Index prediction on the Swedish stock market using natural language processing methods on Swedish news

exploring topic modelling, sentiment analysis and machine learning methods to extract information from swedish financial newspaper data and predict the direction of swedish stock market indices

Erik Ris, Axel Sjöberg

Master's thesis
2021:E53

Lund Institute of Technology
Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

Centrum Scientiarum Mathematicarum

Master's Theses in Mathematical Sciences 2021:E53
ISSN 1404-6342

LUTFMS-3428-2021

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/

Abstract

This master thesis explores if topic modelling and sentiment analysis on Swedish financial newspaper data can be used to predict the direction of the Swedish stock market. A pipeline was set up where full length articles as well as article summaries were fed into a topic model and a sentiment analysis model. Several methods for combining the outputs of these models were explored in order to create data representations. The data representations were fed into four different machine learning models and one deep learning model that predicted the direction of stock index movement for three time periods: daily, weekly and monthly. The performance of the stock market index prediction model showed great promise on the in-sample data, alas, no conclusive answer could be drawn from the results when testing on the out-of-sample data. Allowing for the topic model to be trained on the test period, some encouraging results were obtained that lead to interesting observations which serves as a foundation for future research.

This master thesis was written under guidance of Lund University, Faculty of Engineering, Division of Mathematical Statistics and in collaboration with the company Sanctify Financial Technologies.

Keywords: Stock market prediction, Index prediction, Sentiment analysis, Topic modelling, LDA, Swedish newspaper data, NLP, Machine Learning, RNN

Acknowledgements

We would like to express our gratitude to our supervisor Erik Lindström, head of Mathematical Statistics at Lund University, Faculty of Engineering, with whom we have had many discussions, and who has guided us throughout the process. Thank you!

This thesis is written in collaboration with Sanctify Financial Technologies. We would therefore also like to extend our gratitude to the whole team at Sanctify, helping us whenever we came calling. An extra thank you to our two primary supervisors Oscar Dahlblom and Gustav Johnsson Henningsson, who helped us in defining the problem and investing time and energy in our work. A final thank you to Patrik Elfborg at Sanctify for being patient with us as we were trying to learn Linux and Git.

List of Abbreviations

ANN: Artificial Neural Network

BoW: Bag of Words

CNN: Convolutional Neural Network

EDA: Exploratory Data Analysis

EMH: Efficient Market Hypothesis

GRU: Gated Recurrent Unit

LDA: Latent Dirichlet Allocation

LR: Logistic Regression

LSTM: Long-Short Term Memory

ML: Machine Learning

NB: Naïve Bayes

NLP: Natural Language Processing

NMF: Non-negative Matrix Factorization

PCA: Principal Component Analysis

RBF: Radial Basis Function

RF: Random Forest

RNN: Recurrent Neural Network

SVC: Support Vector Classifier

SVD: Singular Value Decomposition

SVM: Support Vector Machine

TF-IDF: Term Frequency–Inverse Document Frequency

Contents

1

Introduction

This chapter introduces the *scope* and *purpose* of this thesis, presenting a short *background* and stating the *research question* to be answered. The chapter ends with an *outline* of the thesis.

1.1 Background

Trying to predict the direction of stock market movements is nothing new, and attempting to do so has been a project of many. Often, this has been done analyzing financial time series data using traditional time series models. With the digitization of data and the breakthroughs in data science and machine learning, new approaches have risen in priority. The combination of more available data and more sophisticated methods to make sense of said data have resulted in new opportunities for market prediction attempts. This is the case for market prediction using textual data. In this thesis textual data will be used in order to predict the movement of the Swedish market.

Earlier works on stock market prediction utilizing text data on the Swedish stock market focuses, to a large extent, on individual stocks. One of the main contributions of this thesis is to extend the scope of the research on market predictions based on Swedish text data to also include stock market indices.

Sanctify Financial Technologies

This thesis is written in collaboration with Sanctify Financial Technologies, a fintech company leveraging machine learning in order to deliver financial and market related knowledge, where the recipient could be a stock market analyst.

1.2 Research Question

Can topic modelling and sentiment analysis on Swedish financial news paper data be combined and leveraged in order to predict the direction of Swedish stock market indices while keeping interpretability?

1.3 Purpose and Scope

The scope of this thesis is the Swedish stock market and Swedish financial news paper data. The purpose is to explore the usage of financial news data in order to perform stock market index predictions. More specifically, the purpose is to investigate topic modelling and sentiment analysis in combination.

The research can further be broken down into three dimensions. These are the predictive time frame, what part of news articles to include and different stock market indices. For each combination, different feature construction methods and predictive machine learning models are explored.

One of the aspects of the research question is to keep the model and pipeline interpretable. Interpretability is in this thesis defined as the ability to follow the steps in the model and understand how the output is extracted from the input data.

1.4 Delimitations

The research scope of this thesis would be too extensive without delimitations. The first delimitation for this thesis is to only include news paper data from one source and form of publishing: articles published in the paper copy of Dagens Industri. It is also decided that seven stock market indices, a broad market index and six more sector specific indices, is a sufficient number to answer the research question.

1.5 Thesis Outline

This thesis will investigate natural language processing (NLP) approaches such as topic modelling by Latent Dirichlect Allocation (LDA) and sentiment analysis on financial news data in order to predict the future direction of different stock market indices. The purpose of using sentiment analysis is to find the tone of a text and the purpose of using a topic model is to find topics in a corpus. The output of these models, a sentiment score per article and a topic distribution per article are combined in order to create data representations. The market predictions, formulated as a classification problem, are conducted by both traditional machine learning models and a deep learning model. In *chapter 2* the necessary theory will be covered,

including financial markets, machine learning and natural language processing. In *chapter 3* recent previous research will be covered, focusing on NLP within a financial context with a market prediction application. In *chapter 4* the data and data sources are described. In *chapter 5* the research methods are presented and described in order to facilitate understanding and replication. In *chapter 6* the results are presented along with a discussion of their implications. In *chapter 7* the conclusions are presented, research questions answered and potential future research approaches explained.

2

Theory

In this section the relevant theory required to better understand the research question and analysis conducted is presented. The theory is divided in to three parts: *Financial Theory*, *Machine Learning* and *Natural Language Processing*.

2.1 Financial Theory

2.1.1 Stock Markets and Market Indices

There are many different stock markets around the world, and *the stock market* can be said to be a collective expression for all of these where stocks change hands. The stock market is divided into a primary market, where public companies sell shares in their company to the public, and a secondary market, where private investors and institutions trade these shares between one another. The secondary market is what *the stock market* usually refers to. The performance of a specific stock market, for example Nasdaq Stockholm, is often quantified by the performance of a given market index. An index consist of different stocks traded on the stock market and can be structured based on for example size or sector. When talking about the performance of the Swedish market one often refer to the performance of the OMXS30 index, consisting of the 30 most traded companies on Nasdaq Stockholm.

2.1.2 The Efficient Market Hypothesis

In his paper from 1965 Eugene Fama presented the Efficient Market Hypothesis and the idea that financial markets are completely random, and hence, not predictable. An investor should therefore not be able to consistently realize above average risk adjusted market returns given the information available at the time of investment, as the price should reflect all available information. (Fama, 1965) Due to shortcomings of his original hypothesis, Fama revised the hypothesis to instead include three levels of efficiency, strong, semi strong and weak in 1970. (Fama, 1970) For markets that

are classified as weakly efficient, historical information can not be used to generate above average risk adjusted market returns. In markets with a semi-strong efficiency current public information cannot be used to generate above average risk adjusted market returns, and in strongly efficient markets not even insider information can generate above average risk adjusted market returns.

This thesis extends the standard evaluation of the weak to semi-strong form of the efficient market hypothesis. This by considering "alternative data", in form of text data, from a financial newspaper believed to carry relevant information.

2.1.3 Behavioural Finance

Behavioural finance is a subbranch of behavioural economics. Behavioural economics unlike neoclassical economics, does not rely on people being rational, but rather the opposite. Deviations from rationality are seen as common and effectual, and the theories aim to explain these deviations. This is mainly done through decision making heuristics, mental fallacies and cognitive biases, such as: overconfidence, sunk cost fallacy and anchoring and adjusting. (Angner, 2016) Investors and stock market analysts are also subject to these cognitive biases and other human reasoning errors. (Friesen and Weller, 2002)

2.1.4 Stock Market Prediction

Markets that are weakly efficient are by definition predictable to some extent. Behavioral economics theory disregard the assumption that market participants are rational and suggests that market participants' errors in their decision making process make them operate on emotional assumptions and as a consequence the actual underlying value is not always analysed enough. Historically the two main branches of stock market analysis and furthermore prediction are *technical analysis* and *fundamental analysis*.

Technical Analysis

Technical analysis is devoted to studying patterns in market graphs and via mathematical models trained on historic data infer probabilities of future market movements. Some techniques used in technical analysis include moving average rules, filter rules and relative strength rules. The effectiveness of these techniques is quite a debated topic in the research world, yet it remains a common practice amongst market participants. More advanced financial models for market predictions is popular research topic and various different machine learning algorithms have been used in this effort.

Fundamental Analysis

The second school of market prediction analysis is called fundamental analysis and it is a field in which fundamental data from different sources are analyzed to guide assumptions. The fundamental data origin from various sources but common ones include financial data e.g. a currency in the foreign exchange market, a balance sheet for a company, index-data from different markets or information about government and central bank actions. It can also be other types of data such as news about meteorological incidents like natural disasters. Determining underlying value for an asset by using these data is a difficult task as the required analysis is subjected to plenty of uncertainty and due to the fact that the amount of available data is extremely vast and usually unstructured. This makes it hard for humans, and even more so for automated systems to successfully process and make accurate predictions based on the data.

2.2 Machine Learning

Machine learning is a sub-field of Artificial Intelligence and can be described as the study of methods for training computers to learn to solve specific tasks without being explicitly programmed. Mitchell (1997) formally defined the algorithms studied within ML as: *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E"*. The predictive ability in machine learning models is rooted in statistical analysis. Ergo, the machine learning discipline is highly similar to the statistics discipline, the main difference between the two is generally attributed to their terminology (Murphy, 2013) and to their application, where methods in statistics is more focused on inference and methods in machine learning is more focus on prediction (Bzdok et al., 2018).

Generally, the problems studied in machine learning can be divided into three categories based on the feedback the learning system uses: supervised learning, unsupervised learning and reinforcement learning. (Murphy, 2013) Of these, reinforcement learning methods which uses learning based on rewards will not be used throughout this thesis.

2.2.1 Supervised Learning

Supervised learning methods use labeled data, i.e. data with an input and output, to train models so that they are able to predict the output from a given input. If subjected to sufficient training, and a robust relation over time, these models, which have been trained to map input data to output data, can be used to make predictions on unseen data, i.e. predict the output given the input. Two common types of supervised learning algorithms are classification and regression. In classification the objective is to predict the categorical output, e.g. male or female, whereas in

regression the objective is to predict a numerical output, e.g. weather temperature.

2.2.2 Unsupervised Learning

Unsupervised learning methods use unlabeled data, i.e. data with no known output, with the objective of finding structure in the data. The systems that builds on unsupervised learning do not seek to predict outputs from the data, instead they are used to draw inferences on the data set and find patterns among the observations. As unsupervised learning algorithms are unable to compare a predicted output to the corresponding true output, model evaluation is less straight forward in unsupervised learning than in supervised learning. One of the most common tasks in unsupervised learning is clustering analysis, where the goal is to assign objects into a larger group where the members are similar on certain characteristics.

2.2.3 Traditional Machine Learning Methods

In this thesis, three different machine learning models are used to predict the outcome on the stock market. These are a logistic regression (LR), a support vector classifier (SVC) and a random forest (RF). Below, the models are described briefly.

Logistic regression

Logistic regression is a binary classifier, meaning that the response variable is $y \in \{0, 1\}$. The probability of positive/successful outcome is drawn from a Bernoulli distribution, see equation (2.1).

$$p(y = 1|\boldsymbol{x}, \boldsymbol{w}) = p = \sigma(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}) \tag{2.1}$$

where $\sigma(\cdot)$ is the sigmoid function, defined in equation (2.2) and $\boldsymbol{w}$ is the one-dimensional vector of weights that are fitted to $\boldsymbol{x}$, the vector containing the input features.

$$\sigma(x) = \frac{e^x}{1 + e^x} \tag{2.2}$$

Letting $\hat{y}$ denote the estimation of y, a general decision rule in binary classification is that, $p > 0.5 \Leftrightarrow \hat{y} = 1$.

The logistic regression can be extended to a multi-class setting by using the softmax function, given in equation (2.3) instead of the sigmoid function.

$$\sigma(\boldsymbol{\eta})_i = \frac{e^{\boldsymbol{\eta_i}}}{\sum_{j=1}^{C} e^{\boldsymbol{\eta_j}}} \tag{2.3}$$

where $\boldsymbol{\eta} = [\boldsymbol{w_1^T x}, \ldots, \boldsymbol{w_C^T x}]$, $\boldsymbol{\eta_i} = \boldsymbol{w_i^T x}$ and the number of classes is C. The probability of y belonging to class k can therefore be stated as $p(y = k|\boldsymbol{x}, \boldsymbol{w}) = \sigma(\boldsymbol{\eta})_k$.

Random Forest

Random forest is an ensemble method classifier that builds upon decision trees and bootstrapping. An ensemble method is a predictive method that creates and combines several predictive models in order to produce a result that is better and more robust than that would be obtained by any of the models constituting the ensemble, alone. For example combining n classifiers and predicting the most common prediction given from these n separate classifiers. A decision tree is a simple classification method that relies on a set of decision rules to make a prediction, see figure 2.1 for an informal example. Bootstrapping refers to random sampling methods with replacement. (Breiman, 2001)

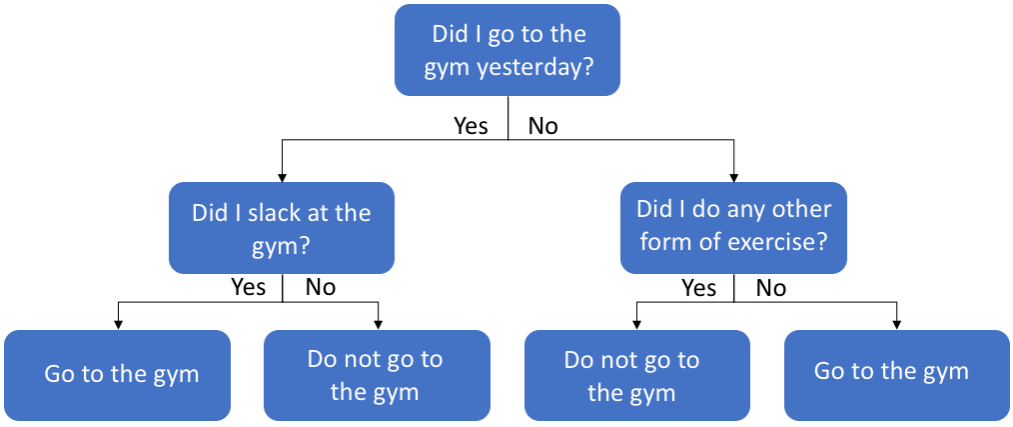

Figure 2.1: Example of a decision tree, answering the question "Should I go to the gym?"

A random forest classifier, is a combination of a predefined number of decision trees, each fitted to a bootstrap sample of the input data. The classification of the random forest classifier is the most common class predicted by the decision trees that the "forest" consists of. (Breiman, 2001)

An illustrative example of a Random Forest classifier fitted with n decision trees can be seen in figure 2.2.

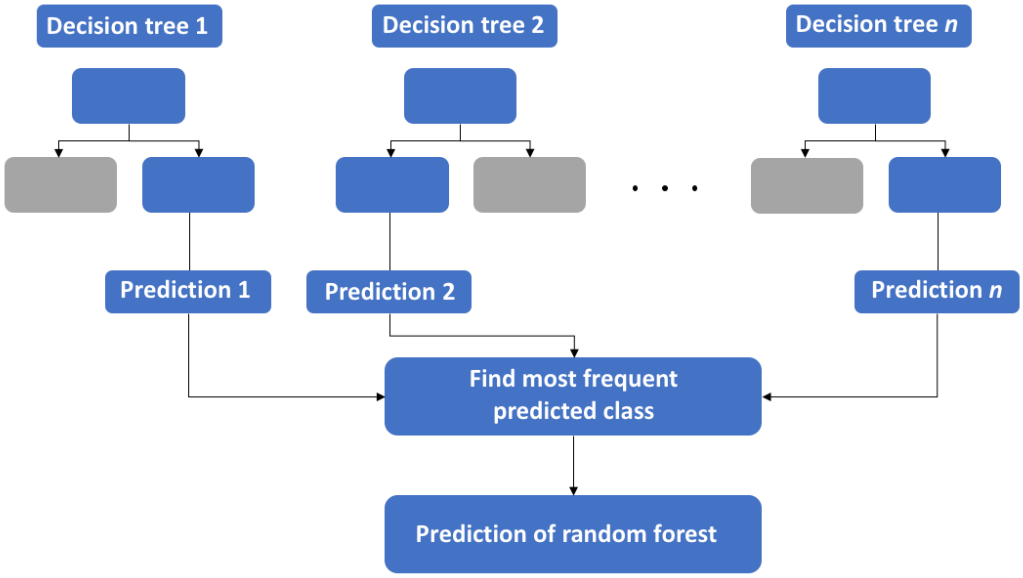

Figure 2.2: Example of a random forest classifier

Support Vector Machine

Support vector classifiers (SVC) are support vector machines that perform classification. The goal of a support vector machine (SVM) is to find the separating hyperplanes, normally referred to as decision boundaries, that maximize the smallest distance to the data points in either class, while still successfully separating the data points which belongs to different classes. Depending on which side of these separating hyperplanes future observations lie, their corresponding class can be predicted. If the data is not linearly separable, the data is subjected to transformations into higher dimensions or represented by pairwise similarity of the observations via the kernel trick. Methods relying on the kernel trick, do not explicitly transform each observation into a higher dimension, rather the data is represented by a kernel matrix where the pairwise similarity of the observations are calculated by a kernel function. A common kernel function to use is the radial basis function (RBF) defined in equation (2.4). (Murphy, 2013)

$$K(x_i, x_j) = \exp(-\frac{d(x_i, x_j)^2}{2l^2}) \tag{2.4}$$

where $d(x_i, x_j)^2$ is the euclidean distance between x_i and x_j, and l is a hyperparameter called the length scale of the kernel. A higher value of l will give a higher value for the pairwise similarity, thus acting as a smoother for the kernel function. (Murphy, 2013)

To account for non-perfect separation, slack variables can be used which puts a penalty on the objective function over which we optimize, in cases of wrongly classified observations. The magnitude of the penalty is controlled by a chosen hyperparameter C, where a low value of C gives less complex boundaries as the slack

variables punish the objective function little and vice versa for larger values of C. (Murphy, 2013)

2.2.4 Artificial Neural Network

Artificial Neural Network (ANN) is a model class inspired by the distributed learning process in human brains. An ANN is composed of a series of neurons which are accompanied by a weight and a bias that work in symbiosis to transform an input data representation to a prediction of the output. Normally the neurons are arranged in different layers where the input layer has as many neurons as the features of the input data, and the output layer has as many neurons as the features in the output representation. In between these two layers, multiple different layers, normally called hidden layers, can exist. An illustrative example of how an ANN is constructed can be seen in figure 2.3. The n input features are fed through the hidden layers of the network resulting in four output signals. In figure 2.3 the process of how signals are fed into a neuron and transformed into an output signal is illustrated. This process is called forward propagation. The output signals from the neurons in the previous layers are all multiplied with their corresponding weights. Thereafter the products and the bias term of the neuron are added together and the resulting sum is passed on the activation function which will return the output signal of the neuron. There are several activation functions that can be used, but an important thing is that it has non-linear properties, otherwise the neural network would essentially perform regular linear regression. One common activation function is the sigmoid function defined in equation (2.2), another one is the rectified linear unit activation function defined in equation (2.5). (Schmidhuber, 2015)

$$\phi(x) = \max(0, x) \tag{2.5}$$

The training of an ANN requires labeled data and is therefore a supervised learning method. The difference between the predicted output and the true output is called the loss. ANNs are trained by reducing this loss by adjusting the weights and biases in the model through a process called back-propagation, which calculates the gradients for each node in the network. The correction of the weights is also affected by the learning rate, which is a scalar term. The new weights and biases are calculated by subtracting the old weights and biases with the learning rate multiplied with the gradient. A larger learning rate thus means a larger correction factor, and generally tend to yield faster but also more noisy and less accurate learning. (Schmidhuber, 2015) A normal approach is to use a decaying learning rate, meaning that the learning rate is reduced over the course of the training to avoid stagnation and premature convergence. In training, oscillating situations may arise, when the parameter values of the model jump back and forth over a local minimum. Setting a lower learning rate will give new optimal weights, thus breaking the oscillating cycle. (Nagib et al., 2020)

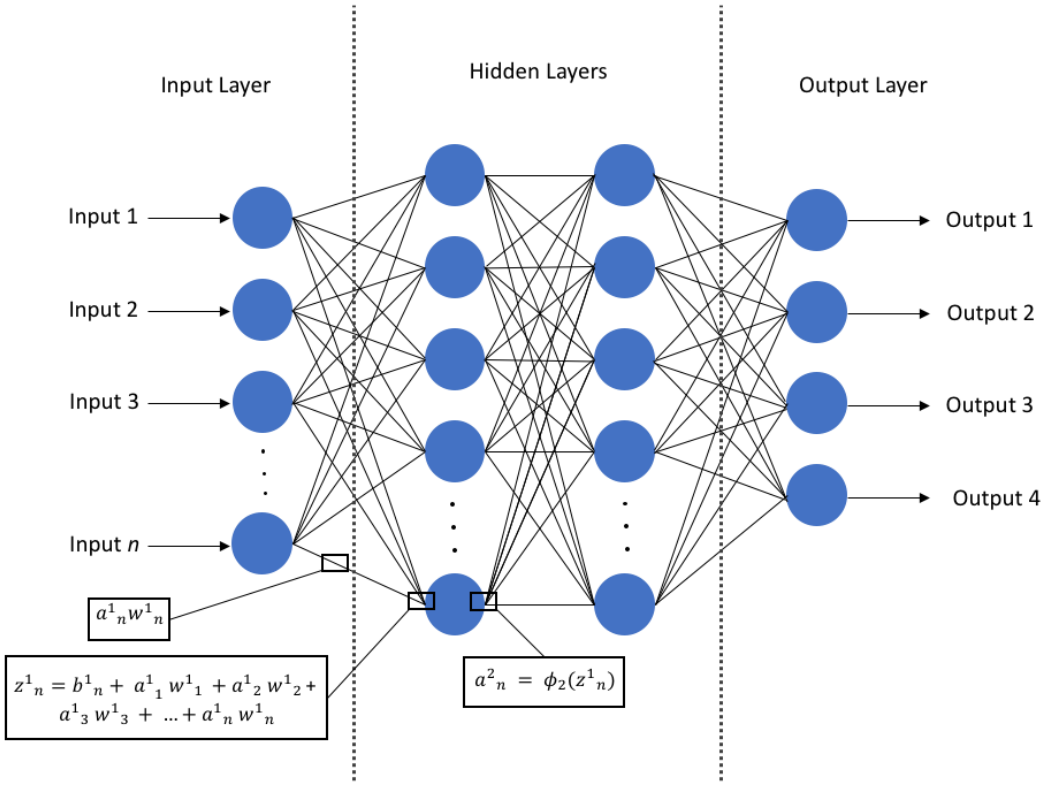

Figure 2.3: Example of an artificial neural network which takes n input features and predicts the output out of four target variables. ϕ_2 is the activation function used in the first hidden layer, b^1_n is the bias term of the n^{th} node in the first hidden layer.

Recurrent Neural Network

A special form of ANNs is the so called Recurrent Neural Network (RNN) which is designed for modelling sequential information. In RNNs information contained in the previous time step is passed to the next time step. (Hopfield, 1982) A problem with ANNs is the so called vanishing gradient problem, which is that the gradients used to make corrections of the weights gets smaller and smaller when moving down the layers from the output layer to the input layer. This could mean that barely any learning takes place in the early layers of the network. In ANNs there are methods to counter this issue, for instance using a different activation function. To train RNNs a modified version of the back-propagation method called back-propagation through time is used. The problem with RNNs are that the vanishing gradient problem is amplified when also modelling over sequential time steps. This means that the long term dependencies are harder to learn, just like the learning in the early layers were hard in the normal feed-forward network. In order to mitigate the problems of vanshing gradients in RNNs two model architectures are commonly used, the long-term-short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and the gated recurrent unit (GRU) (Cho et al., 2014). These model architectures rely on something called gates which learn what parts of the information should be added or removed from the hidden states. What these gates enables are shortcuts for gradients to be passed back in time without decreasing in magnitude for each

time step. The problem of vanishing gradient is therefore reduced for these model architectures. These model architectures are therefore normally better at learning long-term dependencies than ordinary RNNs are.

2.2.5 Data Processing

Before feeding the data into machine learning models, various techniques are used to process the data. In this section data processing methods used in this work are explained.

Feature scaling

Models relying on some form of gradient descent in the training phase generally tend to improve when features are scaled appropriately. E.g, for neural networks using the sigmoid function as activation the function will saturate for high values, i.e, it plateaus, meaning that the gradient becomes close to zero. Since backpropagation essentially works on correcting the weights and biases based on the gradients for these, having covariates large in magnitude will result in longer training time. Moreover if the loss function includes regularization and the features are unscaled the coefficients will be penalized differently for the different covariates.

Analogously, for models which rely on optimizing a cost function derived from the euclidean distance between two points, covariates that have large range and magnitude impact the estimation of the model parameters more than covariates with a smaller range and magnitude. For SVC which sets out to maximize the distance from the observations in different classes to the decision boundaries, feature scaling becomes very important.

In feature scaling it is common two differentiate between normalization and standardization. A common definition is that normalization scales and shifts the features so that they fall in a predefined range, normally [-1,1] or [0,1], whereas standardization maps the features to a zero mean distribution with unit variance, not necessarily Gaussian. Equation (2.6) is the formula for normalizing a feature $\boldsymbol{x}$

$$x = a + \frac{(x - \min\{\boldsymbol{x}\})(b - a)}{\max\{\boldsymbol{x}\} - \min\{\boldsymbol{x}\})} \tag{2.6}$$

where a, b constitutes the limits of the range the feature $\boldsymbol{x}$ is mapped to. Equation (2.7) is the formula for standardizing the feature.

$$x = \frac{(x - \bar{x})}{\sigma} \tag{2.7}$$

where $\bar{x}$ is the average of $\boldsymbol{x}$ and σ is the standard deviation of $\boldsymbol{x}$

Principal Component Analysis

Principal component analysis or (PCA) is a linear orthogonal transformation of a data set (matrix X), used in order to reduce the dimension of the data set while preserving much of the variability. This is a common dimensionality reduction process, and the method was first published in the beginning of the twentieth century. (Pearson, 1900) PCA results in a transformed data set with orthogonal features called principal components, each component explaining some of the variance. PCA can for example be conducted through singular value decomposition, where the the eigenvectors and their corresponding eigenvalues are computed. The eigenvectors of X are the principal components, where the first principal component explains most of the variance followed by the second component explaining second most etc. In order to reduce the dimension of X, one can settle for having a data set explaining a certain fraction of the original variance only keeping as many principal components as needed. (Jolliffe and Cadima, 2016)

Sparse Principal Component Analysis

The principal components often being linear combinations of a high number of the original features, thus reducing the interpretability of the data set, is an issue with the method. Sparse PCA solves this by making the linear combinations more sparse, i.e. each linear combination containing fewer features. (Zou et al., 2006)

2.2.6 Model and Parameter Optimization

Cross-Validation

Cross-validation is a method to ensure good fit and reduce over fitting for statistical models. For ordinary k-fold cross-validation the training data set is partitioned in K subsets or folds. A statistical model is then trained on $K-1$ subsets of the training data and validated on the last remaining subset. This is repeated for every combination, i.e. K times, and the validation score is given as an average of the K validation scores. Often K is set to five, but can be set to any integer, the procedure can be seen in table 2.1.

5-fold cross-validation					
Iteration	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
1	Train	Train	Train	Train	Validate
2	Train	Train	Train	Validate	Train
3	Train	Train	Validate	Train	Train
4	Train	Validate	Train	Train	Train
5	Validate	Train	Train	Train	Train

Table 2.1: Description of five fold cross-validation

Sliding Window Cross-Validation

In order to be able to utilize cross-validation and the benefits for time series data, where the time dependencies are to be preserved, another form of cross-validation can be performed. Sliding window cross-validation have the same underlying foundation as K-fold cross-validation but the subsets of the training data will not be shuffled. Like in the K-fold cross-validation the training data set is partitioned into K subsets. Table 2.2 illustrates the validation method. The size of the first fold, in this work referred to as the initial training length, is normally larger than the other folds while the rest of the folds are of equal size.

5-fold sliding window cross-validation					
Iteration	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
1	Train	Validate	-	-	-
2	Train	Train	Validate	-	-
3	Train	Train	Train	Validate	-
4	Train	Train	Train	Train	Validate

Table 2.2: Description of five fold sliding window cross-validation

Grid Search

A common way to perform hyperparameter optimization is to conduct a grid search. For a given statistical model with some predefined hyperparameters a "grid" of possible combinations of hyperparameters is created. For each combination of hyperparameters a cross-validation score is computed, and after comparing the scores for each combination, the most feasible combination of hyperparameters is deemed the one with the highest cross-validation score. When having a large number of hyperparameters within a model this method can be highly time consuming.

Feature Selection

The motivation of feature selection, stem from the fact that features are sometimes either irrelevant or redundant. When training a supervised learning model, coefficients are iteratively readjusted in order to achieve the objective of mapping the input data to the target variable with as high accuracy as possible. When the feature space is larger the model tend adapt to much to the training data and consequently the model will have poor performance on unseen data, this is called overfitting.

There exist several feature selection methods, e.g the LASSO algorithm, elastic net regularization and wrapper methods. In this work the choice of feature selection method falls on the latter of these. Two common wrapper methods are forward and backward selection. In forward selection the features which increases the performance of the model most are added one by one and in backward selection, features are removed one by one from the full feature space.

2.2.7 Comparing Results and Effectiveness

Baseline

A frequently used and basic initial baseline for a classification problem is the zero rule (ZeroR) baseline, where the baseline accuracy is set to be the most prevalent class. This would be the best guess for a classifier that did not apply any rules to the data set. This can be used as a baseline both for a binary classification problem and for a classification problem with multiple classes.

Leverage & Lift

Lift and leverage are two ways to quantify a result from a classification task by comparing to a baseline. Leverage is defined as the performance increase or decrease in percentage points and Lift is the ratio of the performance and the baseline.

$$\text{Leverage} = \text{Classification Accuracy} - \text{Baseline} \tag{2.8}$$

$$\text{Lift} = \frac{\text{Classification Accuracy}}{\text{Baseline}} \tag{2.9}$$

2.3 Natural Language Processing

Languages that have naturally evolved in humans without conscious planning and instead by use and repetition are defined as natural languages, e.g. Swedish, English or Hindi. Unlike humans, communication with computers requires using a formal language like Pascal or Scala. Natural language processing (NLP) is an interdisciplinary field that draws from linguistics, computer science and AI. The discipline is devoted to processing natural language data and from these create structured representations that enables a computer to process the input and analyze the contents of the data. (Kumar, 2011) This section provides the theoretical background of methods in NLP that are used throughout this thesis.

2.3.1 Text Pre-processing

In order to create representations that can be processed by a computer, the raw text data is pre-processed. Several different methods exist for this purpose and ranges from very simple to more advanced methods that are very computationally demanding. As an initial preprocessing step in many NLP tasks, it is common to remove special characters, punctuations, commas, URL-links and stop-words. Numerous studies have indicating an improved performance of NLP models when using these preprocessing methods. (HaCohen-Kerner et al., 2020)

In a corpus, some words occur frequently in all documents and therefore carry little information, e.g., *the, a, it, all*. These words are known as stop-words and a common approach is to remove these from the data set. This is done with a predefined list of stop-words. Other common filtering approaches include removing terms that only occur in very few documents, and removing terms that occur either too frequently or too infrequently in the corpus. The removal of stop-words and the other filtering approaches are active fields of research. (Gerlach et al., 2019)

In the raw format of a corpus there will be several inflections of words, e.g., walk, walked, walking or foot and feet. In order to decrease the size of the vocabulary of the corpus it is therefore common to map all the inflected words to a single word. Two different techniques can be used to do this, stemming and lemmatization. Stemming refers to the process of using algorithms to compute the word stem of an inflicted word. For the above inflicted words, walk, walked and walking all maps back to the word stem walk. Stemming often follows a clear set a heuristics that relies on slicing words by regular patterns and therefore irregular inflections can cause erroneous stemming. An alternative approach is to use lemmatization which is the process of mapping inflected words to their corresponding lemma, i.e., the dictionary form of the word, instead of the word stem. Lemmatization is a more computationally demanding process than stemming and modern lemmatizers often requires position of speech (POS) tags and in addition to this, some types of lemmatization also use information about the context of the words such as the nearby words in the sentence. The more simple lemmatizers are built on lookup dictionaries while the more advanced ones are trained machine learning models. (Juršič et al., 2010) Lemmatization is more computationally demanding than stemming and therefore slower. The benefits of using lemmatization is words with diamentrally different meanings are less likley to map to the same transformed word when using lemmatization, give more interpretable terms and tends to give better performance on Germanic languages. (Haselmayer and Jenny, 2017) A common approach is also to use stemming on the lemmatized terms.

2.3.2 Text Representations

One common method to represent a corpus is the Bag of Words (BoW) model. In the model each document is represented only by the words that are in the text and not by word order or grammar. This representation is constructed using one-hot encoding where the entries in the cell is either a binary indicator showing whether or not if the feature is present in a document, or a numeric figure indicating the frequency of each features in each document. The model therefore has a vocabulary which is the set of all unique terms in the corpus as well as vectors to represent the composition of words in each document. Figure 2.4 illustrates this process.

The first step of the BoW model is to first perform some form of word segmentation, i.e, splitting a written string into the separate words. (Shao et al., 2018) In English, Swedish and other space-delimited languages a common approach is to use the space as word divider.

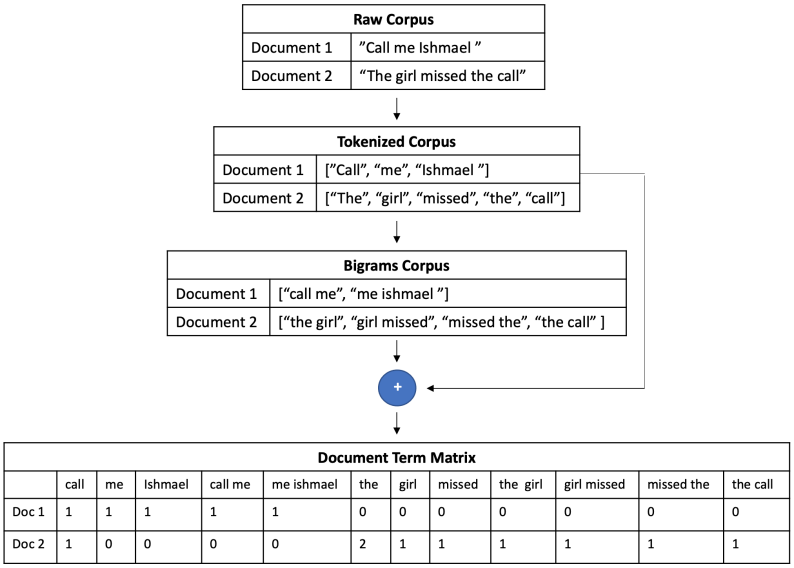

Figure 2.4: Bag of words representation from a raw corpus using the preprocessing methods tokenization, lowercase transformation and adding bigrams

There are several issues with the BoW model. One problem is that the vocabulary does not capture common phrases or expressions of more than 1 term, such as *stock market, hedge funds* or *financial crisis*. To counter this issue it is common to add sequences of words to the corpus vocabulary, so called n-grams, where n indicates the length of the word sequences. A bigram or 2-gram is the collection of word sequences of length two, trigrams or 3-grams are the collection of word sequences of length 3, etc. Figure 2.4 illustrates how bigrams are formed from the documents in a corpus. There is a trade-off however, adding n-grams could capture information that is important for the semantics of a text but it also increases the dimension of the vocabulary, and thus so also makes the vectors representing the articles even more sparse.

Another issue with the BoW model is that in certain cases, documents with diametrically different meanings could have the same representation. For instance the sentences *She is fast, not slow* and *She is slow, not fast* have opposing meanings but will have the same representation. For some tasks this is less of a problem, e.g in topic modelling where the goal is to infer the underlying themes of the documents.

2.3.3 Term Frequency-Inverse Document Frequency

The term frequency-inverse document frequency (TF-IDF) is the product of the term-frequency, a statistic indicating the frequency of each term in all documents, and the inverse-document-frequency, a statistic showing how significant and unique a term is to each document. The term-frequency can be defined in several ways, e.g by using the raw count or the Boolean count. Throughout this thesis the normalized term-frequency given by equation (2.10) will be used.

$$TF(t,d) = \frac{f_{t,d}}{\Sigma_{t' \in d} f_{t',d}} \tag{2.10}$$

The document-frequency is the measure of the normalized number of documents that contain a term. Taking the logarithm of the inverse of the document-frequency is defined as the inverse-document-frequency. In the literature there exist several different notations and definitions of the inverse-document frequency. In this work we will follow the basic notation given by Robertson (2004), see equation (2.11).

$$IDF(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{2.11}$$

The idea behind the inverse-document frequency measure is that terms that occur frequently in the corpus receive a lower score, meaning they are not very significant or unique for a single document. Taking the logarithm will make the IDF score for a certain term move towards zero as the number of documents containing that term increases. The final TF-IDF score is given by taking the product of the TF and IDF measure, see equation (2.12)

$$\textit{TF-IDF}(t,d,D) = TF(t,d) \cdot IDF(t,D) \tag{2.12}$$

2.3.4 Embedding Representations

Distributed representation vectors of words can be used to capture the semantic relationship between words, meaning that words that are close in the vector space have similar meaning. (Mikolov et al., 2013) Word embedding transforms the data from a high dimensional space into predetermined low dimensional space which usually is in the range 50-1000, depending on the task at hand. Outside of clustering words with similar meaning and reducing the dimension of the representation, embedding representations have another surprising property, which is that vector addition provide meaningful results.

2.3.5 Topic Modelling

Topic modelling is an unsupervised learning method that detects themes in the corpus by exploiting the statistical relationship between the terms in the documents. In topic modelling it is assumed that the corpus contains M documents, V unique words and K topics that are determined a priori. The objective of topic modelling is to infer the term distribution over each topic as well as the topic distribution for each document. In essence, a topic is a probability distribution of the words in the vocabulary, meaning that all the topics contain all the words in the vocabulary but with varying weights. Even though all words technically belong to a distribution, it is common in topic modelling to only present the five to ten most important words

for each topic. Moreover, each document is a composition of different topics, e.g., a document in the form of newspaper article could be both about trade and at the same time about politics. The term distribution over topics is usually represented by a $K \times V$ matrix and the topic distributions of the documents is represented by a $M \times K$ matrix. (Blad and Svensson, 2020)

The starting point of topic modelling algorithms is Latent Semantic Analysis (LSA). The idea behind LSA is create a semantic space by factorizing a term document matrix using singular value decomposition (SVD). The decomposed matrix is then truncated by, by only retrieving the K largest singular values and their corresponding singular vectors, meaning that the K most important dimensions are kept. Modern topic modelling algorithms have extended on the LSA using two different approaches. One of them has its foundation in linear algebra and the other one has its foundation in probability theory. (Blad and Svensson, 2020) The state-of-the art topic models in these respective categories are Non-Negative Matrix Factorization (NMF) and Latent Dirchlet Allocation (LDA). Of these two the LDA algortihms will be used in this thesis as it has been found to yield more interpretable and coherent topics for topic modelling on Swedish news articles. (Blad and Svensson, 2020)

Latent Dirchlect Allocation

LDA is a latent variable model that belongs to a family of models called generative statistical models. In statistical modelling there are two main approaches to compute classifiers, a generative approach and a discriminative approach. The terminology and definitions of these approaches can be somewhat confusing as they differ in literature. Following Murphy (2013) we distinct these two approaches on whether they model the joint distribution of an observable and target variable or if they model the conditional probability of the target and observable variable. Generative models and thus LDA belong to the former of these. As aforementioned, LDA is also a latent variable model, which means that the model allows for a set of variables that are unobserved. The objective of an LDA model is to infer the conditional distribution of the latent variables given the observable variables, via the joint probability distribution of the latent and observable variables. In the LDA model these latent factors are what human readers would consider to be topics. The goal of LDA is thus to approximate the conditional probability of the topics given the word arrangements in the documents. The LDA model assumes that each documents is composed out of different topics and that each word in that document is assigned to one these topics. The input in an LDA model is a bag of words representation and the output is the term distribution over topics, represented by a $K \times V$ matrix, as well as the topic distributions of the documents, represented by a $N \times K$ matrix.

Formally a corpus consists of M documents, the number of unique words in the corpus is V and the predetermined number of topics is K. z_{ij} denotes the topic of the j^{th} word in document i, and w_{ij} is the j^{th} word in document i. The word distribution for topic k is denoted φ_k and the topic distribution for document m is denoted θ_m. Both φ_k and θ_m are assumed to follow a multinomial distribution and they are drawn from the Dirichlet distributions, $Dir(\alpha)$ and $Dir(\beta)$. The Dirichlet

distribution is the conjugate prior of the multinomial distribution. By definition the posterior distribution is therefore also a Dirichlet distribution. The total probability of a given corpus is given by equation (2.13).

$$P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\Theta}, \boldsymbol{\Phi}; \alpha, \beta) = \prod_{k=1}^{K} P(\varphi_k; \beta) \prod_{m=1}^{M} P(\theta_m; \alpha) \prod_{t=1}^{N_m} P(z_{m,t}|\theta_m) P(w_{m,t}|\varphi_{z_{m,t}}) \tag{2.13}$$

Where N_m is the length of document m. The objective of the LDA model is to approximate the posterior distribution of the hidden variables stated in equation (2.14).

$$P(\boldsymbol{Z}|\boldsymbol{W}; \alpha, \beta) = \frac{P(\boldsymbol{Z}, \boldsymbol{W}; \alpha, \beta)}{P(\boldsymbol{W}; \alpha, \beta)} \tag{2.14}$$

Calculating the posterior defined in equation (2.14) can be done in several ways. As the number of documents in the corpus is large, calculating the total probability for each possible combination of latent factors is unpractical due to sheer size of the number of different combinations. Instead, some approximate inference algorithm needs to be used. One could either use an optimization-based algorithm like Online Variational Bayes or a sampling-based algorithm such as Gibbs sampling. A common approach is to use collapsed Gibbs sampling which is a Markov chain Monte Carlo (MCMC) algorithm that marginalizes over some variables. In the case of equation (2.13) this means that $P(\boldsymbol{Z}, \boldsymbol{W}|\alpha, \beta)$ is calculated by integrating out $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$, which gives the expression presented in equation (2.15).

$$\begin{gathered} P(\boldsymbol{W}, \boldsymbol{Z}; \alpha, \beta) = \\ \int_{\boldsymbol{\varphi}} \prod_{k=1}^{K} P(\varphi_k; \beta) \prod_{m=1}^{M} \prod_{t=1}^{N_m} P(w_{m,t|}|\varphi_{z_{m,t}}) d\boldsymbol{\varphi} \int_{\boldsymbol{\theta}} \prod_{m=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N_m} P(z_{m,t|}|\theta_m) d\boldsymbol{\theta} \end{gathered} \tag{2.15}$$

The expression in equation (2.15) can be divided into two separate integrals which in turn can be simplified separately. Moreover since, θ_i is independent of θ_j for all $i \neq j$ and φ_i is independent of φ_j for all $i \neq j$, the integral sign inside product expression can be moved, according to equation (2.16) and equation (2.17).

$$\begin{gathered} \int_{\boldsymbol{\varphi}} \prod_{k=1}^{K} P(\varphi_k; \beta) \prod_{m=1}^{M} \prod_{t=1}^{N_m} P(w_{m,t|}|\varphi_{z_{m,t}}) d\boldsymbol{\varphi} = \\ \prod_{k=1}^{K} \int_{\varphi_k} P(\varphi_k; \beta) \prod_{m=1}^{M} \prod_{t=1}^{N_m} P(w_{m,t|}|\varphi_{z_{m,t}}) d\varphi_k \end{gathered} \tag{2.16}$$

$$\int_{\boldsymbol{\theta}} \prod_{m=1}^{M} P(\theta_j;\alpha) \prod_{t=1}^{N_m} P(z_{m,t|}|\theta_m) d\boldsymbol{\theta} = \prod_{m=1}^{M} \int_{\theta_m} P(\theta_j;\alpha) \prod_{t=1}^{N_m} P(z_{m,t|}|\theta_m) d\theta_m \tag{2.17}$$

The term count for the i^{th} term in the vocabulary, assigned to topic k, in document m is denoted as $n^k_{m,i}$. The product expressions in the equations (2.16) and (2.17) which are the furthest to the right can then be written on the form given by equation (2.18) and equation (2.19).

$$\prod_{m=1}^{M} \prod_{t=1}^{N_m} P(w_{m,t|}|\varphi_{z_{m,t}}) = \prod_{i=1}^{V} \varphi_{k,i}^{n^k_{(\cdot),i}} \tag{2.18}$$

$$\prod_{t=1}^{N_m} P(z_{m,t|}|\theta_m) = \prod_{k=1}^{K} \theta_{m,k}^{n^k_{m,(\cdot)}} \tag{2.19}$$

where $n^k_{(\cdot),i}$ is the term count for the i^{th} term in the vocabulary assigned to topic k, over the whole corpus, and $n^k_{m,(\cdot)}$ is the total word count for topic k in document m. Using the expression given in equation (2.18) and the true distribution expression for the Dirichlet distribution, the right side of equation (2.16) can be simplified according to the calculations shown in equation (2.20).

$$\begin{aligned}
&\prod_{k=1}^{K} \int_{\varphi_k} P(\varphi_k;\beta) \prod_{m=1}^{M} \prod_{t=1}^{N_m} P(w_{m,t|}|\varphi_{z_{m,t}}) d\varphi_k = \\
&\prod_{k=1}^{K} \int_{\varphi_k} \frac{\Gamma(\Sigma_{i=1}^{V}\beta_i)}{\prod_{i=1}^{V}\Gamma(\beta_i)} \prod_{i=1}^{V} \varphi_{i,r}^{\beta_i-1} \prod_{i=1}^{V} \varphi_{k,i}^{n^k_{(\cdot),i}} d\varphi_k = \\
&\prod_{k=1}^{K} \int_{\varphi_k} \frac{\Gamma(\Sigma_{i=1}^{V}\beta_i)}{\prod_{i=1}^{V}\Gamma(\beta_i)} \prod_{i=1}^{V} \varphi_{k,i}^{n^k_{(\cdot),i}+\beta_i-1} d\varphi_k = \\
&\prod_{k=1}^{K} \frac{\Gamma(\Sigma_{i=1}^{V}\beta_i)}{\prod_{i=1}^{V}\Gamma(\beta_i)} \frac{\prod_{i=1}^{V}\Gamma(n^k_{(\cdot),i}+\beta_i)}{\Gamma(\Sigma_{i=1}^{V} n^k_{(\cdot),i}+\beta_i)} \int_{\varphi_k} \frac{\Gamma(\Sigma_{i=1}^{V} n^k_{(\cdot),i}+\beta_i)}{\prod_{i=1}^{V}\Gamma(n^k_{(\cdot),i}+\beta_i)} \prod_{i=1}^{V} \varphi_{k,i}^{n^k_{(\cdot),i}+\beta_i-1} d\varphi_k
\end{aligned} \tag{2.20}$$

Naturally, the sum of all probabilities will sum to one for any distribution. For the Dirichlet distribution this is expressed in equation (2.21).

$$\int_{\varphi_k} \frac{\Gamma(\Sigma_{i=1}^{V} n^k_{(\cdot),i}+\beta_i)}{\prod_{i=1}^{V}\Gamma(n^k_{(\cdot),i}+\beta_i)} \prod_{i=1}^{V} \varphi_{k,i}^{n^k_{(\cdot),i}+\beta_i-1} d\varphi_k = 1 \tag{2.21}$$

.

The final expression in equation (2.20) can therefore be written in the form given by equation (2.22).

$$\prod_{k=1}^{K} \frac{\Gamma(\Sigma_{i=1}^{V}\beta_i)}{\prod_{i=1}^{V}\Gamma(\beta_i)} \frac{\prod_{i=1}^{V}\Gamma(n_{(\cdot),i}^{k}+\beta_i)}{\Gamma(\Sigma_{i=1}^{V}n_{(\cdot),i}^{k}+\beta_i)} \int_{\varphi_k} \frac{\Gamma(\Sigma_{i=1}^{V}n_{(\cdot),i}^{k}+\beta_i)}{\prod_{i=1}^{V}\Gamma(n_{(\cdot),i}^{k}+\beta_i)} \prod_{i=1}^{V} \varphi_{k,i}^{n_{(\cdot),i}^{k}+\beta_i-1} d\varphi_k = \prod_{k=1}^{K} \frac{\Gamma(\Sigma_{i=1}^{V}\beta_i)}{\prod_{i=1}^{V}\Gamma(\beta_i)} \frac{\prod_{i=1}^{V}\Gamma(n_{(\cdot),i}^{k}+\beta_i)}{\Gamma(\Sigma_{i=1}^{V}n_{(\cdot),i}^{k}+\beta_i)} \tag{2.22}$$

Analogously, the expression in equation (2.17) can be written in the simplified form given by equation (2.23).

$$\prod_{m=1}^{M} \int_{\theta_m} P(\theta_j;\alpha) \prod_{t=1}^{N_m} P(z_{m,t|}|\theta_m) d\theta_m = \prod_{m=1}^{M} \frac{\Gamma(\Sigma_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \frac{\prod_{k=1}^{K}\Gamma(n_{m,(\cdot)}^{k}+\alpha_k)}{\Gamma(\Sigma_{k=1}^{K}n_{m,(\cdot)}^{k}+\alpha_k)} \tag{2.23}$$

Thus the total probability given in equation (2.15) can be written in the simplified form stated in equation (2.24).

$$P(\boldsymbol{W},\boldsymbol{Z};\alpha,\beta) = \prod_{k=1}^{K} \frac{\Gamma(\Sigma_{i=1}^{V}\beta_i)}{\prod_{i=1}^{V}\Gamma(\beta_i)} \frac{\prod_{i=1}^{V}\Gamma(n_{(\cdot),i}^{k}+\beta_i)}{\Gamma(\Sigma_{i=1}^{V}n_{(\cdot),i}^{k}+\beta_i)} \times \prod_{m=1}^{M} \frac{\Gamma(\Sigma_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \frac{\prod_{k=1}^{K}\Gamma(n_{m,(\cdot)}^{k}+\alpha_k)}{\Gamma(\Sigma_{k=1}^{K}n_{m,(\cdot)}^{k}+\alpha_k)} \tag{2.24}$$

In Gibbs sampling the topic assignments for all the words in an entire corpus are assumed to be known except for one single word in a single document. Denoting the topic assignment of the n^{th} word in document m as $z_{m,n}$ and the topic assignments for all the words in the corpus except for topic assignment $z_{m,n}$ as $Z_{-m,n}$, the probability that word n in document m belongs to a certain topic can be calculated by equation (2.25).

$$P(z_{m,n}|\boldsymbol{Z_{-m,n}},\boldsymbol{W};\alpha,\beta) = \frac{P(z_{m,n},\boldsymbol{Z_{-m,n}},\boldsymbol{W};\alpha,\beta)}{P(\boldsymbol{Z_{-m,n}},\boldsymbol{W};\alpha,\beta)} \tag{2.25}$$

The objective of the LDA algorithm is to approximate the posterior distribution given by equation (2.14). In this expression for the posterior distribution, the denominator of the right hand side, $P(\boldsymbol{W};\alpha,\beta)$ is invariable to Z. This means that the posterior is proportional to the numerator of the right hand expression, i.e $P(\boldsymbol{Z},\boldsymbol{W};\alpha,\beta)$. Moreover terms that do not depend on the topic assignment of $z_{m,n}$ can be removed without loss of proportionality. This means that the conditional distribution in equation (2.25) is proportional to the expression stated in equation (2.26).

$$P(z_{m,n}|\boldsymbol{Z_{-m,n}},\boldsymbol{W};\alpha,\beta) \propto P(z_{m,n},\boldsymbol{Z_{-m,n}},\boldsymbol{W};\alpha,\beta) \propto \prod_{k=1}^{K}\Gamma(n_{m,(\cdot)}^{k}+\alpha_k) \prod_{k=1}^{K} \frac{\Gamma(n_{(\cdot),v}^{k}+\beta_v)}{\Gamma(\Sigma_{i=1}^{N_v}n_{(\cdot),i}^{k}+\beta_i)} \tag{2.26}$$

where v is the index in the vocabulary of the n^{th} word in document m. The probability in equation (2.26) can be simplified further by leveraging properties of the gamma function and thereafter removing all the factors that do not depend on $z_{m,n}$.

$$\prod_{k=1}^{K} \Gamma(n_{m,(\cdot)}^{k} + \alpha_k) \prod_{k=1}^{K} \frac{\Gamma(n_{(\cdot),v}^{k} + \beta_v)}{\Gamma(\Sigma_{i=1}^{N_v} n_{(\cdot),i}^{k} + \beta_i)} \propto (n_{m,(\cdot)}^{k,-(m,n)} + \alpha_k) \frac{n_{(\cdot),v}^{k,-(m,n)} + \beta_v}{\Sigma_{i=1}^{N_v} n_{(\cdot),i}^{k,-(m,n)} + \beta_i} \tag{2.27}$$

where, $n_{(\cdot),(\cdot)}^{(\cdot),-(m,n)}$ is the same measure as $n_{(\cdot),(\cdot)}^{(\cdot)}$, only without $z_{m,n}$.

$$P(z_{m,n} = k | \boldsymbol{Z_{-m,n}}, \boldsymbol{W}; \alpha, \beta) \propto (n_{m,(\cdot)}^{k,-(m,n)} + \alpha_k) \frac{n_{(\cdot),v}^{k,-(m,n)} + \beta_v}{\Sigma_{i=1}^{N_v} n_{(\cdot),i}^{k,-(m,n)} + \beta_i} \tag{2.28}$$

Using the probability given in equation (2.28) we can calculate how likely it is that a word is assigned to the K different topics. This probability distribution is normalized so that it sums to 1 and then the topic is drawn from this normalized probability distribution, and the topic assignment is updated. This process is repeated for every word until we reach a maximum probability, a low enough tolerance or have run the algorithm for a fixed number of iterations.

Intuitively, the $n_{m,(\cdot)}^{k,-(m,n)}$ term is the importance of topic k in document m and $n_{(\cdot),v}^{k,-(m,n)}$ is the importance of word with index v in the vocabulary for topic k. The probability can therefore be seen as the smoothed product of these two importance scores, where α_k acts as a smoothing term for $n_{m,(\cdot)}^{k,-(m,n)}$ and β_v acts as a smoothing term for $n_{(\cdot),v}^{k,-(m,n)}$.

Both α and β needs to be determined a priori and prior knowledge about the documents in the corpus should be utilized when tuning these parameters. If it can be assumed that the documents in a corpus carry roughly the same amount of topics, a symmetric α should be used, meaning all α_k are the same. Analogously, if there is no reason to believe that certain topics should contain more words than other topics, a symmetric β should be used. Standard practice is to always use a symmetric β as the using asymmetric β tend to yield little to no improvement of the model. The asymmetric α parameter on the other hand have in plenty cases shown to outperform the symmetric α (Syed and Spruit, 2018), but this choice is not as conspicuous as it is for β. High values for symmetric α, means that each topic contains most of topics, while a low α means that each document is composed of a single or a few topics. Similarly, a low beta means that the topics are composed of a few words and vice versa for a high β.

2.3.6 Sentiment Analysis

The idea behind sentiment analysis or opinion mining is to find the tone in textual data. When referring to the tone in sentiment analysis it is the tone of the language used and not the underlying context of the text that is of primary concern. (Shapiro et al., 2020)

There are two main objectives or problems within the field of sentiment analysis, domain-specificity and complexity. The sentiment of a given text can be different depending on the context (or domain) from where the text inherits. This could for example be the case for different industries. Complexity refers to the level of complexity in the text that is being analyzed, on a basic level an example of increasing complexity are negations and on a more difficult level is sarcasm. Sentiment analysis is primarily conducted in two different ways: using a lexical approach or using a machine learning approach. (Shapiro et al., 2020) In this thesis the more basic and less time consuming approach of using a predefined dictionary is used.

Lexical Sentiment Analysis

This approach builds on using a predefined dictionary consisting of words or n-grams that are classified as either positive or negative a priori. Each document in the corpus is then searched for these classified words and a sentiment score is given to each document. There are many possible ways to define the sentiment score, based on the counts of positive and negative words.

There is a complexity problem with handling negations in sentiment analysis based on a dictionary, as the word *lethal* might be classified as negative but *not lethal* should of course not be classified as negative and possibly even classified as positive. In order to combat complexity problems when using a dictionary for sentiment analysis different heuristics can be inferred in the model. This could for example be switching the sign of the sentiment score when the preceding word is *not*, or increasing it when the preceding word is *very* or a synonym to these. A dictionary based model that have taken this into account is the VADER model, which uses a lexical approach in combination with five heuristics giving sentiment scores on a sentence level.(Hutto and Gilbert, 2015)

There are many different sentiment dictionaries available, some more general and some more domain specific. In 2011 Loughran and McDonald released a finance specific dictionary. (Loughran and Mcdonald, 2011) Another more general dictionary is the Harvard IV-4 psychological dictionary. Both of these dictionaries are originally in English, and have to be translated if they are to be used on text in another language.

ML-based Sentiment Analysis

Sentiment analysis conducted with machine learning methods is not restricted by a small set of rules for drawing conclusions about the sentiment of a text. There are possibly an infinite amount of different rules a machine learning model can pick up and use for predicting the sentiment of a text. But the downside of using a machine learning approach is that it requires predefined labeled texts in the corpus for the algorithm to learn from. Texts are often manually labeled which is time consuming and can be misleading as two different individuals might have different subjective opinions of the tone of a text. (Shapiro et al., 2020)

3

Previous Research

In this section we lay out an overview of relevant recent research carried out in the domains touched upon in this thesis. The chapter is divided into two main parts: *Work on News Data* and *Work on Other Forms of Textual Data*.

3.1 Natural Language Processing in Finance

In a review article from 2014 the current state of natural language processing based on online text mining used for market prediction is reviewed. This is divided into three separate fields of study: linguistics, machine learning and behavioral economics. Various textual sources have been used to perform both sentiment analysis as well as topic modeling, in order to create representations for market prediction models. (Khadjeh Nassirtoussi et al., 2014) In another review article from 2016, 51 studies on predicting stock prices and market activity by leveraging text mining are covered. The majority of these were performed as classifications, often so with a binary classification output. Various predictive machine learning techniques and methods have been used, with a focus laying on more standard predictive approaches. (Fisher et al., 2016)

3.2 Work based on News Data

In this section previous research in the field of NLP primarily within the domain of market prediction, where the text analyzed is either financial news articles, article summaries or news headlines, are reviewed.

3.2.1 Importance of News Data

It has been showed that investors' behaviour is impacted by their level of optimism respectively their level of pessimism about the future market. (Bollen et al., 2010) News releases by the media both report about the current state of the financial market and impact the market dynamics simultaneously. (Wisniewski and Lambe, 2010) This is also shown in Tetlock (2007), where the tone in an informative column on the market activity in the Wall Street Journal is shown to correlate to the market activity.

3.2.2 Topic Modelling on Swedish News Data

In the master thesis Blad and Svensson (2020) LDA and Non-negative matrix factorization (NMF) were compared in order to provide topics for a data set consisting of Swedish news articles. The thesis raises the difficulties with working with Swedish data in NLP, and highlights for example that lemmatization should be preferred over stemming for the Swedish language, and other Germanic languages. The thesis states that using nouns or nouns in combination with proper nouns in order to describe topics is equally "preferred" by experts in the field. The thesis also conclude that LDA using Gibbs sampling is to be preferred in terms of model performance over LDA using online variational Bayes. In the thesis both quantitative as well as qualitative analyses of the different techniques are performed.

3.2.3 Topic modelling on Norwegian News Data

In a series of papers (originally working papers from Norges Bank) a large data set consisting of news articles from Dagens Næringsliv, Norways largest business news paper is used to perform several tasks. In Larsen and Thorsrud (2017) the data is used to predict asset returns and in Thorsrud (2020) it is used to nowcast quarterly GDP of Norway. The data set included all news articles published during about 25 years (mid 1988 - 2014), and totaled 459,745 news articles. LDA (using Gibbs sampling) is performed on the data set and 80 are concluded to be the ideal number of topics, out of these 80, 40 are deemed informative and used for further modeling. Selection of the topics used is described in Larsen and Thorsrud (2019), and these are then inferred in the following papers. Sentiment analysis using a lexical approach are performed using the Harvard IV-4 psychological dictionary translated into Norwegian.

3.2.4 Sentiment Analysis of News Data

In Shapiro et al. (2020) different methods of performing sentiment analysis in an economical context are discussed and compared. Traditional lexical based methods are the focus of the article but machine learning methods are also discussed. The

text data consisted of financial and economical news paper articles. It is shown that the size as well as the domain-specificity of the lexicon used have a high impact of the final accuracy, which was evaluated by humans.

3.2.5 Stock Price Prediction

Concerning stock price prediction, a plethora of earlier research is available. The main aspects investigated in earlier research mainly touch upon the performance of different predictive models, whether to include sentiment and/or topics in the model, and which text should be used in the model, the full text of of the article or only the headlines.

Evidence from Hájek and Barushka (2018) indicates that using the combination of sentiment analysis and topic modeling are proven to increase the predictive power for individual stock prediction both for deep neural networks and traditional machine learning models. The article also suggested that more information is carried in the topic data than the sentiment data, but some information that is not conveyed in the topic data are to be found in the sentiment, and the combination is therefore preferred.

In Hájek and Barushka (2018), the authors also found that deep neural networks outperformed SVM for all combinations of the inputs, meaning that when using only either the topics or the sentiments as input as well as when using both topics and sentiments in the input, the deep neural network outperforms the SVM. This result is reinforced by Mohan et al. (2019) in which the authors compared different traditional machine learning methods, statistical models and deep learning models. Their result showed that the best model was a RNN with LSTM cells combining stock prices and new sentiment. Based on previous research the current state-of-the-art model for stock prediction seem to rely on neural networks rather than traditional machine learning methods.

There exist several approaches when using a prediction model based on neural networks. The most promising approach is to use some sort of RNN usually with LSTM cells. A deep learning model that have shown some promise is a hybrid model composed out of a RNN as well as convolutional neural network (CNN). This hybrid model was used in Vargas et al. (2018) which investigated the feasibility of using textual data in combination with technical-indicators. In this model the RNN-model was used to process the technical features and the CNN was used to process the news text data.

There is no standard evaluation metric in earlier research. In many articles, precision, recall and/or geometrical scores on categorical predictions are used to measure the performance of the models. (Hájek and Barushka, 2018) Other articles have trained a trading agent and where the realized return acts as a proxy for the model. (Vargas et al., 2018) In Mohan et al. (2019) the evaluated models performs regression rather than classification and to measure the performance of these models mean average percentage error (MAPE) is used.

Previous research on stock price prediction that exploits news data news have used different segments of news articles to create the data representations. In Hájek and Barushka (2018) the authors use full text articles, Mohan et al. (2019) uses an approach where only sentences in the vicinity of the word position where the stock is mention is used. In Vargas et al. (2018) news headlines and not full text articles were used following recommendations in previous work. In most cases only news that were related to the stocks the models sought out to predicted the movement of, were used.

3.2.6 Market Index Prediction

Research questions and problems connected to individual stock prediction are usually the subject of study in research examining market indices as well. As were the case in individual stock prediction, news sentiments combined with technical indicators outperformed models that only relied on one of these (Li et al., 2020), (dos Santos Pinheiro and Dras, 2017) and deep learning models outperformed more traditional machine learning models (Li et al., 2020), (Sousa et al., 2019). For research in market index prediction the performance metric used were mainly precision and recall or some average metric of these two (Li et al., 2020), (Ratto et al., 2019), but event studies where a trading agent had been trained were also used (dos Santos Pinheiro and Dras, 2017).

Several different ways to infer the news sentiment exist in the previous research. In Li et al. (2020) both manually and automatically annotated sentiment dictionaries are used. The manual sentiment dictionaries used in the study were the Loughran and Mcdonald financial dictionary, the Harvard IV-4 dictionary and the Vader sentiment dictionary. The other category of sentiment dictionaries, the machine learning annotated sentiment dictionaries were the SenticNet 5 and SentiWordNet 3.0. The study found that Loughran and Mcdonald financial dictionary performed best on accuracy.

Other form of sentiment analysis that have been used in market prediction includes embeddings from a pre-trained BERT model which had been fine tuned using a manually sentiment labeled set of 582 finance-specific news articles. In Liu et al. (2020) BoW (TD-IDF) and GloVe embeddings were used in machine learning models to predict the daily movement of the DIJA (Dow Jones Industrial Average).

As were the case for individual stock prediction, previous research on market prediction using news have used different segments of news articles to create the data representations. In Li et al. (2020), Ratto et al. (2019) and Sousa et al. (2019) full text articles are used and in dos Santos Pinheiro and Dras (2017) and Liu et al. (2020) only headlines are used. In Li et al. (2015) the difference in performance when feeding a stock movement prediction model with summarized news articles and full text news articles is compared. For stock index prediction summarizing is shown to improve the result. For individual stock predictions however, using the full text outperformed using the summarized articles. In dos Santos Pinheiro and Dras (2017) the authors focus on the headlines of the news articles instead of the

full text body as earlier work had shown that this produced better results.

Most predictive models used for market prediction in recent research are RNNs with LSTM, for instance Li et al. (2020) and dos Santos Pinheiro and Dras (2017). In Ratto et al. (2019) the authors used a feed-forward nerual network, but fed it with technical indicators meaning that information in earlier timesteps are forwarded into future timesteps. The difference becomes that the model with this method learns how to use the information conveyed in previous time steps in a more explicitly given way.

3.3 Work based on other Forms of Textual Data

Similarly to using news data as basis to form stock market predictions, other text sources have been exploited in this quest as well. This section provides a brief background to other common sources of text information for stock market prediction outside of news data.

3.3.1 Corporate Disclosures

Corporate disclosures have been used for a variety of stock market prediction tasks, with promising results. Much of the previous research where corporate disclosures are used, focuses on stock prediction. Usually the prediction model is trained on all the stocks listed on large exchange as is the case in Ahnve et al. (2020) and Lutz et al. (2020) and in some previous work the stocks are listed on a smaller niche exchange, as in Kim et al. (2018). Some work have also used corporate disclosures to train stock market indices prediction models (Feuerriegel and Gordon, 2018).

The time lag between the publication of the disclosure and the measurement of the prediction target variable, varies to a high extent in earlier research. In Feuerriegel and Gordon (2018) they create several models that are fed with different stock market indices representations and have time lags varying from 1 week to 24 months. In this work the authors found that long term prediction tended to yield the most significant results. Several earlier work have been successful in outperforming the set baseline, e.g in Ahnve et al. (2020) where the time lag is 24 hours and in Kim et al. (2018) where the target is the classification of the movement between closing price of day the disclosure was released to opening price of the next day.

Just as where the case for stock market movement predictions models using news data, the performance of models fed with disclosures are measured in primarily two ways. For stock movement classification models accuracy, or some derived measure of it, is used in Kim et al. (2018) and Ahnve et al. (2020) and from stock movement regression MAPE is used (Feuerriegel and Gordon, 2018).

TF-IDF have been the dominant method used in previous reviewed work for creating the data representations from corporate disclosures, see Ahnve et al. (2020), Feuer-

riegel and Gordon (2018) and Kim et al. (2018). In Ahnve et al. (2020), TF-IDF were found to outperform embeddings produced by a word2vec model, a pre-trained doc2word model and a pre-trained BERT model when feeding the representations from corporate disclosures in Swedish into a logistic regression.

Reviewing earlier works, no state of the art model for stock market movement prediction using corporate disclosures is identified. For instance in Ahnve et al. (2020) logistic regression was preferred over several variants of support vector classifiers and naïve Bayes while in Kim et al. (2018) several variants of support vector classifiers and naïve Bayes outperformed logistic regression and random forest. In both of these works the models had been fed TF-IDF distributions, and the results are conflicting.

3.3.2 Social Media

Topic modeling and sentiment analysis is used in combination on twitter data in Nguyen and Shirai (2015) and the output is used to predict the movement of individual stocks using a SVM-model. The work reinforces the findings of Hájek and Barushka (2018) that topic modeling in combination of sentiment analysis is preferred over only using topic modeling, but in a new setting where text data from social media is used.

4

Data

4.1 Text Data - Swedish Financial News

The textual data used in this thesis consists of news articles from Dagens Industri, the largest business news paper in the Nordics (Di, 2020). The data set consists of approximately 239 000 news articles published in the daily paper publication from the beginning of 2007 to the end of 2020. The advantages of using printed newspaper data is that the data is released to the public before the stock markets open each day of publication and it always comes in similar shape and form.

4.1.1 Data Collection

The textual data are downloaded from the database Retriever Mediearkivet (Retriever, 2021), each download in the format of a text file (.txt) consisting of 500 news articles. The text files were read into a pandas data frame, separating *title*, *preamble* and *textual body* in separate columns together with metadata for each article. The metadata tags collected for each article are presented in table 4.1.

Metadata	
Tags	Explanation
Date	Date of publication
Page	Page of the article
Word count	word count of the article (title excluded)

Table 4.1: Metadata collected for each article

4.1.2 Exploratory Data Analysis of Text Data

Figure 4.1 present the number of articles per day. In this figure an anomaly in the news data set can be observed. During the whole period the published number of

articles is relatively stable, with the exceptions of the time period before 2010, when the number of printed articles were slightly lower as well as August 2010, when the number of published articles often exceed 125 articles/day, which is never observed for any other time period. There is no apparent explanation in the data set for the former anomaly. The explanation of the latter anomaly is that during august 2010 the retriever database contains several duplicates where the title contains a formation index. These erroneous article entries are identical in every other fashion, and are therefore removed from the data set. Figure 4.2 shows the number of entries per date after filtering out duplicates, and as can be noted in the figure, the number of articles published every day is more stable. Another anomaly detected is the lack of data for several Mondays during the summer each year. This anomaly is a result of Dagens Industri not publishing a printed newspaper on Mondays for a number of weeks each summer. A third anomaly discovered regards the preambles, from 2017 to 2020 the preambles have been cut off after the first rows. This anomaly is assumed not to affect the full article approach of the study but is assumed to have effect on the summary approach as the preamble is a large part of each summary. The data set used for examining summaries of news articles is therefore all data before 2017-01-01. In order to examine if the anomaly have an effect on the full article approach as well, a secondary full article study is conducted on the reduced data set.

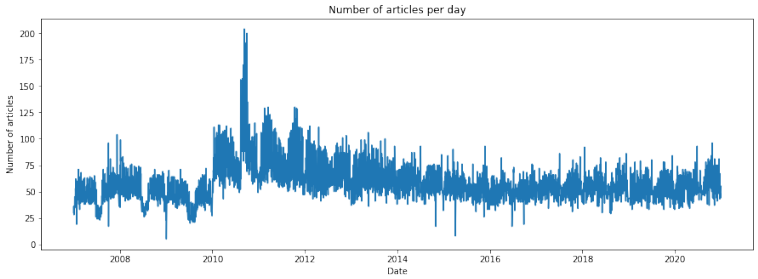

Figure 4.1: Articles per day initial data set

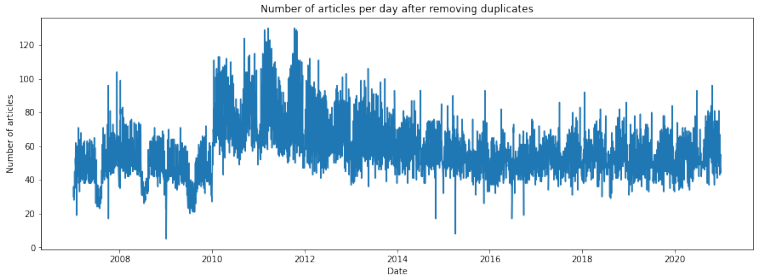

Figure 4.2: Articles per day after removal of duplicates

4.2 Stock Market Data - Swedish Indices

4.2.1 Data Collection

The price data used are retrieved from Nasdaq Nordic's web page. The indices, their ticker symbol and an explanation can be seen in table 4.2. The data stretches from the beginning of 2007 to the end of 2020, like the news paper data, and amounts to just over 3500 trading days with daily price data. The data sets are downloaded in comma separated values format (.csv).

Ticker	Description
OMXS30	Broad Swedish market index
SX10PI	Price index OMX Stockholm Technology
SX20PI	Price index OMX Stockholm Health Care
SX3010PI	Price index OMX Stockholm Banks
SX35PI	Price index OMX Stockholm Real Estate
SX4020PI	Price index OMX Stockholm Consumer Products and Goods
SX5010PI	Price index OMX Stockholm Industrials

Table 4.2: Ticker and description for each index

OMSX30 is chosen as it can be seen as a market index for the Swedish stock market and the other six indices are chosen in order to examine the effectiveness and feasibility of a sentiment per topic prediction approach in different sectors.

4.2.2 Exploratory Data Analysis of Index Data

The different stock indices are inspected and some outlier data points are discovered and examined. The data sets are plotted, cross referenced against the same data set from a different source, and the dates of the data entries are examined. For example, the OMXS30 data set includes some dates that should not be in the set, one Saturday, one Sunday and one Swedish National day. These dates are duplicate entries, and their associated data points are therefore omitted from the data set. Moreover, after cross referencing data from Nasdaq Nordic and Avanza Bank, a missing data point was identified in the OMXS30 data set. This missing value was imputed into the data set using the corresponding data point from Avanza Bank.

5

Methods

In this chapter the methods used throughout the thesis are described and motivated. These methods can, on a high level, be divided into *processing*, *feature construction* and *model construction*. The methods are implemented using *Python*.

5.1 Data Processing

The data are collected and analysed as described in chapter 4. The objective of the exploratory data analysis (EDA) is to find potential anomalies in the data sets, and to ensure quality of the data. When potential anomalies in the data have been evaluated and addressed, the data is subjected to the more fine grained pre-processing outlined below. This is carried out for both full length news articles as well as for article summaries, constructed as the title and preamble of the article. The methods in this section are explained only for full length articles, in order to avoid repetition.

5.1.1 Textual Data Pre-processing

The pre-processing results in two slightly different data sets, one text data set that is suitable for topic modelling and one text data set that is suitable for sentiment analysis. Each article is already split into three parts: title, preamble and text-body. All of these are subjected to the same pre-processing. The first step of the pre-processing is to remove special characters and words that contain special characters except for dots e.g: %, @ and URL-links. The dots are kept in order to facilitate the introduction of sentence-based bigrams in the topic modelling data set. Sentenced-based bigrams are bigrams where the constituents always come from the same sentence. This means that in a document where the string *"...after the financial crises. Social networks influenced..."* is present, *(crises, social)* is not a part of the bigrams corpus whereas if dots were filtered out, and then the bigrams were collected, *(crises, social)* would be a part of the corpus. The logic of only keeping

sentence-based bigrams is that consecutive words in a sentence relate more to each other, and carry more semantic valuable information such as idiomatic expression and non-compound words, than a word that ends a sentence, and the word that begin the following sentence do.

After removing the special characters the next step is to remove numbers and Swedish stop-words. Each article in the data set is then tokenized in order to identify the POS-tag for each word to subsequently lemmatize the words. This is done using a pre-trained Swedish model from UDPipe (Straka and Straková, 2017) via the spacy-udpipe package in python (TakeLab, 2020). The data set is then duplicated and the two versions are pre-processed separately.

Sentiment Analysis Pre-processing

For each tokenized article lemmatization of each word is carried out using the pre-trained Swedish lemmatizer from the lemmy package, the lemmatized words are then converted into only containing lower case characters. This results in the final data set used for sentiment analysis, as can be seen in the the sentiment model bubble in figure 5.1.

Topic Modelling Pre-processing

Following the findings of Blad and Svensson (2020), tokens that are not nouns are filtered out. Moreover, bigrams that contain at least one noun are created and added to the list of unigrams representing each article. The bigrams are created by connecting two words with an underscore. All the unigrams and the words in the bigrams are lemmatized using the pre-trained Swedish lemmatizer from the lemmy package. The lemmatized unigrams and bigrams are then converted into only containing lower case characters. Finally, unigrams and bigrams that occur less than five times or more than 50 000 times in the text of the whole corpus are removed and a final bag-of-words representation and a corresponding dictionary is formed. This process is illustrated in the topic model bubble in figure 5.1. The lower and upper limit which constitute the range of the term frequency allowed, are determined after examining the term-frequencies cumulative distribution. This reduced the size of the vocabulary between 90 and 95 percent depending on if full length articles or article summaries were used. Reducing the size of the vocabulary reduces noise in the final topic representations and decrease the training time of the topic models.

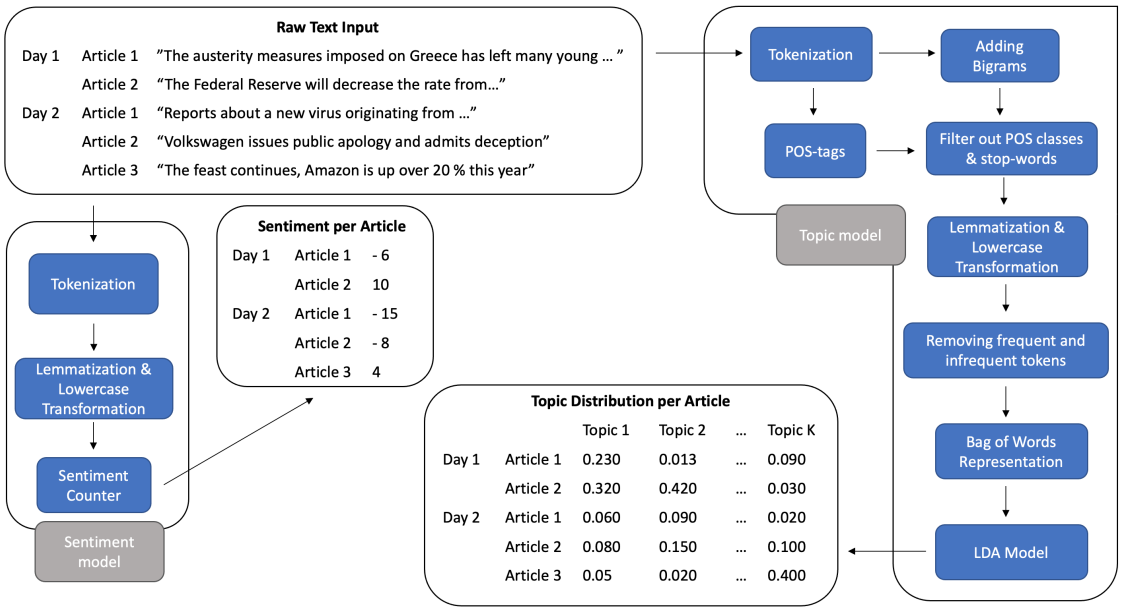

Figure 5.1: Raw text to sentiment per topic

5.2 Feature Construction

The pre-processed data is used to create two different representations. One of these is a matrix with the inferred topic distribution of the articles, and the other one is a matrix with the inferred sentiment of the articles. These representations are merged and manipulated in order to generate the final representation, which will be fed to the stock prediction model. The final representation is a matrix with the daily topic sentiment. This section explain how the topic and the sentiment representations are created as well as how they are merged when forming the final input representation with daily topic sentiments. As in the previous section the methods used are explained for full length articles only, to avoid repetition.

5.2.1 Topic Modelling

For each article a topic distribution is inferred using an LDA-model. The final Bag-of-words representation of the data set from the pre-processing step is fed into an LDA model from *Mallet* (McCallum, 2002) which was accessed via a python wrapper from the *Gensim* package. The choice of hyperparamters except for K, the number of topics, were set to the default, following the results of Blad and Svensson (2020), which means that the number of iterations in the Gibbs sampling is set to 1000 and the smoothing factors are symmetrical and set to, $\alpha = 5/K$ and $\beta = 0.01$. The number of topics, K is therefore the only hyperparameter connected to topic modelling that will be changed when different models are fitted.

The LDA models are trained in two different fashions. In the first of these, the LDA models are trained on articles from the training partition of the data set, and topics

then inferred on the unseen data in the test partition. In the second of these the LDA models are trained on articles from the full data set. The motivation behind this is to investigate the predictive capability of the topic model. If the stock prediction model, using topics from a topic model trained on the full data set, is better than the stock prediction model where the topics also had to be predicted, this could indicate that better methods for topic estimation is needed. It could also indicate that iterative retraining of the topic model is required due to shifts in the topics covered over the years, e.g *covid-19* or *brexit*. Naturally, training the topic model on the full data set is compromising from a test point of view as the model is allowed to look forward in time which cannot be done in a real life setting.

This training is carried out for different values for K, the number of topics parameter. Each model can then be fed seen and unseen articles and make predictions on the distribution of the K topics for the content of a given article. This is performed with each model for the all of the articles in the data set, and results in a topic distribution matrix over the articles. The topic modelling bubble in figure 5.1 illustrates this process.

Topic Exploration

As the LDA model does not assign names to the topics, but rather a distribution of importance for every word, it is common to represent the topics by their 5-10 most important words. In figure 5.2, 20 random topics for a fitted LDA model with 80 predefined topics are presented in the form of word clouds. The words relating to the different topics are translated into English from Swedish and the original version is presented in appendix A.

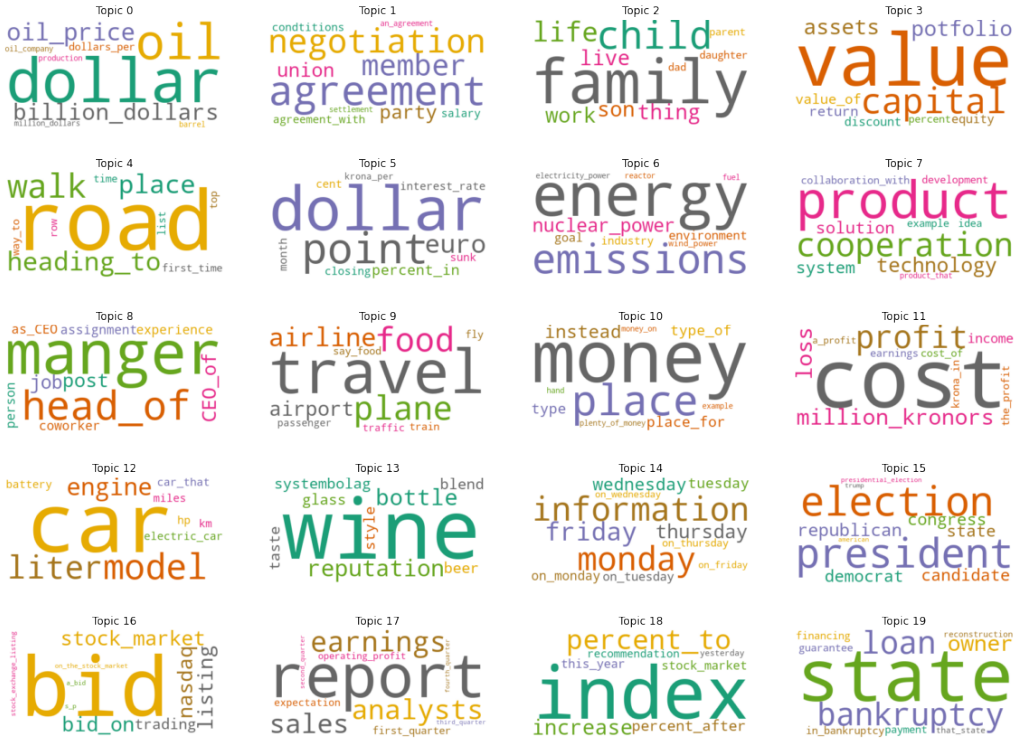

Figure 5.2: 20 topics from the DI-corpus, generated by an LDA model with $K = 80$. The topics are represented by word clouds of the 10 most important words for each topic

Evaluating the performance of the topic models is difficult due to it being an unsupervised machine learning model. There exist methods for evaluating the performance of the models, such as coherence score, but a human reader can also evaluate these by studying the most important words for the different topics. From looking at figure 5.2, the topics presented are coherent and logical. Moreover, an interesting feature of the topic models, that can be observed, is that certain topics are prevalent when the K hyperparameter varies for the topic models. For instance, it is noted that there is always at least one topic relating to the oil industry and that there is always a topic related to central banks, see figures 5.3 and 5.4. This is a pleasant property as the model produces more fine grained topics when increasing the predetermined number of topics while at the same time not being sensitive to the choice of predetermined topics, so that a change in K from 60 to 70 does not produce diametrically different topics.

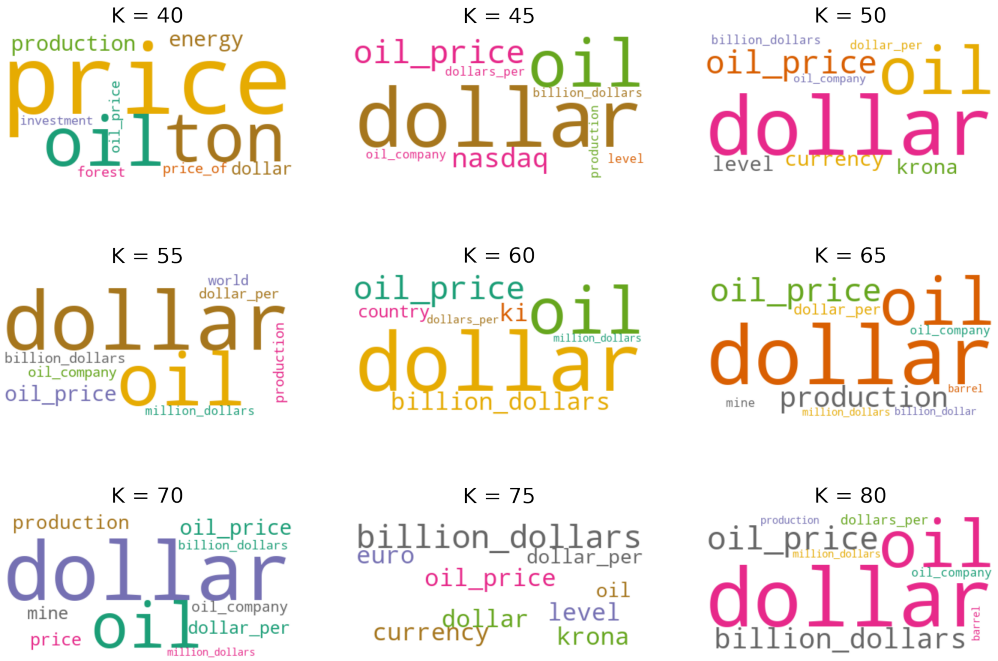

Figure 5.3: Topic representation of inferred topics relating to the oil industry from the LDA models. The topics are represented by word clouds

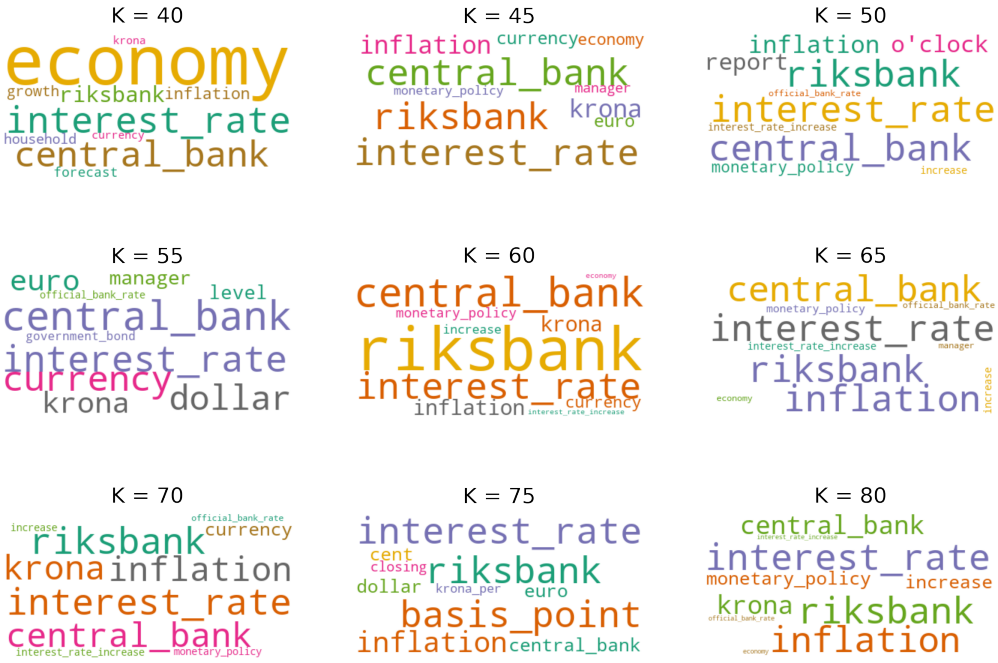

Figure 5.4: Topic representation of inferred topics relating to central banks from the LDA models. The topics are represented by word clouds

5.2.2 Sentiment Analysis

The sentiment analysis is conducted on an article level by a lexical approach. This, as a lexical approach to sentiment analysis have been shown to be both effective and sufficient, meaning that it has yielded good results in earlier research at the same time as it is easy to implement. (Li et al., 2020) (Shapiro et al., 2020) Throughout the thesis, this method based on one lexicon is used.

The lexicon used is the Loughran & Mcdonald financial dictionary (Loughran and Mcdonald, 2011). This dictionary consists of financial specific words labeled either positive or negative and is originally in English. In order to be able to use the dictionary to conduct sentiment analysis on our Swedish data set the dictionary is translated into Swedish. To ensure better accuracy the entries of the lexicon are lemmatized to fit the format of the lemmatized articles.

The sentiment score used is the crude measurement of the number of positive words subtracted by the number of negative words. This implies that all of the words have the same weighting when calculating the sentiment score. For all of the articles the sentiment score is calculated for the full articles and the articles summaries. These sentiments are in the next step coupled with the topic distributions of the articles.

5.2.3 Sentiment per Topic

On a high level, the frequency of positive and negative financially important words are used to calculate the sentiment score for each article, and topic distributions are inferred for every article using several LDA models with different predefined number of topics. The sentiments scores for every article is collected in a sentiment vector for both the full articles and the article summaries, and the topic modelling results in unique article-topic distribution matrices for each of the topic models. The distribution matrix for a given topic model is then combined with the matching sentiment vector, in order to produce the daily sentiment per topic signals. This combination is performed for all of the topic models, on a daily level and in three different ways. In one of these, all the articles and their topic distributions and sentiment scores are used, and in two of these, only the most important article for every topic and the corresponding sentiment score is used. The three different approaches are explained in detail below and all of these allocation methods corresponds to the *Topic & Sentiment Weighting* block in figure 5.5.

> **First approach:** As each row in a distribution matrix is the distribution of the K number of topics for a given article, multiplying the row with the sentiment score of the article, gives a sentiment per topic score for each article. On a daily level, the topic sentiments for each article are summed for each of the topics. This process is illustrated in figure 5.5, where in day 1 there are two articles which have the sentiments -6 and 10 as well as the topic distributions 0.230 and 0.320 for topic 1. The total daily sentiment for topic 1, on day 1, thus becomes $0.230 \cdot (-6) + 0.320 \cdot 10 = 1.820$.

Second approach: This approach to produce the daily sentiment per topic is an approach identical to the one used in Thorsrud (2020). The topic distribution for each article is first calculated. The most important article for each topic on a daily level is determined by retrieving the article which has the highest topic distribution weight for each of the articles on a specific day. The sentiment of that topic is then set to be equal to the sentiment of its most important article. This method is illustrated in 5.5 where for day 1 there are two articles. For topic 1 the topic distributions for the two articles are 0.230 and 0.320. As article two has the higher topic distribution weight, article 2 is the most important article for topic 1 on day 1. Thereby the topic sentiment on day 1 for topic 1 is the sentiments of article 2, which is 10.

Third approach: The third approach is an extension to approach two. The sentiment per topic is again inferred like in the second approach from the most relevant article, but is also weighted by the distribution weight for the given topic. In the example in the second approach the daily topic sentiment would therefore be $0.320 \cdot 10 = 3.2$.

The reason for multiplying with the topic distribution weight is that the most important article for a specific topic could have a very low topic distribution weight for that specific article. This would be the case if the topic is not very relevant in the news issue of that day. If the article that happened to have the highest topic distribution weight for a topic has a very high or low sentiment-score but is not well covered in during that day, the daily topic sentiment will be misleading and show that there is a very strong sentiment score for that specific topic where in reality the topic is just poorly covered in the newspaper. The idea behind this method is that multiplying the sentiment score of the most important article with the corresponding topic distributions can offset this issue.

It is worth noting that all of these methods have some flaws. For example, the methods will not be able to represent an article which talks about two topics where only positive things are said about topic one and only negative things are said about topic two. Instead the total sentiment of the article is calculated and this sentiment score is used to create the daily topic sentiment signal.

The daily sentiment per topic matrix is the final input representation fed to the stock movement classification model. To summarize, a news paper issue containing a number of news articles have been reduced to K sentiment scores for each date in the data set, as is illustrated in figure 5.5.

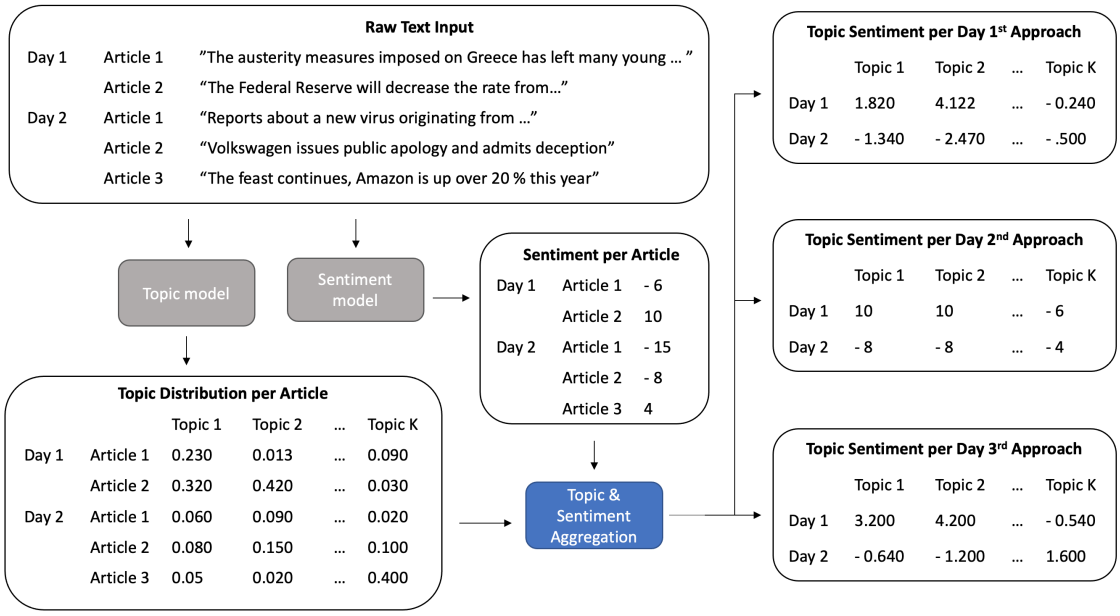

Figure 5.5: Daily Sentiment per Topic Score Pipeline

There is a problem with how to handle the news data published on Saturdays when the stock exchange is closed. News published on Saturdays contain information that could influence the stock market, just like news published any other day. In order to include this information in the data set, a weighted average between a Saturday's sentiment scores per topic and the following Monday's sentiment scores per topic is used for entries on Mondays. The resulting sentiment score per topic is 0.25 times the data for Saturday and 0.75 times the data for Monday, when there are data for both days. This in order to assign a higher importance to more recent data, while still including new data from when the stock market is closed. The ratio could be fit as a hyperparameter but is deemed to increase the dimension of the problem too much in relation to the potential upside. This weighted average approach also solves the anomaly detected for some weeks of the summer period, when there are no news publicized on Mondays. By giving full weight to the sentiment scores for the previous Saturday, for each such Monday, more days can be used in the data set. When there are news on Monday but not for the previous Saturday, full weight is given to the Monday. Table 5.1 presents how the weights for Saturdays and Mondays are assigned depending on the publication history.

Sentiment Weights for Saturdays and Mondays for the weighted average				
News Issues		Weights		Occurrences (2007-2020)
Saturday	Monday	Saturday	Monday	
Yes	Yes	0.25	0.75	567
No	Yes	0.00	1.00	38
Yes	No	1.00	0.00	103

Table 5.1: Weights for the sentiment scores of Issues published on Saturday and Monday

Sentiment per Topic Exploration

In figure 5.6 an example of a daily topic sentiment over the training and validation period is presented. The topic for which the daily sentiments are plotted encompass themes about the economy and the five most terms (i,e single words and bigrams) of the topic are *growth, economy, forecast, recovery* and *state of the market*. Note that some of these are not unigrams or bigrams and include stop-words. This is due to the terms being translated from Swedish, in which compounded words are more common than in English. The plot shows a positive sentiment score bias for the topic and behaves reasonably stable throughout the time period, but experiences a slight drop during the aftermath of the financial crisis in 2008 before recovering around 2010. In order to infer meaningful long term interpretations of the topic sentiments, a rolling average can be used. In figure 5.7, a 100 day rolling average of the daily topic sentiment presented in figure 5.6 is plotted with the development of the OMXS30 index. The daily predictions in figure 5.6 are quite noisy however when a rolling average is applied to the time series a more interpretive plot is generated . As topic 27 is about economic development, it is reasonable that the daily sentiment seem to correlate with the returns of the OMXS30 index.

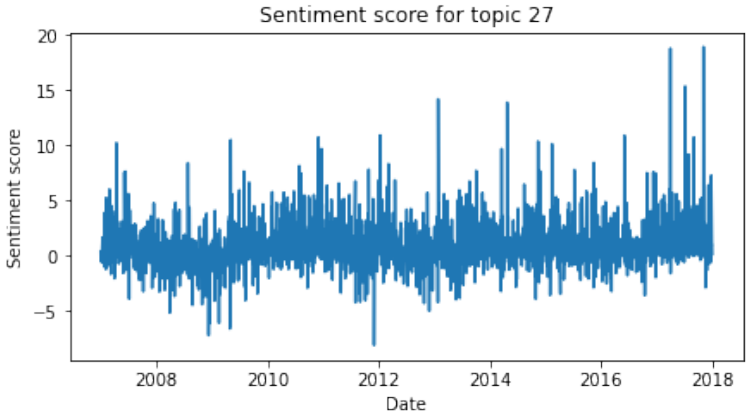

Figure 5.6: Example of daily topic sentiment inferred from an LDA model with 80 predefined topics and a daily sentiment weighting using the third approach defined in section 5.2.3 on 42

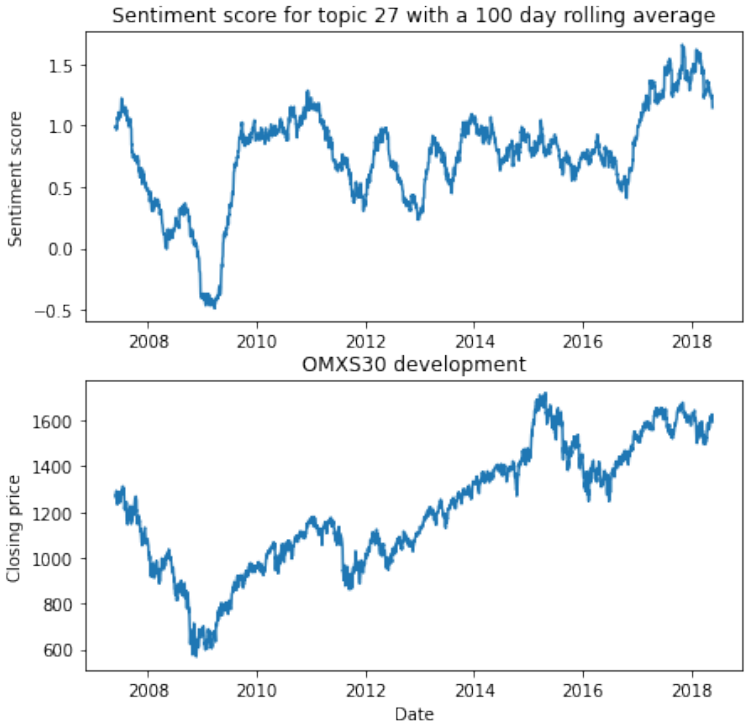

Figure 5.7: 100 day rolling average for the daily sentiment of an economic development topic and the OMXS30 development. The daily topic sentiment was inferred from an LDA model with 80 predefined topics and a daily sentiment weighting using the third approach defined in section 5.2.3 on 42

Another interesting property that is captured by the daily sentiment signals when filtering them though a 100 day moving average can be observed in figure 5.8. Topic 15 encompass American politics whereas topic 60 encompass Swedish politics. The clear spikes which can be seen in both of the plots coincide with the time of the presidential election in the US and the general elections in Sweden.

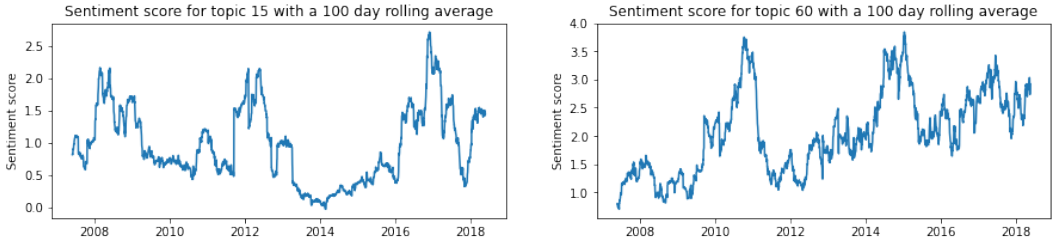

Figure 5.8: 100 day rolling average for political election topics. Topic 15 encompasses American politics and topic 60 encompasses Swedish politics. The daily topic sentiments were inferred from an LDA model with 80 predefined topics and a daily sentiment weighting using the third approach defined in section 5.2.3 on 42

The purpose of the daily topic sentiment signals is to be fed to a stock index prediction model, which will predict the direction of the stock market for three different time periods. In order to deduce to what extent sentiment signals from several days back should be used, when predicting the direction of daily, weekly and monthly price change. An operating hypothesis in this work is that when predicting the direction

of the stock price change today, a smaller look-back window is required compared to when predicting the direction of the price change several days forward. In order to determine if there is any structure in the input signals that can be exploited if we want to use topic sentiments from several time lags, the autocorrelation and the partial autocorrelation are plotted for several daily topic sentiments. In figure 5.9 some of these plots are presented. For most of the topic signals the autocorrelation and partial autocorrelation plots indicated that the input topic signals behave as white noise (as is the case for Topic 0 in figure 5.9), for some of the autocorrelation and partial autocorrelation plots there were some marginal structure that could, as is the case for topic 17 in figure 5.9, be exploited. Moreover, for some of the topic signals the autocorrelation and partial autocorrelation plots indicate a very strong lag 5 seasonality. The reason for the 5 day seasonality for some of the topics is believed to be that those topics are recurrent themes in the Saturday's issue. These topics include cars as well as wine and other alcoholic beverages reviews, which correspond to the topic 12 and 13 in figure 5.2. Most other daily topic sentiment behave as white noise however, and those topic sentiment signals that do not, either have a lag 5 seasonality or have very idiosyncratic autocorrelation and partial autocorrelation plots. This makes it difficult to use some general method for determining how long the backward looking window for the input signals should be, and how these should be modeled.

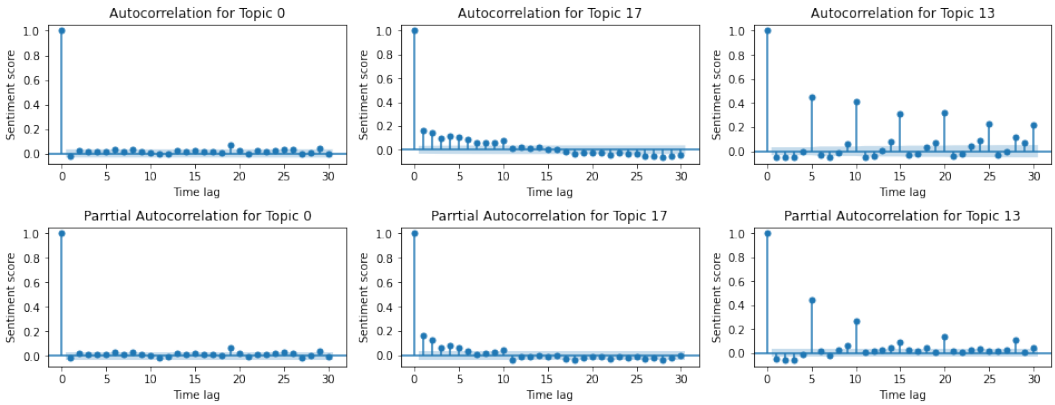

Figure 5.9: ACF and PACF plots for sentiment per topic time series

To conclude the input has a high dimensionality and the time dependencies between lags seem to either not be very significant, exhibit idiosynratic relationships and in those cases when there is a strong seasonality the topics are believed to not impact the outcome on the stock market too much. The time dependencies of topics relating to lifestyle themes and certain topics could carry meaningful information when feeding the model input signals spanning over several days, however to get a reasonable scope and reduce this problem, these relationships are not taken into account moving forward. Due to having high dimensional data and fitting several models with different number of predefined topics in combination to using various aggregation methods, the input signals are modeled as rolling average defined in equation (5.1), where the back-looking window, w is a predefined parameter. As a last pre-processing step the input features are standardized according equation (2.7).

$$RA(s_k) = \frac{s_k + s_{k-1} + ... + s_{k-w+1}}{w} \tag{5.1}$$

5.2.4 Data Labeling

The corrected and extended data set, now consisting of a sentiment score per topic on a daily basis matrix, is combined with an index data set and the dates for which both index price data and news data are kept. The label inferred for each day is a either a binary variable (0,1), taking the value 0 if the direction of price movement was downwards and 1 if the direction of price movement was upwards or as a multi class variable (up, stable or down) for a given percentage change tolerance, for example, ± 0.1 %. This yields the final data frame used for the predictive modelling. The initial approach is to test the predictive power over three time horizons: 1 day (daily), 5 days (weekly) or 20 days (monthly). This is done by altering the way in which the price change and thereby label of each data point is calculated and inferred. For day i, the price change is computed in the following ways:

Daily: Closing price of index day(i) - closing price of index day($i-1$)

Tolerance for multi class labeling = ± 0.1 %

Weekly: Closing price of index day($i+4$) - closing price of index day($i-1$)

Tolerance for multi class labeling = ± 0.5 %

Monthly: Closing price of index day($i+19$) - closing price of index day($i-1$)

Tolerance for multi class labeling = ± 1.0 %

5.2.5 Combining Input Data - Traditional ML-Models

When predicting the over night price change from day $i-1$ to day i, the input data originates from the news paper published in the morning of day i, and as aforementioned the data are transformed into sentiment per topic for day i. When classifying weekly or monthly price change more input data are used, looking further back in time. For predicting the weekly price change a window of 5 days of news paper data are combined to predict the price change five days ahead. The news paper data used to make a weekly prediction at day i are the data from the days: $i-4$, $i-3$, $i-2$, $i-1$ and i. The data are aggregated as an average of the sentiments for each topic, again resulting in one row consisting of one sentiment score per topic. An illustrative example of this can be seen below where the data in table 5.2 after being aggregated turns into table 5.3. The same approach is used when predicting the price change 20 days forward, but instead of a backward looking window of 5 days, a backward looking window of 20 days is used. The length of the backward looking window were determined after a few tries on the validation data and by studying the the sentiment per topic signals. Naturally the back-looking

window could be included as a hyperparamter that needs to be fitted, but in order to maintain a feasible scope within our time frame, the back-looking window was held constant.

Date	Topic 1	Topic 2	...	topic K-1	Topic K
2001-01-01	1	1	...	1	1
2001-01-02	2	2	...	2	2
2001-01-03	3	3	...	3	3
2001-01-04	4	4	...	4	4
2001-01-05	5	5	...	5	5
2001-01-06	6	6	...	6	6
2001-01-07	7	7	...	7	7

Table 5.2: Sentiment per topic example

Date	Topic 1	Topic 2	...	topic K-1	Topic K
2001-01-01	-	-	...	-	-
2001-01-02	-	-	...	-	-
2001-01-03	-	-	...	-	-
2001-01-04	-	-	...	-	-
2001-01-05	3	3	...	3	3
2001-01-06	4	4	...	4	4
2001-01-07	5	5	...	5	5

Table 5.3: Sentiment per topic example, five days aggregated

Dependency Reduction

The sentiment per topic time series correlate to a varying extent depending on the sentiment assignment approach, defined in 5.2.3, used. In table 5.4 the highest correlation between two input signals when using different LDA models and sentiment topic weighting are used is presented. For the first approach the correlation between the input signals were a lot higher than when using the second or third approach. Normally the correlation rarely exceeded 0.1 when using second or third approach but very often exceeded 0.1 when using the first approach. In order to circumvent eventual issues with multicolinearity and with the added effect of reducing the feature input space, PCA can be used. In this work two different types of PCA are used, normal PCA and sparse PCA on the input features to see if this improves the accuracy of the fitted models. For the different LDA models it was generally found that for a normal PCA 50 % of the components explained 80 % of the variability in the topic sentiment signals. When fitting a logistic regression, feature reduction methods that deal with highly correlated features are necessary as it tends to increase the variance of the parameters of the model. For any model reducing the number of features also avoids the problem of overfitting and can speed up the training time.

Highest correlation between two input signals									
	40	45	50	55	60	65	70	75	80
Approach 1	0.837	0.814	0.801	0.805	0.799	0.798	0.799	0.829	0.803
Approach 2	0.186	0.169	0.196	0.244	0.271	0.281	0.228	0.374	0.266
Approach 3	0.252	0.318	0.275	0.272	0.372	0.373	0.325	0.386	0.422

Table 5.4: Highest correlation between input signals for different number of topics (K) and sentiment per topic allocations

5.3 Model Construction

This section describes how different predictive models and alternations to the input data set are tested and compared, henceforth denoted as hyperparameters. In order to find the subset of hyperparameters to use in construction of the input data set, what kind of features to use and how many should be included, a grid search over the factors is conducted. Focus lies on narrowing down the scope by comparing cross-validation results for the different hyperparameters affecting the model. This analysis is conducted in several steps, starting out with logistic regression as the predictive model and the broad OMXS30 index as price data. Ideally the analysis would be conducted for each combination of classification model, predictive time horizon, type of text data and stock index, in order to tune every hyperparameter to the specific classification task. Due to the increasing complexity of such an approach and the time factor, this is deemed unfeasible to fit into the scope of this thesis and a more greedy approach is taken. There are also several factors kept constant throughout the thesis that also could affect the end result, but again, due to the time constraint these are decided upon before the grid search. Examples of such factors are the back-looking window, the sentiment analysis approach and the pre-processing of the text data.

The hyperparameter validation phase is carried out on the training partition of the data sets. For the full articles, there are two different data sets for which the analysis is carried out. The first of these spans from 2007 to 2020, where 2007-2018 constitute the training data set and 2019-2020 the test data set. The second data set spans from 2007 to 2016 where 2007-2015 constitute the training partition and 2016 the test data partition. The latter partition of the training and test data mentioned in this paragraph is also used for article summaries. The data sets of different lengths for the full articles are kept separate but due to the sequential nature of the data, there are some overlap in the data sets.

5.3.1 Hyperparameter Evaluation

The parameters evaluated are: number of topics in the topic model, different approaches for assigning sentiment to the topics, types of dimensionality reduction, type of labels inferred on the data and manual selection of topics. These are further described in table 5.5. The search is conducted for three predictive time horizons

(daily, weekly and monthly) and for the full news articles and news article summaries, this in order to investigate the feasibility of the method for different time horizons and parts of news articles. For each parameter combination forward selection is performed in order to find the best performing subset of features.

Parameter	Values
Number of topics	40, 45, 50, 55, 60, 65, 70, 75, 80
Sentiment per topic allocation method	Approach 1, Approach 2, Approach 3
Dimensionality reduction technique	No PCA, PCA, Sparse PCA
Labeling procedure	Binary, Multiple classes
Manual feature selection	Yes, No

Table 5.5: Hyperparameters to be tested

The hyperparameter tuning is carried out in several iterations each consisting of a grid search for the parameters stated above where some parameters are kept constant and some are compared. The scope is further reduced by omitting parameters and parameter values, and narrowing the grid from one iteration to next.

The hyperparameter tuning is conducted for the full articles and article summaries separately. For full articles two different time frames are tested and compared, 2007-2016 and 2007-2020. For summaries on the other hand only the one time period is tested for, 2007-2016. The reason for this is that the preambles normally constitute a large part of the article summary. These are missing from 2017 and there is therefore a large shift in the data that form the topic sentiment signals when using article summaries before and after 2017. The preambles contribute relatively little to the full article compared to the article summary. Thus, the data that form the topic sentiment signals are more coherent over time for full articles than for summaries for the period 2007-2020. It is worth noting this discrepancy in data between the time periods and considering this when comparing results from different time periods. The reasoning behind testing the full text article approach on both data sets is to be able to compare the summary approach to the full text approach, and to check if the anomaly in the data set affect the full article approach as well, on top of testing the approach for a stable time period not affected by the market turbulence in 2020.

The first iteration of the grid search is conducted for the training set of full articles (time frame 2007-2020), where the topic model was trained on only the training data. The search is performed over number of topics, sentiment per topic allocation, type of dimensionality reduction and labeling type. As stated above this is conducted with a logistic regression as classifier, and for the three different time horizons. For each combination the cross validation results are compared in order to reduce the dimension of hyperparameters, this is primarily based on leverage and lift over training data baselines and by comparing hyperparameter combinations.

After the primary grid search the grid is narrowed down by omitting parameters and parameter values. For example the range of topics is decreased by half only keeping K from 60 to 80, PCA and Sparse-PCA is deemed unnecessary and the first sentiment per topic allocation approach is dropped. Moreover, the features to be searched over in the forward selection is set to $\frac{K}{2}$ as maximum. This yield a more

narrow grid for the next search iteration.

After narrowing down the number features to be tuned as hyperparamters in the predictive models, a finer grid search is conducted separately for the traditional machine learning models and for the RNN. This finer search is carried separately for the two different time periods for the full articles as well as for the article summaries. The objective of the final grid search is primarily to find a suitable regularization factor for the machine learning models and to find a suitable model architecture and regularization factor for the RNN.

For logistic regression, a search over suitable regularization parameters (C) is conducted. A similar brief tuning of the model specific hyperparameters are conducted for SVC with a linear kernel and SVC with a RBF-kernel. For Random Forest the number of classifiers (decision trees) as well as the depth of each tree are the hyperparameters tuned. Otherwise the default settings of Scikit-learn are used (Pedregosa et al., 2011). The final model specific hyperparameters are presented in table 5.6.

Classification Model	Non-default model specific hyperparameters
Logistic Regression (LR)	Regularization Parameter C = 0.01
Linear SVC (LSVC)	Regularization Parameter C = 0.01
SVC RBF-Kernel (SVC RBF)	Regularization Parameter C = 0.1
Random Forest (RF)	# classifiers = 25, maximum depth = 2

Table 5.6: Model specific hyperparameter settings for traditional ML classifiers

For the RNN a grid search was carried out in a similar fashion as described above. The difference here is that since there are time dependencies of the feature inputs, a normal cross-validation can not be used and instead a sliding-window cross validation is used. The RNN models were trained using tensorflow with the default parameters (Abadi et al., 2015). The full setup of the initial grid search on the RNN is presented in table 5.7 and the parameters evaluated in this grid search is listed in table 5.10.

The same grid search setup defined in table 5.7, except for the kernel regularization factor, was later used for a second and final grid search for each of the models. In this grid search the the optimal regularization was determined before the testing. This was done by varying the kernel regularization between 0.01, 0.005 and 0001. Unlike the traditional machine learning models, the data was not retrained for the entire training period before testing, instead two thirds of the data was used for training and the final third was used as validation. This was done in order to be able to use an early stop approach on some unseen data. As the training accuracy converges towards 100 % rather quickly while the validation accuracy remains stable after a few epochs, an early stop is applied to minimize the risk of overfitting. This approach has the obvious drawback of making the model more biased to the validation set. It boils down to a trade-off between risk of overfitting on one hand and a biased model on the other hand. We determined that due to the discrepancy in the training and validation loss, the risk of overfitting overshadows the problems of having biased model to the validation set, ergo, the RNN models are not re-trained on all the training data.

Parameter	Values
Random seed	100
Optimizer	Adam with default parameters
Loss	binary cross entropy
Epochs	200
Batch size	54
Initial training length	67 % of training data
Validation folds	3
Early stop type	validation loss
Early stop patience	15
Kernel regularization type	L2
Kernel regularization factor	0.001

Table 5.7: Initial grid search setup for the RNN

Parameter	Values
Hidden layers	1, 2, 4, 5, 6
Dropout	0, 0.25, 0.5
Units	32, 64, 128
Manual feature selection	Yes, No
Sentiment per topic allocation method	Approach 2, Approach 3

Table 5.8: Hyperparameters to be evaluated for the RNN in the initial grid search

5.3.2 Final Hyperparameters

After iterating through different grid searches omitting unfeasible hyperparameters a final set of hyperparameters for full articles and summaries for both the traditional machine learning methods and the deep learning method are decided upon. These are presented in table 5.9 and table 5.10. When iteratively shrinking the hyperparameter space by grid searching and evaluating validation results several patterns are noticed. More topics in the topic model provide better results, dependency reduction by PCA and sparse PCA did not improve the accuracy of the model and results in a decreased interpretability of the input features and consequently, dependency reduction using PCA or sparse PCA were not used. Sentiment allocation approach 1 resulted in worse validation accuracy compared to the other approaches, and combined with the fact that the features were correlated to a much higher extent than for approach 2 and 3, approach 1 was omitted. Approach 2 was found to work best for article summaries and approach 3 for full article representation. Representing the classification task as a binary classification is preferred over having multiple classes. This was the case even after trying different tolerance levels for creating the multiple classes as described in section 5.2.4. As the validation results and feature selection suggest fewer features in the data set a final type of feature engineering was introduced. By manually selecting topics from each topic model, vague topics and topics referring to non market related factors are removed in advance, reducing the dimension of the feature selection problem. As an example this reduced 80 topics to 50 topics, keeping all topics that are in some way related to the industry, financial

markets or the economy as a whole. Topic 14 in figure 5.2 is for example omitted, as it mainly contains information about the days of the week.

Final Hyperparameters: Full Text Articles and Article Summaries - Traditional ML Models	
Parameter	Values
Number of topics	80
Dimensionality reduction technique	No PCA
Labeling procedure	Binary
Manual feature selection full text	Yes
Sentiment per topic allocation method for full articles	Approach 3
Sentiment per topic allocation method for article summaries	Approach 2

Table 5.9: Final hyperparameters for full length article and article summaries text representation

Final Hyperparameters for RNN Summary and Full text	
Parameter	Values
Hidden layers	80
Dimensionality reduction technique	No PCA
Dropout	0.25
Units	32
Labeling procedure	Binary
Manual feature selection	Yes
Sentiment per topic allocation method for Article Full Text	Approach 3
Sentiment per topic allocation method for Article Summaries	Approach 2

Table 5.10: RNN Hyperparameter settings used in the testing

6

Empirical Analysis

In this section the *results* are presented and *discussed*, focusing on the out-of-sample results i.e. the test data predictions. The performance on the out-of-sample data was very different from the in-sample data, as the models always performed well above the baseline on the in-sample data.

6.1 Results

6.1.1 Traditional Machine Learning

For the 2007-2020 time period there were no test results that were consistently above baseline, with one exception: weekly prediction for the bank index *SX3010PI*. This is true both for input data where the topic model was trained on the full data set and for the input data where the test data was left out of the topic model.

For the 2007-2016 time frame the test results were more frequently above the baseline, especially for the input data set where the topic model was trained on the full data set, but still, most of the predictions fell short of the baseline. The test results for the 2007-2016 time frame for which the topic model was trained on the full data set is presented below. The results are presented for each combination of predictive time frame, type of text, stock market index and classification model, and can be seen in table 6.1 to 6.6. Test results in bold were above the test set baseline, test results that were on par with the baseline are often explained by the classifier predicting only the most frequent class. For each predictive time frame, type of text and index the feature selection is based first on manual selection and then forward selection for a logistic regression, after which all models were trained on the same subset of features.

Daily predictions - Full Article					
Index	LR	SVC RBF	LSVC	RF	Baseline
OMXS30	**0.530**	0.506	**0.538**	**0.5138**	0.506
SX10PI TECHNOLOGY	0.510	**0.542**	0.510	0.490	0.510
SX20PI HEALTH CARE	0.494	0.498	0.478	0.470	0.510
SX35PI REAL ESTATE	0.557	0.557	0.546	0.542	0.557
SX50PI INDUSTRIALS	0.549	**0.553**	**0.553**	0.538	0.549
SX3010PI BANKS	0.506	0.486	0.498	0.510	0.514
SX4020PI CONSUMER P&S	0.498	0.490	0.506	0.478	0.530

Table 6.1: Daily prediction test results for the input data set spanning from 2007 to 2016 for each classification model and index. The topic model was trained on the full articles and on the test data. Results in bold outperform the baseline

Weekly predictions - Full Article					
Index	LR	SVC RBF	LSVC	RF	Baseline
OMXS30	0.466	0.542	0.463	0.510	0.542
SX10PI TECHNOLOGY	**0.553**	**0.593**	**0.546**	**0.538**	0.51
SX20PI HEALTH CARE	0.482	0.494	0.478	0.526	0.553
SX35PI REAL ESTATE	0.498	0.613	0.494	**0.648**	0.64
SX50PI INDUSTRIALS	**0.609**	0.593	0.601	0.569	0.593
SX3010PI BANKS	0.470	0.470	0.459	0.506	0.561
SX4020PI CONSUMER P&S	0.455	0.565	0.451	0.514	0.585

Table 6.2: Weekly prediction test results for the input data set spanning from 2007 to 2016 for each classification model and index. The topic model was trained on the full articles and on the test data. Results in bold outperform the baseline

Monthly predictions - Full Article					
Index	LR	SVC RBF	LSVC	RF	Baseline
OMXS30	**0.727**	**0.763**	**0.708**	**0.692**	0.668
SX10PI TECHNOLOGY	0.494	0.510	0.498	0.420	0.542
SX20PI HEALTH CARE	**0.636**	**0.636**	**0.644**	0.601	0.601
SX35PI REAL ESTATE	0.455	0.474	0.443	0.470	0.573
SX50PI INDUSTRIALS	0.617	0.712	0.585	0.719	0.763
SX3010PI BANKS	0.498	0.617	0.498	0.518	0.692
SX4020PI CONSUMER P&S	0.470	0.467	0.467	0.502	0.66

Table 6.3: Monthly prediction test results for the input data set spanning from 2007 to 2016 for each classification model and index. The topic model was trained on the full articles and on the test data. Results in bold outperform the baseline

Daily predictions - Article Summary					
Index	LR	SVC RBF	LSVC	RF	Baseline
OMXS30	**0.546**	0.506	**0.561**	**0.522**	0.506
SX10PI TECHNOLOGY	0.506	**0.514**	0.506	0.482	0.510
SX20PI HEALTH CARE	0.506	0.490	0.502	0.498	0.510
SX35PI REAL ESTATE	0.553	0.557	0.553	0.549	0.557
SX50PI INDUSTRIALS	0.5415	0.549	**0.561**	0.541	0.549
SX3010PI BANKS	0.498	0.510	0.502	0.474	0.514
SX4020PI CONSUMER P&S	0.494	0.470	0.494	0.494	0.530

Table 6.4: Daily prediction test results for the input data set spanning from 2007 to 2016 for each classification model and index. The topic model was trained on the article summaries and on the test data. Results in bold outperform the baseline

Weekly predictions - Article Summary					
Index	LR	SVC RBF	LSVC	RF	Baseline
OMXS30	0.502	0.542	0.498	0.526	0.542
SX10PI TECHNOLOGY	**0.549**	**0.546**	**0.542**	**0.522**	0.51
SX20PI HEALTH CARE	0.534	0.553	0.542	0.534	0.553
SX35PI REAL ESTATE	0.573	0.640	0.565	0.617	0.64
SX50PI INDUSTRIALS	0.534	0.593	0.534	0.565	0.593
SX3010PI BANKS	0.494	0.553	0.494	0.522	0.561
SX4020PI CONSUMER P&S	0.538	0.585	0.542	**0.613**	0.585

Table 6.5: Weekly prediction test results for the input data set spanning from 2007 to 2016 for each classification model and index. The topic model was trained on the article summaries and on the test data. Results in bold outperform the baseline

Monthly predictions - Article Summary					
Index	LR	SVC RBF	LSVC	RF	Baseline
OMXS30	0.546	0.609	0.542	0.553	0.668
SX10PI TECHNOLOGY	0.502	0.538	0.522	**0.546**	0.542
SX20PI HEALTH CARE	0.526	0.502	0.518	**0.629**	0.601
SX35PI REAL ESTATE	0.553	0.534	0.530	0.538	0.573
SX50PI INDUSTRIALS	0.553	0.613	0.542	0.620	0.763
SX3010PI BANKS	0.601	0.534	0.617	0.549	0.692
SX4020PI CONSUMER P&S	0.569	0.534	0.589	0.494	0.66

Table 6.6: Monthly prediction test results for the input data set spanning from 2007 to 2016 for each classification model and index. The topic model was trained on the article summaries and on the test data. Results in bold outperform the baseline

For the input data consisting of full news articles, weekly prediction for the technology index *SX10PI* and monthly prediction for the broad market index *OMXS30* resulted in all classifiers predicting better than baseline. For daily prediction of the *OMXS30* index, all the models fed with article summary representations, outside the SVC RBF-kernel, predicted above the baseline as well as outperformed the models that were fed the full article representations. In the tables 6.7, 6.8 and 6.9 the

topics used to construct the features are presented for these predictive time frames and indices. The features constructed by the topics presented in the tables were selected by forward selection. The Swedish counterpart of these tables are presented in appendix B.

Feature	Words in topics used to construct features
1	transaction, purchase, sell, business, buyer
2	business, company_as, entrepreneur, company_and, Swedish_company
3	stock_exchange, analyst, on_stock exchange, stock_market, investors
4	last_year, last_year, this_year, sale, increase
5	employee, connection, connection_with, in_relation, salary
6	profit, loss, earnings, million_krona, income
7	customer, turnover, employee, owner, earnings
8	medium, television, advertising, programs, networks
9	research, researchers, foundation, drugs, research_and
10	riksbank, central_bank, inflation, interest rate, currency
11	founders, investors, shareholders, venture_capital_company, capital
12	stock_exchange, trading, listing, on_stock exchange, change
13	opportunity, development, condition, need, future

Table 6.7: Topics used to construct features for the final model that predicted the weekly direction of SX10PI (technology index). The topics were inferred from the full articles and are represented by the five most important words translated into English. Original version in Swedish available in appendix B. The topics sentiments were selected in a forward selection approach.

Feature	Words in topics used to construct features
1	fund, return, fund_manager, general_pension_fund, saver
2	service, telephone, mobile, operator, Microsoft
3	order, contract, port, boat, ship
4	customer, turnover, employee, owner, earnings
5	product, technology, system, example, world
6	opportunity, development, condition, need, future
7	job, labor market, unemployment, work, person

Table 6.8: Topics used to construct features for the final model that predicted the monthly direction of OMXS30. The topics were inferred from the full articles and are represented by the five most important words translated into English. Original version in Swedish available in appendix B. The topics sentiments were selected in a forward selection approach.

Feature	Words in topics used to construct features
1	business_cycle, profitability, CEO, group, foundation
2	sales, sales_of, store, trend, listing
3	bid, turnover, ladder, bid_on, large_owner
4	message, goal, unemployment, people, message_of
5	top, Swedish, yesterday, project, research
6	Thursday, on_Thursday, expectation, oil, weekend
7	company, business, survey, battle, Swedish_company
8	capital, man, children, school, percent_of
9	Exchange, rise, Stockholm_exchange, Wednesday, European_exchange
10	bankruptcy, oil company, subsidiary, press_release, a_press_release
11	task, actor, trust, gold, auction
12	customer, view, term, credit_rating, door

Table 6.9: Topics used to construct features for the final model that predicted the daily direction of OMXS30. The topics were inferred from the article summaries and are represented by the five most important words translated into English. Original version in Swedish available in appendix B The topics sentiments were selected in a forward selection approach.

6.1.2 Deep Learning

Similarly to the traditional machine learning models, there were no test results that consistently outperformed the baseline for the time period 2007-2020. This was the case when using article summaries and full articles for topic models that was trained on the full data set and topic models that had only was trained on the training data set. Moreover, for both the 2007-2016 and 2007-2020 time periods, the accuracy was better when the model was fed daily topic sentiments, that was inferred from a topic model trained on the training data set together with the test data set. For the OMXS30 index in the 2007-2016 period, the data set where the topic model was trained on only the training data, the RNN always fell short of the baseline for full articles and outperformed the baseline with a very low margin, 0.003, for weekly predictions using article summaries. As was the case for the traditional machine learning models, the time period and setting that yield the best results were the 2007-2016 period when the topic model also was trained on the test data.

In tables 6.10, 6.11 and 6.12 the test result of the RNN model fed with the full articles and the article summaries for the time period 2007-2016, where the topic model was trained on the test data is presented. For monthly prediction, the predictive models never outperformed the baseline neither when the model was fed article summaries nor when it was fed the full text articles. In two cases the predictions were on par with the baseline as the models were poorly fitted and always predicted that the stock would go up. For daily, and weekly prediction the results are less clear. Some observation from the tables are that for *SX4020PI*, the consumer product and services index, the baseline is never outperformed. For daily and weekly prediction the full articles were better than the article summaries for the *SX10PI* index and vice versa for the *SX3010PI* index. For *SX3010PI* however the weekly predictions for both summaries and full articles were not above the baseline. For daily pre-

dictions, full articles outperformed the baseline three times and article summaries outperformed the baseline five times. For daily predictions, article summaries were better than full articles five times and full articles were better than article summaries three times. For weekly prediction the baselines were exceeded 3 times for both full articles and article summaries and they outperformed one another 3 times each.

RNN test results on OMXS indices 2016 daily predictions			
Index	Full Text	Summary	Baseline
OMXS30	0.486	**0.526**	0.506
SX10PI TECHNOLOGY	**0.538**	**0.518**	0.510
SX20PI HEALTH CARE	**0.514**	0.49	0.510
SX35PI REAL ESTATE	**0.569**	**0.569**	0.557
SX50PI INDUSTRIALS	0.534	**0.569**	0.549
SX3010PI BANKS	0.498	**0.557**	0.514
SX4020PI CONSUMER P&S	0.470	0.53	0.530

Table 6.10: Daily test results for the input data set spanning from 2007 to 2016, full data set was included in the topic model, for each index and article representation

RNN test results on OMXS indices 2016 weekly predictions			
Index	Full Text	Summary	Baseline
OMXS30	**0.557**	**0.553**	0.542
SX10PI TECHNOLOGY	**0.542**	0.494	0.51
SX20PI HEALTH CARE	0.522	**0.561**	0.553
SX35PI REAL ESTATE	0.49	0.565	0.64
SX50PI INDUSTRIALS	**0.601**	0.561	0.593
SX3010PI BANKS	0.514	0.526	0.561
SX4020PI CONSUMER P&S	0.569	0.569	0.585

Table 6.11: Weekly prediction test results for the input data set spanning from 2007 to 2016, full data set was included in the topic model, for each index and article representation

RNN test results on OMXS indices 2016 monthly predictions			
Index	Full Text	Summary	Baseline
OMXS30	0.668*	0.632	0.668
SX10PI TECHNOLOGY	0.462	0.462	0.542
SX20PI HEALTH CARE	0.553	0.593	0.601
SX35PI REAL ESTATE	0.51	0.502	0.573
SX50PI INDUSTRIALS	0.763*	0.751	0.763
SX3010PI BANKS	0.403	0.466	0.692
SX4020PI CONSUMER P&S	0.609	0.585	0.66

*Table 6.12: Monthly prediction test results for the input data set spanning from 2007 to 2016, full data set was included in the topic model, for each index and article representation. Entries marked with * yielded poorly fitted prediction models that always predicted up*

6.2 Discussion

Results were better when the topic model was trained on the full data set. The generated topic signals fed into the predictive modelling were thus compromised in a testing context, but this was necessary in order to infer topics for the test period that better reflected the contents of the newspaper for the test period. Topics covered in the news vary over time. For the data set covering 2007-2020 *Covid-19* or *Corona virus* were not mentioned in the training set, however it has been a factor that has been covered in the news and affected stock markets a lot during 2020. When the topic model is only trained on the training period, the topic model cannot detect emerging themes in the test data. When allowing the topic model to be trained on the test data, the topic sentiment signals constructed are representative of the information published in the news issues during the test period. If this approach would lead to bad results, topic sentiments could be disregarded as a suitable input for a stock prediction model. If, on the other hand, the results are promising when training the topic model on the test period, the method could be modified in order to capture new themes in the corpus. This could be done by retraining the topic model frequently. Training the topic model on the training and test period can to some extent be seen as a proxy for using a topic model that is retrained over time.

It is important to note that in this thesis only the topic model was at some point fed the test data set. The stock prediction models were of course not trained on the test period, meaning that if the predictive models work well, topics obtained in news paper data combined with sentiment analysis are useful for stock market prediction to some extent. As the results indicate that using a topic model that had been trained on the full data set improves performance compared to using a topic model that had not been trained on the test data, this could suggest that retraining of the topic model could improve predictive performance. Adding to the theme of retraining models as time progresses, another dimension that would be important, is to retrain the predictive model as well. Simply retraining the topic model will let the system detect new emerging topics. However, the predictive model will not be able to classify whether these topics should have a positive influence of the stock market or not.

On a general level the results were poor when the topic model had not been trained on the test data. Training the topic model on the full data set including the test data yield some promising results and in the remaining part of this section, these are discussed.

An observation that can be made by studying the tables 6.1 - 6.6 is that, for those indices and time horizons when the baselines were outperformed, the baseline was often outperformed by all the traditional machine learning models. This implies that the data representations fed to the models carry some valuable information and a test result over baseline is not attributed to chance.

It is important to note that the analysis that yielded the final set of hyperparameters were mainly carried out by examining validation results obtained for a logistic regression on the *OMXS30* index. In many cases the SVC RBF, LSVC and RF

outperform LR. For daily and monthly predictions when using full text articles, and daily predictions when using article summaries, the ML models were on par with or outperformed the baseline for *OMXS30*. It is therefore plausible that if the same analysis, as were carried out for the LR on the *OMXS30* index, were conducted for every index and model combination, more index - model combinations could outperform the baseline. This was however deemed to be an unreasonable scope for this thesis. It could also be that the *OMXS30* is an index which is easier to predict than smaller indices, but in the current setting this is impossible to establish.

Due to the high dimension of the input parameters affecting the end result, a large part of these were decided upon and kept constant throughout the thesis. This was done in order to reach a reasonable scope, fit the time frame and test the most interesting hyperparameters. A downside to this is of course that the these hyperparameters set a priori might have had a meaningful effect on the end result. Some examples of factors decided upon in advance: the decision to concatenate input signals for Saturdays and Mondays in order to utilize more data, the priors, α and β, in the topic modelling, the length and weighting of the back-looking window.

Due to the sequential nature of the input data (daily publications of news articles and daily closing prices of stock market indices) some overlap in the training and test sets is present for the 2007-2020 data set and the 2007-2016 data set. The initial grid search where some parameters were omitted was conducted on the training set of the 2007-2020 data set and inferred for the summaries as well as the full articles for the shorter time period 2007-2016. One could potentially argue that this is bad practice as test data for the shorter time period is not completely unseen. However, this was just the case for the initial coarse grid search and was followed by separate fine grained grid searches and hyperparameter tuning for the two different full article data sets, as well as for the full articles and article summaries data sets.

One hyperparameter that was determined a priori was the topic document prior of the topic model, i.e the α. In this work a symmetrical α where $\alpha = 5/K$ was used as a heuristic, as it has been proven to be a good choice when conducting topic modelling on Swedish news articles in previous research. Earlier works on Swedish news data have used articles which contain at least 300 words. When performing topic modelling on the article summaries, i.e. the title plus the preamble, the heuristic used for choosing the document topic prior might be insufficient, as the summary of an article might contain less topics than the full article.

For the RNN, the article summaries seem to give promising results for stock market predictions with a shorter time horizon but discouraging results for stock market predictions on a longer time horizon. For full articles, a similar, less strong pattern can be observed in the results. Correspondingly, the traditional machine learning models that use the full text articles generally outperformed those that used article summaries on a longer time horizon, while the article summaries generally outperformed the full articles for daily predictions. This similar pattern for traditional machine learning models that were fed representations with a fixed back-looking window, and for the RNN which learns how the information in sequential time steps should be modelled, could suggests that summaries are better for short time predictions and that the full articles are better for long time predictions. The observed

pattern could however also be the result of using different sentiment allocations for article summaries and the full articles. For article summaries approach 3 was used and for full articles approach 2 was used. It could be that different sentiment allocation methods are suitable for different predictive time horizons.

As aforementioned, one advantage of using an RNN, is that the model can learn structure in sequential information, without having to explicitly give the model the length and weights of a back-looking window. With the traditional machine learning models the length and the weighting of the back-looking window are a hyperparamters that are set a priori, but for the RNN these parameters are learned in the training. For daily predictions conducted with traditional machine learning models, only one day of topic sentiments were used, whereas for the RNN, information in the topic sentiments time series spanning over longer time periods could be learned and exploited, even for daily predictions. As the daily predictions for the RNN often outperformed the baseline as well as the traditional machine learning models, this could indicate that including a back-looking window for the traditional machine learning models could improve performance.

One major drawback of the RNN is the risk of overfitting. Due to this, training on the full training data set was deemed an unfeasible approach due to the necessity of having unseen data that could trigger an early stopping. This approach yielded promising results for daily predictions but is also problematic from a model construction point of view, where the norm is to re-train the model on all the training and the validation data before testing the model. If the data converges to 100 % accuracy on the training set but remains stable on the validation set, one could argue that there is a need to improve the grid search of the hyperparameters. RNNs however have a tendency to overfit and the data set is quite noisy. In earlier works concerning stock market prediction where RNNs have been successful, the task have normally been stock prediction where the model were fed news tagged with the stock ticker. This often yielded a large data set, as several individual stock could generate a direction in each time frame, while a stock market index generates only one. This could be the reason why earlier research have been more successful with using RNNs without having to use an early stopping approach. In addition to a larger data set, the models were also oftentimes fed technical indicators which might be an other reason why RNNs have been more successful in previous work.

The predictions on the data set for the time period 2007-2016 were better than those for the data set including the time period 2017-2020. This is in alignment with Larsen and Thorsrud (2017) which documented an underreaction in market prices to news on the Norwegian stock market. The authors of the paper show how the predictive power decreased over time, and that the news paper data used needed to be complemented with online media data to retain the predictive performance in more recent years. This decaying predictive power could be the reason for the decrease in performance when comparing the predictions made for the data set 2007-2016 and 2007-2020. It could however also be a consequence of the anomaly detected for the preambles in the time period 2017-2020 or by a change in the nature of the articles published in the newspaper.

A reason for the somewhat poor results could be attributed to the data source.

Using only newspaper articles from Dagens Industri, the coverage of the newspaper might be insufficient. This could be due to several reasons, for example using only one newspaper source there will be less reinforcement of a specific events and topics as a newspaper rarely publish two articles covering the same event in a single issue. Using several news data sources would on the other hand likely mean that if the same topic is covered to a high extent in several newspapers, the topic is very important for that day and that representations that rely on several news issues could reflect this. Moreover, the predictive power of the information published in Dagens Industri might not be as high as expected, even after improving the methods and using iterative retraining. Another possible reason for the poor results could be the lack of robustness analysis in form of detecting and handling outliers after the initial exploratory data analysis. Having outliers in the sentiment per topic signals could skew the input data and hence, reduce the predictive power.

7

Conclusion

In this chapter the *conclusions* reached in this thesis are presented. These are introduced with a short *summary* of the thesis, followed by the answer to the *research question*, the *contributions* of the thesis and finally *reflections* and suggestions for *future research*.

7.1 Summary

In this work, several methods have been explored to build an interpretable system that leverages topic modelling and sentiment analysis in order to predict the direction of Swedish stock market indices. We successfully managed to create a pipeline that transforms raw text data of a Swedish news corpus into daily topic sentiment signals. From the perspective of a human reader the topics generated by the model are coherent and reasonable. The sentiments of the topics seem to exhibit reasonable characteristics and reflect events that have occurred during different time periods. Unfortunately, the results were discouraging when predicting the topics and the direction of the index on the test period. By also allowing for the topic model to be trained on the test period, some encouraging results were obtained that allowed for some interesting observations. These observations lay a strong foundation for how future research should leverage topic modelling and sentiment analysis when predicting the direction of stock market indices.

7.2 Research Question

Can topic modelling and sentiment analysis on Swedish financial news paper data be combined and leveraged in order to predict the direction of Swedish stock market indices while keeping interpretability?

The method we have presented generates an interpretable model, however the results are insufficient to fully answer the research question. In the discussion we have described necessary steps that need to be taken in order to reach a more conclusive answer. These steps will be elaborated on when suggesting potential future research and in the reflections section below.

7.3 Contribution

This work has made several contributions to the research field. The literature review provides a foundation for future work on topic modelling combined with sentiment analysis on Swedish text data, especially in a financial context.

This work has also introduced a new, promising, sentiment allocation method for topic and sentiment modelling on daily news issues. For this new sentiment allocation method, the results suggest that the method outperforms the earlier used approach for full length news articles. This however should be tested in a more rigorous setting. In the results there are also indications that using summaries are better for short term predictions and full length articles on long term predictions.

In this work we have reinforced earlier findings on Swedish news data, that an LDA model used on the Swedish news corpus generate coherent topics, and also highlighted the importance of good methods for topic estimation and or topic model retraining, when a topic model is used for market prediction.

To conclude, the most prominent contribution of this work is paving the road for research focused on stock market index predictions based on Swedish news data.

7.4 Future Research

In the validation phase it was found that sentiment allocation method three was better for full articles and allocation method two was better for article summaries. In the results it was furthermore found that article summaries were better on short term prediction and full text articles were better for long term predictions. The better performance of the article summaries could potentially therefore be a consequence of the different sentiment allocation methods. Future work could test using both sentiment allocation method two and three on both the article summaries and on the full articles to verify if sentiment allocation method two actually is better than sentiment allocation method three on article summaries and vice versa for the full articles. This test setup could also verify that article summaries are better than full articles for long term prediction, without having the difference in sentiment allocations affecting the results. A test similar to this was carried out in the validation phase and to some extent on the test data, but should be done in a more rigorous test setting. Future work could also try new sentiment allocation methods as there are still plenty of options left to be explored in this area.

Future research could include more advanced sentiment analysis methods. For instance an approach similar to the Vader algorithm could be used, where the words preceding the positive or negative words, i.e. the dictionary words, are being taken into account. In case of a negation before the dictionary words the sentiment score of the dictionary word could be multiplied by a negative factor, and for a positive reinforcement word, the sentiment score of the dictionary word could be multiplied with a positive factor. A field that is currently expanding and where sentiment analysis has shown big promise is semi-supervised learning. Methods that rely on semi-supervised learning could be used and evaluated against the lexical approach used in this work as the method for sentiment analysis. In addition to using more advanced methods for inferring the sentiment of an article, it could be worth to investigate if different parts of the text should be given a different weight when calculating the sentiment of the article, e.g. give the title and the preamble twice the weight. Moreover, the articles that occur in different parts of the news issue could be given a different importance weighting, e.g. weighting differently based on the page number.

In this thesis the default symmetric priors were used in the topic models as it has been shown in previous work to yield satisfactory results on Swedish news articles. When evaluating the topics generated by the models these also look reasonable at glance but we think there could be a difference in how the priors of the topic models should be chosen when having full text articles or article summaries. With the assumption that an article summary covers less topics, a lower document-topic prior, i.e α, should be used. Efficient heuristics for establishing how the priors should be set when using the full text article, the article summary or only article headlines is an area of research that could improve the methods presented in this work.

As suggested by the results of this thesis, it could be beneficial to include sequential data in form of a back-looking window for daily predictions conducted with traditional machine learning methods. We encourage research on how this back-looking window should be formed, i.e. the length of such a window and how to determine the weighting for each data point included. This suggestion also covers weekly and monthly predictions, where more research is needed on how much news data from the past should be included when predicting the future movements of stock indices. In this thesis these back-looking windows are kept, more or less, constant with an equal weighting on each data point. This needs to be explored further, and is believed to improve the performance of time lengths presented in this thesis, not just for daily predictions.

The results in this work show that a topic model that has been trained on the training data as well as test data yields topic sentiment signals that a stock market prediction model is better at capturing when predicting on "unseen" data. We hypothesize that it could be due to shifts in themes that are covered in the news issues. Future work could try to establish and quantify the relationship between frequency of retraining the model and improvement of the predictive model. More specifically future work could test daily, weekly, and monthly re-training of topic model to see this relationship. In addition to retraining the topic model, the classification model should be retrained as well, so that new themes are not just caught by the system, but so that it also learns how these new occurring themes in the

news affect the stock market. On a similar note, when the topic model is trained only on the training part of the data set, the validation results are better. This could indicate that training the topic model on more data, from a long time period, counter-intuitively makes the topic signals worse. Usually in data science the more data the better is the motto to adhere to, but this might not be the case for topic modelling of news data as there are shifts in the themes covered. An approach that could increase the performance of the predictive model, and should be tested, is to use a sliding window for the news data to be included in the topic model, combined with continuous retraining.

To test if the predictive performance of printed Swedish newspaper data on the Swedish market data actually decay over time, as has been observed in previous work for the Norwegian market, one could redo the research and tests performed in this thesis over several time periods. On the same note, one could include online news data, to see if this combat the decay in predictive power of the printed newspaper data.

One especially pressing suggestion for further research is to perform similar research as in this thesis and including other forms of text representations such as embedding representations. Using topic modelling and sentiment analysis in combination might be a good solution to keep the interpretability of the text data throughout every step of the method. But, there might very well be methods separated from topic modelling and sentiment analysis that can capture the information in the newspaper data better. We strongly suggest that such methods are researched further in the field of stock market index prediction on the Swedish market using Swedish news data.

In this work we have explored several methods for sentiment analysis and topic modelling and how these can be combined and leveraged for stock market predictions. These methods are potentially interesting for other areas of research. Examples include using topic sentiment signals created from Swedish news data to predict outcomes of political elections, unemployment or GDP.

As a final note, no significance test was used when analyzing the results of this work. Future work where there is a more rigorous test setup and the results outperform the baseline could benefit from using significance test when presenting the results.

7.5 Further Reflections

Writing this thesis, it became evident to us that the scope of this thesis was overreaching given the time frame. There were many potential branches to explore and to evaluate, increasing the dimensions of the research as the work progressed. A more narrow scope would reduce the methods explored but would likely yield more conclusive answers to which methods are best in different settings. As an advice for others attempting similar research we suggest to narrow the scope more than what is believed necessary.

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

F. Ahnve, K. Fantenberg, G. Svensson, and D. Hardt. Predicting stock price movements with text data using labeling based on financial theory. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4365–4372, 12 2020.

E. Angner. *A course in behavioral economics*. Palgrave Macmillan, London, 2016. ISBN 978-1137512925.

J. Blad and K. Svensson. Exploring nmf and lda topic models of swedish news articles. Master's thesis, Uppsala University, Uppsala, 12 2020.

J. Bollen, H. Mao, and X.-J. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2:1–8, 10 2010.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 10 2001.

D. Bzdok, N. Altman, and M. Krzywinski. Statistics versus machine learning. *Nature methods*, 15(4):233–234, 4 2018.

K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing)*, pages 1724–1734, 10 2014.

Di. Dagens industri: Om oss. URL https://www.di.se/nyheter/om-oss/, 2020. Accessed: 2021-05-03.

L. dos Santos Pinheiro and M. Dras. Stock market prediction with deep learning: A character-based neural language model for event-based trading. In *Proceedings of the Australasian Language Technology Association Workshop 2017*, pages 6–15, 12 2017.

E. Fama. The behavior of stock market prices. *Journal of Business*, 38:34–105, 1 1965.

E. Fama. Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25:383–417, 5 1970.

S. Feuerriegel and J. Gordon. Long-term stock index forecasting based on text mining of regulatory disclosures. *Decision Support Systems*, 1:88–97, 8 2018.

I. Fisher, M. Garnsey, and M. Hughes. Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research. *Intelligent Systems in Accounting, Finance and Management*, 23:157–214, 07 2016.

G. Friesen and P. Weller. Quantifying cognitive biases in analyst earnings forecasts. *Journal of Financial Markets*, 9:333–365, 10 2002.

M. Gerlach, H. Shi, and L. A. N. Amaral. A universal information theoretic approach to the identification of stopwords. *Nature Machine Intelligence*, 1:606–612, 12 2019.

Y. HaCohen-Kerner, D. Miller, and Y. Yigal. The influence of preprocessing on text classification using a bag-of-words representation. *PLOS ONE*, 15(5):1–22, 05 2020.

M. Haselmayer and M. Jenny. Sentiment analysis of political communication: combining a dictionary approach with crowdcoding. *Quality & Quantity*, 51:2623 – 2646, 11 2017.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 4 1982.

C. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, 1 2015.

P. Hájek and A. Barushka. Integrating sentiment analysis and topic detection in financial news for stock movement prediction. In *ICBIM '18: The 2nd International Conference on Business and Information Management*, pages 158–162, 9 2018.

I. Jolliffe and J. Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374:20150202, 4 2016.

M. Juršič, I. Mozetič, T. Erjavec, and N. Lavrač. Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16(9):1190–1214, 5 2010.

A. Khadjeh Nassirtoussi, S. Aghabozorgi, T. Ying Wah, and D. C. L. Ngo. Text mining for market prediction: A systematic review. *Expert Systems with Applications*, 41(16):7653–7670, 11 2014.

M. Kim, E. Park, and S. Cho. Stock price prediction through sentiment analysis of corporate disclosures using distributed representation. *Intelligent Data Analysis*, 22:1395–1413, 12 2018.

E. Kumar. *Natural Language Processing*. I.K. International Publishing House Pvt. Limited, 2011. ISBN 9789380578774.

V. Larsen and L. Thorsrud. The value of news for economic developments. *Journal of Econometrics*, 210:203–218, 5 2019.

V. H. Larsen and L. A. Thorsrud. Asset returns, news topics, and media effects. Working Papers No 5/2017, Centre for Applied Macro- and Petroleum economics, BI Norwegian Business School, 10 2017. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=305795.

X. Li, H. Xie, Y. Song, Q. Li, S. Zhu, and F. L. Wang. Does summarization help stock prediction? a news impact analysis. *IEEE Intelligent Systems*, 30:1, 6 2015.

X. Li, P. Wu, and W. Wang. Incorporating stock prices and news sentiments for stock market prediction: A case of hong kong. *Information Processing & Management*, 57:102212, 2 2020.

Y. Liu, J. Trajkovic, H.-G. H. Yeh, and W. Zhang. Machine learning for predicting stock market movement using news headlines. In *2020 IEEE Green Energy and Smart Systems Conference*, pages 1–6, 11 2020.

T. Loughran and B. Mcdonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66:35 – 65, 2 2011.

B. Lutz, N. Pröllochs, and D. Neumann. Predicting sentence-level polarity labels of financial news using abnormal stock returns. *Expert Systems with Applications*, 148:113223, 6 2020.

A. K. McCallum. Mallet: A machine learning for language toolkit. URL http://mallet.cs.umass.edu, 2002. Accessed: 2021-03-20.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2, page 3111–3119, 12 2013.

T. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997. ISBN 9780071154673.

S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, and D. C. Anastasiu. Stock price prediction using news sentiment analysis. In *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 205–208, 4 2019.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2013. ISBN 9780262018029. Available Online. URL https://mitpress.mit.edu/books/machine-learning-1.

A. E. Nagib, M. Mohamed Saeed, S. F. El-Feky, and A. Khater Mohamed. Neural network with adaptive learning rate. In *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference*, pages 544–548, 10 2020.

T. Nguyen and K. Shirai. Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 1354–1364, 7 2015.

K. Pearson. On lines and planes of closest fit to points in space. *Philosophical Magazine*, 2:559–572, 11 1900.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

A. Ratto, S. Merello, Y. Ma, L. Oneto, and E. Cambria. Technical analysis and sentiment embeddings for market trend prediction. *Expert Syst. Appl.*, 135:60–70, 11 2019.

Retriever. Retriever mediearkivet. URL https://www.retriever.se/tag/mediearkivet/, 2021. Accessed: 2021-02-21.

S. Robertson. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60:2004, 10 2004.

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 1 2015.

Y. Shao, C. Hardmeier, and J. Nivre. Universal word segmentation: Implementation and interpretation. *Transactions of the Association for Computational Linguistics*, 6:421–435, 12 2018.

A. Shapiro, M. Sudhof, and D. Wilson. Measuring news sentiment. *Journal of Econometrics*, 11 2020.

M. G. Sousa, K. Sakiyama, L. d. S. Rodrigues, P. H. Moraes, E. R. Fernandes, and E. T. Matsubara. Bert for stock market sentiment analysis. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1597–1601, 11 2019.

M. Straka and J. Straková. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, 8 2017.

S. Syed and M. Spruit. Exploring symmetrical and asymmetrical dirichlet priors for latent dirichlet allocation. *Int. J. Semantic Comput.*, 12:399–423, 9 2018.

TakeLab. Spacy-udpipe. URL https://pypi.org/project/spacy-udpipe/, 2020. Accessed: 2021-03-10.

P. Tetlock. Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, 62:1139–1168, 2 2007.

L. A. Thorsrud. Words are the new numbers: A newsy coincident index of the business cycle. *Journal of Business & Economic Statistics*, 38(2):393–409, 4 2020.

M. Vargas, C. Anjos, G. Bichara, and A. Evsukoff. Deep learning for stock market prediction using technical indicators and financial news articles. In *2018 International Joint Conference on Neural Networks*, pages 1–8, 7 2018.

T. Wisniewski and B. Lambe. The role of media in the credit crunch: The case of the banking sector. *Journal of Economic Behavior & Organization*, 85:163–175, 10 2010.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

Appendix A

(Word clouds in Original Language)

Figure A.1: Original version of 5.2 showing the topic representations of the inferred from the LDA models

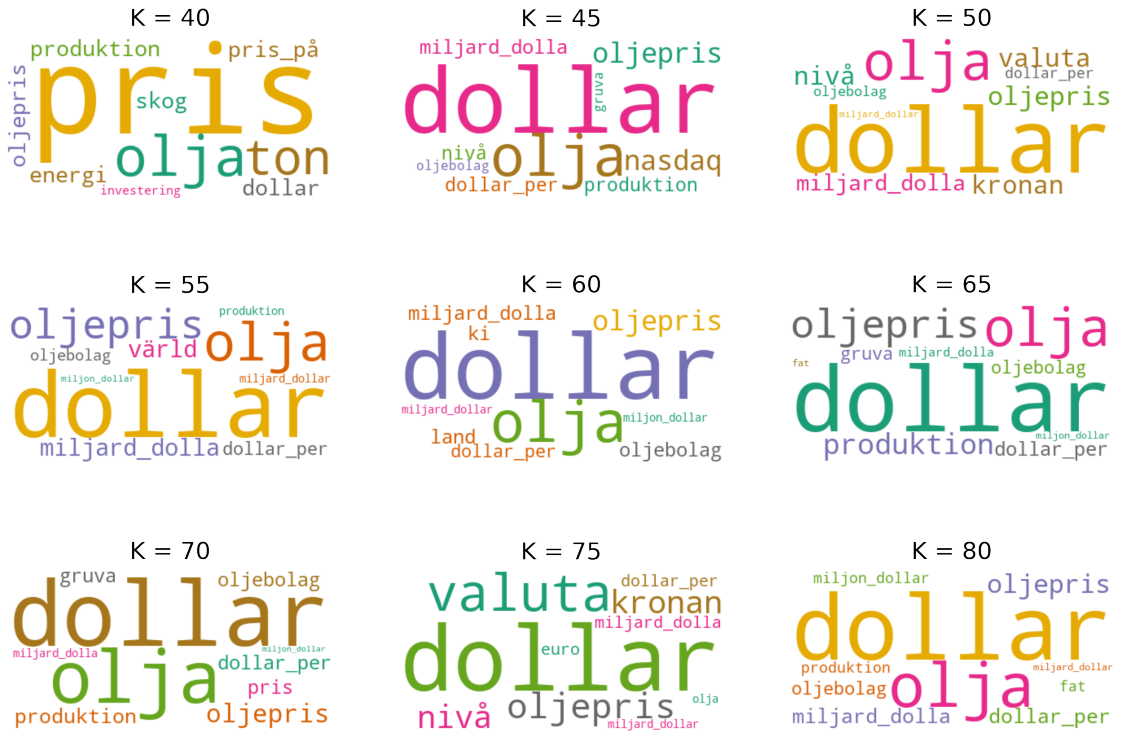

Figure A.2: Original version of 5.3 showing the topic representations of the inferred from the LDA models

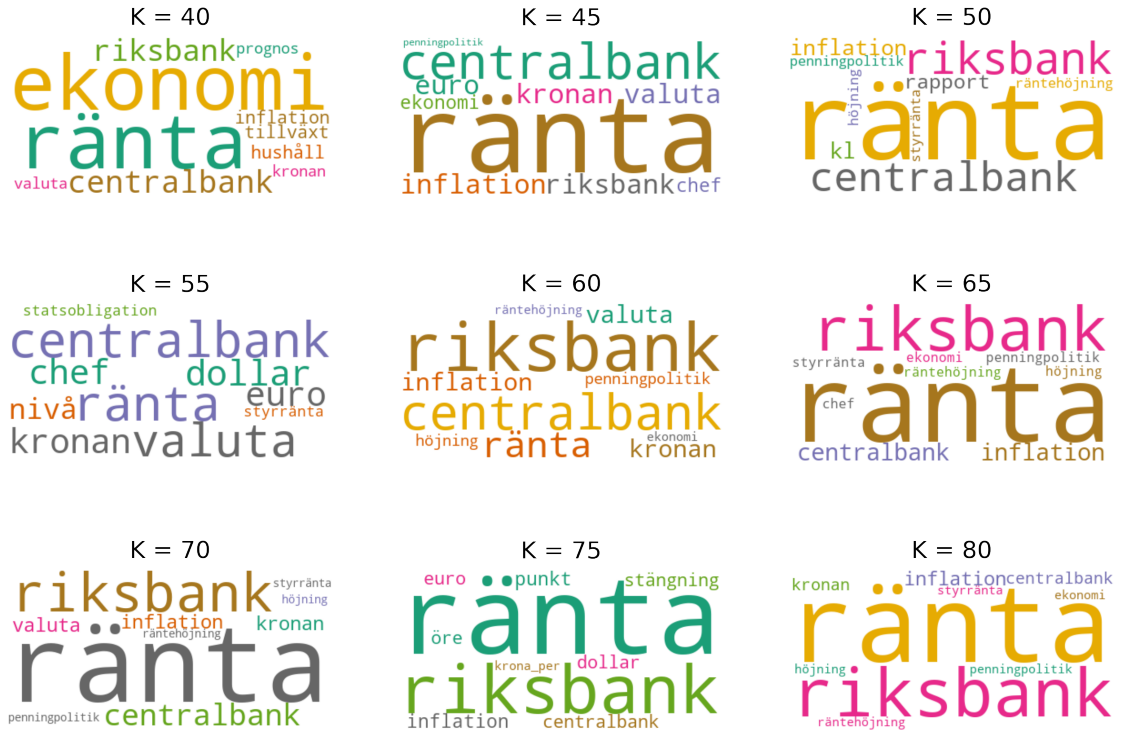

Figure A.3: Original version of 5.4 showing the topic representations of the inferred from the LDA models

Appendix B

(Topics in Original Language)

Words in topics used to construct features
affär, köp, försäljning, verksamhet, köpare
näringsliv, företag_som, företagare, företag_och, svensk_företag
börs, analytiker, på_börs, aktiemarknad, placerare
fjol, i_fjol, i_åra, försäljning, ökning
anställd, samband, samband_med, i_samband
vinst, förlust, resultat, miljon_krona, intäkt
kund, omsättning, anställd, ägare, resultat
medium, tv, reklam, program, nät
forskning, forskare, stiftelse, läkemedel, forskning_och
riksbank, centralbank, inflation, ränta, valuta
grundare, investerare, delägare, riskkapitalbolag, kapital
börs, handel, notering, på_börs, förändring
möjlighet, utveckling, förutsättning, behov, framtid

Table B.1: Topics used to construct features for final model used to predict the 5 day movement for SX10PI (technology) index for full article representation, described by top five words per topic

Words in topics used to construct features
fond, avkastning, förvaltare, ap_fond, sparare
tjänst, telefon, mobil, operatör, microsofta
order, kontrakt, hamn, båt, fartyg
kund, omsättning, anställd, ägare, resultat
produkt, teknik, system, exempel, värld
möjlighet, utveckling, förutsättning, behov, framtid
jobb, arbetsmarknad, arbetslöshet, arbete, person

Table B.2: Topics used to construct features for final model used to predict the 5 day movement for SX10PI (technology) index for full article representation, described by top five words per topic

Words in topics used to construct features
konjunktur, lönsamhet, koncernchef, koncern, stiftelse
försäljning, försäljning_av, butik, trend, notering
bud, omsättning, stege, bud_på, storägare
besked, mål, arbetslöshet, folk, besked_om
topp, svenska, gårdag, projekt, forskning
torsdag, på_torsdag, förväntning, olja, helg
företag, verksamhet, undersökning, strid, svensk_företag
kapital, människa, barn, skola, procent_av
börs, uppgång, stockholmsbörs, onsdag, europeisk_börs
konkurs, oljebolag, dotterbolag, pressmeddelande, en_pressmeddelande
uppgift, aktör, förtroende, guld, auktion
kund, utsikt, sikt, kreditbetyg, dörr

Table B.3: Topics used to construct features for final model used to predict the 1 day movement for OMXS30 index for article summary representation, described by top five words per topic