show why this approach would not work with respect to news or topic filtering. However this dissertation was able to show that stock predicting is feasible from the

geopolitical and tone information provided by GDELT. Even over a simple portfolio, this approach and this style of modelling appears to be profitable, and thus useful to investors in the real world, and this should be explored further.

Appendix A Phrases to Test K-Means

The phrases with which the K-Means algorithm was tested are shown below:

— coronavirus hits remote utah

— scottish wind power success

— trump passes bill

— covid

— aboriginal peoples australia complain

— nikkei closes 90 points down

Appendix B K-Means Clustering k=3

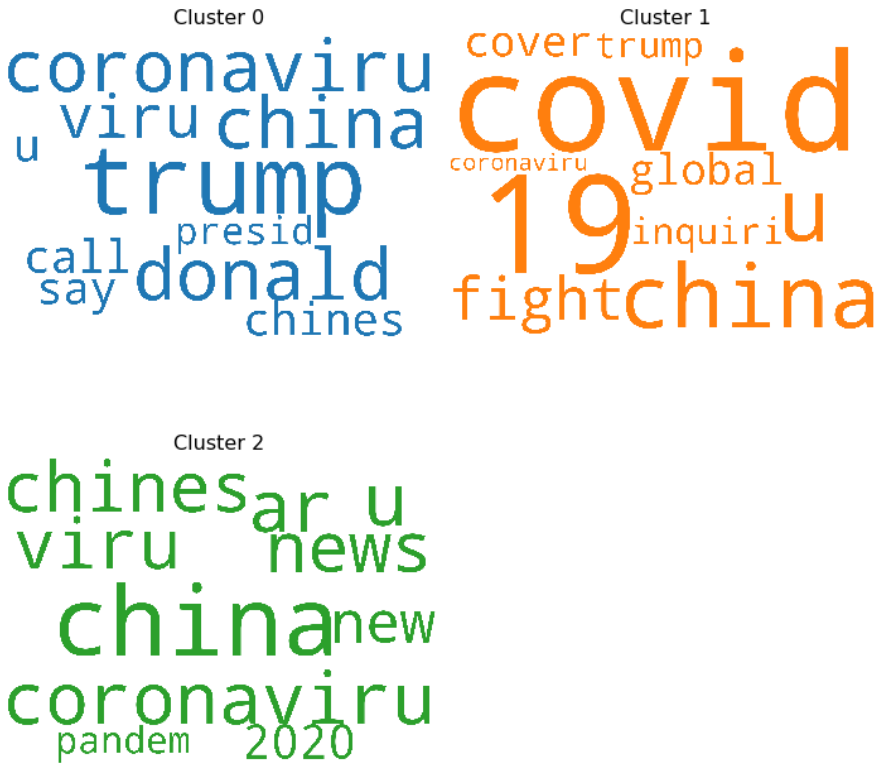
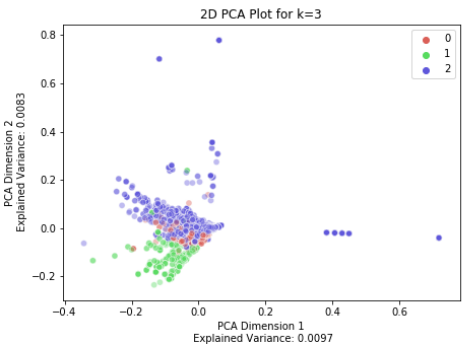
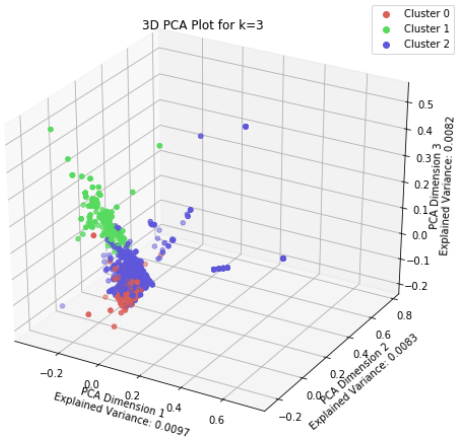


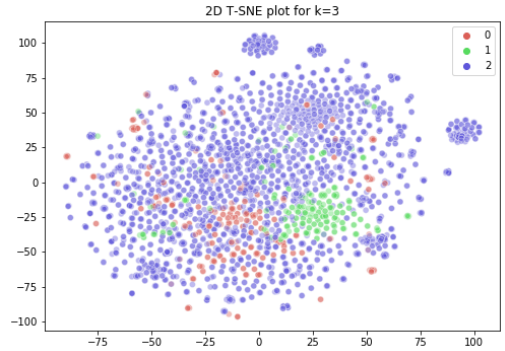
Figure 22: Word Cloud for k=3 clusters



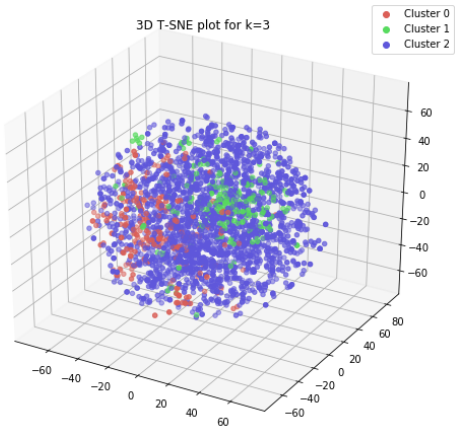
(a) 2D PCA k=3



(b) 3D PCA k=3

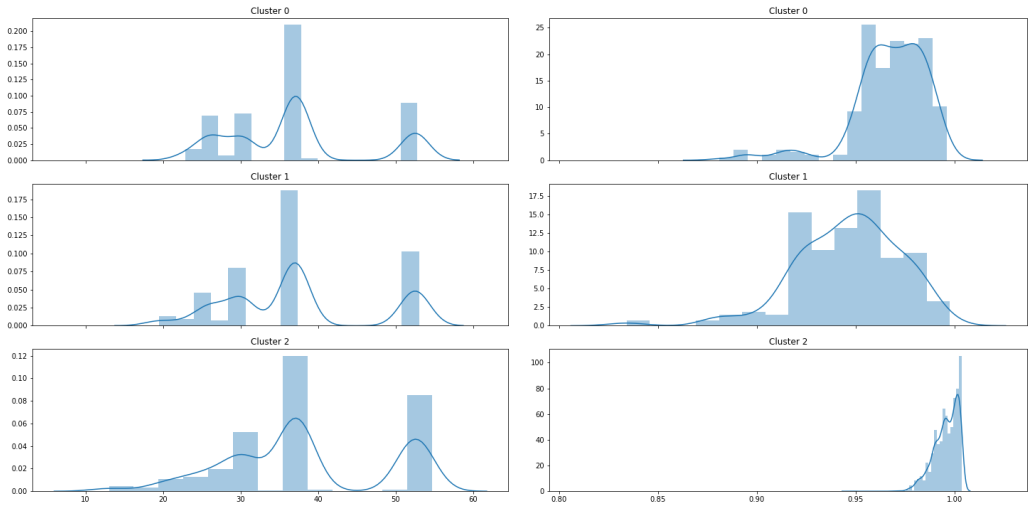


(c) 2D T-SNE k=3



(d) 3D T-SNE k=3

Figure 23: Decompositions of the clusters in 2 and 3 dimensions using PCA and T-SNE for k=3



(a) Mahalanobis Distance

(b) Euclidean Distances

Figure 24: Cluster Distances (Mahalanobis and Euclidean) for k=3 clusters

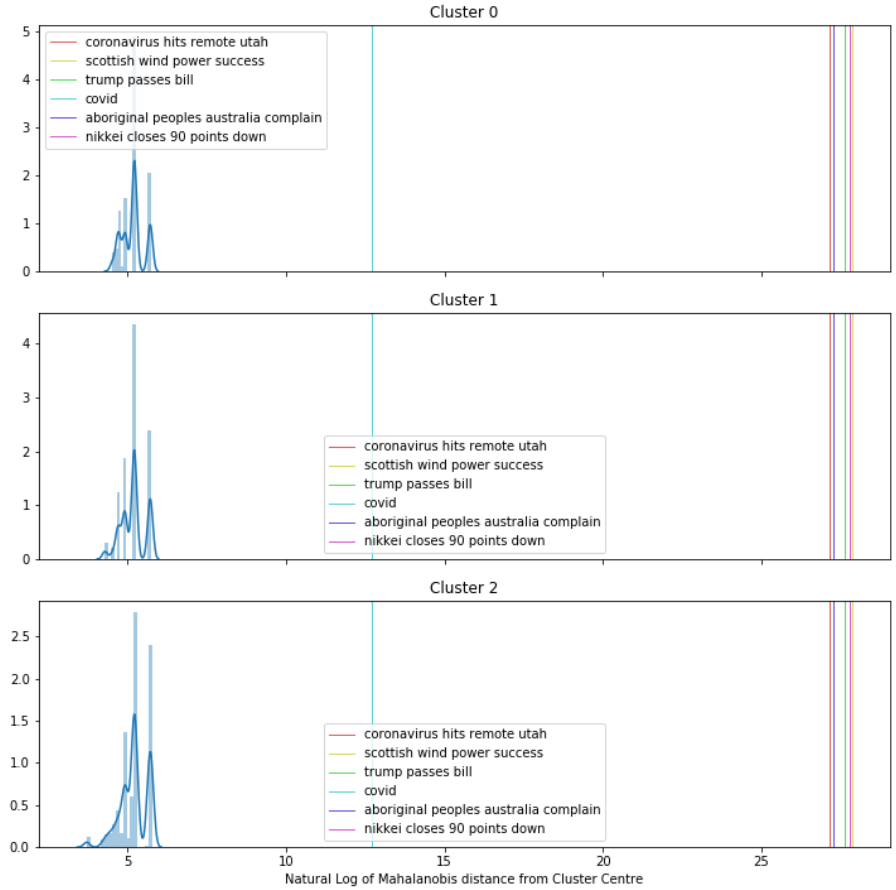


Figure 25: Log of Mahalanobis Distances for Clusters and Selected Phrases for k=3 clusters

Appendix C K-Means Clustering k=4

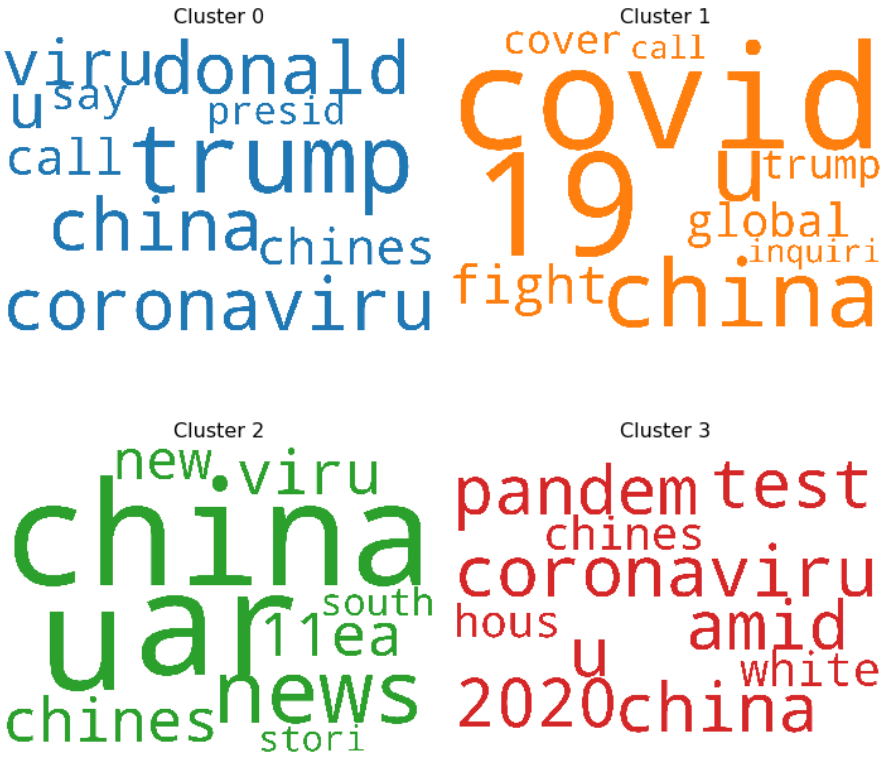
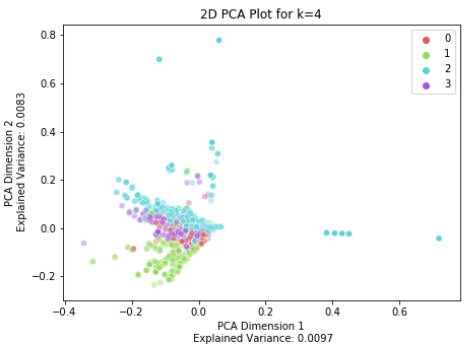
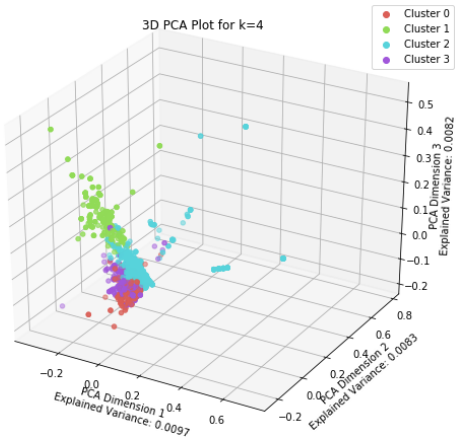


Figure 26: Word Cloud for k=4 clusters

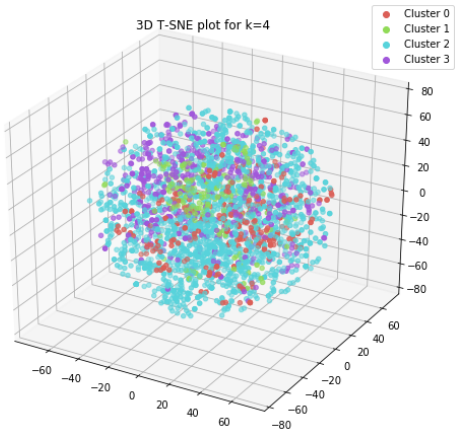


(a) 2D PCA k=4



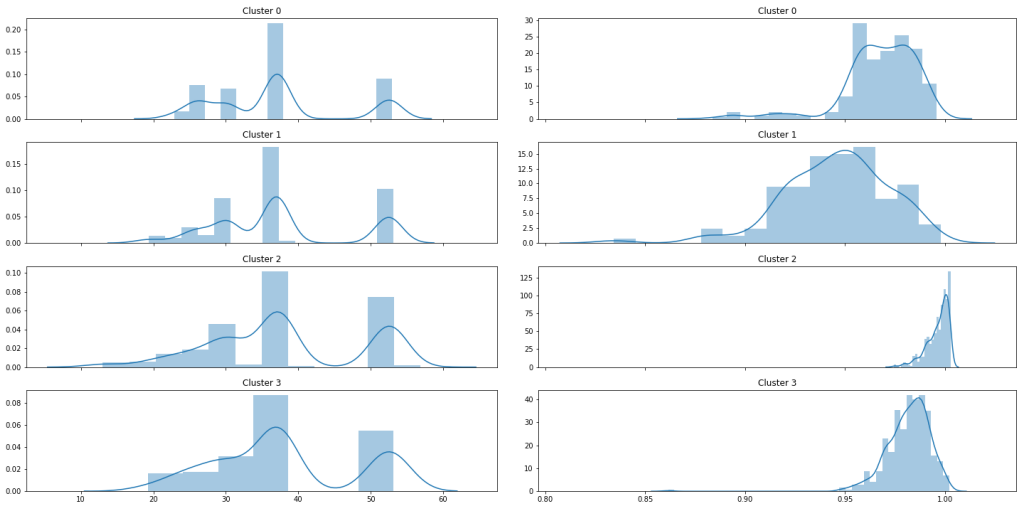
(b) 3D PCA k=4

(c) 2D T-SNE k=4



(d) 3D T-SNE k=4

Figure 27: Decompositions of the clusters in 2 and 3 dimensions using PCA and T-SNE for k=4



(a) Mahalanobis Distance

(b) Euclidean Distances

Figure 28: Cluster Distances (Mahalanobis and Euclidean) for k=4 clusters

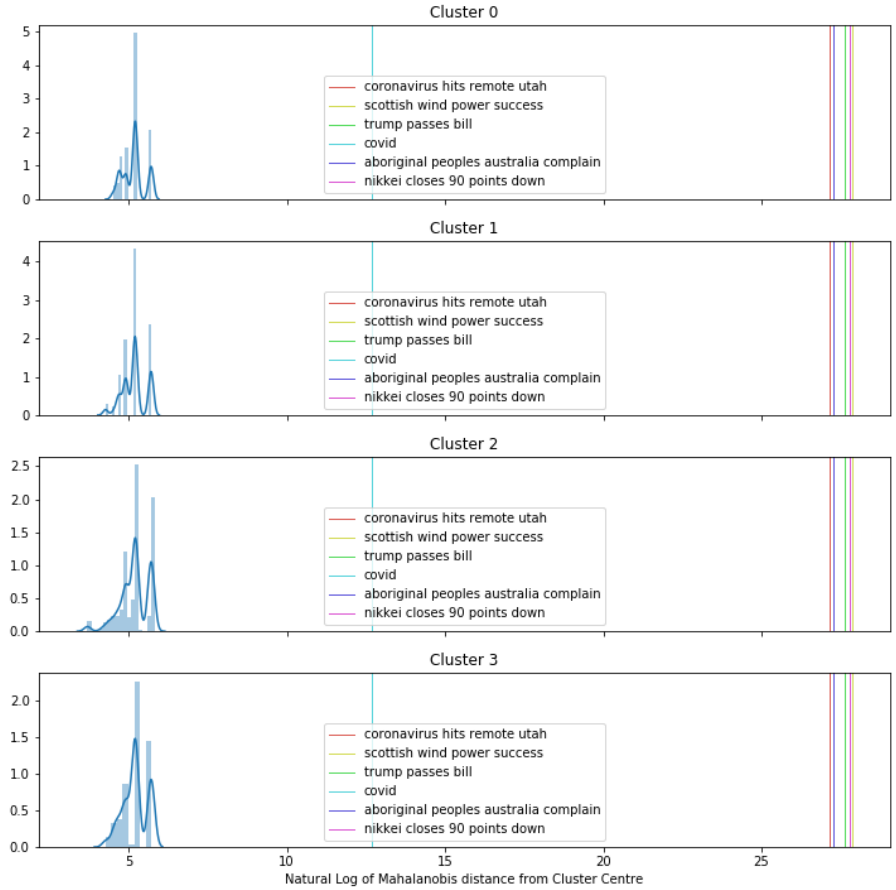


Figure 29: Log of Mahalanobis Distances for Clusters and Selected Phrases for k=4 clusters

References

[1] Algorithm for computing lda. https://medium.com/@jonathan_hui/machine-learning-latent-dirichlet-allocation-lda-1d9d148f13a4.

[2] Posterior probability for computing lda. https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158.

[3] S and p annualised returns. https://www.macrotrends.net/2526/sp-500-historical-annual-returns.

[4] Alamro, R., McCarren, A., and Al-Rasheed, A. Predicting saudi stock market index by incorporating gdelt using multivariate time series modelling. In *International Conference on Computing* (2019), Springer, pp. 317–328.

[5] Arias, M., Arratia, A., and Xuriguera, R. Forecasting with twitter data. *ACM Transactions on Intelligent Systems and Technology (TIST) 5*, 1 (2014), 1–24.

[6] Ariyo, A. A., Adewumi, A. O., and Ayo, C. K. Stock price prediction using the arima model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation* (2014), IEEE, pp. 106–112.

[7] Beel, J., Gipp, B., Langer, S., and Breitinger, C. paper recommender systems: a literature survey. *International Journal on Digital Libraries 17*, 4 (2016), 305–338.

[8] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research 3*, Jan (2003), 993–1022.

[9] Bollen, J., Mao, H., and Zeng, X. Twitter mood predicts the stock market. *Journal of computational science 2*, 1 (2011), 1–8.

[10] Buana, P. W., Jannet, S., Putra, I., et al. Combination of k-nearest neighbor and k-means based on term re-weighting for classify indonesian news. *International Journal of Computer Applications 50*, 11 (2012), 37–42.

[11] Caldara, D., and Iacoviello, M. Measuring geopolitical risk. *FRB International Finance Discussion Paper*, 1222 (2018).

[12] Cao, L. Support vector machines experts for time series forecasting. *Neurocomputing 51* (2003), 321–339.

[13] Cerioli, A. K-means cluster analysis and mahalanobis metrics: a problematic match or an overlooked opportunity. *Statistica Applicata 17*, 1 (2005), 61–73.

[14] Egeli, B., et al. Stock market prediction using artificial neural networks. *Decision Support Systems 22* (2003), 171–185.

[15] Fama, E. F. *Efficient market hypothesis.* PhD thesis, Ph. D. dissertation, University of Chicago, Graduate School of Business, 1960.

[16] Feuerriegel, S., Ratku, A., and Neumann, D. Analysis of how underlying topics in financial news affect stock prices using latent dirichlet allocation. In *2016 49th Hawaii International Conference on System Sciences (HICSS)* (2016), IEEE, pp. 1072–1081.

[17] Goldstein, J. S. A conflict-cooperation scale for weis events data. *Journal of Conflict Resolution 36*, 2 (1992), 369–385.

[18] Khaidem, L., Saha, S., and Dey, S. R. Predicting the direction of stock market prices using random forest. *arXiv preprint arXiv:1605.00003* (2016).

[19] Khan, R., Qian, Y., and Naeem, S. Extractive based text summarization using k-means and tf-idf. *International Journal of Information Engineering & Electronic Business 11*, 3 (2019).

[20] Khedr, A. E., Yaseen, N., et al. Predicting stock market behavior using data mining technique and news sentiment analysis. *International Journal of Intelligent Systems and Applications 9*, 7 (2017), 22.

[21] Kullback, S., and Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics 22*, 1 (1951), 79–86.

[22] Maaten, L. v. d., and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research 9*, Nov (2008), 2579–2605.

[23] Malkiel, B. G. The efficient market hypothesis and its critics. *Journal of economic perspectives 17*, 1 (2003), 59–82.

[24] Melnykov, I., and Melnykov, V. On k-means algorithm with the use of mahalanobis distances. *Statistics & Probability Letters 84* (2014), 88–95.

[25] Memari, M. *Predicting the Stock Market Using News Sentiment Analysis*. Southern Illinois University at Carbondale, 2017.

[26] Mitchell, A. F., and Krzanowski, W. J. The mahalanobis distance and elliptic distributions. *Biometrika 72*, 2 (1985), 464–467.

[27] Nguyen, T. H., Shirai, K., and Velcin, J. Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications 42*, 24 (2015), 9603–9611.

[28] Philip, S., Shola, P., and Ovye, A. Application of content-based approach in research paper recommendation system for a digital library. *International Journal of Advanced Computer Science and Applications 5*, 10 (2014).

[29] Qiao, F., Li, P., Zhang, X., Ding, Z., Cheng, J., and Wang, H. Predicting social unrest events with hidden markov models using gdelt. *Discrete Dynamics in Nature and Society 2017* (2017).

[30] Raykov, Y. P., Boukouvalas, A., Baig, F., and Little, M. A. What to do when k-means clustering fails: a simple yet principled alternative algorithm. *PloS one 11*, 9 (2016), e0162259.

[31] Řehůřek, R., and Sojka, P. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (Valletta, Malta, May 2010), ELRA, pp. 45–50. http://is.muni.cz/publication/884893/en.

[32] Saunders, E. M. Stock prices and wall street weather. *The American Economic Review 83*, 5 (1993), 1337–1345.

[33] Vattani, A. The hardness of k-means clustering in the plane. *Manuscript, accessible at http://cseweb. ucsd. edu/avattani/papers/kmeans_hardness. pdf 617* (2009).

[34] Warren, R., Smith, R. F., and Cybenko, A. K. Use of mahalanobis distance for detecting outliers and outlier clusters in markedly non-normal data: a vehicular traffic example. Tech. rep., SRA INTERNATIONAL INC DAYTON OH, 2011.

[35] Wikipedia. Lda model.

[36] Xu, R. Pos weighted tf-idf algorithm and its application for an mooc search engine. In *2014 International Conference on Audio, Language and Image Processing* (2014), IEEE, pp. 868–873.

[37] Yonamine, J. E. Predicting future levels of violence in afghanistan districts using gdelt. *Unpublished manuscript* (2013).

[38] Zhang, D., and Li, S. Topic detection based on k-means. In *2011 International Conference on Electronics, Communications and Control (ICECC)* (2011), pp. 2983–2985.