Learning Algorithms for Information Extraction from Foreign Exchange Market News

A Non-Stop Prediction
of
Economic-News-Sentiment

Thomas Wiesner, Bsc.

Master's Thesis

Learning Algorithms for Information Extraction from Foreign Exchange Market News

Thomas Wiesner

Institute for Information Systems and Computer Media (IICM)
Graz University of Technology

THE UNIVERSITY OF WESTERN AUSTRALIA

Advisor:
Univ.-Doz. Dipl.-Ing. Dr. techn. Christian Gütl

Co-Advisor:
Prof. Wei Liu

Graz, May 2014

Diplomarbeit

Lern-Algorithmen für die Informationsgewinnung von Nachrichten des Fremdwährungsmarktes

Thomas Wiesner

Institut für Informationssysteme und Computer Medien
Technische Universität Graz

THE UNIVERSITY OF WESTERN AUSTRALIA

Betreuer:
Univ.-Doz. Dipl.-Ing. Dr. techn. Christian Gütl

Mitbetreuer:
Prof. Wei Liu

Graz, Mai 2014

Abstract

With a daily trading volume of more than 4 trillion US Dollar, according to the Bank for International Settlements, the Foreign Exchange (Forex) market is the largest market worldwide. It is also, beside being one of the most risky markets, a highly available market, with the possibility to trade 24h a day, excluding weekends. Compared to the New York Stock Exchange (NYSE), where stocks worth 153 billion US Dollar are exchanged daily on the floor, the Forex-Market has about 25 times higher daily turnover.

While the Stock market is a highly researched field, the Forex market seems to be of no such great interest. One of the biggest challenges for traders on the Forex market is the continuous flow of new information with a possible impact on the market. Furthermore it is not possible that a Forex trader is always online and prepared for sudden changes in the economic situation, which leads to sudden shifts of exchange rates. Therefore a way of filtering news on the fly is required, and, furthermore, a way of automatically trading on the Forex market based on the sentiment of news articles is desired.

This thesis researches an approach to extract information from continuously available news from the internet on a non-stop basis. It shows various ways to harvest news from different sources, how to filter unwanted topics utilizing text-clustering methods and how to prepare them with currently available and open Natural Language Processing tools. After preprocessing, machine learning techniques are used to do a sentiment-labeling on new articles using Support Vector Machines (SVM). The tasks are perfomed on 208.989 news articles between January and February 2012. The topic filtering, to get only *interesting* news, shows an accuracy of 70.5% on random samples. The classification based on the remaining samples shows an accuracy of 75.19%.

Kurzfassung

Mit einem täglichen Handelsvolumen von mehr als 4 Trillionen US Dollar, ist der Fremdwährungsmarkt (Forex-Market), laut der Bank für International Zahlungsausgleich, der größte Handelsmarkt weltweit. Er ist nicht nur einer der risikoreichsten Märkte, sondern auch hochverfügbar, mit der Möglichkeit 24 Stunden pro Tag zu handeln, ausgenommen Wochenends. Verglichen mit der New York Stock Exchange (NYSE), wo täglich Aktien mit einem Volumen von 153 Milliarden US Dollar gehandelt werden, hat der Fremdwährungsmarkt einen 25-mal höheren Umsatz.

Während der Aktienmarkt gut erforscht ist, scheint der Fremdwährungsmarkt kein so großes Interesse für die Forschung zu wecken. Eine der größten Herausforderungen für Händler am Frendwährungsmarkt ist der kontinuierliche Fluss an neuen Nachrichten mit einer möglichen Auswirkung auf die Wechselkurse. Weiters sind die Händler nicht immer bereit auf plötzliche Änderungen in einer Ökonomischen Situation, welche zu plötzlichen Änderungen in den Wechselkursen führt, zu reagieren. Deshalb ist eine neue Filtermethode gebraucht, die Nachrichten spontan filtern kann. Weiters ist es gewünscht, dass ein Weg gefunden wird, wie man am Fremdwährungsmarkt automatisch handeln kann, basieren auf der Stimmung die aus den Nachrichten hervorgeht.

Diese Arbeit erforscht eine Möglichkeit der ununterbrochenen Informationsgewinnung von ständig verfügbaren Nachrichten aus dem Internet. Die Arbeit zeigt verschiedene Wege um Nachrichten zu gewinnen, ungewünschte Themen mittels Text-Clustering Methoden auszufiltern und die Nachrichten mit aktuellen und offenen Natural Language Processing Werkzeugen aufzubereiten. Nach der Vorbereitung wird mittels maschinellen Lernens, im Speziellen Support Vector Machines (SVM), die Stimmungskennzeichnung auf neue Artikel vorgenommen. Die Aufgaben werden auf 208.989 Artikel von Jänner und Februar 2012 angewandt. Themenfilterung, um nur *interessante* themenrelevante Artikel zu erhalten, funktioniert mit einer Genauigkeit von 70,5% bei zufällig ausgewählten Proben. Klassifikation, basierend auf den restlichen Problen, funktioniert mit einer Genauigkeit von bis zu 75.19%.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Ort	Datum	Unterschrift

Acknowledgements

Encouragement, unlimited support and persuasion are words I connect with what I received from my family for years and years. Therefore my deepest gratitude is expressed first and foremost to my family. To my mother having always an open ear, to my father knowing always a great way to overcome times of frustration. In particular I want to acknowledge the way my grandmother was, is and will always play a main part in my life.

I want to express my deepest appreciation to my professor and supervisor Dr. Christian Gütl, who gave me the opportunity to develop my ideas overseas in Australia. You have been an outstanding mentor for me. Not only was I blessed meeting my far beyond outstanding co-supervisor in Australia, but I was also privileged to meet my future wife and one of the best companions and friends during my stay. Therefore I want to express my deepest gratitude to Dr. Wei Liu, lecturer at the SCSSE at the UWA in Perth, for her open door, open ear and great comments. Without her guidance I wouldn't have been able to finish this thesis. Furthermore I want to thank Lau Tsz Lam for spending sleepless nights and supporting me, cheering me and keeping the harmony at all times. I want to thank Dean Clemente for spending countless hours with me, having always an open ear for problems, responding with great ideas, or finding an unconventional way to overcome a challenge.

Last but not least I want to thank Andreas Schwarz, Johannes Löffler and Christian Unterkofler, who are companions and highly respected colleagues for years, as well as my unconditional supporting friends, Gerulf Schnitzler, Martin Gutmann, and my employer Martin Moschitz. Without their support I would not even come remotely close to writing these lines.

Graz, May 2014 Thomas Wiesner

"I am always doing that which I cannot do, in order that I may learn how to do it."

Pablo Picasso

Contents

List of Figures

List of Tables

1. Introduction

"The way to make money is to buy when blood is running in the streets."

[John D. Rockefeller]

1.1. Motivation and Background

While the Stock market is often a myth and only few common people are active traders on that market, an enormous amount of people were already active as Foreign Exchange traders without even actively taking notice. If one travels to a country with a different currency other than the currency used in the home country, a currency conversion is needed. This can be for example Euro to US Dollar, or to Australian Dollar, or Japanese Yen (¥), just to mention a few. The home currency is converted to the target currency with a specific exchange rate, while the rate changes continuously. During writing these lines, the Euro has against the US Dollar an exchange rate of 1:1.3033, which means: in exchange for 1 Euro one get 1 US Dollar and about 30 US Dollar-Cents. One main question naturally rises: Is this the best time for the exchange? In contrast to professional trading, the small amount which is usually exchanged for traveling to foreign countries and the proportional high fees that occur during a typical transaction, a small change in the exchange rate doesn't have a high influence in the amount of money which is available for traveling. In other words: a huge amount of money must be transferred to get a reasonable profit (or loss) if the exchange rate changes.

The Foreign Exchange market seems to be something that many people find easy. Access is made comfortable. People can trade from home using their own computer with access 24 hours a day, on every working day, with the possibility to buy, or sell one currency against another. Compared to almost any other profession, this seems like the holy grail for a money making machine without the need to study for a long time, or to study at all. On the other hand it is a highly competitive market. There are various numbers regarding the failure rate of traders in numerous forums and blogs, most of them claim a 95:5 theory: 95% of the traders fail, while the other 5% actually make profit. The numbers for loss versus profitability vary mostly between 80:20 and 99:1, meaning the common opinion that the Internet community suggests is that, on

average, traders are making between 80% to 99% loss. While, indeed, these numbers seem to support the common sense when browsing through different forums, the actual percentage of profitable active traders is higher (see Figure 1.1). The Forex Magnates (2011) website gave quarterly insights from major brokers, which are regulated and publish their statistics. These numbers show that, on average, 60.9% of the 24,068 active trading *accounts* at the broker Oanda are not making any money. Also there are rumors that the 95:5 theory is related to all, ever existed, trading accounts, while Forex Magnates (2011) are calculating their statistics based only on active accounts.

	Q4 2011			Change from Q3 2011		
	% Profit	% Loss	Total Accounts	% Profit	Accounts	Accounts Growth
Oanda	39.1%	60.9%	24,068	3.6%	(5,676)	-19.1%
GFT	39.0%	61.0%	9,656	13.7%	(447)	-4.4%
Interbank FX	38.0%	62.0%	9,234	10.0%	(1,775)	-16.1%
PFG	36.7%	63.4%	1,959	14.6%	(234)	-10.7%
FXDD	33.7%	66.3%	6,055	6.3%	(509)	-7.8%
Gain Capital	34.0%	66.0%	11,404	4.0%	(1,443)	-11.2%
FXCM	30.0%	70.0%	19,787	0.0%	(721)	-3.5%
Alpari	32.5%	67.5%	2,237	1.7%	(29)	-1.3%
FX Solutions	32.0%	68.0%	4,640	5.0%	(237)	-4.9%
MB Trading	31.8%	68.2%	4,546	8.0%	(513)	-10.1%
FX Club	29.5%	70.5%	1,656	8.9%	161	10.8%
TradeStation	28.0%	72.0%	1,919	1.0%	139	7.8%
Advanced Markets	42.2%	57.8%	45	0.0%	-	0.0%
Average profitability change				6.4%		
Weighted profitability change				5.2%		
Total change in number of accounts					(11,284)	
Total number of accounts in the US			97,206			

Figure 1.1.: OANDA is still the most profitable broker with 39% of its clients being profitable in Q4 2011. The "Q4 2011" Column shows that the loss versus profitability theory of 99:1 or 95:5 is not supported. In fact even the last Broker on the list, "TradeStation" still has 28% profitable and 72% failing traders across 1919 accounts, which would be 72:28. [Image taken from Forex Magnates (2011)]

Among a lot of different tools, choices and possibilities that are available for Forex trading, one of the most dangerous and underestimated, for new traders, is the ability to use a leverage, according to Rosenstreich (2005). The leverage gives the trader the possibility to use more money for trading than the amount he actually has in his account. For example a trader has in his account 100 US Dollar and uses a leverage of 100:1, so it is possible to trade with 10.000 US Dollar, or in other words, the market is controlled with 10.000 US Dollar. This looks especially good for new traders who think that they can easily gain a huge profit. In the eyes of a new trader, with a leverage, the amount of money can easily be doubled. What they don't see is, while their chances

of profit raise, their chances of loss raise as well. Mistakes are made. Especially to new traders it is a matter of emotions when they see their amount of money fluctuate (Toshchakov, 2006). As a result, new traders often lose a lot of money in a relatively short amount of time and walk away utterly disappointed.

According to Sager and M. P. Taylor (2006, p. 83), the Foreign Exchange market is usually divided in two types: broker (or *dealers*) and *customers*. While dealers, also known as market makers, are responsible for about 59% of the daily market turnover, the remaining 41% are customers which can be further divided in various ways. Generally speaking customers are in need of a broker to access the interbank market. So, when a customer wants to trade currency pairs on the Forex market, there is almost always a broker between him and the bank. Normal people have, since the introduction of electronic systems in the Forex market, direct access to the intrabank market through multibank electronic trading portals like FX Connect or FXAll, where the portal acts as a broker. Now, not only normal human beings are customers, but also banks them self can become customers of the market, when they trade. Furthermore the customers can be classified in two groups: "informed" and "non informed" customers. An informed customer usually refers to someone who has more information about the market than other traders. Central banks are the main participants on the market as "informed" customers. These banks trade on the market, but also provide access and can act as a broker. They have not only access to the data series of their own currency in advance to their customers, but also, in addition, they have access to, at least, the same information as their customers in the Foreign Exchange sector. They could act before the customers can, can influence the market and also stabilize it. To observe the reaction and movements of central banks in the Foreign Exchange sector is one of the most interesting informations. Unfortunately, instead of publishing information of the movements and reaction of banks to the public, the information is made available anonymously only to insiders (Sager & M. P. Taylor, 2006, p. 87).

When new information and news is available, different customers react within a different time frame. One of the fastest are hedge funds. While hedge funds are not only within the quickest, they also use a higher leverage than other customers. Sudden shifts of their assets can amplify shocks to the Foreign Exchange market. The Relational Expectations Hypothesis (REH) is *"the hypothesis that an economic agent will make full use of all available information when forming expectations, especially with regard to inflation, and not just past values of a particular variable."* according to Dictionaries (2013). This means, if the REH holds and if economic agents would have full access to all available information, they would form their expectation unbiased (Miah, Hassan, & Rahman, 2004) and the information is reflected immediately in exchange rates (Sager & M. P. Taylor, 2006). But not all customers trade on their own. Different companies provide services to trade for large clients, these are called *currency overlay firms*. Practical observations showed that currency overlay firms, for example companies who

trade for large institutions like pension funds, reduce active hedges gradually. This is eventually also due to the desire to minimize the impact of the transfer of huge assets and to maximize the consequential return. (Sager & M. P. Taylor, 2006)

When traditional asset price models are taken under consideration, like

$$S_t = \beta' F_t + \alpha S^e_{t+1}, \tag{1.1}$$

then the current spot rate is reflected mainly with the future spot rate S^e_{t+1} with the information available at time t. A spot rate is a market expectation for future price movements. A common mistake is the misinterpretation of asset models as strict mathematical models. The future spot rate (S^e_{t+1}) is only true, if the market reflects all available information instantaneously and consensually, which is simply not true and can also be seen in practical observations. If seen as strict mathematical model, under the condition that the market reacts instantaneously, there would be jumps in the exchange rates between two currencies rather than movements.

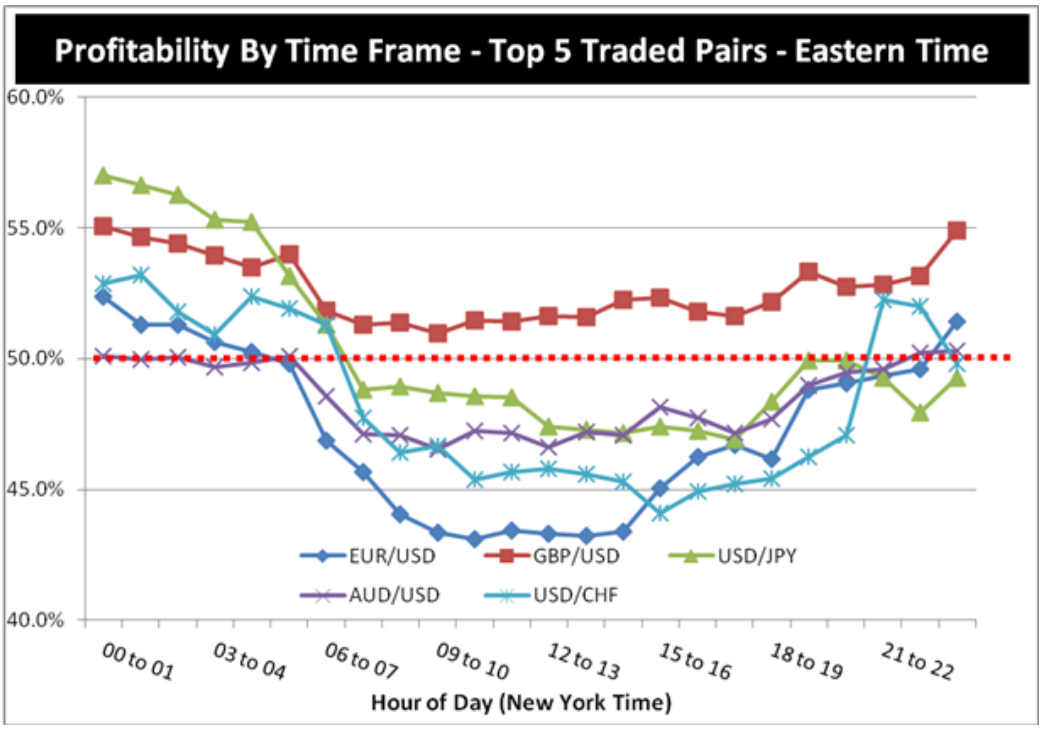


Figure 1.2.: Average profitability by time frame by FXCM Clients from 2009-2010. [Image taken from DailyFX (2011)]

The website DailyFX analyzed over 12 million real trades with regard to the profitabilitiy by the time of the day (DailyFX, 2011). They tried to answer a simple question:

when is the best time of the day to trade Forex? Their results show, while patterns vary on a day to day basis, the average pattern over the course of the year is stable. Especially in the morning hours traders seem to be more profitable. DailyFX show different trading strategies, how to profit from the knowledge of the high activity during certain hours a day (see Figure 1.2; 1.3). Like many of the trading suggestions which can be easily found on the Internet, they also make base assumptions on a time series, which is, seen from a signal processing point, mostly random. It is not surprising that it is not possible to learn from random data, or to gain enough information from this data which mostly consists of noise. Instead it makes more sense to see the problem from a practical side: *Most* trades are stil made by humans who make their trading based on news announcements.

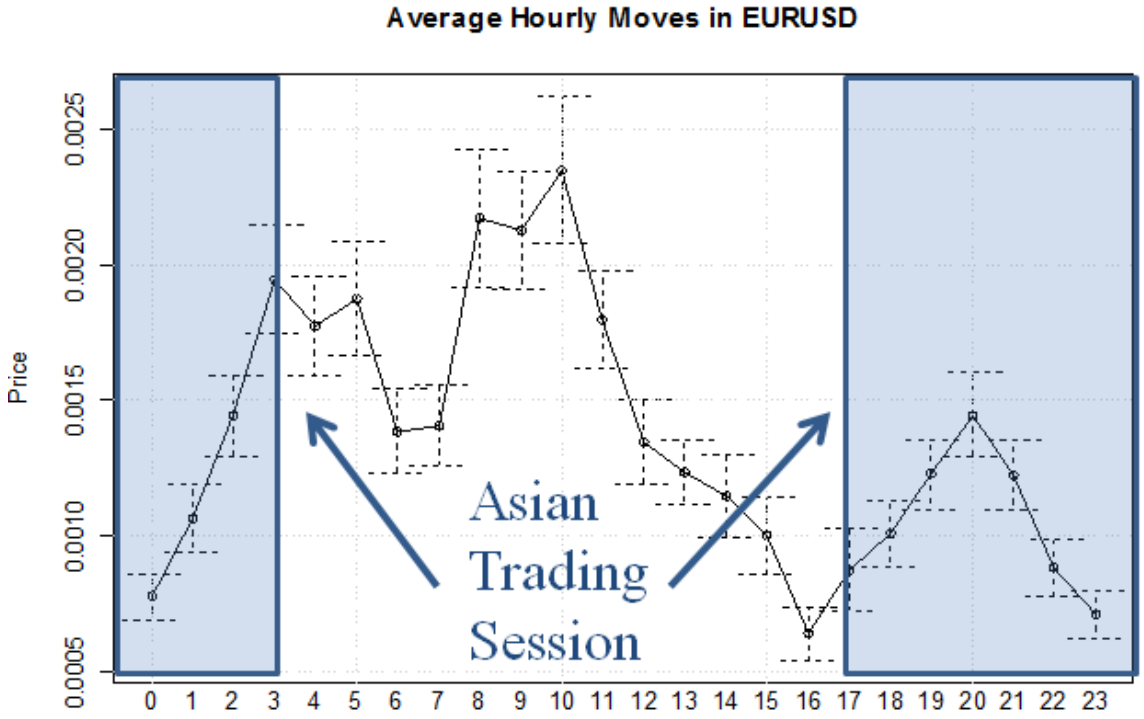


Figure 1.3.: Average hourly moves of EURUSD (Eastern Time). [Image taken from DailyFX (2011)]

Bauwens, Omrane, and Giot (2005) analyzed the impact of news from the Reuters news alert screen on the Foreign Exchange market, specifically EUR-USD quotes, from 9 different categories between May 15 and November 14 2001. They show that the market consists of a public and a private component. While the public usually reacts directly after a news release, the market often shows a rise in volatility before a news release. This higher market pre-announcement activity is explained due to insider information. Also before scheduled news they could see a rise in volatility, for example

regular speeches of the Chairman of the European Central Bank or economy and finance ministers. This is explained to avoid surprise effects for market participants, meaning traders bring their assets to a safe haven before a surprising bad news arrives. The situation for unscheduled news is slightly different. Especially for rumors of central bank interventions there is a high increase (+48%) in market volatility before the announcement, because usually news agencies react to news if they are already widespread. Only after reaching a certain level of importance to the market the news are treated seriously and, as a consequence, gets broadcasted. Which means in other words that the news is already well known and the market reacted already so newsagencies actually report because the market showed a certain reaction. In contrast to the pre-announcement rise in volatility, they find only a light market reaction change in the post-announcement phase (Bauwens et al., 2005).

A lot of work has been done trying to predict exchange rates based on previous market movements. Using multiclass support vector regression (Premanode, Vonprasert, & Toumazou, 2013), decision support systems (DSS) and artificial neural networks (ANN) (Lai, Yu, & Wang, 2005) or using genetic algorithms (GA) (Evans, Pappas, & Xhafa, 2013), just to mention a few. While some approaches show promising results under some circumstances, this thesis tries to find another way of predicting market movements. Thus, the purpose of this thesis is rather to find a way to scrape, process and learn from web based information and news than from previous market movements, or news and signals from (often manually) preprocessed Forex market specialized agencies. Furthermore it shows different ways to gain news from various sources, how to do a sentiment labeling of news, how to filter unimportant data, how to preprocess data utilizing the Apache OpenNLP Framework and then how to predict the market impact of future news articles using LibLinear. It tries to close the gap between machine learning techniques used for text processing and automated trading.

1.2. Outline of this Thesis

In chapter two the Foreign Exchange Market fundamentals will be explained. It is a brief history of the Forex Market introduced and the fundamental workflow explained. Furthermore the chapter introduces which information is driving the Forex market, who are the main players and ends with a summary about the different kinds of algorithmic trading and explains the efficient market hypothesis.

In chapter three basics for sentiment analysis are explained. Machine learning fundamentals are introduced, for example the different ways of building a feature vector and the most important algorithms are explained. Where text corpora can be found and how text processing and machine learning is then applied is the main part of the chapter.

In chapter four the existing systems are analyzed and compared to each other. While the Stock Market is relatively well researched, the Forex Market is not yet part of the main research. Work is reviewed from the first attempts to do predictions on financial markets to sophisticated systems. The major insights are summarized and lead to the ideas in the next chapter.

In chapter five ideas for a new system based on the findings in chapter four are developed. First the main ideas are defined, which are modeled then into a more granular detail. The platform requirements are given, the used libraries are defined and the general architecture is documented.

In chapter six the prototype is built based on the ideas and requirements from the previous chapter. A more in-detail description about the usage of classes and their combination which lead to the results is given.

In chapter seven the results are presented, while they are separated in two sections: First the results from the filtering module are presented. This includes the first big part, if articles can be filtered on the fly, to get only those articles which are from the financial domain. The second part dedicated towards the actual Forex prediction based on macroeconomic news sentiment.

In chapter eight the lessons learned are presented, while in chapter nine a short conclusion, a summary and an outlook for future work is given.

2. The Foreign Exchange Market

"It's not the having, it's the getting."

[Charles Spurgeon]

2.1. Historical Evolution and Overview

The Foreign Exchange (*Forex or FX*) market is a global currency pair trading market. One currency, for example US Dollar (*USD*), is traded against another currency, for example Euro (*EUR*), which is expressed as a currency pair *USDEUR*, at a specific exchange rate at a specific time. Beside the obvious function as a market to exchange international currencies for trading with goods, it also allows and supports speculation in difference of the value of currency pairs. It is defacto the most liquid market in the world with a reported daily turnover of around $4 trillion US Dollar according to the BIS (2010). Marketwatch (2010) summarized that British based banks are responsible for 36.7% of the daily turnover, followed by the US with 18% and Japan with 6%. Looking at the most important currency pairs are EURUSD (Euro - US Dollar) with 28% transaction share and the USDYEN (US Dollar - Yen) with 14%.

The Foreign Exchange market, as we got used to it today, was formed during the 1970s, after countries gradually switched from fixed exchange rates to floating exchange rates. Between World War II and the 1970s the Bretton Woods system was established, which aimed to rebuild the international economic system. When in 1971 the US Dollar became a fiat currency, which means to some extension that the US Dollar was no longer directly bound to gold, the Bretton Woods system came to an end, the US Dollar became a reserve currency for many states and many other fixed currencies also became free floating. This is also known as the Nixon shock. 1973 introduced Reuters computer monitors for trading quotes. (Giancarlo, 2002; Baillie & McMahon, 1990)

Trading on the FX Market is possible 24 hours a day except weekends, from 20:15 GMT on Sunday to 22:00 GMT on Friday. The exchange rate of a free floating currency is determined by its supply and demand and is quoted on financial markets mainly by banks around the world. It is an uncontrolled over the counter (OTC) market where an

exchange of currencies is usually negotiated directly between dealers and the customer. With 36.7% of the daily turnover, the United Kingdom is the most dominant trading center worldwide according to the BIS (2010). London's market price usually drives the quoted currency price around the world. The *spread* is the difference between the bid and ask price, which varies among the different levels of market participants.

2.2. Workflow

On the top level is the interbank (or interdealer) market. The market participants, nearly only banks, deal either directly or via two main electronic brokering platforms, the Electronic Broking Services (*EBS*) or the *Thomson Reuters Dealing 3000 Xtra*. If the market is broadly divided in the two major participants, then there are the dealers on the one hand and the customers on the other hand (Sager & M. P. Taylor, 2006). Dealers are the dominant group of the Forex market with about 51% of the total daily market turnover (in 2002). Newer reports, like from the BIS (2010) state a 39% top tier interbank market participation.

Dealers

Dealers are the market makers, leverage traders and senior risk traders. Market makers facilitate access for customers to the interdealer market. Although they can be further divided, it would exceed the scope of this thesis. A typical trading volume of senior risk traders is about 100 to 200 million US Dollar. The interdealer market is dominated largely by banks, where according to Sager and M. P. Taylor (2006) 17 banks in London and 13 banks in the USA account for 75% of the turnover. Most interbank trading occurs electronically, providing anonymous limit order bid-ask pricing to dealers. With a limit order (section 2.2), the buyer (or seller) has somehow the control of the price at which he wants to buy (or sell), but the buying (or selling) itself is not guaranteed. A bid-ask price (section 2.2) is the current highest price at which someone is willing to buy, or the lowest price at which someone is willing to sell, which has usually about 5 percentage in point (PIP, section 2.2) difference.

Customers

Customers are the group which interact with dealers to access the interbank market. Dealers who interact with customers are also often called brokers. Customers have, generally speaking, two possibilities: They can use voice trading, or, what the majority

does, use one of the electronic systems which mainly interact as gateways between the top level market and the customers' computer. While within the interbank market the difference between the bid and the ask price is between 0 and 1 PIP (*percentage in point*), the difference raises the more the average trading volume goes down. As stated above, the Foreign Exchange market is an over the counter market (OTC), where the bid and ask price is directly negotiated within the customer and the dealer (or broker). Which also means that the dealer, who faciliate the access through his electronic systems, can take advantage of being the man in the middle and utilize information about the movements of his customers.

PIP

Especially in FX markets the unit of change is *percentage in point* (PIP). For all major currencies a PIP is a 1/100th of one cent, except for the Japanese Yen. All these currencies are quoted with four decimal points, the Japanese Yen up to two decimal points.

An example would be the rise of the exchange rate EURUSD from 1.3000 to 1.3015, which would be a price ratio increase by 15 PIP. Although this is the smallest move a currency pair can make, some platforms have brought greater price transparency to the Forex market by adding an additional decimal place to their quotes (Informed Traders, 2008). This is then known as fractional PIP, where instead of 4 decimal places quoted it is extended to 5 decimal places (3 decimal places for JPY instead of 2 decimal places).

Process of Trading

Trading is the process of converting one currency with a specific amount to another currency and holding it for a specific time. While it would be theoretically possible to hold the foreign currency forever, it would make no sense in terms of Forex trading. A successful trade would be the conversion of €10.000 to US-Dollar for an EURUSD exchange rate of 1:1.3000, having $13.000, waiting until the EURUSD rate *falls* to 1:1.2500, and converting it back to Euro giving a net profit of €400 (or 500 PIP).

Bid-Ask Price

Also known as the buy-sell spread, where the difference is the spread. A dealer buys a currency at a higher price than he sells it. If there is a spread of 10 PIP, the difference of the exchange rate is 0.0010, meaning that for example the EURUSD ask-price is

1:1.3005 and the EURUSD bid-price is 1:1.2995. This means, if at the same time someone tries to change €10.000 to US-Dollar, the resulting amount would be $12.995. But with the same exchange rate one would need $13.005 to change it to 10.000 Euro.

Limit or Market Order

There are several ways to purchase (or sell) currencies, which is basically referred as "to make an order". While the market order guarantees a buy (or sell), it cannot guarantee the price. The limit order gives the trader the guarantee to buy (or sell) at a specified price or better, but does not guarantee that the order is processed. There are also take profit orders which closes the trade once a certain level of profit is reached, as well as stop loss orders which limit the amount of loss.

Leverage

The leverage gives the trader the possibility to control the market with a larger amount of money than he has in his deposit. If a broker offers a leverage of 100:1 the trader is able to trade for each 1€ in his deposit 100€ on the market. This means in a more realistic example: for a 10.000€ deposit, a profiterable trade without any leverage of 10 PIP would mean a profit of 10€ (or 0.1%). With a leverage of 100:1 the amount in the deposit would still be 10.000€, the amount of the trade would be 10.000€ * 100 = 1.000.000€ . Now, the same 10 PIP profitable trade would mean a profit of $10 * \frac{1}{100}$ of 1 percent of the traded volume, which would be 1000€.

2.3. Electronic Forex Trading

The Forex trading, as an over the counter market trading, is directly negotiated between the customer and the dealer. While the price could be theoretically independent, it is mostly influenced by quotes of banks which dominate the market. The electronic Forex trading, or short *etrading*, has various impacts: it reduces the costs, and due to the local independence, it also brings greater competition and liquidity. One of the most important impacts for this thesis is the possibility for *algorithmic trading*.

Reuters, EBS and Bloomberg

Three of the most widely used electronic services are Reuters, EBS and Bloomberg (Pareek, Saha, & Ghosh, 2011). In general these services are broker services. They try

to match bid-ask orders together. Supply and demand are forming the price. Basically the systems provide an electronic limit order book, which means: the highest bid and the lowest offer order are crossed, the spread between the bid and the ask price is the fee the Forex broker takes. As an example[1] the Deutsche Börse provides such an order book graphically with a 15 minutes delay.

Data Format

Tick data is a real-time snapshot of the broker-data. Most brokers do not provide tick-data, they provide at least M1 (1 minute) data, meaning the data has a resolution of one minute. The range can vary: M1 (one minute), M5 (five minute), etc. , H1 (hourly), H2 (two hours), etc. , D1 (daily), etc. , W1 (weekly), MN (montly). If the data is not provided as tick-data, but as data aggregated to a certain time-range, it has a high value (the highest tick-data value within the time range), a low value (the lowest tick-data value), a open value (the first tick-data value) and a close value (the last tick-data value) along with the volume traded. Commonly summarized in a OHLC chart (open-high-low-close chart) or candlestick chart. If the open value is below the close value it is called *bullish*, otherwise called *bearish* (see Figure 2.1). The only conclusion which can be drawn from aggregated data is: a rate was fluctuating between high and low, started with open and ended with close. Therefore the goal is to have a high resolution of the data as possible.

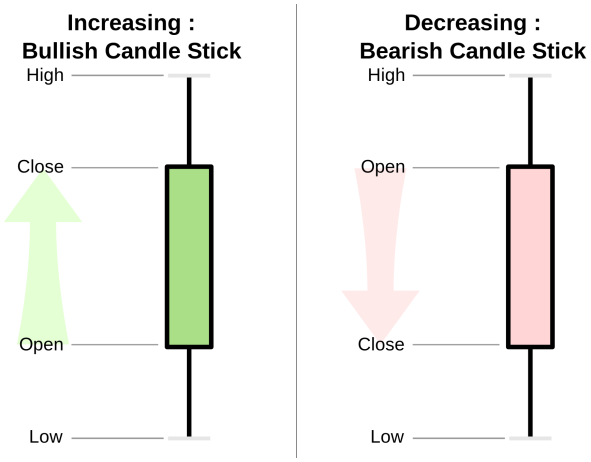


Figure 2.1.: Candlestick chart scheme, bullish and bearish. [Image extracted from Probe-meteo.com (2013)]

[1] http://www.boerse-frankfurt.de/en/deutsche_boerse/orderbuch.m?isin=DE0005785604

Historical Data

Many brokers provide their users with historical data in some way, although the quality of the data differs in many ways. MetaTrader (section 2.5) provides data with a maximum resultion of M1, NinjaTrader (section 2.5) provides a higher resolution, but splits the data in bid and ask quotes. Some distortions can happen in the given timescale, or for example, brokers omit several random hours in a day. Usually, if tick-data is provided, a timestamp down to the miliseconds is given. Is the data shifted for 10 minutes in one direction it can give highly inaccurate results. Generally speaking it is not easy to get accurate historical data.

Volatility and Market Activity

Market activity is measured by the number of quotes on the market (Bauwens et al., 2005). Return volatility is the uncertainty or risk about the amplitude of change, meaning, if a rate changes over a long range, the volatility is high (Investopedia, 2013).

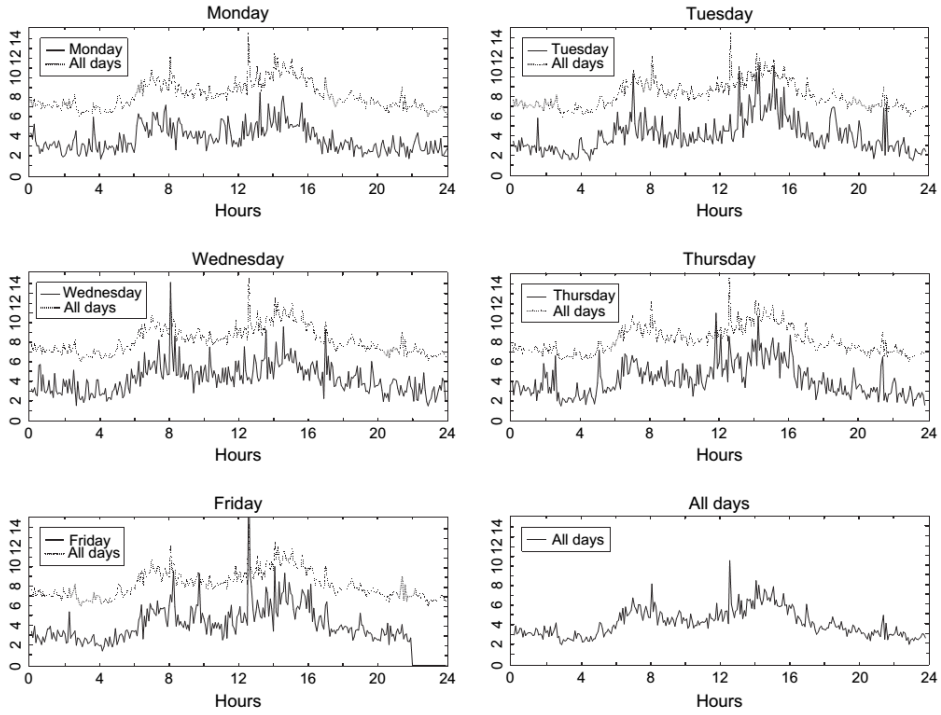


Figure 2.2.: Intraday and average volatility. The all-day pattern is shifted upwards by 4 units. Hours in GMT. [Image extracted from Bauwens, Omrane, and Giot (2005)]

Difference to the Stock Market

On the surface the way the Stock and the Forex market work seem to be very similar, although there are several major differences. The IBTimesFx (2011) summarized the five biggest differences of the two markets: Firstly the Stock markets are globally bound and secondly they operate 8 hours a day, the Forex market operates globally in 3 shifts throughout the entire year. In the Stock markets there is always a middleman (mostly a broker) who charges a fee, while with spot trading a trader can buy or sell directly. Although mostly a Forex broker is in between, at least, the spread is transparent and there are no additional fees required. The speed is another big advance of the Forex market: orders are almost instantaneously executed, while with the Stock market an order has to be done on the floor and can take up to several minutes until it is executed. The last big difference is the complexity: While Stock traders often have their eye on hundreds (or even thousands) of Stocks, in the Forex market it is enough to concentrate on one, or several currency pairs.

Timing of the Foreign Exchange Market

Although the Euro - US Dollar currencies are traded continously, the market activity and volatility changes its pattern noticeable. When looking at the average pattern (see Figure 2.2) then one can find some parallels between the market activity and opening/closing times, as well as lunch breaks: The Singapore and Hong Kong markets open around midnight GMT, where an increase in volatility is visible. This flats out at around 4am GMT, when the lunch break for the Asian financial markets begin. It increases afterwards again, where it reaches a local maximum at around 7am to 8am GMT, which is right after the opening of the European markets. Inventory control is absolutely essential in FX Markets (Bjønnes & Rime, 2005). Mostly dealers have to close their positions at the end of a day (Bauwens et al., 2005), therefore the volatility increases also during closing times. The lunch break in Europe causes another decrease in volatility around 11.30am GMT, which highly rebounds around at 12pm GMT, exactly then, when the New York markets open for trading. According to Bauwens et al. (2005), the big spike at 12.30pm GMT is explained by Friday news announcements in the US. The general high volatility between 12pm and 4pm GMT is explained due to the simultaneous activity of the European and the US markets. It decreases during the New York lunch break and increases slightly at the closing time of the European markets around 5pm GMT and around the closing time of the New York markets around 9pm GMT.

Triangular Arbitrage

As triangular arbitrage in the Forex market is the situation explained, where a trader can make profit by exchanging his initial currency via two other currencies back to the initial currency. This can be used especially by algorithmic trading and is investigated by Chaboud, Hjalmarsson, Vega, and Chiquoine (2009) and is further explained in chapter 4.2. An example would be the exchange of $1 to 1.3800€, exchanging 1.3800€ to £1.1663 and these £1.1663, at another institute and at the same time for a different exchange rate, back to $1.0005, making a profit of $0.0005 - for every Dollar.

2.4. Information driving the Forex Market

Market activity is considered as a proxy for volatility (DeGennaro & Shrieves, 1997; Melvin & Yin, 2000; Bauwens et al., 2005) and it is assumed that news affects both, volatility and market activity. Due to the fact that customer-dealer orders are not observable by other market participants, the market activity and volatility are considered as a source of private information.

How the market reacts on news still remains poorly understood (Andersen, Bollerslev, Diebold, & Vega, 2007), although for dealers one of the most important sources for private information is how customers place orders, since it may signal the interpretation of public news (Bjønnes & Rime, 2005).

DeGennaro and Shrieves (1997) examine how news impacts on the volatility before, during and after the news arrival. Information is split into a *private* and a *public* part. Their base data is the Japanese Yen (¥) - US dollar (JPYUSD) exchange rates from October 1, 1992 trough September 30, 1993, filtered and with a sampling frequency of ten minutes. Both, the exchange rate as well as the news source is Reuters. The Reuters money news-alerts provided in this time 105,065 news headlines which they extract and categorize by keyword combinations into *regularly scheduled macroeconomic news*, *unscheduled economic policy news* and *unscheduled interest rate reports*, leaving, after further filtering, 2,140 news items.
First they examine the relation between dealer spreads and quote frequency on news arrival. They can show a statistical significance between the dealer spread and *unexpected* quotes, while in contrast, there is no significance in *expected* quote arrival and dealer spreads, which also supports their theory that informed traders hide their actions during high market activity. In other words the spreads increase when it is surprising news, nothing happens if the news is already expected by dealers.

Bauwens et al. (2005) follow up a similar idea and analyze *"the impact of a more*

refined and extended set of nine categories of news announcements on Forex volatility in the new Euro/Dollar market", as well as they investigate *"the volatility dynamics before, during and after scheduled and unscheduled news announcements"*. They split the time around the arrival of news in three periods: a pre-announcement, a contemporaneous and a post-announcement phase and hypothesize the traders behavior between these phases and between scheduled and unscheduled announcements. They use Euro - US Dollar (EURUSD) exchange rates from May 15 to November 14 2001 with a resolution of five minutes (M5). Their news headline source is also Reuters, although they consider a much larger set of news events.

A relatively new work is from Marshall, Musayev, Pinto, and Tang (2012). They examine the impact of 16 scheduled US macroeconomic announcement indicators on the foreign exchange implied volatility. Although they measure the implied volatility, which means they examine if the volatility prediction is stable or changes suddenly, they give some interesting insights in how the market reacts on news. They examine the impact of news in a timeframe of 5 days before and after the news release.

More interestingly Laakkonen and Lanne (2010) study the effects of good news versus bad news on exchange rate volatility in times where the economy is considered as good versus the economy is considered as bad. They examine the problem that studies regarding the impact of good news versus bad news are made at different times in different states of the economy, resulting to different kinds of impact on volatility. Their dataset is also the most comprehensive, also incorporating findings of previous studies. They use M5 EURUSD rates between 1 January 1999 and 31 December 2004. Rather than only looking at US macroeconomic news (headlines from Reuters), they use news published the Bloomberg World economic calendar from both, the US as well as the European business cycle.

Scheduled Announcements

Scheduled news announcements are such announcements where the release time and the topic are known in advance, like regular speeches, or, for example, planned releases of US macroeconomic figures.

Unscheduled Announcements

Bauwens et al. (2005) use forecasts of key institutes and specialized organizations, as well as rumors of central bank interventions or other extraordinatry events, like desasters, terrorist attacks, etc., as unscheduled news. Basically all authors categorize these events as unscheduled news.

Impact of Announcements

DeGennaro and Shrieves (1997) can show that volatility is increased after the news release for scheduled news and for unscheduled news. For scheduled news the volatility highly increases when the news announcement arrives and continues to be high after the news announcement for their measured 10 minutes interval. For unscheduled news the situation is slightly different. Volatility decreases at news announcement and it is claimed that it is either a calming effect of the news itself, or it may reflect endogeneity of policy related announcements. Although they cannot find any concluding data that show a tendency for informed trading in the two preceding hours to the news announcement, it is mentioned that informed trading can also happen even earlier.

Bauwens et al. (2005) clearly show an overall volatility increase for scheduled news during the pre-announcement phase by 10% up to 20%, depending on the news topic, which is explained by speculations of the traders, not because of informed trading. They can also show a volatility increase for *unscheduled* news during the pre-announcement phase between 8% and 48% (for rumors of central bank interventions). Such news is mostly already widely known and it circulates for a certain time until it reaches a certain level of seriousness. Then agencies will announce the news.
A post-announcement volatility increase is explained by the heterogeneity of the interpretation of the news, so that some surprised trader reactions take place and also just some positions get closed. Although it is also shown that unscheduled announcements have a statistical zero total impact (with some exceptions), almost all news cause an increase in volatility in the pre-announcement phase. A question they addressed was if there is a difference between *positive* and *negative* news. It is no strong evidence shown that there is a difference.

The examinations of Marshall et al. (2012) are rather interesting on the sensitivity of news to a currency pair than to the impact itself. After studying the impact of news on the implied volatility, they conclude that the Euro is far more sensitive to US macroeconomic figures on the announcement day than the Deutsche Mark was. Their results conclude that the implied volatility is not significantly changing before, during and after the announcement day, which means that any private information of market participants is already incorporated in the calculation.

The comprehensive work of Laakkonen and Lanne (2010) shed light on the relationship between the state of economy and the impact of news announcements. It is indicated that news releases causes jumps in the level of exchange rates as well as the volatility. The impact is asymmetric: good news has less impact in bad times than bad news in good times. In general, news in bad times is less influencial than in good times, although espeially bad news in good times has a high impact. This is mostly explained due to the way the asset price is calculated by risk-averse investors in good times, or

on the other hand, bad times. This means that good news in bad times still brings the additional risk of a bad economic situation, even the news is good.

2.5. Algorithmic Trading

Automated, or *algorithmic trading*, is the exchange of currencies without a human interaction, based on algorithmic decisions. Basically it can also be described as *bot trading*. This can be based on a variety of rules. One of the scenarios would be the observation of the market until a triangular arbitrage trading (see 2.3) can be done. If this is not done automatically, but by humans, it would cost extensive concentration on a continuous basis. The trader would have to obeserve the market continuously and would have to continuously calculate the arbitrage. Beside this special scenario of algorithmic trading, almost every kind of trading strategy can be implemented in software to automate it utilizing specialized software, or special APIs (*Application Programming Interfaces*) by brokers. Investopedia (2014) defines algorithmic trading as *"A trading system that utilizes very advanced mathematical models for making transaction decisions in the financial markets."*.

In this section the topics are described which are associated with algorithmic trading, as well as the most popular software is presented.

High, Medium and Low Frequency Trading

HFT or *high frequency trading* is the trading of a large quantity of securities in a short amount of time, typically far less than one second. HFT is further divided into several topics and different strategies can be applied to it, but it will exceed the scope of this thesis.

The opposite of HFT is LFT, or *low frequency trading*. Beside the risk factors of both ways, there is an ongoing discussion what is better: high frequency trading based on technical analysis of historical data, or low frequency trading based on news analysis. (Barber, Lee, Liu, & Odean, 2010; Learn to trade the Market, 2014; Chan, 2012). It is assumed that HFT traders are usually high-risk traders, where they rationally infer the trading abilities by the amount of success or failure they have. A Bayesian learning model can ideally represent this behavior and it could be shown that traders are either getting better with experience or, in most cases, quit after unsuccessful trades. The two most interesting strategies for HFT are triangular arbitrage trading and liquidity-redistribution, which is nothing else than taking advantage of price discrepancies between different platforms. The goal of high frequency traders is also a

low-latency execution of trades within 1ms to 10ms.

Low frequency trading typically includes strategies with holding times from more than one day (Stack Exchange: Quantitative Finance, 2011). Between HFT and LFT is MFT (*medium frequency trading*) with a holding period of typically less than, or up to one day. To develop MFT and LFT strategies, different brokers offer a variety of ways to execute and automate these trades for their end-users. Beside the direct access to their platforms via APIs, some offer propriety software, especially designed to do technical analysis for the strategies. The software packages are usually bound to one, or only several brokers. But they offer an easy way to *backtest* your strategy easily by re-playing past market-data.

Trading Software

Some of the most spread trading software packages are *MetaTrader*[2], *JForex*[3] and *NinjaTrader*. Numerous Forex brokers try to develop their own proprietary software, others build their services based on existing software packages and yet others provide standardized APIs to access their services. MetaTrader version 4 is one of the pioneers in automated trading and achieved to build a robust community around their software. They provide automated trading plug-ins with the name *Expert Advisors*. All of the software packages usually give the user the possibility to back-test their programmed strategy against past trading data. This runs in a simulation and gives the programmer the opportunity to debug and improve the strategy that the autoed trading is doing.

While the advantages of an all-inclusive system like MetaTrader are appealing to start off, there are several drawbacks which are going to be discussed now. First of all, the foremost limitation is the language of the software-packages itself. Although some do provide C# native language support, in general the access to lower level language properties is rather small. Secondly, the software architecture is built to automate trading strategies which make use of available trading data from the broker. So the flow of the simulators is to "feed" the algorithm with the data from the Forex market and let the algorithm, that was programmed by the trader, "predict" the future market moves. External data, as from news sources, including crawling, text-processing and classification is not the primary goal of any of the software packages.

[2] http://www.metaquotes.net/en
[3] http://www.dukascopy.com/wiki/

2.6. Efficient Market Hypothesis

The *efficient market hypothesis* (EFM), in its original strong form, states that market (or security) prices always reflect all available information in efficient capital markets (Fama, 1970). The problem is that by applying this hypothesis to the real world, it would be impossible to gain profits by trading on news. Also, the EFM can only be applied when there are no costs for information and trading, which suggests that the EFM in its pure form is false. A less strict and more reasonable version of the EFM states therefore that *"prices reflect information to the point where the marginal benefits of acting on information (the profits to be made) do not exceed the marginal costs"* (Fama, 1991).

A vast amount of research was done on this topic, where the results could not be more different. The range, where researchers found that all information is finally reflected by the market ranges from several seconds to 60 minutes after a news release, including research from Schumaker, Zhang, Huang, and Chen (2012), Gidófalvi and Elkan (2001), Peramunetilleke and Wong (2002), Marc-André Mittermayer and Gerhard Knolmayer (2006), Lavrenko et al. (2000b). The reason can be found in the source of their news and to which securities on which market they are applied to. Some use news from newswire services, some use information which was gained from social networks or blogs, others use information from classical edited news sources such as newspapers. The markets vary from the New York Stock Exchange to other markets in China, or Japan. Throughout the research the following numbers seem to emerge: information seems to be reflected by markets starting from the time it was released up to 20-30 minutes after the release.

2.7. Summary

This chapter gives the reader a basic understanding of the workflow of the Forex market and the surrounding terms and definitions. I starts off with a brief historical evolution, explains how the Forex market evolved and who are the main players on the market. The Forex market is an over the counter market, where there can, but not necessarily has to be a broker between two parties exchanging currencies. This is one of the most important differences between the Forex and the Stock market. Due to the global operation of the Forex market, the working hours of the different traders on different continents can be directly observed in the pattern of the market activity. Furthermore in this pattern can be observed what the market drives and which information is resulting in an increased trading activity. It can be seen that insider trading is taking place due to the increased trading activity prior normally surprising news announcements. Although this is *probably* still caused by human trading interaction, a constellation

called triangular arbitrage, where brokers have different exchange rates, can especially be used for algorithmic trading. High Frequency triangular arbitrage trading can easily be modeled in software utilizing access to several dealers. The available software leveraged furthermore the term *algorithmic trading*, which is mostly understood to utilize past trading data to predict future market moves. Finally, research around the efficient market hypothesis gives insight that information is fully reflected by the market after up to 20-30 minutes after the news release.

3. Methods for Sentiment Analysis

> *"Everything we hear is an opinion, not a fact. Everything we see is a perspective, not the truth."*
>
> [Marcus Aurelius]

3.1. Background and Overview

When people are about to make decisions, they usually look for other opinions about the topic of interest. If one wants to buy something, to vote for someone, or to book a hotel to go on holidays, it is often relied on others opinions and tried to utilize any available information to gain knowledge *what others think* about the target of interest. People are reading information in forums, blogs, newspapers or product reviews before the decision is made. It is not surprising that travel review sites like tripadvisor.com count over 40 million unique monthly visitors, according to High Scaleability (2011). Not very different is the opinionbuilding about public entities when we gain information from news. Godbole, Srinivasaiah, and Skiena (2007) point out that news is seldom *neutral*, it is either *good* or *bad*. In the past years the automatic analysis and extraction of the sentiment regarding a specific entity from newspapers, blogs or reviews has been getting an incredible amount of attention (Yi, Nasukawa, Bunescu, & Niblack, 2003; Godbole et al., 2007; Pang & Lee, 2008). So, one can say the sentiment of a specific entity is the summary of all the positive and negative opinions and emotions. To put it another way: "*Sentiment Analysis* is the task of identifying positive and negative opinions, emotions and evaluations" (Wilson, Wiebe, & Hoffmann, 2005).

What is Sentiment?

According to Princeton WordNet 3.1 (2012), a *"[...] sentiment, opinion, persuasion, view, thought [...]"* is *"[...] a personal belief or judgment that is not founded on proof or certainty"*. The field of the sentiment analysis is not only challenging, it also faces a vast amount of difficulties and, in addition, is a broad subject of research for all kinds of

applications. Undoubtedly an opinion can vary from one person to another, but furthermore sentiment analysis tries to classify a variety of domains and users to relatively few classes, how Pang and Lee (2008) describe it. While there are in other disciplines, like topic categorization, a group of different clusters which rely somehow to the topic or the content, there are mostly just two classes (positive and negative) or a small range of polarity in sentiment analysis, which have probably not much to do with the actual topic.

Sentiment analysis is also known as *opinion mining*, *review mining*, *subjectivity analysis* and has some connections to *affective computing*, where the goals are to enable computers to learn and express emotions according to Pang and Lee (2008). Within these fields of research are a lot of difficulties connected, where some of the most challenging tasks are the *disambiguation*, *negation tagging* and *contextual polarity*. While sometimes huge corpora of hundred billion words is subject of the analysis task (how Turney and Littman (2002) work with), it is generally better to split a document in sub-documents and work on smaller portions. (Pang & Lee, 2008)

Wilson et al. (2005) give some examples that rise especially with negation of sentences, to demonstrate some challenges: Negation may be local ("**not** *good*"), or eventually the subject is negated ("**no one** *thinks that it's good*"), or the negation is used as intensifier ("**not only** *good but amazing*"). All these examples are subject to negation tagging and contextual polarity. If the topic is known, then it is often useful to know which exact feature the writer is expressing his opinion about, how Mejova (2009) summarizes. These features can be *explicit* ("Battery life too short") or *implicit* ("Camera is too large"). (Jindal & Liu, 2006)

Applications

Sentiment analysis is a broad topic where applications can be found in a variety of segments. One, and probably the most famous application, is the analysis of product and movie reviews (Zhuangi, Jing, & Zhu, 2006; Pang, Lee, & Vaithyanathan, 2002). Within the area of reviews, normally the topic is known (the entity the opinion is about) and mostly there are either two classes (for example "thumbs-up" / "thumbs-down"), or something like a five star rating. Furthermore there is no need to restrict the analyzing task to product or movie reviews. It could also include opinions about political candidates or other public entities. Another application from the same sector, but in a different way, would be finding errors in user reviews, where users are accidentally giving a one star rating, but their review is absolutely positive. There are applications (Pang & Lee, 2008) where user ratings can be biased, or are in need of correction.

Mejova (2009) also gives the example as an application for brand tracking. Companies like OpSec Security[1] provide services that help customers to track consumer feedback across different blogs, articles and other sources on the Internet.

Another possibility for sentiment analysis is the role as a sub system for other systems. A *recommendation system* might rely on a sentiment analysis to avoid recommending items with a lot of negative feedback. Advertising systems like Googles AdSense work not very different from recommendation systems. AdSense or AdROSA (Kazienko & Adamski, 2007) try to provide ads related to the content of the website. A sentiment sub system can help avoiding ads on websites that are inappropriate for ads placement, like Jin, Li, Mah, and Tong (2007) do. Furthermore it could help to show only ads for such products, where relevant positive sentiments are detected. (Pang & Lee, 2008)

This thesis, to give another example, focuses on the global economic sentiment from news, trying to identify upcoming trends in exchange rates. Due to the fact that textual financial information has a direct impact on the exchange rates (Niederhoffer, 1971) and volatility (Bauwens et al., 2005), it is certainly another great application to analyze financial textual news. While the term *sentiment analysis* in financial economics refers to analyzing the market confidence with indicators from proxies such as Stock prices and trading volumes (Devitt & Ahmad, 2007), the term is, within this thesis, purely used to do sentiment analysis tasks based on global financial textual news articles.

Goals

Since sentiment analysis has such a broad range of applications, there are also a variety of goals where researchers try to solve different kinds of problems. With regard to classification of movie or product reviews, the most important task is sentiment polarity detection or generally *polarity classification* (Pang & Lee, 2008). This thesis focuses especially on this task and tries to find solutions by combining different approaches. The underlaying problem has an opinionated text, which is about a single entity or topic and the goal is to detect the sentiment polarity between two opposing polarities, which is sometimes gradually between "good" and "bad".

Koppel and Shtrimberg (2006) try to label Stock news based on the market movements. After the labeling a supervised sentiment detection is done. The goal is to predict future Stock market movements. They used two different methods, where the first method models an informed customer, who is utilizing insider information, the other model reflects the "honest investor" who has no access to insider information. The results are unsurprisingly good for the model that utilizes information *before* it was released, with

[1] http://www.opsecsecurity.com/

over 70% accuracy, especially if one reconsider that there is a huge increase in volatiliy *before* a news is released (Bauwens et al., 2005). For the model that represents the *honest* investor, who has access to news only *after* they were released, the results show a 52% accuracy. The authors state that the margin is probably too small to overcome the cost of trading.

In general it is necessary to have some form of data where text mining and sentiment analysis algorithms can be applied to. To achieve the final step and categorize textual data in two, or several categories, a certain amount of pre-processing is necessary. In turn to do pre-processing on textual data, the data itself must be imported or crawled somehow, which has other underlaying challenges. If the textual data is obtained from uncategorized news and one wants only the interesting articles with highly discussed topics, then yet again other techniques have to be applied. These steps are outlined within this chapter and the theoretical background is explained.

3.2. Textual Data Sources

Unstructured data like textual data (e.g. news articles), with certain assorted topics, already prepared for further processing, with correct labels, is unfortunately not easy to get. There are several corpora, which can be freely downloaded. Among others, three of the largest most interesting for the domain of Forex market prediction based on news articles, are the *Reuters Corpus Volume 1* (RCV1) , *Reuters Corpus Volume 2* (RCV2) and the *Thomson Reuters Text Research Collection* (TRC2) corpora[2]. The RCV1 contains 810.000 Reuters news stories in English language from 20th August 1996 to 19th August 1997. The RCV2 contains 487.000 news stories in multiple languages from the same period. The newer TRC2 corpus contains 1.800.370 news stories from the period between 1st of January 2008 to 28th of February 2009. Unfortunately, to the authors best knowledge, a corpus with newer data or from different sources (other than Reuters) could not be found. Although these are interesting options, it is considerable to crawl and import articles from different sources. Therefore this section will start by briefly explaining the challenges of web-crawling and web-scraping. Then the traditional further processing steps are explained.

[2] http://trec.nist.gov/data/reuters/reuters.html

Crawling versus Scraping

Web crawling, generally, can be considered as following links from one website to another and building iteratively an index of the path and the containing information. In other words, it is like building a big table of contents of the web, where one can retrieve certain topics by querying the index and finding the pointer (link) to the original source. Web scraping, on the other hand, is considered as processing a (web) document to extract only certain information. These are typically specialized modifications to extract only certain data of a website, where the information itself was meant to be accessed by a humans in first place. Scraping can, in contrast to crawling, violate legal rights (easier), as often site owners specifically do not want their content to be extracted. Crawlers like Google's indexer take that into account and try to index only the "visible" web and intentionally do not try to overcome obstacles which make sure that a certain website is readable only to (certain) humans. Examples would be login-forms, or special sections on a website behind a CAPTCHA. Web scraping means also sometimes to overcome automatically web-forms (like registering forms), CAPTCHAs and other mechanisms to extract the information behind. On the other hand, indexing a whole website during web crawling is nothing else than web scraping, but by trying not to violate any rights by restricting it to the visible web (James, 2012; Teare, 2006).

In other words, web scraping can be done without crawling, meaning, it is not necessary to follow links to extract certain information from a web document. Crawling cannot be done without scraping, meaning, it is not possible to follow links from a website without extracting them in first place. The typical problems that are occurring include broken HTML Markup, so that a normal parser cannot extract certain elements, or heavy use of JavaScript, so that a user has to interact with a website before the content gets generated.

3.3. Text Processing

Text processing can include a variety of topics. Text in general is unstructured data, which needs further processing before it can be fed to a machine learning algorithm. The general approach is to do *information extraction* (IE) to get and transform certain features of the unstructured textual data into structured data. IE is, in comparison to "document management", only interested in certain parts of a document, not all of them (Zhou & Su, 2002).

This section will introduce all common techniques and definitions around information extraction for unstructured textual data. There are different underlying challenges

to fulfill the tasks. One of them is to divide text into its sentences. Intuitively a sentence-separator is the punctuation at the end of the sentence. It can be ambiguous to divide the text at every punctuation mark, since not every point marks the end of the sentence (e.g. "*etc.*"). This is generally known as *sentence boundary detection* and will be covered first in this section. After the boundaries of the sentences are known, they are usually further divided in smaller parts by separating for example at the whitespace between words, or according to some rules. This is known as *tokenization* and will be covered next. After knowing the boundaries of each token, they are usually tagged according to certain attributes, which can be nouns, verbs, adjectives, named entities, or other attributes. This is generally called *Part of Speech Tagging* (POS tagging). When talking about POS tagging, it is mainly meant to tag the tokens according to the Penn Treebank (M. Marcus et al., 1994). POS tagging, formulated as a general approach, is marking parts of the speech in a text according to special tags (Schumaker et al., 2012), which will be covered next. A special form of the part of speech representation are the named entities, which will be covered briefly in a separate section after the POS tagging. After having explained how a structured text is divided and separated, it will be covered how the according data can be saved as a *feature vector* in a structured way for further machine learning tasks and how that is accomplished.

Sentence Boundary Detection and Tokenization

Textual documents as a whole form need to be separated in a way that they become process-able by machine learning algorithms. Basically the fist step is to tokenize the text, where a sub-category of tokenization is the detection of the boundaries of the sentences within a text document, called *sentence boundary detection* (Ratnaparkhi, 1998). Naturally, as a first step, it is intriguing to postulate a sentence always ends at a punctuation mark, like ".", "?" and "!". However, especially the "." leads to ambiguities, because it is used as a decimal point, in e-mail addresses and for abbreviations.

To overcome this problem several solutions are available. One of the solutions is the direct model, where a routine, inspired by regular expressions is used. This is usually limited to one single language and changing the underlaying system is cost intensive and error prone. Another solution would be a Lexically-based classifier. The drawback is that the classifiers dictionary will (and has to) grow over the time, or, even worse, eventually multiple rules will interact with each other in a bad way. Also, manually extending the rule-set appears not only to be difficult, but also time-consuming. Therefore other options are required, where machine learning solutions are suggested. Statistical classifiers, like the Bayesian Framework, or Maximum-Entropy models can be used to identify sentence boundaries, by formulating a statistical classification problem (Walker, Clements, Darwin, & Amtrup, 2001). Rather than comparing different

frameworks to do this task, this thesis will concentrate on the Maximum Entropy (*ME*) framework, as this already proofed to be a state of the art classifier in NLP libraries like Apache's OpenNLP[3].

In short, within the Maximum Entropy framework, a linguistic context $b \in \beta$ is mapped to a class $a \in A$ which is done by a classifier $cl : \beta \rightarrow A$. The classifier in turn can be formulated as a conditional probability distribution p, such that $p(a|b)$. b may be single words or several words and their associated labels. While in large corpora there is information about the co occurrence of a's and b's, it will never be enough information for all possible (a, b) pairs. The challenge for the ME framework is now to find a method for using the partial evidence about the a's and b's to reliably estimate the probability model p (Ratnaparkhi, 1998).

The typical problems that are connected to the sentence boundary detection are explained by the following practical example by Pal (2011a). Given is the text in listing 3.1, where the sentences should be detected:

Listing 3.1: Sample sentence for sentence boundary detection

```
Mrs. Smith was here earlier. At 5 p.m. I had to go to the bank.
```

Naively separating the sentence at any punctuation-mark followed by a whitespace, would split the example into these sentences:

Listing 3.2: Sentence boundaries detected wrongly

```
sentence: Mrs.
sentence: Smith was here earlier.
sentence: At 5 p.m.
sentence: I had to go to the bank.
```

Before the classification can take place with a statistical classifier, the classifier has to be trained with existing, labeled (annotated) text. To overcome this time-consuming task, most libraries offer pre-trained models based on a diversity of corpora and for different languages. If the authors of library itself do not offer those models, they can be downloaded in most cases for a specific target corpus. For example, for bio-medical literature for LingPipe [4], or for general news text from Reuters for OpenNLP [5].

If the above sentence is trained with the LingPipe models, the same mistakes are made and there is no performance gain. If the OpenNLP pre-trained models are used

[3] http://opennlp.apache.org/
[4] http://alias-i.com/lingpipe/demos/tutorial/sentences/read-me.html
[5] http://opennlp.sourceforge.net/models-1.5/

with the Maximum Entropy framework, the two sentences are correctly detected. This means, even though a very sophisticated framework like LingPipe is used, if the models are badly trained, the outcome will not be satisfying.

POS Tagging

According to Mitkov (2003) part-of-speech tagging (*POS tagging*) is "*assigning contextually appropriate grammatical descriptors to words in text*". That means nothing else than assigning a certain tag to an input token, which describes what the token represents grammatically within the sentence. A typical example would be the tagging of nouns, adjectives, adverbs, etc. in sentences. Ratnaparkhi et al. (1996) show that a POS tagger with a maximum entropy model reaches a performance of 96.5% on the corpus of the Wall Street Journal from the Penn Treebank project. This performance is state-of-the-art and will probably be not higher due to inconsistencies in the training set. The reason for their work is that recent papers at that time hypothesize that a better use of context will improve the overall performance.

Initially, the Penn Treebank project has the goal to construct a large annotated corpus, called *the Penn Treebank*, consisting of over 4.5 million words of American English (M. P. Marcus, Marcinkiewicz, & Santorini, 1993). Beside the annotation of the whole corpus for part-of-speech tagging, half of the corpus is also annotated for skeletal syntactic structure. The Penn Treebank tags consist of 48 different tags, where the most important are listed in the appendix in table A.4. The ones that are not listed in the table describe punctuation, mathematical symbols as well as currency symbols. The project was in operation from 1989 - 1996 and produced approximately 7 million words of part-of-speech tagged text, 3 million words of skaletally parsed text and over 2 million words of text parsed for predicate-argument structure. The materials are wide-ranging, from computer manuals, nurse notes to Wall Street Journal articles and transcribed telephone conversations. The whole materials are distributed by the Linguistic Data Consortium (A. Taylor, Marcus, & Santorini, 2003). Due to the huge efforts that the Penn Treebank project made in creating an annotated corpus, it has become the state-of-the-art benchmark for POS tagger and is usually referred to, when one speaks about POS tags.

Named Entity Extraction

Names entity extraction (or named entity recognition) is the task of finding proper Names, Dates, Times, Places, etc. within a text. Tjong Kim Sang and De Meulder (2003) describe it with the following example:

Listing 3.3: Sample sentence for named entity recognition

```
U.N. official Ekeus head for Baghdad.
```

This sentence has three different named entities: The organization *U.N.*, the person *Ekeus* and the location *Baghdad*. According to Chinchor, Brown, Ferro, and Robinson (1999), a named entity is either a person, a location or an organization. Although the name *Named Entiy Recognition* suggests that only *Named Entities* are subject of the search, also number expressions such as monetary, distance, speed, etc. expressions are recognized. Thus, the following three different elements with their tags are subject of recognition and extraction by state-of-the-art entity extractors:

- Named Entities (PERSON, ORGANIZATION, LOCATION)
- Temporal Expressions (DATE, TIME, DURATION)
- Number Expression (MONEY, MEASURE, PERCENT, CARDINAL)

N-Grams

Tokens usually represent a word within a text. And n-grams are a sequence of n items from a sequence. If the items are based on tokens then n-grams are n items in a row of a sequence of tokens. This sequence can be n-grams with $n = 1$, as *unigrams*, $n = 2$ or *bi-grams*, $n = 3$ or *trigrams*, *four-grams*, *five-grams*, etc. Considering the following sample sentence:

Listing 3.4: Sample sentence for N-Grams

```
I like reading
```

The sentence has, with n-grams based on tokens, 3 unigrams: "*I*", "*like*" and "*reading*". And two bigrams: "*I like*" and "*like reading*". If the n-grams are based on characters, then the trigrams would be, "*I l*", " *li*", "*lik*", "*ike*", etc.

Google released a corpus which is based on their activity of crawling and indexing. The corpus contains one trillion words from public Web pages and has around 24GB in a compressed format. After some filtering, around 13.5 million unique words (unigrams on word basis) remain, 314.8 million unique bigrams, 977 million unique trigrams, 1.3 billion fourgrams and 1.1 billion unique fivegrams[6].

[6] http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html

The Feature Vector Representation

To feed textual information into a machine learning algorithm, the data must be in a structured form. The textual representation has to be re-mapped to a numerical representation and there are various ways of doing this. One famous way is a *bag-of-words* approach, where the word-order is lost. This loss is negligible, since the word order carries only little information and is not relevant in text-categorization (Silva & Ribeiro, 2003). In this example the feature vector contains in each row the weight between 0 and 1 based on the relative number of occurrences of a word within this document, encoded with the dictionary from table 3.1:

Listing 3.5: Sample sentence to convert to a feature vector

```
The weather report for today forecasts great weather.
```

Therefore the above text can, after encoding with the dictionary from table 3.1, also be represented as the following sequence of IDs: 5 7 2 1 6 3 4 7. The feature vector now maps each of the features (words in this case) to their relative occurrence within the document. It weights the IDs from the dictionary between 0 and 1: $0 \leq w_{ij} \leq 1$, which is also depicted in table 3.1.

There are several different ways to represent a word within a text in a numerical

Word	ID	Frequency
for	**1**	**0,5**
forecasts	**2**	**0,5**
great	**3**	**0,5**
report	**4**	**0,5**
The	**5**	**0,5**
today	**6**	**0,5**
weather	**7**	**1**

Table 3.1.: Example of a dictionary, and a sample feature vector in bold

form. The simplest one is the *binary* representation: If a word from the dictionary appears it is marked with a 1, if not with 0. Usually this is replaced with the *term frequency*, where, like in the above example, the words are weighted according to their occurrence within a document (Silva & Ribeiro, 2003).

TF-IDF

As mentioned above, to represent text in a way that a it can be used for machine learning, it has to be remapped to a numerical representation. "*Each document is thus represented as a point in a vector space with one dimension for every term in the vocabulary*" (Silva & Ribeiro, 2003). The simplest one is the binary representation, where a word in a document either exists, or not. This is usually replaced by the frequency the word occurs in a document, the *term frequency* (*tf*). To gain more information, it could be interesting to weight the terms according to their occurrence within the whole document collection. Usually words, which are occurring in fewer documents are better selectors. The *document frequency* (*df(t)*) is the number of documents the term appears in. Thus, the *inverse document frequency* (*idf(t)*) is:

$$idf(t) = \frac{|D|}{df(t)} \tag{3.1}$$

$|D|$ is the total number of Documents. $df(t)$ the number of documents the term t appears in. Now, vector components can be weighted according to the idf. Additionally, to avoid giving highly frequent words more importance than appropriate, the log or the square root can be applied to the tf.
The most common weighting is called the *Term Frequency - Inverse Document Frequency* (*Tf-IDF*):

$$\text{TF-IDF} = tf \times \log(idf) = tf \times \log\left(\frac{|D|}{df(t)}\right) \tag{3.2}$$

TF-CDF

A potential useful concept is the category frequency (CF) (Peramunetilleke & Wong, 2002), where for every possible category (e.g. dollar up, down, steady) the CF of a keyword is the times the keyword occurs in that category. The Category Discrimination (CDF) is then

$$CDF_i = \frac{\max(CF_{i,up}, CF_{i,down}, CF_{i,steady})}{DF_i}, \tag{3.3}$$

where DF_i is number of time windows containing keyword tuple i at least once. The weight $w_i(t)$ is then calculated by multiplying the term frequency $TF_i(t)$ with the CDF_i. Finally $w_i(t)$ is normalized.

Cosine Similarity

Having the terms already in a vector space model, it could be beneficial to know the similarity between two documents. If each document is represented as a vector, the angle between two vectors can used as a measurement of divergence between two vectors. If the angle is known, the cosine of the angles can be used as the numeric similarity between the two vectors, which varies between 0.0 (nothing in common) and 1.0 (exactly the same). An alternative to the cosine similarity is often the dot-product between two vectors. When the vectors are normalized to unit length, then the cosine of the angle between two vectors is the same as the dot-product (Singhal, 2001). The similarity between two vectors $\mathbf{D}$ and $\mathbf{Q}$ is defined as

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{Q} \cdot \mathbf{D}}{\|\mathbf{Q}\|\|\mathbf{D}\|} = \frac{\sum_{i=1}^{n} \mathbf{Q_i} \times \mathbf{D_i}}{\sqrt{\sum_{i=1}^{n} (\mathbf{Q_i})^2} \times \sqrt{\sum_{i=1}^{n} (\mathbf{D_i})^2}} \tag{3.4}$$

Stop Words

A typical text-collection consists of numerous features which have little representative importance regarding the documents content. These are usually words which occur very frequent in a given text corpus. Fox (1989) constructed such a stop word list based on 1,014,000 words from a corpus containing a broad range of literature in English. Alone the first three stop words ("the", "of" and "and") within this list of 421 words account for 13% of the whole corpus. Since the stop words are adding little to no information to the documents content, they should be removed in the feature set. The relational database engine MyISAM from MySQL has a stop-word list built into their system which contains 434 most frequent words. This list is automatically applied to the search index. While it makes perfect sense on first glance to remove all stop-words from a query to gain the containing information, it would reduce the results for some queries to zero. For example, the following sentence "to be or not to be" would, after removing all stop words, contain no more words. The full list of the MySQL built in stop-word list can be found in the appendix A.5.

3.4. Machine Learning

Self learning systems in the discipline of text recognition, clustering, text categorization, etc. have been an active field of research since the mid 1950's (Quinlan, 1986). Especially the ability to *learn* provides the potential for high-performance systems. The field of machine learning is divided into a diverse subset of different disciplines and algorithm types. It borrows its underlying ideas and theories from statistics, computer

science, engineering, cognitive science, optimization theory and many others. Basically it is divided into three to four broad topics. *Supervised learning* has the goal to produce a correct output based on a training on labeled samples. *Reinforcement learning*, where the machine produces a certain action-output and gets rewarded or gets punished, based on the performance the output reaches. The goal is to maximize the reward and minimize the punishment. Basically the same idea applies to the *game theory*. The fourth broad method is *unsupervised learning*, where the goal is to find patterns in unstructured data (Ghahramani, 2004).

Supervised and unsupervised learning methods are heavily used within this thesis, therefore the following section will cover unsupervised learning methods and clustering methods and then supervised learning methods and support vector machines. Other methods like reinforcement learning, semi-supervised learning or transduction are not covered by this thesis.

Unsupervised learning

Unlike any other learning method, unsupervised learning has the goal to learn *something* from unstructured data. The learning phase is usually pattern finding in data which is normally considered as purely unstructured noise. The classic examples of unsupervised learning are clustering and dimensionality reduction. (Ghahramani, 2004) gives a great overview and a comprehensive explanation of all different methods for unsupervised learning.

Especially the latent variable models are interesting. With some special assumptions within the *principal components analysis* (PCA), it is possible to do *singular value decomposition* (SVD). The driving idea is to cluster (web) text documents into their thematic threads and put the documents into meaningful cluster descriptions (Osiński, Stefanowski, & Weiss, 2004).

Text Clustering

When using modern web search engines, the results are presented as a list without further grouping. If a query is used which is not very meaningful, a broad range of documents relating to a variety of topics is returned and the user is forced to go through those topics. For example a search query like "*tiger summer*" would return documents to the animal and to the sports group "Hockey Tigers" and also to "Tiger Woods". At the time of writing, the total search results amount to 144 million search results with

mixed thematic clusters using Google. A commercial website to cluster search results was Vivisimo[7] and is now part of the InfoSphere Data Explorer from IBM.

Lingo

The algorithm Lingo (Osiński et al., 2004) tries to solve the problem of text clustering as well and is open source. Lingo is able to capture thematic threads in search results: First clusters are detected, then related documents are grouped together and the clusters are described in a meaningful way. The background behind is the utilization of a term-document matrix A, which is the relationship between the unique terms t and the number of documents d, expressed as the matrix $A = t \times d$. The element a_{ij} from the term-document matrix A is therefore the (weighted) relationship between the term i and the document j. Having A, the distance between two documents can be calculated measuring the cosine for example.

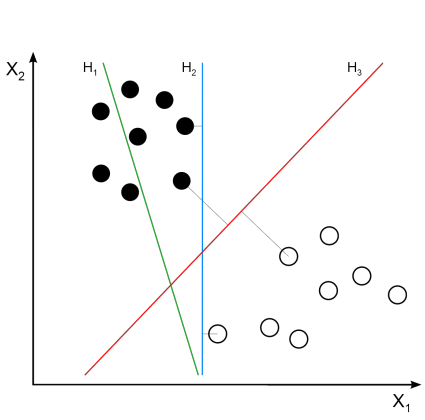
Lingo basically makes use of a technique called *latent semantic indexing* (LSI), where the underlaying idea is to reduce the rank of a term-frequency matrix in order to remove noisy words. To do so, the latent structure of concepts in documents are exploited using an algebraic method of matrix decomposition called *singular value decomposition* (SVD). Before the actual processing can occur, a proper preprocessing is necessary: Stemming and stop words removal, removal of any non-text characters (HTML-Tags, non-letter characters, etc.).

Frequent phrases occur in a document when one talks about a certain topic. These phrases are candidates for cluster labels if they hit certain criteria and parameters chosen in advance. Once the frequent phrases are known, they are used for cluster label induction. Then, after the clusters are found based on the documents in the collection, each document is assigned to a cluster which has the smallest distance. If a document doesn't fit into any cluster, it is assigned to the category "others".

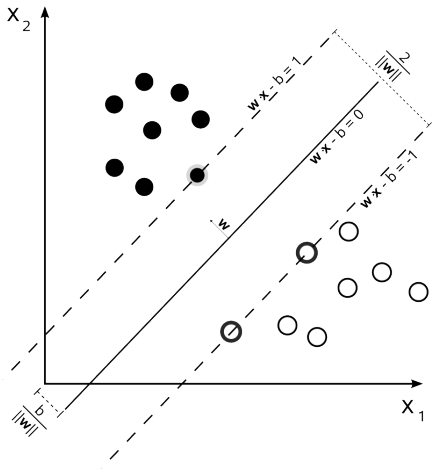
Supervised learning

The goal of supervised learning is that a learner can, based on a training with labeled samples, predict a correct label of an unlabeled sample. This learning scenario is the most common for classification, regression and ranking problems (Mohri, Rostamizadeh, & Talwalkar, 2012). The last years a vast amount of research was running into the classification of textual data, with application domains in a diverse range. From computer vision, speech recognition, biological classification, credit scoring to document classification. The *no free lunch theorem* states that, averaged over all

[7] http://www.vivisimo.com



(a) Graphic showing how a support vector machine would choose a separating hyperplane for two classes of points in 2D. H_1 does not separate the classes. H_2 does, but only with a small margin. H_3 separates them with the maximum margin (Zack Weinberg, 2012).



(b) Graphic showing the maximum separating hyperplane and the margin (Peter Buch, 2008).

Figure 3.1.: Support vector machine process: first finding linear separation hyperplanes and then finding the hyperplane with the maximum margin.

possible cost functions, all optimization algorithms perform the same well (Wolpert & Macready, 1995). In other words, each machine learning algorithm, with regard to a specific problem, has the one or the other trade-off. Thus, the primary goal in a text classification task is to maximize the performance in terms of accuracy.

Aggarwal and Zhai (2012) compared Decision Trees, rule based classifiers, SVM classifiers, Bayesian classifiers, as well as some other classifiers with a text categorization problem. They could show that SVM classifiers are particularly suited to the characteristics of text data.

Support Vector Machines

The idea for the *support vector machine* (SVM) was first introduced by Cortes and Vapnik (1995). The linear variant of the support vector machine was introduced first. It tries to determine linear separators between two different classes. Considering the example shown in figure 3.1a, the SVM tries to separate the training set in the two classes, represented by black and white dots. The best separation is reached, when the margin between the line of separation and the classes is maximized, as shown in the

figure 3.1b. Due to the way the SVM tries to find the optimal hyperplane by examining the appropriate combination of features, it is very robust to high dimensional data. Especially this makes it so suitable for textual data, since dealing with text usually brings up sparse high-dimensional feature vectors (Aggarwal & Zhai, 2012).

Cortes and Vapnik (1995) describe the idea as follows: first the input vector is mapped into some high dimensional feature space Z through some non-linear mapping chosen a priori. Then a linear decision surface is constructed with special properties that ensure high generalization ability of the network.

Optimal Hyperplane

A two dimensional, two class classification problem is shown in figure 3.1. There, a set of labeled training patterns may be from the classes "black" and "white" or -1 and +1 which is given by

$$(y_1, \mathbf{x_1}), \ldots, (y_l, \mathbf{x_l}), \quad y_i \in \{-1, 1\}. \tag{3.5}$$

This training set should be linearly separable if there is a vector $\mathbf{w}$ and a scalar b so that for all elements of the training set the following inequalities are correct:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x_i} + b \geq 1 \quad \text{if} \quad y_i = 1, \\ \mathbf{w} \cdot \mathbf{x_i} + b \leq -1 \quad \text{if} \quad y_i = -1. \end{aligned} \tag{3.6}$$

Combined, these inequalities can be written as

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1, \quad i = 1, \ldots, l. \tag{3.7}$$

The optimal hyperplane separating the training space with a maximum margin is

$$\mathbf{w_0} \cdot \mathbf{x} + b_0 = 0. \tag{3.8}$$

The training samples which are directly on the positive hyperplane $\mathbf{w} \cdot \mathbf{x_i} + b \geq 1$ or on the negative hyperplane $\mathbf{w} \cdot \mathbf{x_i} + b \leq 1$ are called *support vectors* and are bold circled in figure 3.1b. The distance between the hyperplanes on the support vectors is $\frac{2}{\|\mathbf{w}\|}$, where $\|\mathbf{w}\|$ is the Euclidean norm of $\mathbf{w}$. After substituting $\|\mathbf{w}\|$ with $\frac{1}{2}\|\mathbf{w}\|^2$, the optimal hyperplane is the one that minimizes $\mathbf{w} \cdot \mathbf{w}$ under the constraint of equation 3.7, which can be reformulated as maximization problem and is shown later.

If the training set is not separable by a hyperplane, the margin would be arbitrary small. In this case a soft-margin hyperplane is introduced, where one can separate the training set with a minimum number of errors.

Soft Margin Hyperplane

If one processes textual data, the training data is usually inherently noisy, which means the training set cannot be separated. To overcome this problem a soft margin hyperplane is introduces, that allows training samples to be moved to the wrong side of the classification. Additionally a penalty parameter is assigned to such samples (Bacher & Stuckenschmidt, 2012).

Cortes and Vapnik (1995) show that the equation

$$\Phi(\xi) = \sum_{i=1}^{l} \xi_i^{\sigma}, \tag{3.9}$$

where ξ_i is some non-negative variable and subject to the constraints

$$\begin{aligned} y_i(\mathbf{w} \cdot \mathbf{x_i} + b) &\geq 1 - \xi_i, \quad i = 1, \ldots, l, \\ \xi_i &\geq 0, \quad i = 1, \ldots, l, \end{aligned} \tag{3.10}$$

describes the number of training errors for sufficiently small σ. When equation 3.9 gets minimized, the minimal subset of training errors can be found on the training set, which can be excluded, and the training set can then be separated without errors. To separate the remaining part of the training data a penalty parameter is introduced and one has to minimize the function

$$\frac{1}{2}\mathbf{w}^2 + CF\left(\sum_{i=1}^{l} \xi_i^{\sigma}\right), \tag{3.11}$$

subject to the constraints 3.10. To avoid NP-completeness, only the case where $\sigma = 1$ is considered. The minimization problem can solved using Lagrange multipliers and can be reformulated as a maximization problem which also removes ξ_i from the equation

$$\tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{3.12}$$

subject to the constraints

$$\begin{aligned} &0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l \\ &\sum_{i=1}^{n} \alpha_i y_i = 0, \quad i = 1, \ldots, l. \end{aligned} \tag{3.13}$$

The only additional constraint to the Lagrange multipliers is the constant C.

Kernels

The original proposed optimal classifier was, as already mentioned, a linear classifier. Boser, Guyon, and Vapnik (1992) suggested a way to extend the linear SVM classifier by replacing the dot product $k(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ by a nonlinear kernel function. The input data can therefore be transformed nonlinearly into a high dimensional feature space using the *kernel trick*. The optimal hyperplane can be found in the high dimensional feature space (see figure 3.2).

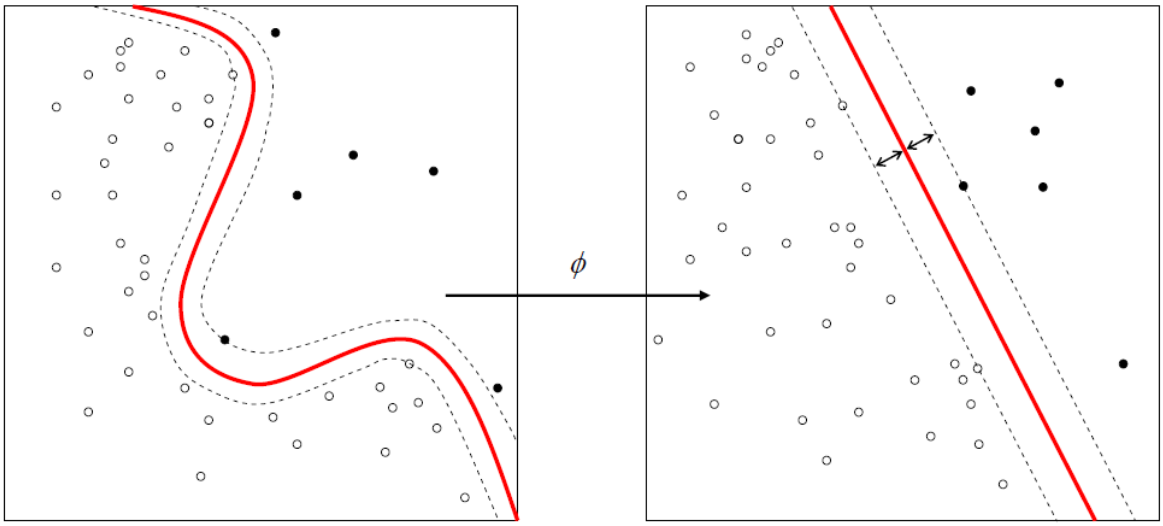


Figure 3.2.: Kernel machines are used to compute a non-linearly seperable functions into a higher dimension linearly seperable function. [Image taken from Alisneaky (2011)]

Hsu, Chang, C.-J. Lin, et al. (2003) created a practical "cookbook" for SVM classification tasks and describe the most common kernels used for every-day classification problems. The most common kernels are

- linear: $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- polynomial: $k(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^d, \quad \gamma > 0$
- radial basis function (RBF): $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \quad \gamma > 0$
- sigmoid: $k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i^T \mathbf{x}_j + r)$.

The parameters γ, r and d are kernel parameters.

LIBSVM and LIBLINEAR

Due to the fact that support vector machines are so popular, several frameworks and packages were developed over the time. One of the most popular software packages

performing classification, regression and other learning tasks with SVMs is LIBSVM by Chang and C.-J. Lin (2011). The package also supports multiclass classification where the "one-against-one" approach is used. This approach constructs for k number of classes $\frac{k(k-1)}{2}$ classifiers and each one trains data from two classes. They implement a "voting" strategy, where each binary classification can vote for a class. The class with the highest number of votes wins, or, if two classes have equal many votes, then the first one in the list.

On of the biggest disadvantages is still the training time. To train large-scale text corpora, a general SVM solver such as LIBSVM would need several hours to train the classifier on huge corpora like the Reuters Corpus Volume 1 (see 3.2) (Fan, Chang, Hsieh, Wang, & Lin, 2008). LIBLINEAR is especially for large sparse data with a huge number of instances, as it is normally the case with textual data, and needs only several seconds to train a classifier on that corpus. It inherits many features of LIBSVM, is very efficient for training large-scale problems and competitive with or even faster than state of the art linear classifiers such as Pegasos (Shalev-shwartz, Singer, & Srebro, 2007) and SVM$^{\text{perf}}$ (Joachims, 2006).

Evaluation

How *good* the classification task is accomplished can be measured in various ways. Basically it is distinguished between how well the task of classification is done and how long it took to execute the task. Although this thesis is also party interested in a fast overall process, at the end of the day the quality of classification is what counts. There are numerous metrics to measure the quality or *effectiveness* of the classification (Sebastiani, 2002).

The two classic information retrieval metrics are *precision* and *recall*. Precision π_i is *"the probability that if a random document d_x is classified under c_i, this decision is correct."*. Recall ρ_i is *"the probability that, if a random document d_x ought to be classified under c_i, this decision is taken"* (Sebastiani, 2002). Basically four different classification metrics are used to calculate precision and recall. The *true positives* (TP) are the samples which should be in category c_{pos} and were classified correctly to the category. The *true negatives* (TN) are the samples which should be in category c_{neg} and were also classified correctly. The *false positives* (FP) are such samples, which should be in c_{neg} but are classified to c_{pos}, and the *false negatives* (FN) are samples which should be in c_{pos} but are classified as c_{neg}. These are usually presented in a confusion

matrix, as depicted in table 3.2. Precision and recall are then calculated with

$$\begin{aligned} \pi &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \\ \rho &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \end{aligned} \tag{3.14}$$

Another measure the performance of ML classifier is the *accuracy* A and the *error* e. The accuracy calculates the number of overall correctly classified samples against all samples and the error all wrongly classified samples against all samples:

$$A = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}, \tag{3.15}$$

$$e = 1 - A = \frac{\mathrm{FP}+\mathrm{FN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}. \tag{3.16}$$

		Actual Category	
		Positive	Negative
Predicted Category	Positive	**true positive** (TP)	**false positive** (FP)
	Negative	**false negative** (FN)	**true negative** (TN)

Table 3.2.: The confusion matrix shows the predicted category versus the actual category of samples between two categories (positive and negative). In the first row are the *true positives* and *false positives*. In the second row are the *false negatives* and *true positives*.

K-Fold Cross Validation

Cross validation is the more simplistic variant of *k-fold* cross validation, where the samples are divided in half. The first half is used for training, the second half for testing and vice versa (Geisser, 1993). K-fold cross validation, sometimes called rotation estimation, is done by splitting a dataset D in random, mutually exclusive k-subsets $D_1, D_2, \ldots, D_k$ of approximately equal size. Training and testing is done k times, where each time $t \in \{1, 2, \ldots, k\}$ the classifier is trained on $D \setminus D_t$ and tested on D_t. A large-scale experiment concluded that ten-fold cross validation is the best method for model selection (Kohavi, 1995). With ten-fold cross validation the sample set is divided in 10 subset of (almost) the same size. If the sample set has 10 elements $D_1, D_2, \ldots, D_{10}$ then at first $D_1, \ldots, D_9$ are taken for training the classifier and D_{10} for validation, then D_9 is used for validation and the other parts $D_1, \ldots, D_8, D_{10}$ are used for training and so on. So each of the parts is once excluded from the training

and used for validation, the process is done for k-times. At the end the average over the results of the validation is calculated.

3.5. Summary

A lot of research is directed towards sentiment analysis. The sentiment itself is defined as the summary of all positive and negative opinions and emotions towards an entity. Applications with a variety of goals are connected with sentiment analysis, where the most famous is sentiment polarity classification for applications like movie or product recommender systems. Most of the applications have in common that the source of information is textual data. The first and foremost problem is the right source of textual data to reduce noise during the training of the classifier. Sometimes it is even better to gather data from the web, if none of the available corpora fits for training, even though with web scraping raise some additional challenges. Before the textual data can be used for machine learning, a transformation to a numerical feature vector has to be done. Sentence detection, tokenization, POS tagging and sometimes even named entity recognition are part of almost every NLP task. Additional stop word removal and choosing the right weighting of the features, for example by TF-IDF, can increase the overall performance. Important for this thesis are the two big learning methods unsupervised and supervised learning. The former is used for example to find topics in a corpus of documents and cluster documents with similar topics together. The latter is used to train a classifier with labeled sample documents in order to identify the right label of an unlabeled document. Support vector machines have proven themselves to be very good for classification of textual data, due to the way they handle large-sparse feature vectors. To evaluate the model, tests in large-scale experiments where made, which yielded that ten-fold cross validation is the best method for model selection.

4. Related Work Predicting Financial Data

"Knowledge is true opinion."

[Plato]

Trying to make predictions on the financial markets is nothing new. Crucial for this work is the review of related work and will be discussed in this chapter. The approaches, as well as the results, are spanning a broad variety between simple techniques and bad results, to sophisticated text mining approaches with questionable results. The survey paper of Marc-André Mittermayer and Gerhard Knolmayer (2006) summarizes comprehensively the Stock and Forex market prediction systems, which are based on a text mining approach and compares the results up to the year 2006. Their discussion will be shortly summarized and supplemented by newer research. The general approach to predict market activity is very abstracted shown in figure 4.1. This section is structured as follow: First work is presented, which is trying to make predictions on the Stock market, then work is presented which is trying to predict exchange rates on the Forex market, each ordered by date.

4.1. Predictions on the Stock Market

Wüthrich et al. (1998) were one of the first one who developed a prototype to predict movements on the Stock market. Five major equity indices are attempted to be forecasted. The predictions are based on textual data from online available newspapers like the Financial Times, Reuters, or the Wall Street Journal. The articles are labeled by 3 categories: up, steady, down. The features are defined manually by experts, which are tuples of words, which can only be seen by examples given, because the original dictionary is not published. This dictionary is used to train three types of classifier: a Naïve Bayes classifier, a Nearest Neighbor classifier and a Neural Net.
The system tries to classify all articles which are published overnight and gives recommendations. In a simulated environment the Stocks are bought (or short-sold) based on these recommendations and the position is hold until the end of the day. The authors

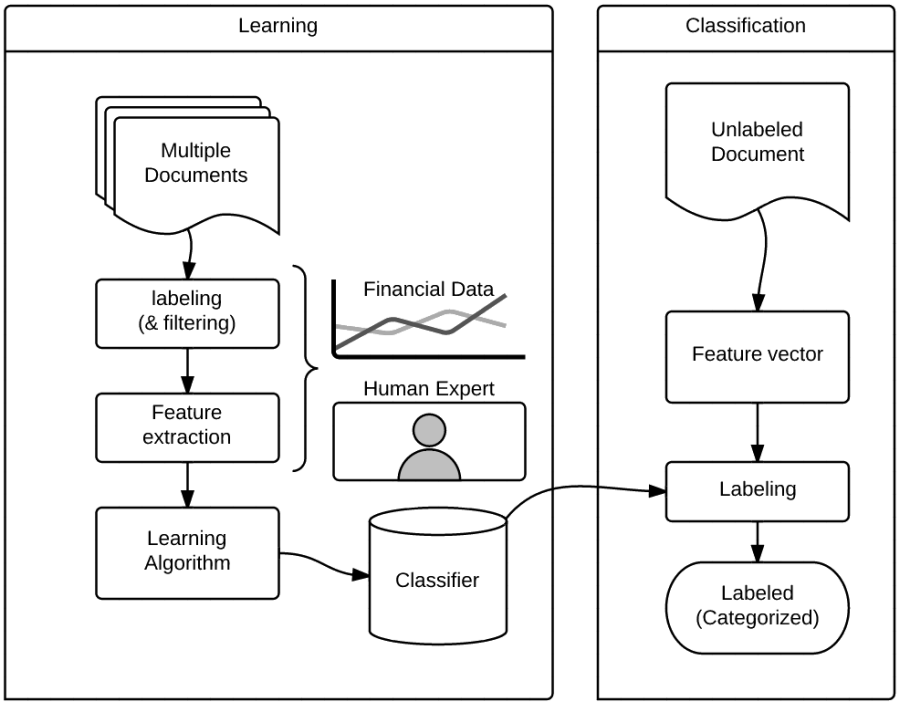


Figure 4.1.: Abstracted prediction flow based on textual data

claim an accuracy of 40% to 46.7%, depending on the news source (33.3% would be random), giving on average 13 bps (1 basis point or *beep* = $\frac{1}{100}$%) profit per round trip. Although there are two problems with this result: First of all the researchers assume that the price is not changing overnight (Marc-André Mittermayer & Gerhard Knolmayer, 2006), second they assume that there are no trading fees. This makes the system very unlikely to be successful in a real world scenario.

Fawcett and Provost (1999)

Fawcett and Provost (1999) try to solve an activity monitoring problem. They demonstrate that the domain of fraud detection can also be applied to different domains like creating Stock market alerts based on news stories. The goal is to create an alarm when there is an "interesting event", which is defined as a 10% change in the companies stock price. The system is tested with tagged news stories and stock prices of approximately 6000 companies. The news stories were lexically analyzed, stemmed and filtered with a stop word list and processed using bi-grams. The performance was evaluated by 10-fold cross-validation. The results are just given as a graphic figure. With an average score of (about) 0.5 one can expect a false alarm rate of 0.1. No more discussions is available

about these results within the paper.

Lavrenko et al. (2000b)

Lavrenko et al. (2000b) developed Ænalyst, a system which tries to predict trends in stock prices based on 38,469 news stories for 127 stocks from Biz Yahoo! (now Yahoo! Finance). The news stories are labeled based on 5 categories (SURGE, SLIGHT+, NO RECOMMENDATION, SLIGHT-, PLUNGES). The categories are based on the slope of the re-described time series of the according stock price, using piecewise linear regression. Then a Naïve Bayes classifier is trained. Based on the classification of news, buy or short-sell recommendations are given. These are evaluated with a Stock market simulation, using the rule that an opened position is closed when either 1% of the profit zone is reached, or at the latest 60 minutes after opening the position. Marc-André Mittermayer and Gerhard Knolmayer (2006) state the asymmetry of the rule: No stop loss limit is defined, which can cause a severe loss in 60 minutes in a real world situation.
The simulation works based on M10 stock prices between March and April 2000. On each position USD 10.000 are invested, giving an impressive result of USD 280,000 after 40 days by performing 12,000 transactions (23 bps). No transaction costs were taken into account, as well as 12,000 recommendations are very unlikely to happen with only 127 stocks in 40 days. Furthermore for the simulation only highly volatile stocks are included and it must be assumed that unlimited funds are available.

Thomas and Sycara (2000)

Thomas and Sycara (2000) began developing a prototype, which aims to compare two approaches to make predictions on financial markets utilizing online available bulletin board posts. The first approach uses a maximum entropy text classification, the second genetic algorithms. 22 stocks are the target to predict with textual data from bulletin boards from January 1, 1999 to December 31, 1999. They claim a remarkable 30% return rate, after combining the two approaches and further filtering out stocks without a great amount of bulletin board entries. Without combining the approaches they have excess returns for the genetic rule learner between -6.22% (all stocks) and 4.95% (>10k bulletin board posts), and for the text learner -8.11% (all stocks) and 6.92% (>10k bulletin board posts). Although it is highlighted that they are only interested in the profitability of the trading rules, they do not make any provision for transaction costs, nor do they give any further numbers about the

amount of trades, the amount of recommendations, or any other details about the trades.

Thomas (2003)

Thomas (2003) developed a more sophisticated system to make forecasts on financial markets. While previous attempts are focusing on stock prices, newer strategies are trying to predict the volatility. The prototype is a manually created rule-based classifier for 39 categories, and is more or less a keyword-pattern finder. It was tested with members of the Russell 3000 index between December 2001 and April 2002. Although no clear performance of system is given (Marc-André Mittermayer & Gerhard Knolmayer, 2006) in a market simulation, the classifier itself has a precision of about 90% and recall of about 70%.

Gidófalvi (2003)

In the year 2003 Gidófalvi developed a system which seems to be very similar to Gidófalvi and Elkan (2001). While the original paper from 2003 cannot be found to the authors best knowledge, it is described by Marc-André Mittermayer and Gerhard Knolmayer (2006) and found also a comprehensive mention in Bacher and Stuckenschmidt (2012). The system tries to predict future stock prices on a minute-by-minute basis based on news. Similar news entries are eliminated and a windows around the news is created where the effect of the news on the stock price is examined. A three label classification (UP, DOWN, NORMAL) with a Naïve Bayes classifier is used. In a market simulation between July 26, 2001 and March 16, 2002 stock prices of the 30 DJIA companies and around 6000 news stories are used. The dataset has a resolution of M10. Transaction costs are not considered. The most profitable setting is, when the news is known 20 minutes in advance to the news-release and the stock is liquidated when the news is released, which strongly suggests the evidence for insider information. 0.10% profit per round trip is achieved.

Schulz, Spiliopoulou, and Winkler (2003)

Schulz et al. (2003) and Spiliopoulou, Schulz, and Winkler (2003) try to classify news stories as relevant or irrelevant to stock price movements. They try to address a problem from market participants from the German Stock market, where the high amount of available news makes it hard to distinguish between relevant and irrelevant news. They use the commercial software SAS Enterprise Miner to label news as

PRICE_RELEVANT or PRICE_IRRELEVANT. An average classification error of 39% is measured, however for the PRICE_RELEVANT tag only a much higher error of 57% is measured. The problem seems to be that the type of news is mainly used by companies for advertisements purposes only (Bacher & Stuckenschmidt, 2012).

G. P. C. Fung, Yu, and Lu (2005)

Three prototypes were developed at the Chinese University of Hong Kong (G. Fung, Yu, & Lam, 2002; G. P. C. Fung, Yu, & Lam, 2003, 2005) which aim to predict stock price trends at the Hong Kong Stock Exchange. For the earlier two prototypes 614 stocks are chosen. The documents are labeled with a very similar approach as done by Lavrenko et al. (2000b): a time window around the news is taken and the labeled with three different categories: Rise, Drop or Steady. The processing of the news was performed with a commercial text mining software from IBM called Intelligent Miner for Text and as a classifier a SVM was used from the University of Dortmund. Unfortunately the authors leave the reader in the dark about essential information, as well as basic chart axis labeling is missing, which makes it difficult to gain insights about the results (Marc-André Mittermayer & Gerhard Knolmayer, 2006).
The latest version (G. P. C. Fung et al., 2005) makes predictions on all intra-day stock transactions of all Hong Kong stocks and news is gathered from the commercial platform Reuters 3000 Xtra. The authors claim, with a performance evaluation in a simulation, an accumulated profit of 18.06% and a rate of correct prediction of 61.6%. The bps cannot be calculated since the numbers of trades are not published. Also the way the authors handle stories with similar content may lead to prediction errors (Bacher & Stuckenschmidt, 2012).

M.-A. Mittermayer and G.F. Knolmayer (2006)

M.-A. Mittermayer and G.F. Knolmayer (2006) based their development on comprehensive research of previous work. The outcome is a prototype called *NewsCATS - A News Categorization And Trading System* to predict intraday stock prices using U.S. press releases. They use news from April 1, 2002 to December 31, 2002 (about 18,000 press releases) associated with only one company which appears during trading hours in a category they consider as relevant. This is news which has a high probability of being distributed simultaneously to the market. In other words: news which has a low private information part, to exclude insider trading. After filtering 9,128 news releases remained. As a feature-set they mix together an automated created bag-of-words list and a handcrafted thesaurus containing features that were assumed to be relevant for

stock price prediction. Classifiers are trained with three different categories: good, bad and neutral on M1 trading data. After training they tested the performance using a 10-fold cross validation as well as a real market situation. The default NewsCATS model reaches an accuracy of 82% (33% would be random). They varied their settings in a number of ways. The best results for the feature selection are with a Collection Term Frequency. For the feature set size it is optimal to use 15% of the number of documents (although it doesn't change much with different sizes). The best document representation is WDFxIDF. While the non-linear SVM is superior over the linear SVM regarding the round trip profit, the linear SVM achieves a better F_1 value. They did not take the costs of trading into account and claim that profits including the costs of trading would probably lead to a zero return rate. However, compared with older prototypes, they claim that their results are remarkably better, taking into account that older systems never obtained more than 50%.

M.-C. Lin, Lee, Kao, and Chen (2008)

M.-C. Lin et al. (2008) focus their work on stock price movement prediction of companies listed in the S&P Index 500 as of September 2008. They gathered 26,255 reports from the EDGAR system from the US Securities and Exchange Commission. The resolution of the financial data is D1. While the whole labeling approach is a rather normal 3-category model with RISE/DROP/NO MOVEMENT labels, an interesting approach is used to build the feature set. They extract quantitative and qualitative features from financial reports. While qualitative features are typical textual features, the quantitative features are some ratios about the performance of the company. They select five financial features: operating margin, return on equity (ROE), return on total assets (ROTA), equity to capital and receivables turnover. To predict short-term stock price movements they propose an effective clustering method called HRK (Hierarchical agglomerative and Recursive K-means clustering). HRK outperforms a SVM and a Naïve Bayes classifier in terms of accuracy and average profit. They also used paired t-tests with a 95% confidence level where they clearly showed that HRK outperforms all of the other classifier.

Robertson (2008)

Several investigations were made (Robertson, Geva, & Wolff, 2006, 2007a, 2007c, 2007b) how the news affects the US, UK and Australian Stock market, where the most recent work ended in a prototype by Robertson (2008). The problem addressed is the news release overflow. Using the program Bloomberg Professional news is downloaded on a

day-to-day basis from over 200 different news sources. Then the difference between the forecasted and realized stock volatility is calculated using a modified GARCH model. If the difference is high, it is very likely that the news contains some information which lead to the difference in the forecast. Therefore news released before the forecast-error with a defined time windows Δt is labeled as INTERESTING. Two classifiers are trained, a SVM and a C4.5 decision tree and achieved an 80% accuracy which was measured with a market simulation.

Li et al. (2011)

Li et al. (2011) try to predict Stock market price movements from the Hon Kong Stock market. They use a novel multi-kernel learning (MKL) technique, utilizing information in market news and stock prices. Also addressed is the problem that good (bad) news not necessarily mean that it leads to prices going up (falling down) immediately, it could also just flatten the price curve, which is different to the traditional view, that is, "*good news means up, bad news means down*". As a representation of the news they choose a bag-of-words approach, as a representation of the price data they use a smoothed price curve with an tick-by-tick resolution. The news articles are from the year 2001 and bought from Caihua, a Hong Kong based Newspaper published in traditional Chinese with a time stamp of the publication. The prototype is validated with a 5-fold cross validation. They compare different approaches and vary the time of the impact-classification from $t = 5$ to $t = 30$ minutes. Accuracy is used as a metric to compare different classifiers, where MKL performs best.

Schumaker et al. (2012)

Another prototype to forecast intraday stock prices based on financial market news articles was developed between 2006 and 2010 (Schumaker & Chen, 2006, 2008, 2009, 2010). This prototype was then modified to take sentiment analysis into account by Schumaker et al. (2012). The news articles are gathered from Yahoo! Finance and feeded into their developed Arizona Financial Text system (AZFinText) which uses Support Vector Regression. Rather than Support Vector Machines, Support Vector Regression is a regression system other than a classification system, which assigns numerical values other than a binary classification. The system makes price prediction 20 minutes after the article has been published, where they claim the first 20 minutes after a release have a weak predictability. As a feature a noun representation is used, which was empirical evaluated and outperforms other representations. The model is generally divided into three partitions. A normal AZFinText system which only takes

proper nouns of an article and the stock price into account. A second one which is also an AZFinText system plus three sentiment features (objective, subjective and neutral), where the OpinionFinder tool is used to make the distinction. A third model which takes three subjective sub-features into account. The experimental period of research is from October 26, 2005 to November 28, 2005, which has 23 trading days. They limited their news to such articles which were released between 18:30am and 3:40pm and contain information about companies which are listed on the S&P 500. The financial data resolution is M1. The threshold was set to 1%, meaning the system invests $1000 worth of stocks if the expected movement exceeds 1% for 20 minutes. Zero trading costs are assumed. They tried to answer two questions. The first is if objectivity/subjectivity impact news article prediction, where they found the best accuracy had subjective articles with 59% and an average trading return of 3.3%. The second question they tried to answer is if polarity has an impact. They found negative polarity has the greatest impact, where they found that their prototype predicted best downswings of price in positive polarity articles and upswings in negative polarity articles, which stands in contrast to earlier work (Tetlock, 2007). They suggest to investigate the role of other parts of a textual representation, other than only proper nouns alone, which could lead to a much higher accuracy.

Hagenau, Liebmann, Hedwig, and Neumann (2012)

Hagenau et al. (2012) examine the impact of unstructured textual information in financial news to stock price movements. The news data corpus are German Adhoc messages from DGAP ("Deutsche Gesellschaft für Adhoc-Publizität") and EuroAdhoc. The resolution of the stock prices is D1. Four different feature types are compared: single words, bi-grams, 2-word-combinations, noun phrases and additionally a dictionary-approach, where single words from the positive and negative word list of the Harvard-IV-4 psychosocial dictionary are used. As classification a linear SVM is trained. Additionally feature selection based on chi-squared (χ^2) is compared to no feature selection, where a χ^2 feature selection improves the classification accuracies for all feature types. The best accuracy is achieved by 2 word combinations with a χ^2 feature selection, giving 65.1% on the validation set.

Bacher and Stuckenschmidt (2012)

Bacher and Stuckenschmidt (2012) developed another approach to forecast stock price movements which incorporates a couple of interesting aspects and will therefore be reviewed a little bit more in detail. In their thesis a comprehensive review of already

existing work is done and the new system is based on several highlighted key findings. To address the noise of news only news regulated by US government is used. As feature set named entities, POS tags and document sentiment are used. The labeling approach with two and three classes are compared to each other and all important classifiers are implemented and also compared. This is evaluated in a market situation. Both, the stocks of interest, which are from the S&P 500 index, and the news are from the time between February 6, 2012 to April 23, 2012 and from May 7, 2012 to June 15, 2012. Tick-data was transformed into data with a lower resolution of 15 seconds. The news is downloaded through the commercial service LexisNexis, that is made temporary available, which aggregates news from the market leaders PR Newswire and Business Wire. Only news was considered in the trading hours and further filtered by a thesaurus. A further news filtering or categorization to a company was not necessary because LexiNexis news is already labeled with the relevancy to each company. The freely available machine learning framework Weka is used. The labeling is similar to M.-A. Mittermayer and G.F. Knolmayer (2006) an the threshold for the return of >+0.3% (<-0.3%) for a BUY (SELL or HOLD otherwise) label is consistent with the one proposed in Li et al. (2011), which results in three categories for BUY/HOLD/SELL (81/778/80 news articles). For the 2 categories model, the threshold is set to 0%, which results in 581 BUY and 368 SELL labels. Additionally to the overall comparison of the results, they compare the automatic labeling approach to a manual approach, where they manually label articles as BUY/SELL recommendation if certain topics occur, which they identify as good/bad news. The automatic labeling is slightly better than a random labeling, where 56% were automatically correctly labeled, 44% are incorrectly classified. Although it is claimed that the bad accuracy will likely cause bad results for the recommendation classifier, they still use the automatic labeled articles and conclude that manually labeled articles would increase an overall system cost. For the feature extraction mostly Weka intern tools are used, except for stemming, where the Porter stemmer is used, and for entitiy extraction, where the system ANNIE is used. To calculate the sentiment the news article is run through a dictionary and sum up numeric values which are in the dictionary for certain words (e.g. "disaster" = -4, "vital" = +1). They train four different classifiers: k-nearest neighbor, decision trees, Naïve Bayes classifier and a SVM with a linear kernel and a SVM with a RBF kernel. Their results for the two class (BUY/SELL) problem show that the SVM with Bigrams as features perform best. Also the dimensionality reduction is compared, where no dimensionality reduction is clearly performing best. A SVM with a RBF kernel performs, considering all metrics they give, similar to a SVM with a linear kernel, where the RBF kernel has a slightly increased accuracy. They address especially in the three class situation the problem that the data is highly unbalanced. This leads for example to a 82.5% accuracy for the SVM giving always HOLD recommendations. Overall in the three class problem the Naïve Bayes classifier performs best with an accuracy of 36.2% (33.3% is random), considering all F_1 values for BUY/HOLD/SELL recommendations. They introduce also a *DefaultLearner*, which gives always BUY

recommendations. In a market simulation the financial performace is evaluated. In the two class problem only the decision tree and the k-NN classifier outperform the DefaultLearner. Concluded is this behavior with the observation that stock prices tend to rise after a news release. In paired t tests they compare all performances to the one of the random learner, which gained no profit. The p level is set to 0.05. No significance of profit differences is shown. In the three class problem only the Naïve Bayes and the decision tree classifier are compared to a random classifier, where the desicion trees outperform slightly the Naïve Bayes classifier with a total profit of 15.2%.

Lauber, Gütl, and Liu (2012)

Lauber et al. (2012) try to predict the Stock market sentiment based on social media, namely stock forum entries from the III Stock market forum using a three category recommendation model (buy, sell, hold). Knowledge-based methods as well as a Naïve Bayes classifier, language models and a SVM with a RBF kernel are compared. Also a hybrid system, which combines the classifiers and tries to balance weaknesses of the classifiers was tested, but did not perform better than the classifiers on their own. The requirements on the system are very large, ranging from filtering of the posts, finding spelling mistakes and trying to make use of negation in the forum posts to word sense disambiguation. This all is mostly done using the framework LingPipe and SentiWordNet. The classifiers are trained on 6002 pre-tagged samples which are distributed among the three categories: 70% buy, 21% hold, 9% sell. In addition to the comparison of the classifiers, different feature sets are used, ranging from unigrams vs bigrams, term frequency bs. term presense to all POS vs. adjective + adverb. All systems are evaluated with a 10-fold cross validation, where the SVM outperforms all others. The lexical classifiers can barely pass the 33% random choice limit, the SVM reaches an accuracy of 76%, the language models are almost equally with the Naïve Bayes classifier around 50%.

4.2. Related Work Predicting the Foreign Exchange Market

To the authors best knowledge, most of the prediction-attempts are done utilizing the Stock market. The differences between the Stock and the Forex market, as depicted in section 2.3, are not very huge in terms of the basic workflow. Given the fact that the first attempts to make predictions on financial systems utilizing textual data was on the Stock market, it is very likely that research followed up on the topic of Stock market prediction rather on Forex market prediction. Nonetheless two systems will be

highlighted in this section.

Peramunetilleke and Wong (2002)

One of the attempts to make predictions on the Forex market based on textual news headlines is from Peramunetilleke and Wong (2002). Between the currency pair US Dollar and Deutschmark, they use a three category model: Dollar moves up, remains steady, moves down. They claim that news headlines only use a restricted vocabulary, so the headlines contain only relevant information. A set of weighted keywords is used, where each set consists of two to five keywords. In total over 400 of these keyword sets exist. A threshold of $\pm$ 0.0023% of the change of the exchange rate is used to separate the news headlines equally among the three categories. They test three different kinds of weighting for their keyword sets: boolean, TFxIDF and TFxCDF. A rule based approach is applied on their weighted set of keywords to make the final prediction. Weighting is done on the change of the exchange rate in a time-window of one, two and three hours. They let their rules test by domain experts at the Union Bank of Switzerland (UBS) in Zurich, where they compare their weights to manually assigned weights from experienced traders. Tested was the system finally for 60 trading hours in the period between September 22, 1993 and September 27, 1993. Their best results were with a TFxCDF for a weighting time-window of 3 hours with an accuracy of 53%.

Chaboud et al. (2009)

Chaboud et al. (2009) investigate the algorithmic trading within the Foreign Exchange market. A M1 data set between September 2003 and December 2007 from EBS is used with EURUSD, EURYEN and USDYEN, where they only use data between 3am and 11am New York time for further investigations, between Monday and Thursday and exclude holidays. They conclude that this is the most active trading time. The dataset itself actually is a tick-data set and is filtered to a lower resolution for their needs. The algorithmic trading is identified based on the idea that trading strategies are very similar across multiple independent market participants in time and direction. Therefore, in addition, algorithmic traders usually do not trade with each other, they trade against human traders. They also investigate the triangular arbitrage (see 2.3), where they use the tick-data dataset, which makes it easier to determine exchange rates out-of-alignment. Their finding supports the general intuition that algorithmic trading is steadily growing since 2003. It causes especially a lowering of the triangular arbitrage frequency from 0.5% between 2003 and 2004 to under 0.03% in 2007 with a profit of at least 1PIP.

4.3. Major Insights

All systems are compared with each other and are summarized in tables 4.1, 4.2 and 4.3. Furthermore the following major insights can be gained after analyzing the systems:

- The higher the resultion of the financial data the better, although tick-data can be considerably downscaled. For example Bacher and Stuckenschmidt (2012) downscale tick-data into 15-seconds intervals. M.-A. Mittermayer and G.F. Knolmayer (2006) use Minute-Data with considerably good results.
- Newer systems widely rely on Support Vector Machines, where the overall accuracy is increased.
- Hybrid systems could be, but are not necessarily better than single systems. Lauber et al. (2012) use a hybrid system where they try to combine classifiers to balance the weaknesses of each single classifier. Although ideas for improvement are given, the single SVM classifier yielded to better results than the combination of the classifiers to a more complex system.
- Feature reduction is not necessarily increasing the accuracy. While Hagenau et al. (2012) could show an improvement when a chi-squared feature reduction is applied, Bacher and Stuckenschmidt (2012) could show that their lerner performs best if no dimensionality reduction is applied.
- Mostly 3 labels (up/down/flat) are used, but the number of documents labeled to each category should be balanced. A 2 label approach could also be interesting. While most systems use a 3 label approach, Bacher and Stuckenschmidt (2012) have great success with a two label approach.
- Unevenly distributed labels can lead to misinterpretation of results. An example would be a two labeling approach, where 900 articles are labeled as "up", and 100 labeled as "down". If a system would always classify as "up" it would achieve a 90% accuracy.
- A Bag-of-words approach is most widely used, where the POS tags used as features can vary (nouns, adjectives, verbs, etc.)
- Some systems use news corpora which are purchased. Such corpora are usually pre-filtered and labeled in various ways, where the publisher makes sure that the labels are correct. Systems with commercially obtained news can easily filter out "uninteresting" news and achieve higher results, whereas systems with

manually obtained news corpora are in need of some kind of preprocessing to filter "uninteresting" news which would lead to noise otherwise. Thus, only "interesting" news from the finance domain should be used. Furthermore a distinction between high-impact and low-impact news is advantageous.

- Most systems rely on the Stock market, where the basic work flow is the same as the Forex market. The key difference is the amount of news and the amount of data to observe. While in the Stock market the system has to observe multiple different categories and assign multiple labels (multiple labels to different companies), in the Forex market a system has to observe macroeconomic news from two different economic regions, for example between the European Union and the United States of America.
- None of the systems take trading fees or transaction costs into account. Even worse, most of the systems claim their profitability is either zero or below zero if transaction costs would be considered.
- Current systems build their models non adaptive, meaning the models are built once and then used for classification of a whole test-set without make any further change on the model.

Author	Type	Labels	Classifier	Resultion	Accuracy Profit
Wüthrich et al. (1998)	Stock price	[3] up, steady, down	Naïve Bayes kNN Neural Net	D1	40% - 46.7%
Fawcett and Provost (1999)	Stock price	[2] noninteresting, interesting	lexical	D1	-
Lavrenko et al. (2000b)	Stock price	[5] Surge, Slight+ No Recomm. Slight-, Plunges	Naïve Bayes	M10	23 bps
Thomas and Sycara (2000)	Stock price	-	Maximum Entropy Genetic Alg.	-	30% return
Gidófalvi (2003) Gidófalvi and Elkan (2001)	Stock prices	[3] up, down, normal	Naïve Bayes	M10	0.10% bps
Peramunetilleke and Wong (2002)	Forex	[3] up, steady, down	rule based	-	53%

Table 4.1.: Overview Prototypes developed for making predictions on financial markets from 1998 - 2002.

Author	Type	Labels	Classifier	Resultion	Accuracy Profit
Thomas (2003)	Stock volatility	39 categories	rule-based	D1	90%/70%[1]
Schulz, Spiliopoulou, and Winkler (2003) Spiliopoulou, Schulz, and Winkler (2003)	News	[2] price_relevant, price_irrelevant	-	-	39% - 57% error
G. P. C. Fung, Yu, and Lu (2005)	Stock price	[3] rise, drop, steady	SVM	D1	61.6%
M.-A. Mittermayer and G.F. Knolmayer (2006)	Stock price	[3] good, bad, neutral	SVM	M1	82%
M.-C. Lin, Lee, Kao, and Chen (2008)	Stock price	[3] rise, drop, no movement	HRK	D1	better than SVM
Robertson (2008)	News	[2] interesting, not interesting	SVM, C4.5	D1	80%
Chaboud, Hjalmarsson, Vega, and Chiquoine (2009)	Investigate Forex	-	-	M1	-

Table 4.2.: Overview Prototypes developed for making predictions on financial markets from 2003 - 2009.

Author	Type	Labels	Classifier	Resultion	Accuracy Profit
Li et al. (2011)	Stock price	[2] up, down	multi kernel	Tick	80%
Schumaker, Zhang, Huang, and Chen (2012)	Stock price	[-] numerical values	SVR	M1	59%
Hagenau, Liebmann, Hedwig, and Neumann (2012)	Stock price	[2] up, down	SVM	D1	65.1%
Bacher and Stuckenschmidt (2012)	Stock price	[3] buy, hold, sell	SVM Naïve Bayes kNN	S15	profit 15.2%
Lauber, Gütl, and Liu (2012)	Stock price	[3] buy, hold, sell	SVM Naïve Bayes knowledge based	D1	best 76% (SVM)

Table 4.3.: Overview Prototypes developed for making predictions on financial markets from 2011 - 2012.

4.4. Summary

In this chapter the related work was introduced and existing systems were examined. It can be observed that early work tries to make use of the Naïve Bayes classifier, while newer work achieves better results using Support Vector Machines, either with a linear kernel, or mixed kernel setups. Furthermore it was examined, if feature reduction and hybrid systems are increasing the overall performance or not. It could be observed that a proper preprocessing and feature selection makes feature reduction obsolete. Based on the findings of Lauber et al. (2012), Hybrid systems are decreasing performance. The news corpora are mostly either from Reuters, or, in most cases, commercially obtained from news agencies like Newswire. Commercial news corpora are often labeled according to their category already, while manually (or freely and automatically) obtained news must be categorized first. Different kinds of filtering is one of the main preprocessing elements, especially when using social media contents. Social media texts, which can be seen as the top category of natural language processing in terms of difficulty, can be hard to process due to the "slang" language and the use of ambiguous writing style (e.g. sarcasm).
Overall, the best systems make use of a proper categorization, filtering and preprocessing. It could not be obtained which features are working best, although a bag-of-words approach with words of different sets of POS tags turn out to achieve results within the highest performing systems.
A common basic problem is the verifiability of the existing work, due to several reasons. Sometimes it is only commercially obtainable data, or unknown sources, or unknown labels, or rule-sets which are not clearly described. Limitations are also given by concentrating only on the Stock market. Macroeconomic news and the Forex market are not yet sufficiently analyzed with tools and procedures known from working prototypes for the Stock market. This will be addressed in the next chapter.

5. Ideas and Requirements

"If you can dream it, you can do it."

[Walt Disney]

The review of existing systems has shown that making predictions on financial markets is no easy task. Although an extensive amount of research was done, there is no information available, if a system exists which is really capable of making profitable predictions on financial markets based on sentiment analysis. There are basically two big areas of research which concentrate on financial markets: Systems which make predictions on Stock markets and systems which try to predict shifts in exchange rates. While most of the systems try to make predictions on the Stock market, the Foreign Exchange market seems to be of not such great interest for research. One possibility for this can eventually be found in the broader domain the Forex market operates. Trying to make predictions on a single Stock usually means to observe the news of a single company. Trying to make predictions on the Forex market means to make predictions on an economic situations which requires to observe all news available and pick such news which eventually can impact the economy of a country, or a whole economic area (like Europe), or both, or even the whole world.

What was shown in previous work is that *breaking* economic news topics have an impact, because in such news articles it is discussed what drives the market. Seen it the other way round: News topics which are impacting the market are usually *breaking* news. Such topics include possible political shifts due to elections, or announcements of possible economical problems including annual reports of banks (or governments) about the economic situation. While there are consent results of research that impact happens immediately, the results for the duration of the impact varies between 10 minutes and some hours, while the majority agrees for a duration of around 20 minutes. What was also observed is that rumors about upcoming bad influences in the economy are often enough to destabilize a currency, while good news often do not impact the market. The reason is that market participants try to bring their money in a "safe haven" before something can happen. Such market participants are hedge-funds and pension funds, which trade for billions of US Dollars in a relatively short time-frame. Having an huge impact from a big player can cascade to smaller participants who follow the movement, where it was seen that big banks trade against a falling exchange rate. That banks trade against this has two reasons, first they balance the economy, secondly

they can gain profit from the comparably good exchange rate.

Generally, to make predictions on the Forex market it is necessary to first identify such news with a possible impact, and then to correctly classify the sentiment in the news. The general idea to make predictions is also shown in figure 5.1 and will be described in more detail in the following sections, as well as the the concrete tasks will be modeled in this chapter. This starts from the underlying idea in the first section to a more granular in-depth description how such news can be applied to make predictions on the Forex market. Furthermore the functional, as well as the non-functional requirements are outlined. After the requirements are defined, it is necessary to investigate available libraries which can be reused. Then the systems conceptual architecture is described in more detail. At the end the research questions, which are tried to be answered, are defined.

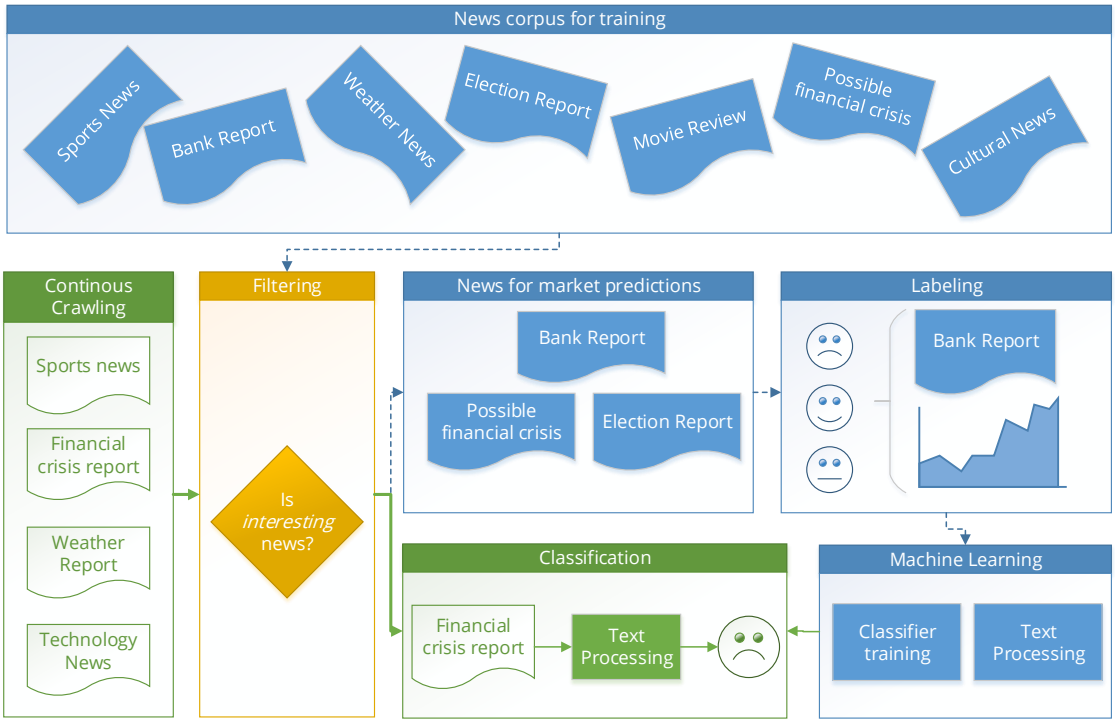


Figure 5.1.: Following the dotted, blue path, first a corpus must be available for the classifier-training. From this corpus, which is a mix of different topics, such news articles should remain after filtering, which are actually interesting in the financial domain. After having such news articles, the labeling process automatically assigns one of three labels based on exchange rates. Then the feature vector is created, which involves a text-processing task. This feature vector is used for classifier training. The created classifier is applied to classify new articles, which, in turn are gathered by a crawler (not dotted, green path), which continuously crawls one or more news sources. News sources (can) contain a lot of irrelevant articles, which are also filtered before the new articles are classified.

5.1. Main Idea

The idea, as well as the motivation behind the project, are based on the relation between news sentiment and economic prosperity. In other words, the project is driven by the fact that a currencies value is directly bound to the economic situation behind the currency and, in turn, changes in the economic situation are usually directly reported via news services. A similar relation could be seen in the Stock market, where a companies stock is directly bound to how profitable a company operates, which is also reported through news, blogs, social networks, annual reports and so on. Both markets, the Stock market, as well as the Forex market have in common that they react immediately to news, where the reaction on the market flattens at around 10 to 30 minutes after a release (DeGennaro & Shrieves, 1997). This is represented in a soft version of the efficient market hypothesis: All available information is therefore reflected in the market price after a certain time. By some researchers it was observed that the impact of news is asymmetric: good news has less impact in bad times than bad news in good times (Laakkonen & Lanne, 2010), other researchers found the opposite: good and bad news shows an equal impact in the Forex market (Bauwens et al., 2005). Generally it was observed that *surprising* news has less impact than news which is scheduled. That means, it is not imperative to use only articles which are reporting bad and/or surprising events, it could be even more beneficial to try go get news-articles about topics which are expected, like bank reports or articles which are reporting about the economic situation of a country which is undergoing a financial crisis (Bauwens et al., 2005). In other words, that there is an impact in the exchange rate, it is necessary that the *content* of the news is surprising, not the *topic* itself, but under the constraint that the topic is interesting to the financial domain.

To make such a system a lot of different things have to play together. For related systems it is necessary to have an appropriate amount of articles available, which have an impact in the financial domain, for learning and testing. By some related systems such a corpus was commercially obtained, whereas within this thesis such a corpus has to be created manually. This means a news corpus has to be created by downloading news articles. Having a corpus of articles with mixed topics, it is not easy to say which news article content is going to be beneficial to make predictions on financial markets. Essentially no system exists to the authors knowledge which can filter articles in such way, that from a corpus of mixed information only such articles remain which are *interesting* at the time of the release. News services provide a variety of different content which *could* have an impact, ranging from news about disasters, terrorist attacks, war, political issues, financial problems, bank reports, etc. Furthermore a lot of these articles are not *breaking* news, meaning news agencies are often just broadcasting summarizing articles without any new information, or articles which are reporting about a past event. A rule based classifier, which determines "interesting" news based on a keywords list

would be useless probably rather quickly: One imagine an article about the financial crisis which was in the 1980s in America. The article would be identified as an article from the financial domain and, furthermore, it would eventually predict that the Dollar is going to be worthless soon. Thus, a way must be found to download and filter topics automatically in such way, that really "interesting" news articles remain. Therefore the aim of this thesis is to classify the sentiment of *such* news articles of a whole corpus, which are driving the Foreign Exchange market for a short period of time. And such task involves: gathering news, extract relevant articles and classify them to make predictions on the market in a 20 to 30 minutes time period. Hence, the first big part is the gathering and extraction of news articles which are driving the Forex market. This is elaborated in more detail in sections 5.5.4 and 5.5.5.

A vast amount of research was done to extract the sentiment of textual data, ranging from unsupervised learning methods like lexical classifiers, sophisticated hybrid systems and classical supervised learning methods like Bayesian classifiers. From the investigation of related systems it turns out that Support Vector Machines are not only providing a top performance within all the other systems, they are also well researched, blazing fast (depending on the kernel-function) and highly recommended for classification of textual data (Fan et al., 2008). Using a supervised learning method requires the training of a classifier, which requires to have labeled data. Therefore it is necessary to not only gather an appropriate amount of articles, but also label the articles based on the movements of the exchange rates for a certain period of time after the release of the article. During the labeling process itself are some things important to recognize. Firstly from the exchange rates a meaningful label should be generated automatically, which will also be representing the sentiment of a news article towards the exchange rate. Secondly, the amount of minutes (the period) from the exchange rate movements used to calculate the label also reflects the prediction: If the training-set articles are labeled with exchange rate movements from the first 20 minutes after each articles release-time, means that any prediction based on that training set reflects 20 minutes of future exchange rate movements. Thus, the label of an article represents the sentiment towards the exchange rate which was used to determine the label. This training-set can then be used to create a classifier which is capable of a sentiment classification for new arriving articles, and in turn, the sentiment prediction is directly bound to the market movement, because the articles in the training set are labeled by using market movements.

The workflow from a high level view is also depicted in the figure 5.1. The blue, dotted path represents the flow which is necessary for training, the green, not dotted path which is necessary for continuous prediction. First articles have to be gathered, then filtered and then gradually labeled and text processed that it can be used to create a classifier. On the other hand, to make predictions, it is necessary to have new news. Also from this new news it is necessary to take only such articles which are relevant

to the financial domain. Then, having found such an article with a possible impact, this article is undergoing the same text processing steps until the article is finally classified by the before created classifier. The label represents the sentiment towards the future market movements. All this should happen within certain boundaries, which are outlined in more detailed in the following sections.

5.2. News Article Corpus

Two different challenges are faced regarding the corpus within this project: First, if a supervised learning method is used for Forex market predictions, then a suitable corpus must be available upfront for learning. Second, if predictions should be made continuously, then a way must be found to get suitable articles on the fly. Although it would be possible to get an appropriate amount of news articles for a supervised learning task upfront, it is still necessary to solve the second problem: How to get continuously articles which are from the financial domain *and* breaking news? The idea which answers this question will be modeled in this section.

Dynamic Similarity Filtering Approach

If articles of a specific domain have to be found, but without knowing which topics, or which content currently represents this domain, some challenges have to be solved. General news agencies (like Reuters, BBC, CNN, etc.) are publishing continuously news of a variety of topics. Within these articles there are *some* which are economically interesting, showing an impact on the financial markets.

An example would be the financial problems in Greece at the beginning of 2012. Some articles would report continuously about the financial problems in Greece and about the ongoing problem and about political decisions which influence the situation and therefore having an impact on the economic situation. A few weeks later, towards the beginning of summer, news agencies eventually report about tourism in Greece. All those articles are about the topic "Greece", but not every article is from the financial domain, and only some of them have content which have an impact. Usually news agencies are not directly publishing economically relevant articles in their "economy" section, they rather bring the "breaking" news first and then, some time later, they bring some kind of a summary article in the economy section. Working with a time-frame of only 10 to 30 minutes until the Forex market would reflect the "breaking" news, it is therefore necessary to directly classify the first published article correctly to the financial (or economic) domain and make predictions on the "breaking" news, rather than on the summary articles.

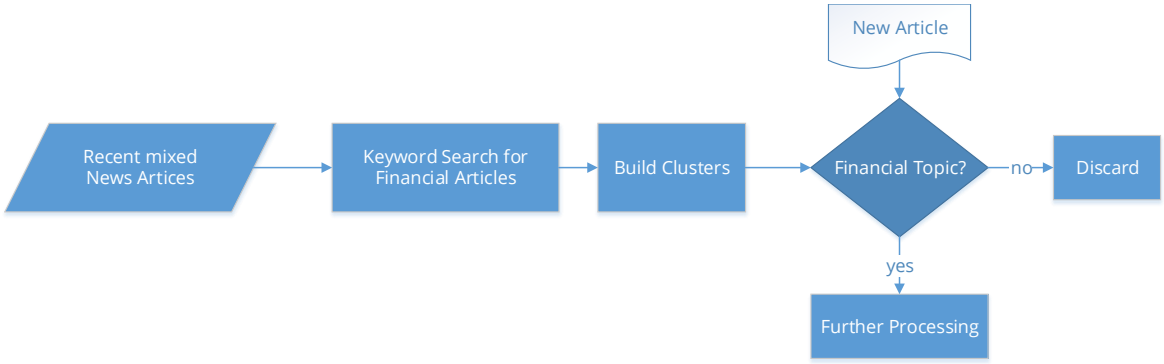


Figure 5.2.: Initial idea of the filtering workflow: From recent news articles of the previous n days financially interesting articles are extracted by a keyword search. Then they are clustered to get the topics, whereas the clusters are representing the topics themself. The more articles in one cluster, the more important the topic. Then new articles can be matched against the articles with a similarity measure.

One does not necessarily need surprising news topics in order to make predictions, which means a baseline can be created by using recent articles. Instead of trying to find articles with surprisingly appearing topics, it is more beneficial to concentrate on "hot" topics. These articles contain eventually the content which changes an economic situation. It seems that people are literally following recent topics and awaiting the outcome: For example a political election, which is usually well discussed prior the actual election day. The goal would be to identify articles which are interesting at the moment of the intended prediction and extract such articles. This could be done in a two-step process: First extract financially relevant articles from the previous days with a keywords list from a mixed news corpus. Such a result set would contain a lot of different articles with different topics, but all from the financial domain. From these articles the main clusters could be found, which would represent the topics. The more articles in one cluster, the more important the topic. After having the top-topics of the last days, one could now use a similarity measure with new arriving articles and see if they are falling into a topic which is currently of interest. This workflow is also depicted in figure 5.2. The challenge here is that there is literally no baseline, so the question is then, how similar an article must be that he is falling into the financial domain? A solution could be a double filtering approach, which is also shown in figure 5.3. Here two sets of clusters are built: One cluster-set with financially interesting articles, and one with articles from every other domain. Then a similarity can be measured against each of the cluster sets, and if the article is falling *more* into the financial cluster, then it is used for further processing.

With this filtering approach it should be possible to crawl *any* news from *one or more* publishing services. Only such news should remain after filtering, which actually has some economic relation and could have an economic impact. The problem is that

there is virtually no list of Forex relevant keywords or such, to get articles which are *currently* macro-economically important. Thus, one of the new ideas behind this thesis is that a list of *interesting* recent economic news can help to find also *new* articles which are interesting for the Foreign Exchange Market sector. News sources, both free and paid, publish on a daily basis usually especially Forex-related summaries and report about the most interesting topics of the day for the financial sector. The idea is to use this daily news summaries and articles which are easy to find with a Forex keywords list to determine the *interesting* topics for the Forex market sector.

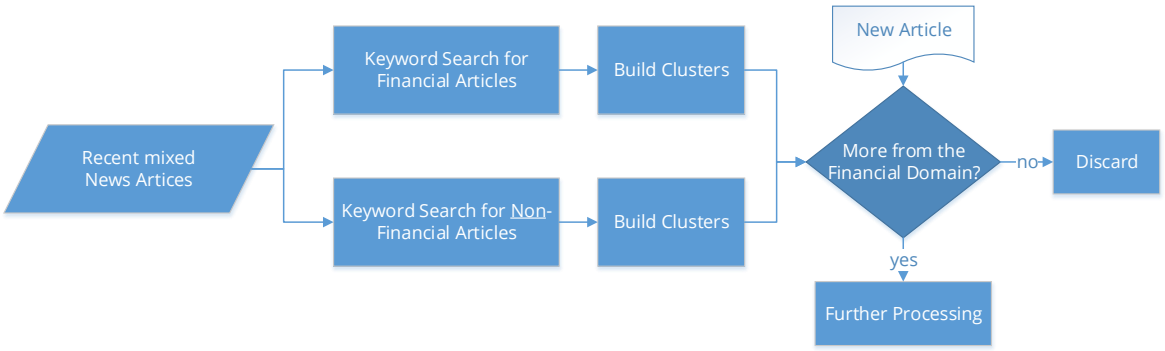


Figure 5.3.: Double similarity filter approach: From recent news articles of the previous n days financially interesting, as well as not-financially interesting articles are extracted by a keyword search. Then they are clustered to get the topics, whereas the clusters are representing the topics themself. Then new articles can be matched against the articles with a similarity measure. If a new article is more in the non-financial domain, he is discarded, if the article is more in the financial domain, he is used for further processing.

5.3. Machine Learning

Already in 1959 a gaming pioneer named Arthur Samuel said that machine learning is the "*field of study that gives computers the ability to learn without being explicitly programmed*" (Simon, 2013). Machine learning is a very broad definition for a category where different techniques like supervised, unsupervised, semi-supervised methods are falling into. In particular this thesis tries to classify news, where related systems with a similar aim applying a variety of machine learning techniques to gain results. These techniques range from Bayesian networks, support vector machines, lexical classifiers, clustering methods, to an almost uncountable amount of different other classifiers. All systems have in common that the best performing methods are supervised learning methods, but only if the data is relatively noise free and only if enough data is available for learning. It can be observed that in early years of research (prior 2003) Naïve Bayesian classifier were very likely used, in recent years of research the popularity of support vector machines (SVM) increased due to its remarkable performance. All of

the systems cannot gain results out of thin air, therefore some sort of data must be available in an appropriate format with suitable content, which was covered in the previous section. In order to use the articles from a news article corpus for supervised learning, a few steps are necessary. First, a training set has to be built in order to train and create a classifier, which means the articles have to be labeled. Then the textual data has to be converted to a feature vector in order to use it for classical supervised machine learning classifier. This section will focus on how this data is used and how machine learning can be applied in order to make predictions on financial markets.

Automated Labeling

When, after filtering, only such news are in the corpus which are from the financial domain, the news articles should be labeled. This process should be done automatically. The idea is, if the exchange rate raises with a certain threshold for x-minutes after the article release, the article should be labeled as "buy", if the exchange rate is steady, the article should be labeled as "hold" and if the exchange rate falls, the article should be labeled as "sell". Therefore exchange rates with a high resolution should be downloaded and labeled against each article. Based on the findings in the last chapter, especially the time between 10 to 30 minutes after the news release is interesting. Using exchange rate movements to label an article automatically makes the label a representation of the sentiment of the content against the exchange rate, which inherently represents the sentiment of the economic situation, because the movement of the exchange rate is directly bound to the economy behind. This means that the classification of a new article into the same three categories is then representing the sentiment of the article against the exchange rate movements and can be used to make predictions. The sentiment is inherently given by buy, sell, or hold recommendations. A similar approach is used by Lavrenko et al. (2000a), where a piecewise linear regression is applied to Stock price movements. This approach can also be used in this thesis for automated labeling.

Labeling using a piecewise linear regression, also called segmented regression, allows to do linear regression on separated subsets of data, instead of the whole set. This is also abstracted shown in picture 5.4, where a dataset of 20 datapoints is separated in two subsets. This allows to get the trend of the exchange rates around the time of a news release. Furthermore, this method can be used to determine the trend not only for a specific amount of time, but also with an easily definable offset. Although it was found that news impacts the Stock market instantaneously, a reasonable delay between the news release time and the impact in the Forex market can be tested, which can be modeled using this offset. More formal: the first parameter of the labeling approach is the offset between the first value for the linear regression and the release of the news-article. The second parameter is the duration used for the linear regression (e.g.

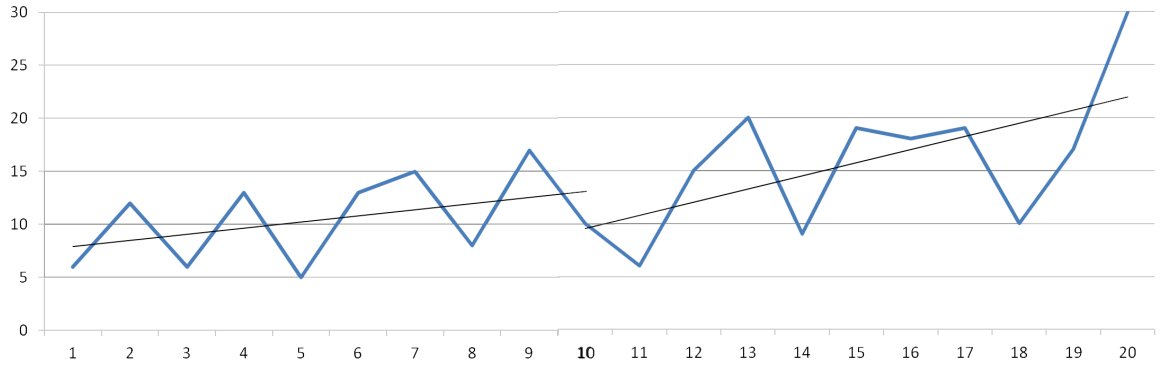


Figure 5.4.: A set of 20 datapoints is separated in two subsets. A piecewise linear regression is then the linear regression applied to the subsets. This method can be used on exchange rates to get a meaningful label.

20 minutes). As an example, given is a news article released at any given day at the following time: 13:30:00. With an offset of 2 minutes and a duration of 20 minutes, the result of the trend analysis of the exchange rates from 13:32:00 to 13:52:00 is used to determine the label for the news article. The label can be distinguished if the trend (the slope) of the linear regression is within a certain threshold.

Feature Vector Representation

Supervised learning requires data for training of a classifier, in order to classify new data. This data cannot be textual data directly, it has to be transformed into a numerical representation, which is referred as a feature vector. Numerous ways are available to transform the textual data into a feature vector representation, where the most famous and most successful by related systems is the bag-of-words approach with a TF-IDF representation. Here, all content from a document has to be split into each words, and the words have to be added to a dictionary, where the dictionary contains then the words (or terms) from all documents. Usually each of the words in the dictionary is given a unique numeric integer ID. Then, after having a complete dictionary with all words from all documents in the training set, the whole dictionary is matched against each document (article): for each word (or term) of the dictionary the amount of occurrences withing each article can be used for example. This refers as the term frequency, the sum how often a term appears. To avoid that frequent terms are overly weighted, one can calculate the inverse document frequency. Furthermore the term frequency - inverse document frequency (TF-IDF) is calculated by setting the term frequency in relation to the frequency the term appears in all documents, which is explained in section 3.3. Some steps are necessary to split each document into the single terms, which are also described in section 3.3 in more detail: Usually

first the document is split into single sentences, called sentence detection. From there the single sentences are split into each token, where a token can represent a word, or a punctuation, a number, etc., which is called tokenization. Then having each token, it could be beneficial to know what the token represents: a noun, a verb, an adjective, etc. Therefore Part-Of-Speech tagging (POS tagging) can be applied to the tokens, whereas the most prominent set of tokens is from the Penn Treebank Project, which assigns each of tokens a specific Part-Of-Speech tag. Furthermore Named Entity Recoginition (NER) can be applied, which extracts named entities, such as names from people, or locations, dates, or currency-values. Knowing which of the words in the dictionary represents which POS-tag makes it possible to use only such tags, which represent a certain feature: for example one could use only adjectives for the training set, or only nouns, or both, or a variation of verbs, nouns and names entities.

Related systems achieved good results using a bag-of-words approach with words of different types of POS tags, although no conclusion could be found which kind of POS tags should be used or not. The basic idea is very similar to related systems: a bag-of-words approach with different types of POS tags, although it should be tested which set of POS tags is mostly suitable for a supervised learning task on the Forex market. Therefore it is necessary to first apply all typical natural-language text-processing steps: sentence detection, tokenization, POS tagging. Eventually it could become handy to have also the Named Entities of the articles and apply them as features for further machine learning. In a second machine learning step it should be evaluated which set of these tags is best suitable for learning: only nouns and adjectives, or verbs, etc.

Learning and Classification

As already mentioned, support vector machines (SVMs) gained an increasing amount of interest for research in the past years, because of the high performance they deliver. Although the underlaying mathematical operations are quite intense, the high popularity is probably due to the libraries which were developed and their easy usage with remarkable results. Basically, support vector machines request data points in space and then try to separate these points with the highest possible margin, which is explained in more detail in section 3.4. As always, the less noisy the input data is, the better the overall performance. This requires some more processing on the corpus during, or before creating the feature vector. Best would be to exclude such words in first place, which deliver zero information. Examples of such words are "the", "and", "a", etc. Such words are called stop words and lists of stop words can be found across the internet with different amounts of stop words. A very famous list of stop words is compiled into MySQL and can be found in the appendix in A.5. After removing stop words to reduce the amount of noise, a support vector machine could be utilized to build a classifier for new news. A support vector machine uses kernels, in its most easy version,

a linear kernel. With a linear kernel, the input data points are tried to be separated linearly, which works great for large sparse feature vectors (Fan et al., 2008). Large sparse feature vectors are such vectors which have a huge amount of data points, but primarily the points are zero, which commonly happens when textual data is converted to a feature vector.

5.4. Requirements

Summarizing and concluding all outlined ideas from the beginning of this chapter, the following steps are necessary: First some functionality that downloads news articles. After articles are gathered, some sort of filter, which removes articles on the fly which are not from the financial domain. Then an automated labeling process, which assigns a label to the downloaded articles based on the exchange rate movements. After the labeling some sort of functionality that converts textual data into a feature vector representation which is then fed into a support vector machine for training of a classifier. On the other hand, it is also necessary that new articles are downloaded continuously and filtered, also converted into a feature vector and then classified using the classifier which was trained before. This is what is also shown in figure 5.1.

Modular Daemon Architecture

Instead of doing all tasks in a row, each of the task should be abstracted into separate modules which are running as daemons. For example, to download news articles a crawling module should be created, which is doing only this task. The crawling module then saves the crawled data into a database. From there, another filtering module should query the database continuously and see if new articles are available. "New articles" in this case represent articles which were not yet filtered. This module then fetches such a new, yet unfiltered article from the database, run the filter and set a specific flag if the article is from the financial domain or not. Then another module is created which is responsible for the labeling process. Such a module has to, yet again, continuously query a database if a new (and already filtered) article is available. The labeling module extracts the articles publishing date and time, downloads exchange rates from that time and label the article depending on what exchange rate movement happened. Yet another module, again, queries the database continuously for new labeled articles, fetches them, does all appropriate natural language processing tasks and save the tokens, the POS Tags and the named entities back to the database. Furthermore, the dictionary itself should already be saved in the database, so that another module for the classifier training only has to query the database and has the feature vector already in an appropriate numerical representation. These described

steps would be necessary to build a training set (and train a classifier). For continuous prediction a very similar approach is implemented. Such a prediction module extracts the articles publishing date and time, and if the article is not older than some minutes, the article is, instead of labeled and used for learning, used for classification and prediction. After a while the article would be found too old to be used for a prediction, and therefore labeled and used for learning. This is also represented in figure 5.5.

From these considerations, that several independent modules are required and interconnected only through a common database, the functional and non-functional requirements can be defined, which is done next. After the requirements are defined, the architectural design is outlined. This will give a more detailed requirement to look for relevant libraries which can be reused.

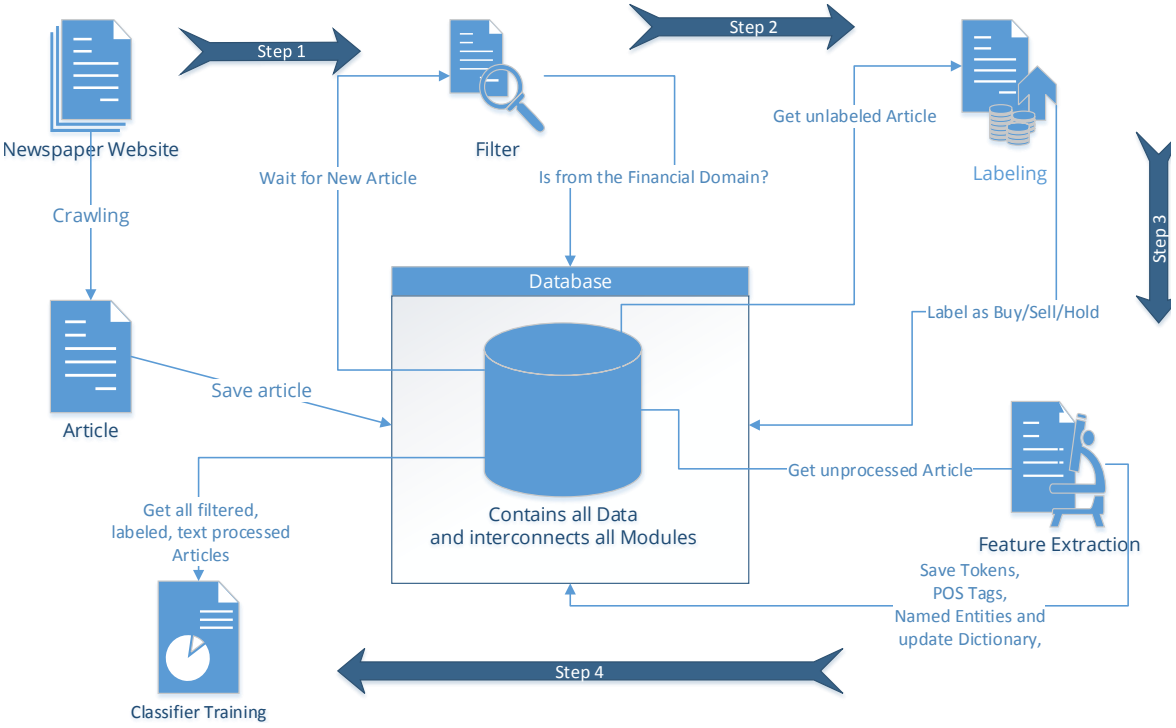


Figure 5.5.: Modules are created which have all a unique, atomic operation. They are interconnected with a database which also holds all information. For example a filtering module runs as a daemon, waiting for new articles and then processing an article as soon as one arrives in the database. This is described in section 5.4 in more detail.

Functional Requirements

The following requirements are defined for each of the functions of the system. First the general requirements are defined, then each of the modules, which represents a function

of the system, is outlined. Each module is interconnected through the database, which also holds all the data.

Persistent Data Store and Interconnector

Purpose: Persistent data store. Handles data flow between modules.
Description: To save and retrieve the data a data layer has to be in place, which can be a conventional relational database or a file-store. Furthermore an interconnector, which makes data interchange possible between the modules should be available.
End result: The modules can interchange data and persistant store and retrieve data from a data store.

Controller

Purpose: Starts and stops instances of the modules, starts simulations and gives basic user interface possibilities.
Description: To interact with the prototype a basic user interface has to be in place. The purpose is to have a central controller which is responsible for starting and stopping the modules and any interaction with the software.

Crawling

Purpose: The purpose of the crawler is to get new articles.
Preconditions: The crawler has no preconditions.
Description: The crawler continuously crawls news sources for new articles. If a new article is found the crawler saves it to the database. Furthermore only the article body-text, the release timestamp, as well as the article title are of interest. The crawler should therefore strip all content from websites which are not part of the main article.
End result: A new article is saved in the database as text, without any further markup.

Filtering

Purpose: Filter articles for further processing. Reduce noise to the system.
Preconditions: A new article in the database.
Description: If a new article arrives it should be filtered if the article is relevant to the economic domain or not.
End result: The article is categorized as economically relevant or not.

Labeling

Purpose: Label articles according to financial data.
Preconditions: The article is relevant to the economic domain. The articles' release-timestamp is more than x-minutes in the past. The article is not in the training set.
Description: An article should be labeled, if there is sufficient financial data available to process automatic labeling for an article. This can only be done if the release date of an article is so far in the past, that sufficient exchange rates can be downloaded to distinguish the articles possible impact on the Forex market. This will likely be only articles which are meant for training of a classifier.
End result: An article is labeled as "buy", "sell" or "hold".

Feature Extraction

Purpose: Extract the natural language features from the article text and saves it.
Preconditions: The article is relevant to the economic domain. The features of the article are not yet extracted.
Description: To process an article and feed it into a machine learning algorithm, the features have to be extracted. This contains sentence detection, tokenization, POS tagging and Named Entity Recognition.
End result: The features of a given text are known.

Supervised Learning

Purpose: Create a classifier.
Preconditions: Labeled, feature extracted articles.
Description: This function will create a new classifier for classification of new arriving articles. The classifier can be saved for later use.
End result: A trained and saved classifier.

Classification

Purpose: Classify a new article.
Preconditions: Features from the article are extracted. A saved classifier.
Description: To classify a new article it is necessary to extract the features. Then the pre-created classifier can be used to classify the new article accordingly to one of the categories "buy", "hold" or "sell".
End result: The new article is classified to one of the categories.

Non-Functional Requirements

To create such a system with the academic environment in mind, some non functional requirements have to be defined as well. These should mostly make sure that the software is running cost free and uses only openly available data. Furthermore it should make sure that the software is easy to handle and efficient in the execution. In more detail:

- *Public Data*
  Publicly available news should be utilized. The corpus should not rely on commercially obtained data. For example the Reuters news archive can be used as a news corpus to gain reproducible insights.

- *Open Technology*
  The system should make use of openly available libraries and should not depend on paid third party libraries.

- *Simplicity*
  With the term "big-data" one inevitably connects terms and tools like Hadoop, Hive, Map Reduce with it. Although the system should be expendable to utilize enterprise-scale systems, this prototype should not be overly complicated. Therefore the prototype is limited by an upper boundary of two Terra-byte of in-memory data.

- *Modularity*
  Each step should be done separately. This means, each step of the system should run separately and each step can be done utilizing more than one server or a cloud service like Amazon's EC2 instances. It can increase the overall processing speed, if needed.

- *Changeability*
  With the simplicity and the modularity it should be possible to change certain modules easily. This includes modules for unsupervised learning (clustering) as well as modules for supervised learning and certain things within these modules like feature-sets.

- *Expandability*
  It should be possible to extend the system in many ways without breaking existing functionality.

5.5. Conceptual Architecture

Having the academic environment in mind, it is necessary to evaluate the results in a meaningful way. Therefore, it is necessary to have a robust and reproducible way of validating the results. In order to gain reproducible results, it is necessary to build a system which uses data that can be obtained publicly for free. Such a system could perform a simulation on the data as if it the data was just released and simulate as if the events were just happening. Therefore it is proposed to use data from a news archive from a time where something interesting was happening in the financial domain and split the system into classical three big phases: *training*, *classification* and *evaluation*.

In order to do the training, first data has to be acquired, which is therefore the first block in figure 5.6. Two different kinds of data have to be obtained: News articles from free news sources and exchange rates with a high resolution. After the data is acquired, the news articles have to be filtered. If the news is not from the financial domain or is not an interesting article, it has to be discarded. From the remaining news, the features have to extracted. Then the news has to be labeled based on the exchange rates after the news release. With labeled training data, a classifier can be trained. In order to train a linear SVM, a parameter search has to be performed. Therefore it is necessary to do a grid-search and validate the parameters based on a 10-fold cross validation (Fan et al., 2008). If the optimal parameters are found, the trained classifier can be saved for later use. Meanwhile, the crawler continuously grabs new news and saves the articles to the database. Then the news is filtered: if it is a current interesting financial topic it is used for further processing. The features are extracted and the previous saved classifier is used to classify the news in one of the three categories "buy", "hold", "sell". in an evaluation phase the predicted category is evaluated with exchange rates from the time after the news release.

Trainig Phase

The training set is based on macroeconomic relevant news and the according financial data of the time of the news release. This data was previously crawled and saved to the database during a data acquisition phase, which is basically the same as the ongoing crawling and described in more detail in 5.5.4. The news is processed and filtered and only news of a certain time of the day is used for further text-processing, which is described in more detail in section 5.5.5. During feature-extraction the news is undergoing comprehensive text processing. That means the news is tokenized and the according POS Tags are assigned to each token, which can be found in more detail in section 5.5.6. Then news is labeled based on the financial data at the time of the news release. For the prediction a bag-of-word approach is used with three labels. A threshold is taken where a balanced distribution of labels is reached. The label for the

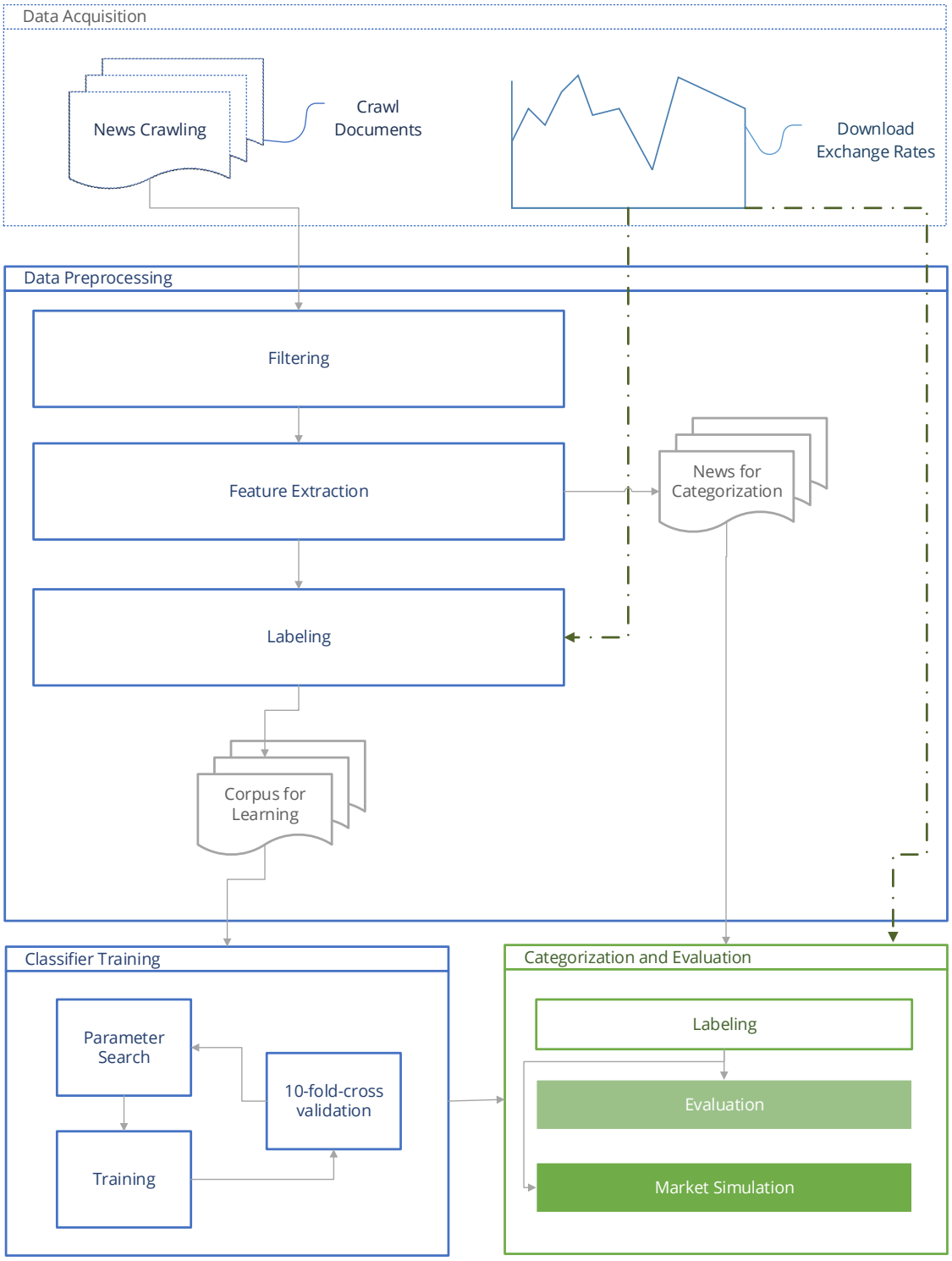


Figure 5.6.: Data acquisition has to be done in order to train a classifier. This data has to be filtered and the features have to be extracted. Supervised learning requires the labeling of the training data. After the optimal parameters for the classifier training are found (Fan, Chang, Hsieh, Wang, & Lin, 2008), classifier is trained and the trained classifier can be used to label new documents. Finally the categorization of new documents is evaluated.

news is based on a piecewise linear regression on the financial data at the time of the release with a certain (variable) duration, this can be found in more detail in section 5.5.8.

Classification Phase

The classifier which was trained during the training phase is saved and can be used to label new arriving articles. If a new, yet uncategorized article, is available, it first has to be preprocessed. First and foremost, within the filtering-phase, the question has to be answered if the article is in the financial domain (see 5.5.5 for more details). If yes, the features have to be extracted (see 5.5.6). Then the previously trained classifier can be used to categorize the article in one of the categories.

Evaluation Phase

The validation phase consists of two different tasks. The first task is to see if the category was chosen right by the classifier for the article. Therefore financial data *after* the release of the news must be downloaded and, using the same parameters for the trend analysis (see section 5.3) as in the training phase, the class of the article has to be validated against the exchange rate. In the second task a market simulation is done. So the amount of profit has to be determined and the trading costs have to taken into account.

5.5.1. General Design Decisions

The general implementation should be based on Java to support code execution across all platforms. Additionally every single module incorporates only publicly available libraries, where the used libraries are outlined in the description of each module. Furthermore it should be possible to run the modules with multiple instances across different servers. This should be possible due to the interconnection with the database. A locking mechanism could avoid race conditions, which happens when two modules of the same type are working on the same problem. An example would be a race condition of the labeling module, where two instances are waiting for new articles and start working at the same time. Furthermore, as already mentioned, all modules are utilizing the same database.

Also it is questionable which data format should be used for saving the data: should the data be saved directly in a relational database, or should a proprietary format be used, like the popular library Lucene? The Lucene library behind Apache Lucene was

originally written by Doug Cutting in 1999 (Goetz, 2000) and has become the state-of-the-art free and open information retrieval software library. The idea of Lucene's architecture is that a document contains fields of text, for example a title and a content. The Lucene index is an inverted index, while each field can be either stored, tokenized, indexed or a mix of all three[1]. Having the documents indexed in Lucene allows a fast operation on several tasks, for example ranked searching, many powerful query types like phrase queries, wildcard queries, proximity queries, range queries, fielded searching (e.g. title, author, contents), sorting by any field, and many more. One of the biggest drawbacks is that a Lucene index is not very *handy* when it comes to understanding big sets of data, because an in-depth look into the stored documents with specific queries if often quite complicated, requires special tools, where a separate, well documented relational database like MySQL is better. Unfortunately often third party tools and libraries do not support a direct connection to the database, instead often such libraries, especially for text processing, were specifically developed for Apache Lucene and require a Lucene index. Thus, textual data is stored in the database within this thesis, and the data in the database will be converted to a Lucene index wherever necessary.

First it will be described how the user can interact with the software, then each of the modules functionality will be outlined, including the workflow, conceptual architecture and used libraries.

5.5.2. User Interface

A user interface should make it possible to interact with the modules in a very simplistic way: start and stop different instances of modules or run a simulation of the classification and gain the end results. It is not the aim of the thesis to provide a nice looking graphical interface, functionality is priority. Therefore any input or output will be limited to textual input and output through a command line interface, which can be extended in a future work. Furthermore all simulation data will be saved in the database for further use, which can be exported directly from the database with appropriate tools for further processing in other programs like MatLab, etc. Therefore a single starting point will be created, which acts as a "Basic Controller". This is the main class which is started first and this provides also the basic user interface. In the future, it can be extended to a more nice looking graphical version.

5.5.3. Database

As the database interconnects all modules, it will be outlined before each of the modules themselves are described. With ease of use in mind and with a strong community

[1] http://lucene.apache.org/core/3_0_3/fileformats.html#Index%20File%20Formats

behind, the MySQL relational database management system (RDMS) is the favorite choice. Although a typical RDMS would probably be avoided in a large scale real time system, it is perfectly suitable for the project and the best choice within this thesis. Free, open source and well maintained, with plenty of available tools which reduces the necessity of creating functions to access the data. Although Java offers a wide range of database abstraction libraries and toolkits, only raw SQL Queries will be used to save to or retrieve data from the database. Two tools from the MySQL Community will be highlighted, which should be obtained in order to make the work on the project more easy: MySQL Workbench[2], a database design and modeling tool, and HeidiSQL[3] a graphical tool to access and edit and view table and more importantly: import and export data in various formats from the database.

5.5.4. Crawling Module

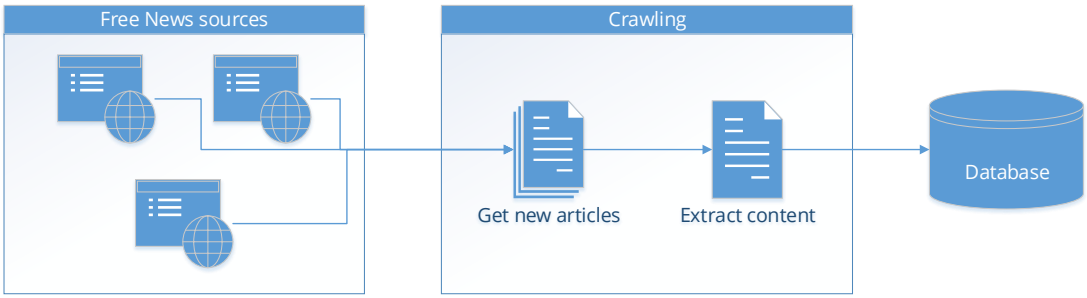


Figure 5.7.: Crawling from one or more news sources, extract the articles content and save it to a database.

The crawlers main purpose is to get information continuously, remove any markup and content which is not part of the main article and save it to a database, which is also depicted in figure 5.7. Each of the crawler instances has to be adapted specially to one news source - which can be websites, RSS-Feeds, Usenet-Groups, Blogs, Facebook. One could call the crawler also a data provider.

Conceptual Architecture

A crawler instance is checking in certain time intervals if new content from a news source is available. If there is new news available, the crawler extracts the content and saves it to the database. This is also depicted in figure 5.7.

[2] http://dev.mysql.com/downloads/tools/workbench/
[3] http://www.heidisql.com/

Libraries

As stated in the section 5.5, the system will be run in a simulation suitable for academic research. Therefore a very specific data set is the target of the crawling module, which is a news archive of a large news agency. This means a crawler has to be adapted to scrape the content of the archive. In a real world environment a crawler like Apache Nutch could be adapted to do the crawling task and extract continuously news. Such a crawler like Nutch is well maintained with a broad community behind it. Although Apache Nutch or a similar crawler is probably adaptable to the very specific task of scraping a news archive, a self developed web-scraping and crawling algorithm to extract the archive data is probably done in a fraction of the time needed to finish such an adaption. Thus, although not advised for real world large scale usage, a news crawling and scraping algorithm will be developed which is capable of extracting such a news archive and with this a news arrival simulation is executed. It is mentioned at this point, that also a MySQL database would probably not be used in a realistic market prediction environment, instead an Apache Lucene index would be suitable for future development as it is better scalable, faster and needs less system resources.

A crawler will be self-developed which is extracting the news articles of a large newspaper. News on a website is not delivered as pure textual data, instead and in general, websites are normally written in Hypertext Markup Language (HTML) version 4 or 5. If the underlying HTML-code has a proper syntax, it is easy to extract information. Unfortunately big press agencies and websites *seem* to try to avoid syntactically correct markup to hamper the automatic crawling of the content. Additionally obstacles like JavaScript enabled content, where the content is loaded asynchronously, are sometimes in place. Most of the HTML-Parser will not work with a broken syntax, whereas the library Jericho[4] especially allows "*analysis and manipulation of parts of an HTML document, including server-side tags, while reproducing verbatim any unrecognised or invalid HTML*". For this reason the free and open source library Jericho is used to extract content from websites where the HTML syntax is possibly broken.

5.5.5. Filtering Module

The purpose of the filter is to reduce noise in the system. The general conceptual architecture is shown in figure 5.8. For the actual classification only "interesting" news is relevant, where the word "interesting" is referred to a news article, which has an impact in the financial domain. If a topic is interesting in a macroeconomic scope, it usually does not change in short periods - like a crisis in a specific county, or political discussions having an impact on the market. In other words, topics about the news

[4] http://jericho.htmlparser.net/docs/index.html

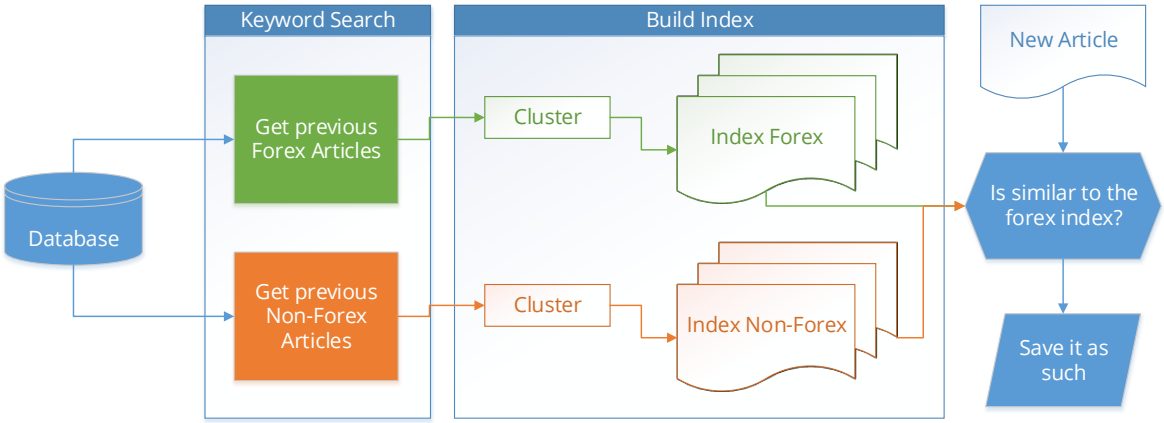


Figure 5.8.: From the database old articles are searched which fit the Forex and the non-Forex domain. Then the main-topics are extracted and a new article is matched against the indices. Then the article is marked (e.g. flag) for further use in the database.

normally doesn't change much in short periods, but the problem is that the underlying news doesn't necessarily contain typical keywords about the Forex market. An example would be the economic crisis in Greece during spring 2012, where it was much discussed about Greece and the government, less about the economy. Another example would be the political elections in France. So a baseline must be found to get these *interesting* topics in news corpora, where these news is not labeled as such. All freely available news sources usually post summary articles about the financial markets during the day which can be utilized to find topics which are *currently* of interest. As already stated in the initial idea in section 5.2, a dynamic similarity filtering approach is proposed, where economic summary articles are utilized from the previous days in order to gain information about which topics are currently from the financial domain. But filtering should do more than only select the interesting articles. For example, if an article is released outside of the main trading hours, even with the most important information, it will probably not show any impact, because there are simply no traders active, who would react to the news. Also, it should be narrowed to the trading hours of only one economic region, like Europe. Furthermore only articles in one language will be supported, although this can be expanded in the future. Thus, the filtering module is responsible for discarding articles which are not from the financial domain, which are outside of European main trading hours and which are not in English.

Conceptual Architecture

The first step, as depicted in the box "Keyword Search" in figure 5.8, is to perform a keyword search to find articles related to the Forex domain, as well as the opposite, the non-Forex domain, in an article-set from the last day (or days). The Forex-keyword

search should bring up articles with topics about the current economic situation. These topics can include articles about political issues, economic problems, released statistics, etc. From these articles the main topics have to be found, which can be achieved using a clustering method. After knowing the main topics for the financial domain, also the non-financial domain topics have to be found. A new article can now be matched using a similarity measure between the article and the financial articles cluster, as well as between the article and the non-financial articles-cluster. Both similarity measures can be taken to determine if the article falls more likely in the financial domain or not. The clustering method can be used to determine on the fly articles which articles currently contain financially interesting content.

Libraries

The idea to use this method comes from the evaluation of the clustering framework and library Carrot[25]. This clustering engine can "*automatically organize small collections of documents (search results but not only) into thematic categories*". Thus, it is perfect to group collections of news-articles into the same thematic group and match new documents against a group of already clustered and separated documents. The clustering has two benefits: First it extracts the topics and matches similar articles to the clusters, and secondly, it ranks articles within the clusters according to the meaningfulness within that cluster. The more similar a topic within the cluster, the higher it gets ranked. The only drawback is that the Carrot^2 library can only import a Lucene index, which makes it necessary to convert the data from the database to a separate Lucene index and process the clustering tasks there.

Having articles in two different clusters which are representing two different thematic groups opens the possibility to match a new article against these groups and measure the similarity. This can, for example, be done using a cosine similarity measure between the feature vectors. It is relatively easy to calculate a cosine similarity when the articles are already in a Lucene index. Pal (2011b) gives good examples of how to use the Lucene capabilities to their full extend, which will be incorporated. In general, it could be interesting for future work to use solely the Apache Ecosystem, including the crawler Nutch, the document save and retrieval library Lucene, the machine learning framework Mahout and all other tools which are available through the Apache Foundation.

5.5.6. Feature Extraction Module

During feature extraction all text-processing steps have to be applied: Sentence detection, tokenization, POS tagging and Named Entity Recognition. A vast variety of

[5] http://project.carrot2.org/download.html

frameworks and libraries with almost equal results are available to fulfill these tasks. Furthermore, to the processing itself, it should later be possible to easily switch between different textual features to create the feature vector for further learning. Thus, the results of each sub-step of the text processing are saved therefore to the database for eventual later use. This means, the most important information is available through a simple query against the dictionary in database, for example in how many articles a word is, in which sentence(s), what kind of POS-tag is applied to that word, etc. One can switch between different features like a subset of POS tags, include named entities, mix those in a variety of ways, or also extend the feature extraction and text processing module to use a completely different set of features

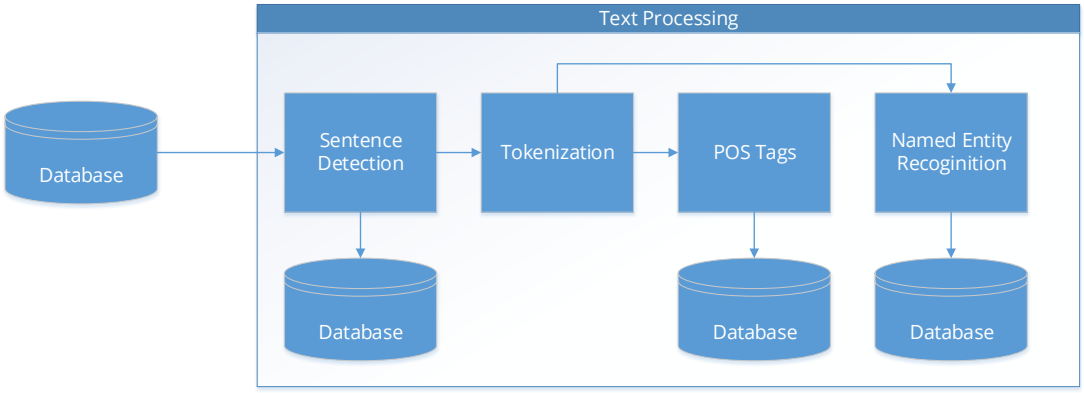


Figure 5.9.: Text Processing during feature extraction: Sentence detection, tokenization, POS tagging and Named Entity Recognition. All results of the steps are saved into the database for further use.

Conceptual Architecture

The feature extraction module is running in the background and continuously looking in the database for articles which were not yet processed. If a new article is available from the financial domain, it should be processed by the text-processing module, as depicted in figure 5.9. First the sentences are detected and the sentence order is saved to the database. Then each sentence is tokenized and a POS tagging is done, which is also saved back to the database. Additionally the Named Entities are recognized and saved to the database as well.

Libraries

As already mentioned, to do this kind of text processing, a vast amount of libraries is available. Recently Apache joined with a new open source NLP Library called OpenNLP. Reading from the apache website, the library "*[...] supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution*". The library is using a maximum entropy method to do any of the NLP tasks. This means that comprehensive learning has to be done first in order to gain meaningful results. Luckily the team who created the library also offers pre-trained models for various languages, trained on a variety of corpora, for free download. The English language models are pre-trained based on the Reuters Corpus [6], which makes it very suitable for using it on news articles. The text-processing task can easily be extended for use with different languages like German, French or Spanish, by switching just the underlaying model.

5.5.7. Labeling Module

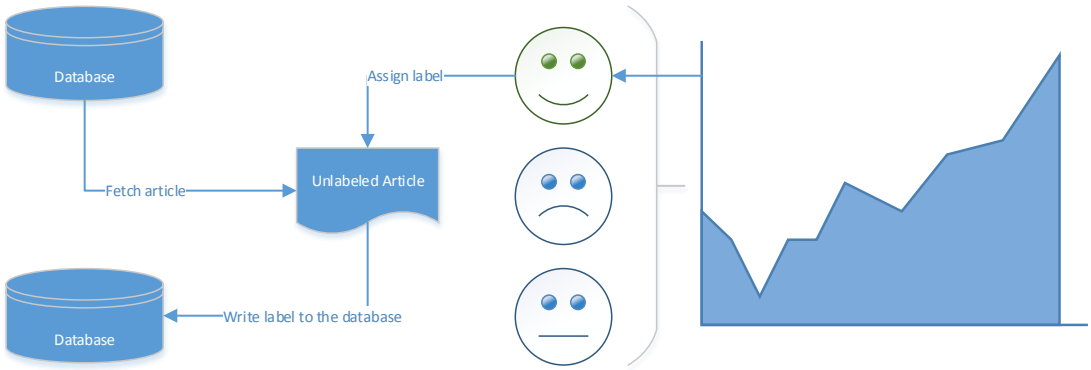


Figure 5.10.: From financial data a meaningful label should be extracted and assigned to the documents in the database. The label is saved back to the database to the according document.

As already depicted, the labeling module should automatically assign a meaningful label to yet unlabeled articles in the database. The module should run in the background as daemon, continuously query the database and look for yet unlabeled articles. If such an unlabeled article is found, from exchange rate movements after the articles release date a meaningful label should be extracted. As already described in section 5.3, a piecewise linear regression could be utilized to gain insights about the movement of the exchange rates. Two variables are introduced: an offset and a duration. The offset is the time between the release time of the article and the first value used for the

[6] http://trec.nist.gov/data/reuters/reuters.html

piecewise linear regression. The duration is the amount of time used for the calculation. This is also depicted in figure 5.11, where the bold red line is the time where an article is released. The light red area is the offset and the movements of the exchange rate in the green area are finally used to calculate a meaningful label. Because the training data is labeled utilizing such an approach, the according classification of new articles represents an exchange rate prediction with the same offset and the same duration. Would a new article be classified as "buy", and the training set would have been labeled with an offset of five minutes and a duration of 20 minutes, it would mean that the prediction is valid in five minutes for 20 minutes.

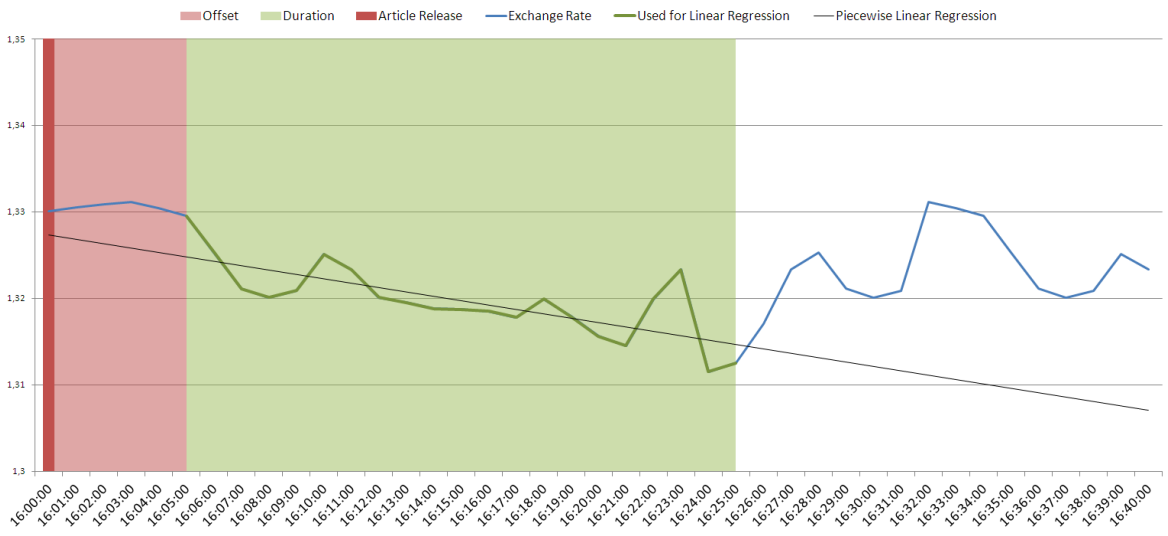


Figure 5.11.: After an article is released, an offset is introduced, which marks the starting point for the linear regression. The linear regression is then done with the data of the exchange rates during the time which is given as duration. In this example, if an article is released at exactly 16:00:00, the offset is 5 minutes and the duration is 20 minutes, then the slope of the linear regression of the exchange rate between 16:05:00 and 16:25:00 is used to label the article.

Conceptual Architecture

The labeling module fetches, if existent, an unlabeled article with a certain age from the database. The article's release date and time is extracted, and exchange rate movements are downloaded from that period. From the article's release timestamp with an offset of n minutes and a duration of d minutes a label is calculated using a piecewise linear regression method. The label is saved back to the database.

Libraries

To calculate a linear regression only basic mathematical operations are needed. Although this could be self developed, the third party library `Math` from the `Apache.Commons` package is used. This has the advantage of a clear defined interface, and can easily be swapped against another method to calculate the label from exchange rates. As the article data is in a MySQL database, a simple queries should be sufficient to find unlabeled articles with a certain age.

5.5.8. Training and Classification Module

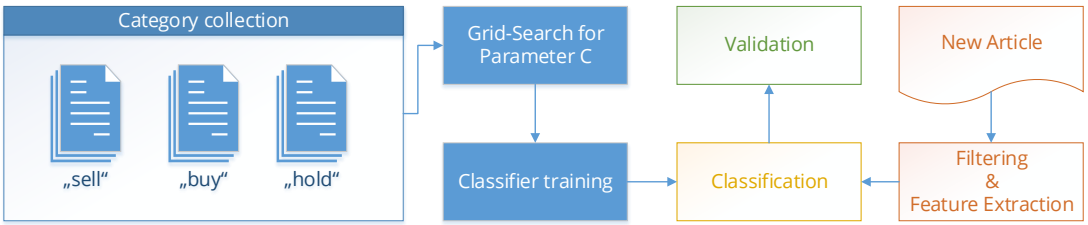


Figure 5.12.: A collection of features of documents are held in three different categories: "sell", "buy" and "hold". On these features a grid search is done to find an optimal penalty parameter C and then the actual classifier is trained. This classifier is used to classify new documents. At the end the classification is validated.

The classification is separated in two stages: first a classifier has to be built, second the classifier has to be applied to new data and this has to be validated. The approach is straight forward: A feature vector with features from the previous section 5.5.6 has to be fed into a machine learning algorithm. For large sparse feature vectors, which are typical when textual data is used, a SVM with a linear kernel is used best (Fan et al., 2008). Beside the weights, which are obtained automatically through learning, also a penalty parameter C has to defined in advance. In order to train a SVM with a linear classifier a parameter search has to be done, typically using a grid search: Iteratively increase the parameter C and do after every increase a 10-fold cross validation on the training set, which is described in more detail in the next section below. Then, after the classifier is trained, it can be used to classify new data. The result has to be validated. This is done by downloading new financial data and label the article accordingly again as described in the previous section 5.3.

Conceptual Architecture

The workflow for classification, also shown in figure 5.12, builds a classifier based on all available, preprocessed information in the database. These articles in the database, which were previously pre-processed in the feature-extraction module and labeled, are used to build up a training set. The training set is then a collection of documents in three categories: "sell", "buy" and "hold. On this training set a grid-search is done to find the optimal penalty parameter C, which is then used to build the actual classifier. If a new document is classified, the previously created classifier can be used to distuingish the correct label for the article. In order to validate the classification, the article has to be labeled with exchange rates which have to be downloaded. The results of validation is usually given as percentage in terms of accuracy. But it is also possible to calculate the actual profit (or loss).

Libraries

Classification is done solely with the library LIBLINEAR[7], which is a library for support vector machine classification with a linear kernel. It is extremely fast, doesn't need a lot of resources and is very similar to the well known package LIBSVM. The library, which is especially very suitable for classification using large sparse feature vectors, as it usually happens with textual data, is the winner of ICML 2008 large-scale learning challenge and was also used to win the KDD Cup 2010. It is originally written in C++ and ported to Java by Benedikt Waldvogel[8].

5.5.9. Summarized Architectural Design

The design decisions are quite comprehensive and will be summarized shortly in this section. First of all, MySQL is used to save all data and interconnect each module. The modules themselves are working as daemons, especially the crawling, filtering, labeling and feature extraction module. They are literally waiting for new articles in the database and getting active as soon as there is something to do for them. To make it possible that more instances of the same module can run, a simple locking mechanism will be implemented. The training module, as well as the actual classification and the evaluation are treated differently. Classification cannot be done without a pre-trained classifier. When a new article arrives, a classifier is trained with previous processed articles in the database and used to predict a label for the new article. The schema shown in figure 5.13, going clockwise from the crawling to the classifier training.

[7] http://www.csie.ntu.edu.tw/ cjlin/liblinear/
[8] http://liblinear.bwaldvogel.de/

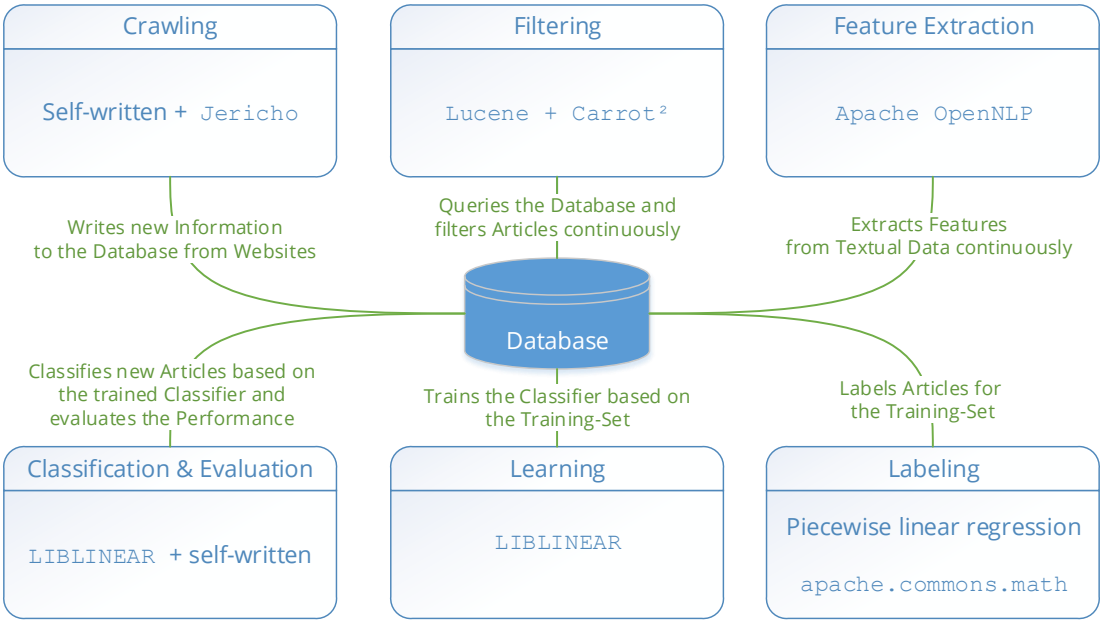


Figure 5.13.: Modules are running as daemon and are interconnected with the database. The workflow has to be seen clockwise: from the crawling until the point where a classifier is trained. The trained classifier can be used to classify a new article. This is described in more detail in section 5.5.9 in more detail.

In this thesis it is assumed that downloading and extracting textual content from HTML encoded websites are functioning well and don't need to be proven. Also the text processing part, done by an external library is functioning well and was the main topic of research while the OpenNLP Framework was created. It is therefore also assumed that text-processing, namely tokenization, POS tagging and named entity recognition works well within the performance claimed by the authors of the library. What is not part of any research yet is how well the filtering performs and how well the market prediction based on news articles performs. Therefore in the next section the main questions, which are tried to be answered by this thesis, are defined.

5.6. Research Questions Addressed

Having the design and the architecture outlined for a system that could potentially make predictions on financial market, leads to plenty of possibilities for testing and research. Some questions which are tried to be answered, although, are especially of interest and should be defined prior the actual prototype. Thus, the following questions are therefore tried to be answered with the gained results of the prototype created within this thesis:

- *Is it possible to determine "interesting" news from the news corpus on an automated basis on the fly?*

  It should be evaluated if the use of unsupervised methods (clustering), to find "interesting" macroeconomic news on a continuous basis, is valuable. The problem which is addressed is that older systems are built on either commercially obtained news corpora, or build on a static publicly available news corpus. The older prototypes and system designs do not address continuously available news. Commercial news are often available with labels to filter uninteresting news, which are, seen from a data processing standpoint, only noise. Systems which are based on publicly available news, use either onetime created keyword sets, or do not address the problem at all and introduce inherent noise into their systems.

- *Across all combinations which are possible with a bag-of-words approach, which is the best feature set?*

  Most prototypes are using a bag-of-words approach to build their feature set. But a bag-of-words approach can be done in various ways, utilizing for example only adjectives, or named entities. Therefore it is necessary to measure the performance with different feature-sets.

- *Is there a measurable delay between the news release and the impact until the news is fully reflected by the market?*

  According to M.-A. Mittermayer and G.F. Knolmayer (2006), Li et al. (2011), Schumaker et al. (2012) the (Stock) market fully reflects the information of a news release within 20 minutes. Therefore interesting news should be identified and the time until the market is saturated should be measured.

- *What if the polarity regarding an entity changes over the time?*

  People's mind regarding certain entities change and with the change sentiment dictionaries become inaccurate (Dragut, Wang, Yu, Sistla, & Meng, 2012). An approach to mitigate this problem could be an adaptive model generation.

5.7. Summary

This chapter defines the general idea and introduces the requirements. Based on the findings in the previous chapter, news articles should be labeled according to three different categories: "sell", " hold" and "buy". Also mainly a time-frame of roughly zero to 20 minutes after the article release is of interest. Therefore 6 different functional requirements are defined: crawling, filtering, feature extraction, labeling, learning and classification. One big advantage is gained through a module daemon design, where instances of the modules are separated and waiting for new articles in the database. Also, the different modules can be switched to newer or different technology and it is easily possible to evaluate new and optimized parameters in the different modules. Especially highlighted is the novel way of finding articles of interest through a combination of clustering and keyword search. For result reproduction purposes only non-paid news sources are utilized. The system is divided into three big phases: training, classification and evaluation. First a crawling module is downloading a news archive, which saves the articles in the database. A filtering module is querying the database for unfiltered articles, and gets active as soon as it finds a new, unfiltered article. The module creates then two clusters based on two keyword searches of all articles from the past days: a financial cluster and a non-financial cluster. Then the module sets a flag if the article is more likely from the financial domain or not. Then a labeling module is running in the background and continuously looking in the database for unlabeled articles, fetches one of them, labels the article based on exchange rate movements and saves the label back to the database. In turn another text processing module is also running in the background, continuously looking for articles which are not yet text-processed and is running a sentence detector, a tokenizer, a POS tagger and a named entity recognition classifier on them. Then, if an article was processed by each of the modules: filtering,

labeling, text processing, the article is ready for creating a support vector machine classifier. The textual features in form of POS tags and named entities of such an article are directly saved in the database and can easily be retrieved with raw queries, which can be used to create a feature vector without any large overhead. After having a new classifier created, a recent article, which was released in the past 2-3 minutes, can also be run through the filtering and text-processing, and a label can be classified, which can be used to predict the future exchange rate movements.

6. Prototype

"All experience is an arch, to build upon."

[Henry Adams]

While the previous chapter outlines the design, in this chapter the actual prototype is described and documented. Furthermore all important aspects of the prototype during runtime are described. The chapter gives an in-depth description about the single implementations of the modules and describes problems during implementation and run-time. First the general implementation requirements are defined to fulfill the requirements for an easy maintainable system. Then the details to each of the modules are given, where also the problems and solutions during implementation-time are described.

6.1. Overview

If one speaks about *big data*, usually enterprise systems like Apache Mahout are connected automatically. The system developed within this thesis will not build upon enterprise systems, but instead use existing systems and libraries where meaningful insights can be gained, like a MySQL database to store and retrieve data instead of a proprietary format like Apache Lucene, or plain queries instead of high level database abstraction tools. It will build upon Java, which can be run across multiple operating systems and is therefore not bound to a platform. Furthermore free and open source libraries are used for the tasks where available. As mentioned, for the database a MySQL server is used, where the design is outlined in the next section 6.2. The web-walking part of the crawler is self developed, as there could not be found any suitable library or existing system which would satisfy the needs to crawl solely a large news archive. For extracting information from a web-site (scraping) the library Jericho is used, where both, the crawling as well as the information extraction module are outlined in 6.5. For filtering a mix between queries and the clustering library from the Carrot^2-framework with the Lingo-algorithm is used, which will be described in section 6.7 in more detail. For text-processing the NLP library Apache OpenNLP is used, which is described in section 6.8 in more detail. The last step, the classification, is separated in two sub-tasks: First, for the labeling, a piecewise linear regression is

applied to the data, then a SVM classifier with a linear kernel is trained. For the support vector machine the library LIBLINEAR is used, which is available for free and for a variety of programming languages. This classifier is then used for prediction and is further validated, which is described in section 6.9.

6.2. Database

The central element of the whole prototype is clearly the database. All single modules depend on a central point for storing and retrieving data. From the the articles to a locking-mechanism, features and financial data, all comes together in the database, which is therefore described first. As already mentioned, the rational database management system (RDMS) MySQL will be used for this system. The design was modeled visually with the freely available software MySQL Workbench[1]. As an database engine InnoDB is used, which allows for ACID[2]-compliant transaction and has foreign key support. Therefore it can be used to integrate a basic locking mechanism directly utilizing the database, to avoid race conditions across modules, without the need to re-invent the wheel of atomic operations. Furthermore the foreign key support can help to make data-integrity easier through cascading actions on update or on delete. The database design is outlined in figure 6.1.

Database Design Description

One of the most important tables of the database is the `Article`-table, which holds a HTML-stripped (plain text) copy of a news-entry on a website. An Article belongs to a `Source`. Crawling is basically walking along a list of links and grabbing its content behind. So, beside of the root article, any articles belong to a `Link` (URL) and, also, an `Article_has_Links` to other websites.

When a new article is in the database, the text-processing module applies several sub-tasks to the article: it marks sentences, does tokenization, POS tagging and extracts the Named Entities which are contained in the article and saves it into the `Entity`, `Dictionary` and `Tokens` tables. After finishing the text-processing part, it is possible to retrieve and entity via `Article_has_Entity`, including the sentence-number of the entity within the article. The single words behind the tokens are saved into the `Dictionary`-table (if they are not yet saved), where they get an unique id through the primary key. Then they are connected via the `Tokens`-table to the article. Basically

[1] http://www.mysql.de/products/workbench/

[2] **A**tomicity. **C**onsistency. **I**solation. **D**urability. https://dev.mysql.com/doc/refman/5.6/en/mysql-acid.html

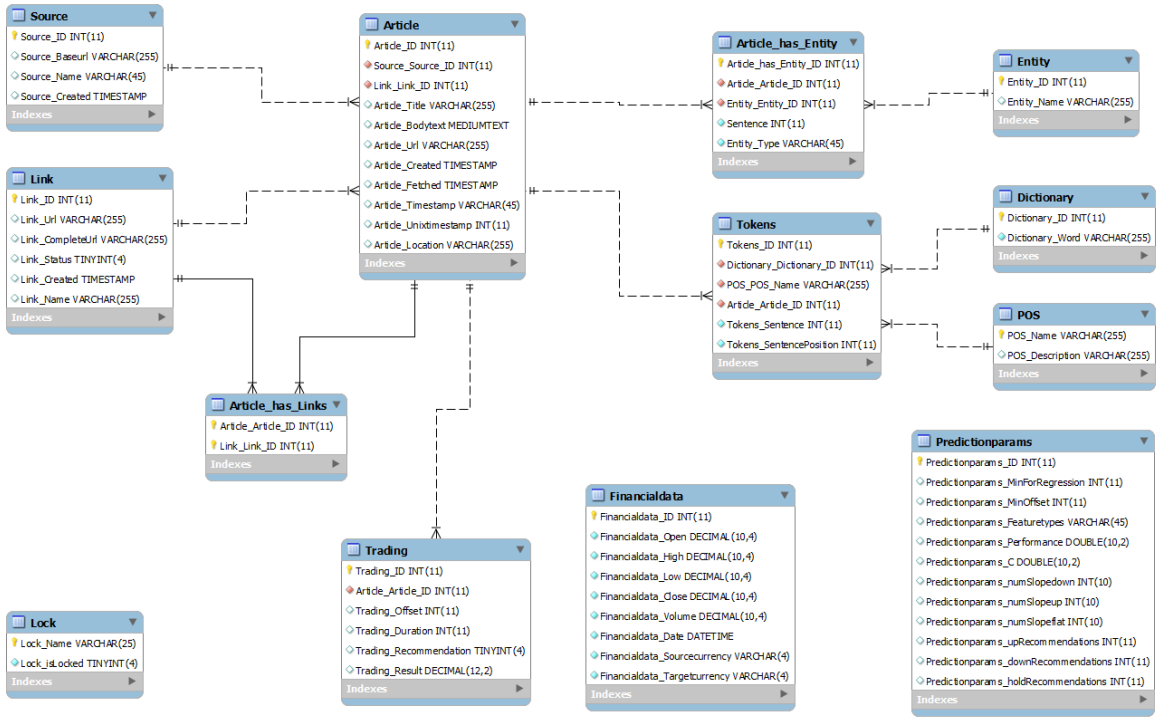


Figure 6.1.: Model of the Database design.

it's possible to completely reconstruct an article just via the `Token` and `Dictionary` table.

The financial data is saved in the `Financialdata`-table and can be saved with a resolution of up to one second. The timestamp of the exchange rate movements, the trading volume, as well as the open, close, high and low value can be saved. Furthermore its possible to store different exchange rates in the same table, where they are defined via the "`Sourcecurrency`" and "`Targetcurrency`" fields.

Different prediction parameters can be verified in a batch-run. The outcome of each run is evaluated with a 10-fold cross validation and is saved in the `Predictionparams`-table. A market simulation evaluation can be saved in the `Trading`-table, where the performance for each Article can be retrieved afterwards.

The `Lock`-table is used to avoid read-before-write across multiple instances of the same module.

6.3. General Class Description

In the previous chapter it was defined that a modular daemon structure is going to be used. This means, that different modules are meant to be running in the background and continuously looking in the database if there is something new to process. From a functionality point of view, all modules are strictly separated and instances of modules are not depending on any run-time data from other modules. But from a programming standpoint, some modules share code with each other: This starts at a, although simplistic, but unified user interface, a common locking class, etc. Therefore a number of classes are available which are used by each of the modules. The class mapping is depicted in figure 6.2 and will be described next.

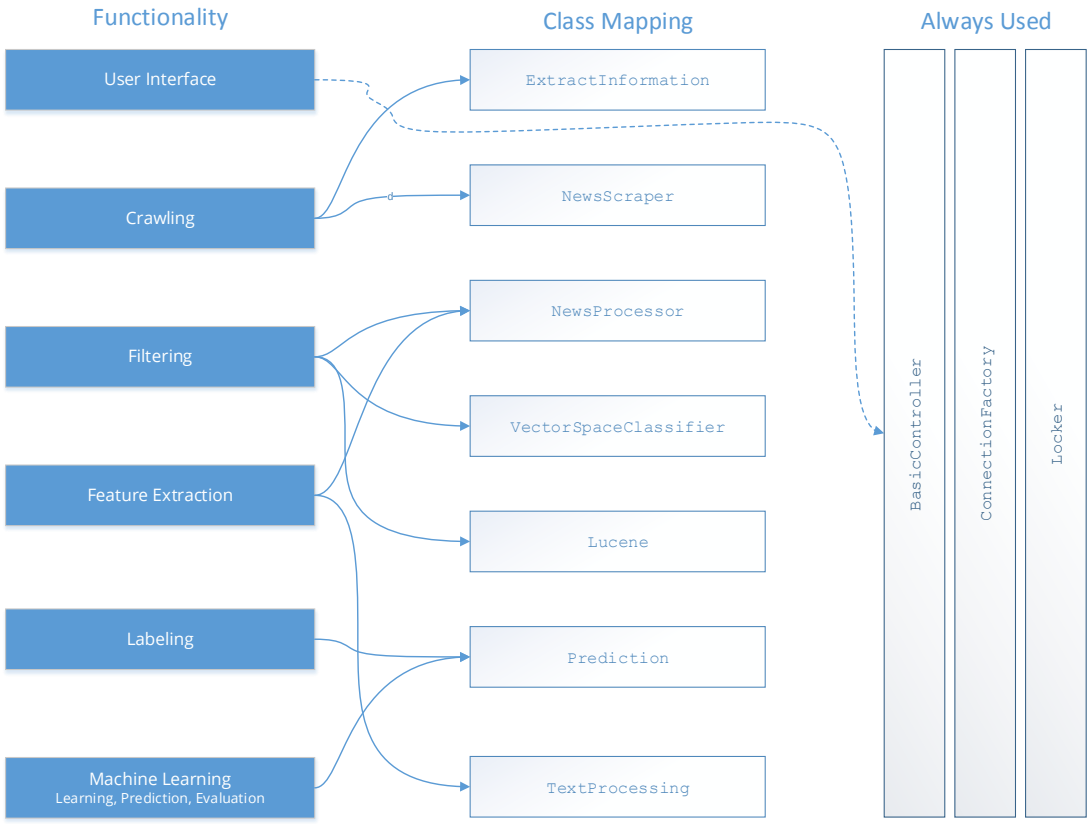


Figure 6.2.: Class mapping between each of the functional modules of the prototype and the classes that were created.

- `BasicController` is the entry-point of the prototype for user interaction purposes. It is, what is referred as the "main" class which is started and a functional user

interface enables the operator to start and stop an instance of a module, or run a prediction simulation.

- `ConnectionFactory` is responsible for opening/closing a connection, as well as all interaction with the database is running across this class. It defines methods for inserting, updating, and retrieving data to and from the database.
- `ExtractInformation`: While the crawling module fetches HTML encoded articles from the news archive, the purpose of this class is to extract the actual article content, as well as the title, the date and time when the article was published and other attributes of the actual article and update the database based on the information found.
- `Locker` is responsible for the locking mechanism and to avoid race conditions between the modules.
- `Lucene` provides a functionality to quickly create a Lucene index from the articles in the database. This is especially important for the clustering library Carrot[2], which requires a Lucene index as input, as well as for the similarity measure between a new article and the articles in the cluster.
- `NewsProcessor`: This class is responsible for connecting the articles in the database with the actual text processing module. This class can be seen as the module, which is continuously looking for unprocessed articles, while the `TextProcessing` class (see below) is then really doing the raw text processing, like sentence detection, tokenization, etc.
- `NewsScraper` is the main module class for the actual crawling of the news archive. It fetches uncrawled website-links from the database, retrieves the website and utilizes the `ExtractInformation` class to extract the information behind an article.
- `Prediction` is doing what the name suggests and is therefore responsible for fetching the articles for training, for creating a classifier, for classifying new articles. Furthermore this class also handles the labeling of articles. This is better described in section 6.9.
- `TextProcessing` is a class which provides methods for sentence detection, tokenization, POS tagging and named entity recognition. It is the link between the OpenNLP library and the way data is stored in the MySQL database.
- `VectorSpaceClassifier` is a helper to do vector space similarity measures.

To create a class diagramm with all classes, private and public methods, as well as the whole interaction diagram, is not enough space available within this thesis. A schematic overview can be seen in the appendix in figure A.1, where the reader will likely see that it advantageous to break the functionality down into the single modules and describe

each of the classes where each module is described. Thus, instead of describing the system as a whole, in the following sections each of the modules implementations are described in more detail, the used classes will be described and how the interaction works.

6.4. User Interface

The user can interact with the whole package through a command line interface. If the prototype is started several options are presented to the user, which is mainly starting a module daemon instance. The first class that is started is the `BaseController`, which opens a menu and allows basic interaction. Depending on the choice of the user, a certain functionality is executed, which is also shown in figure 6.3.

```
Start a Module (enter the number)
==================================
============================== NEWS Crawling options ==============================
1: Start Reuters links extraction from Archive - one-time
2: Start Reuters links extraction from Archive - infinite-loop until user-interruption or error

3: Start Reuters Article Extraction from (1) extracted Links - one-time
4: Start Reuters Article Extraction from (1) extracted Links - infinite-loop until user-interruption or error
============================== NEWS Pre-processing ==============================
5: Start News-Article Classification in Forex/Non-Forex
6: Do the above (nr 5) in an endless-loop for all un-processed articles.

7: Start Text-Processing: Sentence detection, Tokenizing, POS Tagging, Entitiy Extraction.
8: Do the above (nr 7) in an endless-loop for all un-processed articles.

================================ Prediction ================================
9: Start Prediction.
10: Prediction: find best Values (in data-directory then).
11: Run batch-prediction test on feature-types.
============================== Textshortener ==============================
20: Start Textshortener.
Task:
```

Figure 6.3.: A simplistic user interface to start and stop modules or run a simulation.

6.5. Crawling of News

Crawling is the process of continuously retrieving *new* information from publicly news sources. These can be websites, blogs, social networks, RSS feeds, etc. For each of the news sources, there has to be a specialized crawling module. In this thesis the development of a crawler for the archive of a popular news site was accomplished, which enables the simulation of new news arrival. The simulation is necessary due to several reasons: First and foremost with a simulated environment it is possible to

reproduce the results. Second: in the period the project was implemented there was simply nothing interesting happening in the financial domain. And third: The news crawling had to be adjusted several times, leading to a complete removal of already crawled data. Thus, instead of waiting for "interesting" news to arrive, a possibility was created to simulate the situation of new arriving news by downloading news from a period where something financially relevant happened. Specifically the crawler was written to download the news from a publicly available news archive from January and February 2012, to avoid longer waiting times *until something happens* in the financial domain between the Euro and the US Dollar. This data will be used to simulate the events of arrival of new information.

Data Gathering

As described in section 5.5 and section 5.5.4, in order to do anything towards machine learning, the first step is to gather news data. For every news source a specialized crawling module has to be adapted. To simulate financial news arrival and to gain meaningful results, the crawler was adapted to extract the content of the news *archive* of a public news agency from January and February 2012. With this data, all further steps will be simulated as if the news would just arrive. As already discussed in section 5.5.4, to overcome the problem of broken HTML, the parser library Jericho will be used to extract the actual content of an article. The crawler has therefore two main functions:

- Fetch a new link from the database, fetch the websites behind, extract further links from that and update the database accordingly
- Fetch a link to an article from the database, fetch the website and push it through a processing module which utilizes the Jericho library to extract the main content and other attributes

Workflow

The crawler is doing three things: Firstly, getting a new link to a website from the database. Secondly, extracting the content from that website and thirdly, also extracting new links from that website and writing it back to the database. The class diagram is depicted in figure 6.4. In addition to the normal functionality, to support more than one crawling modules at the same time, a locking and staging mechanism is implemented. This works as follows: If a crawler is about to get a new link to a website, it is retrieving and updating the entry in the database as an atomic operation. The module is updating the *new* link as an *in process* link. Only if the operation is successful the actual website crawling process is started. If the operation was not

successful within the module, then it can be assumed that another module was quicker with the locking. This avoids doubled workload and makes it possible to distribute the crawling across several machines. As a crawler is always an adapted version for a special news-source, it is also linked to the database as such. Therefore, for each crawler, the *Source* table in the database has the base seed URL for the crawler. The document behind this source-url will be retrieved, if there are no more new links to websites in the database and will be used to distinguish new links (or articles) and usually points to the main-page of a news-source, or to an RSS-feed.

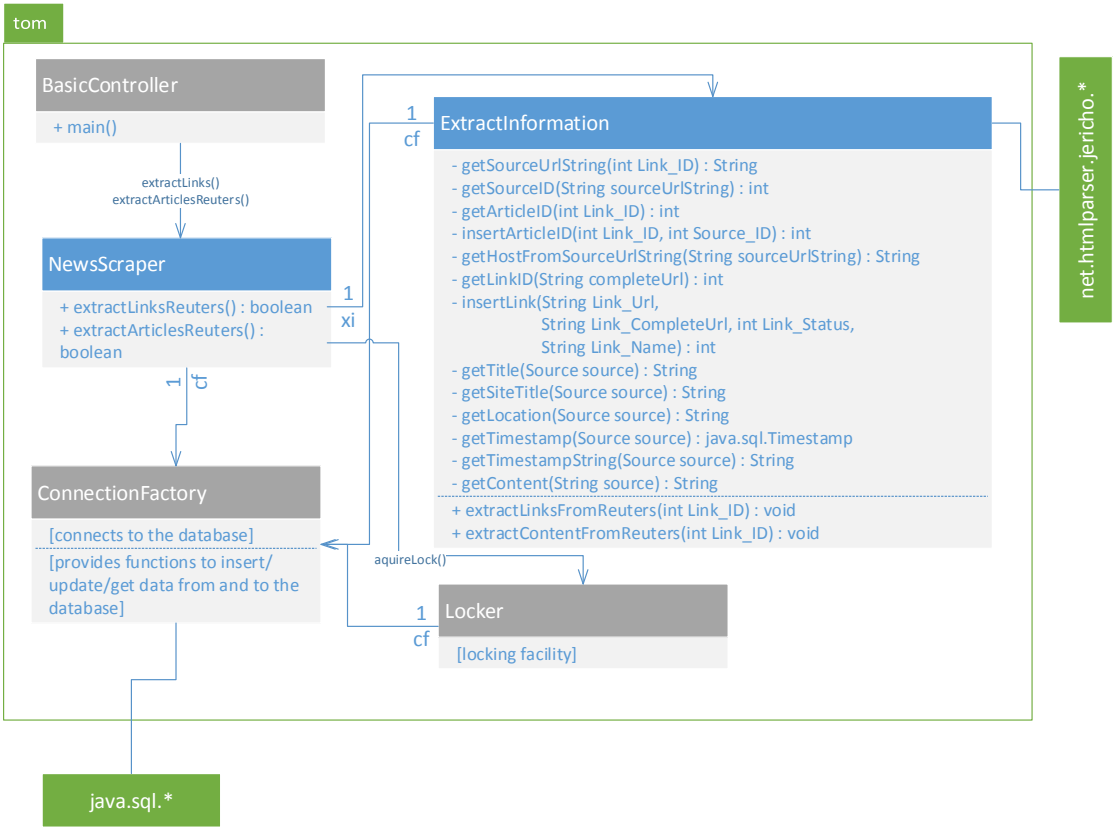


Figure 6.4.: The class diagram for the crawling and extraction of articles and links module.

News Corpus

The news will be saved in the *Article* table in the database, whereas the source-link (where the article comes from) will be saved as an entry in the *Link* table. The Article-table has three important fields: The *Article_Title* which is the extracted main title of the article. This title is usually either the bold text at the top of an article, or

the website-title. In general it depends on how the website-parser was adapted in the crawling module.

Listing 6.1: Part of a typical news article

```
U.S. investors sue Lloyds chiefs over HBOS deal

LONDON | Mon Jan 2, 2012 10:50am EST LONDON Jan 2 (Reuters) - American
    shareholders are suing Britain's Lloyds Banking Group and the bank's
    former executives, saying they were misled over its rescue of fellow
    lender HBOS in the depths of the financial crisis [...]
```

The *Article_Bodytext* has the actual text of the article and shouldn't contain anything else than plain text (no more HTML, or other markup parts). The field *Article_Created* is of the type DATETIME and refers to the exact timestamp, in the timezone GMT, when the article was created (published). This timestamp is usually needed to be extracted from the content somehow, and often given at the beginning of the actual body-text, like in the example 6.1 as "*Mon Jan 2, 2012 10:50am EST*". The rest of the fields are referring for example to the entry-creation timestamp in the database (*Article_Fetched*), to the text which made it possible to extract the timestamp (*Article_Timestamp*) - which is mainly for debug and validation to see if the date-conversion is working correctly, and a location (*Article_Location*), which is filled, if the location was given at the top of the article. A typical example of an news article is given in 6.1. Beside the obvious extraction for the title and the bodytext, the *Article_Created* field would be filled with "2012-01-02 15:50:00" and the *Article_Location* field with "London".

6.6. Exchange Rate Movements

To get *any* financial data on the web is not very complicated. What this project needs are the exchange rates of EURUSD with a resultion of at least M1, from January and February 2012. Mostly data which is offered for free download has a resolution of M5, M10 or H1, or is separated for download in zip-archives on a daily basis, or in a proprietary format. On the other hand such data can be obtained commercially. One way to obtain free exchange rates with a high resolution is to extract them directly from a trading software which offers also back-testing, where a popular example is the software MetaTrader. The idea is to extract the historical exchange rates which is actually used to back-test a trading strategy, save it as a CSV with a timestamp, open, close, high and low value. Although this method raises legal questions, there are several tutorial available which show how to do it[3].

[3] http://www.fxdd.com/us/en/forex-resources/forex-trading-tools/metatrader-1-minute-data/

Luckily, after extensive search, a suitable, free, plaintext dataset could be found for download on the Forextester-website[4]. The 250MB text-file contains the open, close, high and low rates between the US Dollar and the Euro on a minute basis with timestamps in Greenwich Mean Time (GMT) without daylight saving time from the year 2001 until today (updated continuously). This makes it easy to import it to the database, in the "*Financialdata*" table. Unfortunately the trading volume is given always as "4.0" in the file and is therefore not used further in this thesis. The data needs almost no more pre-processing and can be imported using a CSV parser.

6.7. Filtering

Filtering consists of several sub-tasks, which are mainly necessary before clustering the articles with the Carrot[2] clustering library[5]. The class diagram shown in figure 6.5 is an excerpt of the classes of the prototype with all necessary information which is required for the filtering module.The user starts the prototype, where the `BasiicController` acts as the entry point. The `BasicController` creates a new instance of the `NewsProcessor` and runs the function `markNewsRelatedToForex()`. This function acquires a lock for the filtering module and checks the database if yet unfiltered articles are available. Either there is nothing to do, which results in a lock-release and a fallback to the `BaseController`, which, in turn, calls the `markNewsRelatedToForex()` again after a certain time interval. Or, a new article is available for filtering and the function `markNewsRelatedToForex()` gets active: First the article-status is updated from "new" to "in progress" and the lock is released. Then the date and time when the article was published is used to request all articles with a certain period from the database *before* the articles release date and index them into a Lucene index, which is done utilizing the function `indexArticlesToLuceneFromDate` in the class `Lucene`, which accepts two parameters: a start and an end-date as string.

After having news articles from the last days in a Lucene index, it can be loaded into the Carrot[2] clustering library, where this index is searched with two different sets of keywords: One keyword set is to get articles which are from the financial domain (listing 6.2), and one keyword set is used to get articles which are not from the financial domain (listing 6.3). Furthermore, the outcome of the queries are saved back to new Lucene indexes. The searching is done by the function `getClusters` which also accepts, beside the query, two parameters to restrict the number of results and he number of clusters. These numbers are important to adjust for the purpose of the two thematic groups: The goal is to have one Forex-clustering with less clusters, which represent

[4] http://www.forextester.com/data/datasources.html
[5] http://project.carrot2.org/

more the main topics, and one broader, non-Forex clustering with a lot of different topics, which just groups very similar articles together.

Listing 6.2: Find Forex articles in a Lucene index

```
*:* AND (EU OR Euro* OR Market* OR Bank* OR Forex* OR dollar* OR eur* OR bln*
     OR financ* OR govt* OR econom* OR bailout* OR crisis* OR high* OR low*)
    NOT (Game* OR film* OR Football* OR Production* OR REG* OR Kitchen* OR
    Bathroom* OR box office)
```

Listing 6.3: Find Non-Forex articles in a Lucene index

```
*:* OR REG* NOT (Euro* OR JPMorgan* OR Moody* OR crisis* OR nikkei* or FTSE*
    OR Forex* OR dollar* OR FOREX* OR financial* or Euri* OR bailout*) NOT (
    financial* AND market*) NOT (foreign* or exchange*) NOT (goldman* OR sachs
    *)
```

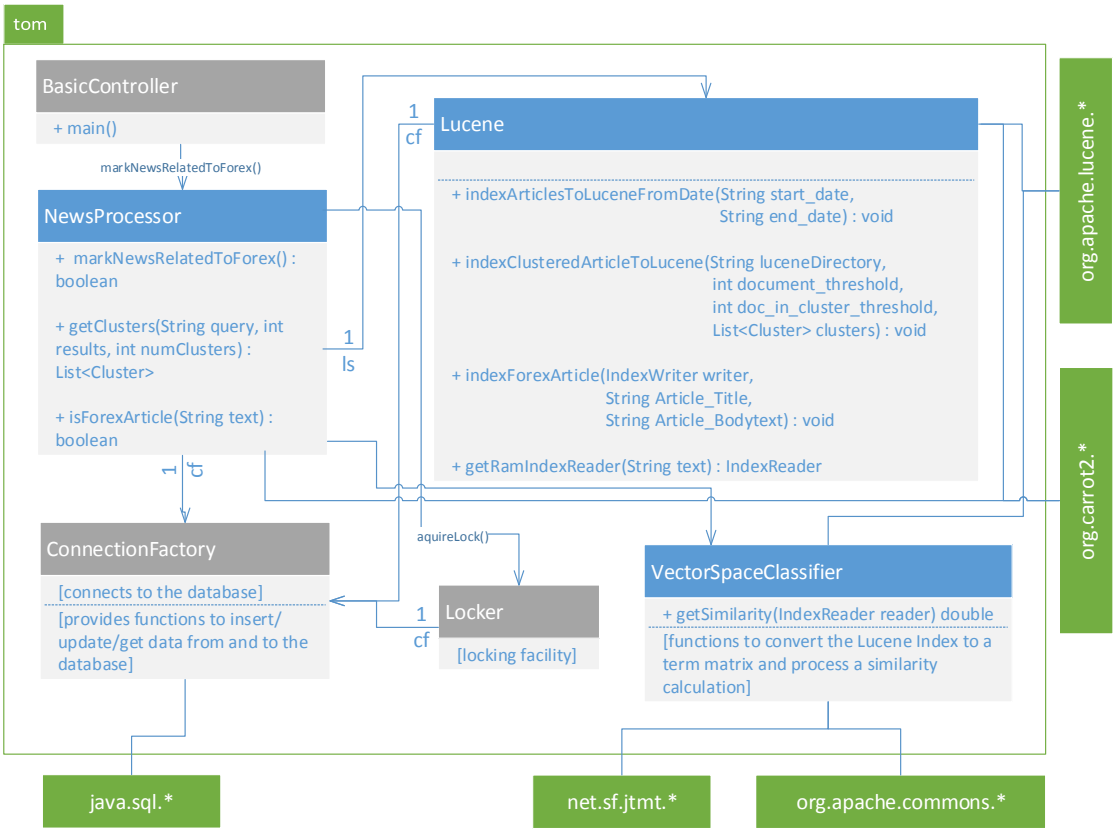


Figure 6.5.: Class diagram of the filtering module including the external packages and the most important operations.

The result-sets are therefore clustered with different cluster sizes. From the financial domain result-set 15 clusters are tried to find, meaning, the results are clustered in 15 different topics. From the non-financial domain 35 clusters are tried to build. The parameters were found empirically through extensive testing, but can be adjusted accordingly in a future work. Then, again, both result sets are indexed into a Lucene index - one Forex-Index and one Non-Forex-Index. A new arriving document can now be matched against these two Lucene indexes using, for example, a cosine similarity, which is done by the `VectorSpaceClassifier`. This is loading the new article into a Lucene index and matching that index against the two prior created indexes. If the document matches more the Forex-Index, the topic in the article is probably of current interest to the financial domain, if the document matches more the Non-Forex-Index, the topic of the article is probably more of a general content. The implementation of the cosine similarity in this thesis is close to the example implementation done by Pal (2011b).

6.8. Text Processing and Feature Extraction

As already described briefly in the previous chapter, text processing consists of several sub tasks. It uses a new publicly available NLP library from the Apache Foundation[6] to perform all the text-processing tasks. The class diagram is shown in figure 6.6 and the workflow will be described in the following section.

Workflow

The entry point is again the `BasicController` which creates a new instance of the `NewsProcessor` class. From there only a single method is started with the rather self explaining name `sentenceDetectionTokenizingPOSTaggingEntityExtraction()`. The method is responsible for looking for new, yet unprocessed articles in the database and, if one found, calling the method `detectSentencesTokenizeEntityPOS(int Article_ID, int Link_ID, String Article_Bodytext)`. Beside the text-processing itself, also a locking mechanism is again in place to allow multiple instances of the same module working at the same time, which works similar as within the filtering module: If a new article arrives a lock is tried to be acquired, the article fetched and a flag is assigned that the article is in progress, then the lock is released. The text processing works within several steps and all actual textual operations as tokenization or POS tagging are implemented in the class `TextProcessing`, which is responsible for two things: utilize the Apache OpenNLP library for the text-processing part and save the results back to the database.Therefore, the `NewsProcessor` class can be seen as

[6] http://opennlp.apache.org/

the controller class for the `TextProcessing` class. First the sentences are detected by

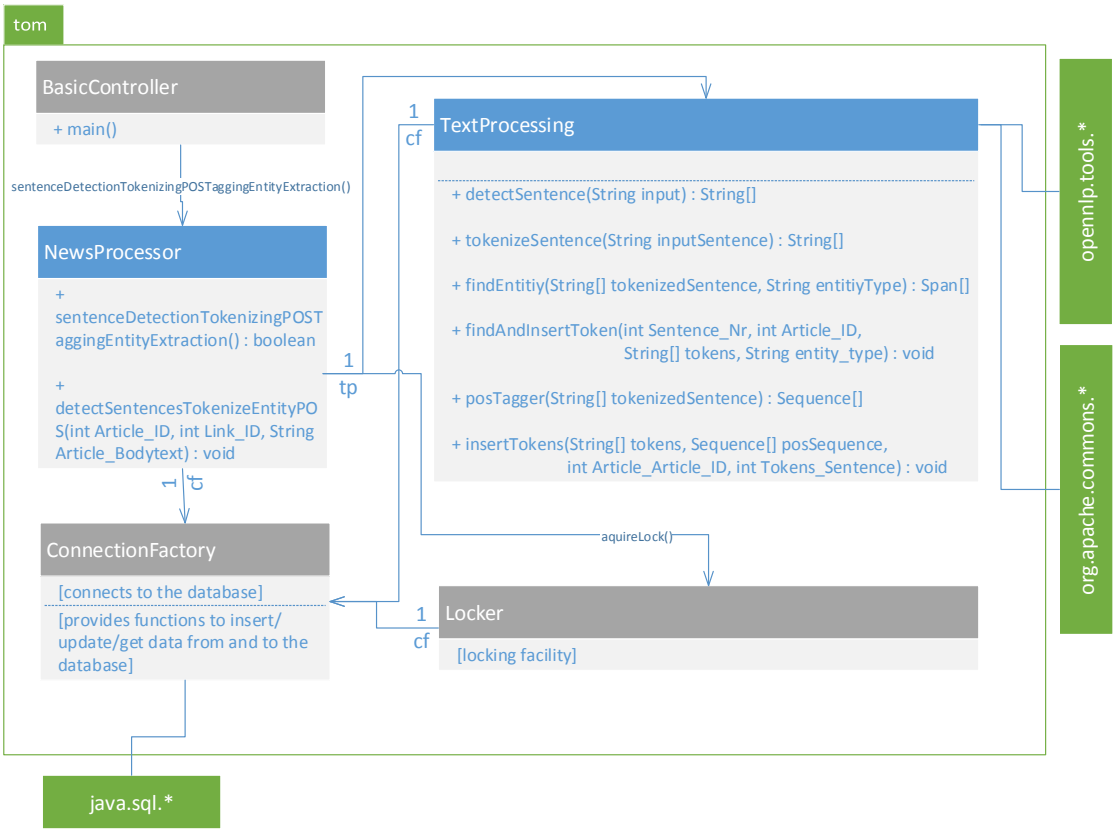


Figure 6.6.: Class diagram of the text processing and feature extraction module including the external packages and the most important operations.

the `detectSentences(String input)` method, which returns an array of sentences. Each of the sentences is then tokenized with the `tokenizeSentence(String inputSentence)` method, which returns an array of single tokens. The single word behind the token gets saved in the "*Dictionary*" table, whereas in the "*Tokens*" table only the link to the word in the "*Dictionary*" table is saved. Furthermore, in the "*Tokens*" table the position of the sentence of the token within the body-text is saved, as well as position of the token within the sentence is saved. So an article-text could be reconstructed completely only with the "*Token*" table and the "*Dictionary*" table. After having the tokens, the methods `findAndInsertToken(...)` are performing a named entity recognition with use of different pre-trained models. Specifically it is searched for Persons, Money, Organization and Location Entities. They are also directly saved back to the database in the "*Entity*" table (if the entity is not yet already saved) and connected to the article via the "*Article_has_Entitiy*" table. Furthermore, each token is POS tagged via the method `posTagger(String[] tokenizedSentence)`, returns

a sequence of tokens matching the input String-array. Then the POS tags including the token and the sentence number are saved to the "*Tokens*" table which links to the "*POS*" table.

Listing 6.4: Sample query to get adjectives from articles

```
SELECT a.Article_ID, t.Tokens_ID, d.Dictionary_Word, a.Article_Bodytext FROM
    Tokens t
JOIN Dictionary d ON d.Dictionary_ID = t.Dictionary_Dictionary_ID
JOIN Article a ON a.Article_ID = t.Article_Article_ID
WHERE
a.Article_Created BETWEEN '2012-01-05 08:10:00' AND '2012-01-05 08:30:00'
AND t.Tokens_Sentence <= 2
AND t.POS_POS_Name IN ('JJ', 'JJR', 'JJS')
ORDER BY t.Tokens_SentencePosition
```

Once the data is in the database in this structured way, SQL queries can be used in a very powerful (and easy) way to obtain certain data of articles. An example can be to extract only the adjectives of articles, and only from the first 2 sentences of the article-text. Furthermore, only adjectives from articles which where released on the 5th of January 2012 between 08:10:00 and 08:30:00 GMT. Such a query is shown in listing 6.4. The result of the query can be viewed in various SQL clients. One of the clients for Windows is HeidiSQL[7], which is very useful to debug the output and export data.

6.9. Labeling and Classification

The labeling and classification task is merged in the `Prediction` class. The reason that that labeling is not separately processed as a module, is the simulated prediction batch processing, which should enable to test if a certain offset and a duration for the labeling has an impact in the predicted market movement. Referring back to section 5.5.7, especially to figure 5.11, outlines the two variables *offset* and *duration* , which are used to calculate a meaningful label from the exchange rate movements. Summarized, a label is calculated from exchange rate movements after the release of an article with n minutes offset and d minutes duration. Classification with this SVM library is a pure supervised learning task, which requires to train a classifier with a training set first. Therefore articles from the last month are labeled first, and then the the classifier is trained with the labeled articles. This classifier is then used to predict the label of a new article, which is simulated in a loop with increasing offset and increasing duration.

The sentiment regarding a topic changes over the time, how Dragut et al. (2012) describe it. This means that the classifier should not be trained with labeled articles

[7] http://www.heidisql.com/

which are *too old*, since the sentiment against a topic discussed in news could change. In this thesis it is assumed that a classifier can be used to label a new arriving article, if the classifier is trained with articles from the previous month. But not only the sentiment-change is important. According to M.-A. Mittermayer and G.F. Knolmayer (2006), Li et al. (2011), Schumaker et al. (2012), the article's content is fully reflected in the market about 20 minutes after the release of the article, which was mainly tested for the Stock market. In this thesis the offset and the duration are therefore variated in the batch processing. This can be done by shifting the time-range for the labeling approach and will be explained next.

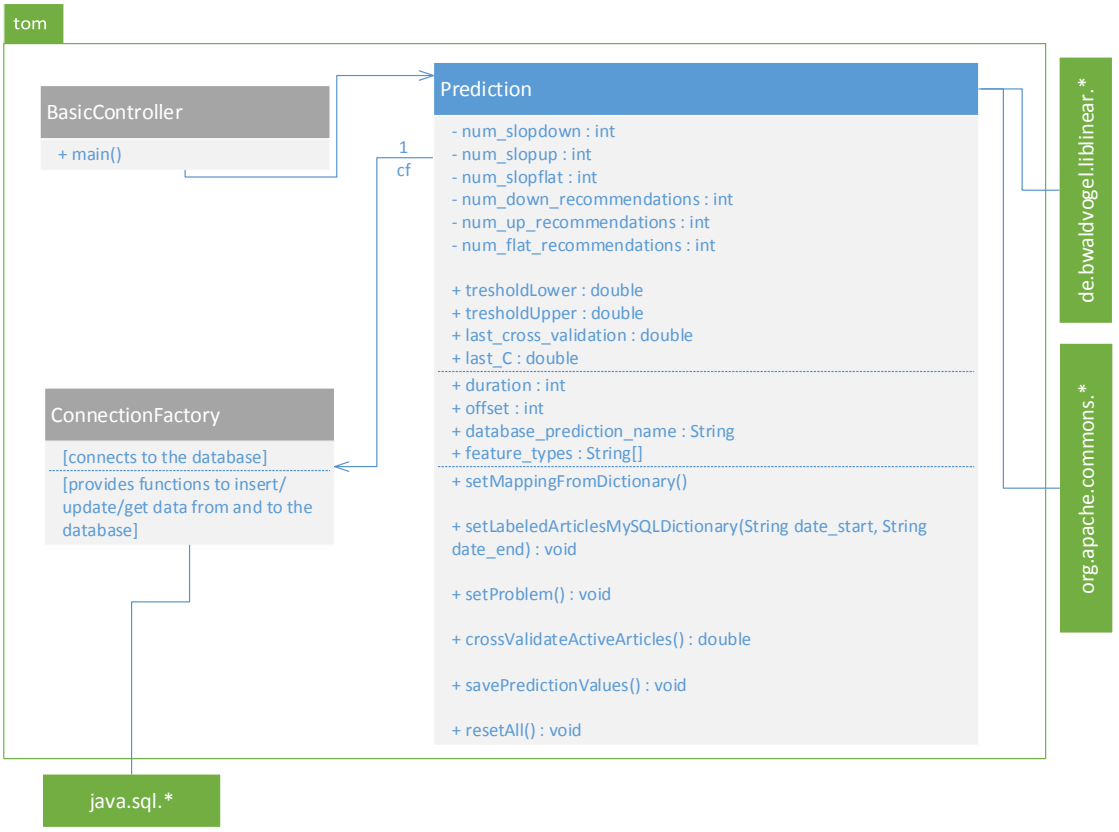


Figure 6.7.: Class diagram for labeling and prediction simulation with all important functions and parameters.

General Description

The class diagram for the prediction, as well as the labeling, is depicted in figure 6.7. After articles are labeled according to the slope of the linear regression (slope goes down = sell, slope goes up = buy, slope is steady = hold), a classifier can be trained.

For training a whole array of different features is available. The classifier, for example, can be trained using only adjectives, or verbs, or nouns, or a combination, or named entities. Therefore all of the possibilities will be batch-processed and the outcome is saved in the database in the "*Predictionparams*" table. This table holds, beside of the performance and which features were used, also information how many articles are in the different categories (sell, buy, hold) and the offset and the duration of the automatic labeling. Also how many sell, buy or hold classifications were given, so it should be easily possible to find good combinations of parameters and exclude such parameters which, for example, lead to unbalanced data.

6.9.1. Workflow

The sequence diagram 6.8 depicts the whole labeling and prediction workflow. The labeling approach, as already mentioned, works slightly different than the other modules for several reasons. Firstly, due to the fact that no suitable source to download exchange rates on a continuous basis could be found leads that exchange rates were downloaded in advance for the simulation-period and imported to the database. The exchange rates for January 2012 and February 2012 are therefore already available in the "*Financialdata*" table. Also, to support a proper evaluation of the classification, the exchange rates have to be known in advance.

The workflow is separated in two sub-tasks. To gain more insights into which parameters are beneficial for a sentiment prediction, a batch-processing is in place, which tries all possible combinations of parameters, ranging from different offsets and durations for calculating the linear regression for labeling, to different POS tag combinations for the feature vector. This is of course only relevant for academic research and cannot be used as such for a real-world prediction. The outcome of each batch-process is saved in the database. More concretely, in the `BaseController` all possible combinations of features (adjectives only, adverbs only, adjectives plus adverbs, nouns, nouns plus adjectives, etc.) are defined in an array. Furthermore the two parameters for offset and period for the labeling are set in a nested loop. A new instance of the `Prediction` class is created within the `BaseController` and all parameters are adjusted according to the parameters in the loop. The classifier is trained with the articles from January 2012 and an evaluation is done on the articles from February 2012, where the outcome of the simulation is saved to the table "*Predictionparams*". The articles which are used for classification were prior tagged as "financially interesting" from the filtering module. Furthermore such articles are excluded which are outside of European trading hours, meaning only articles are used which were released between 09:00 and 19:00, on weekdays. Also such articles are excluded which were released exactly at 13:30 and 14:00, due to plenty of double and summary news releases through news agencies at exactly these timestamps.

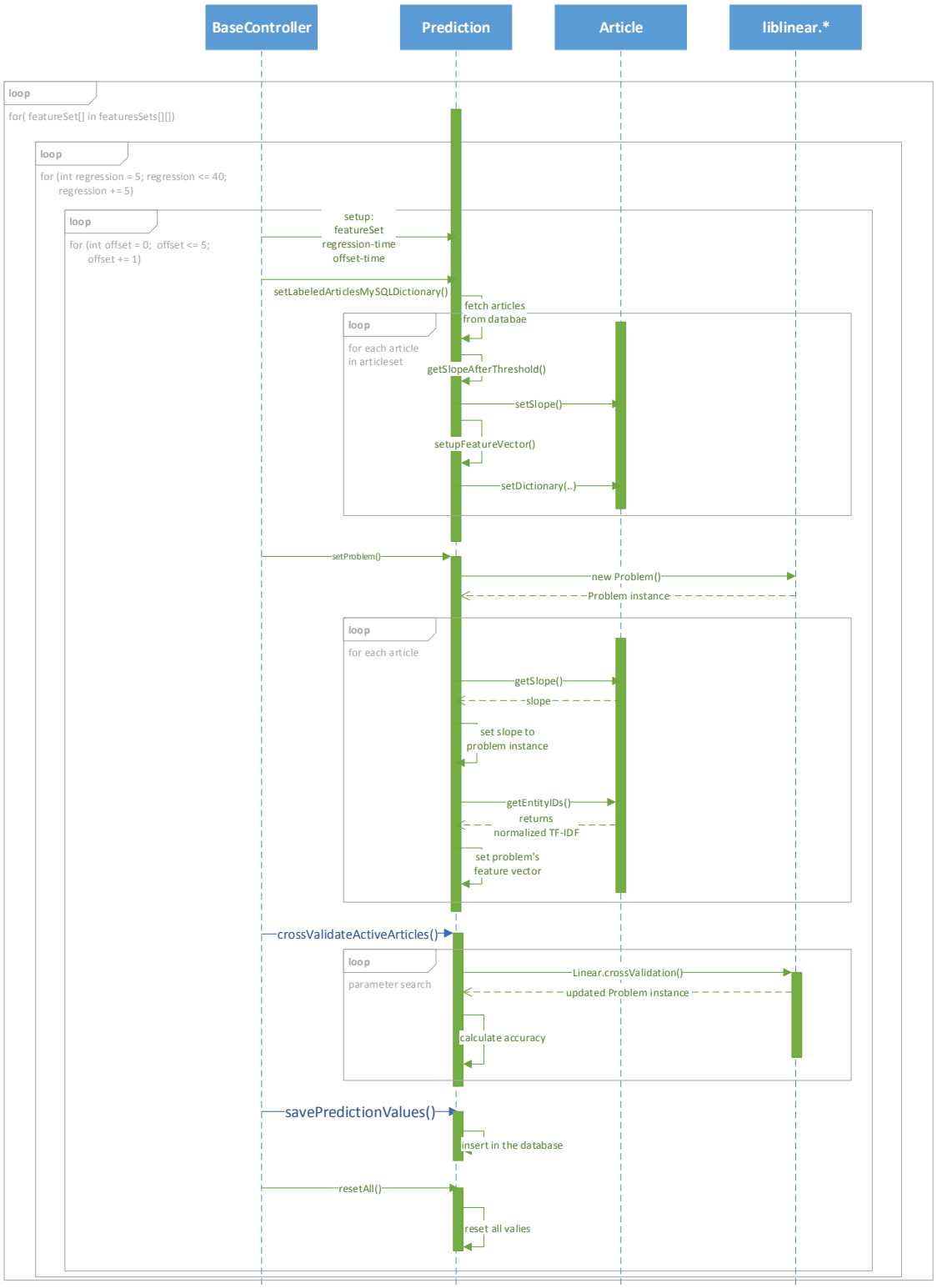


Figure 6.8.: Sequence diagram for the labeling and prediction of the batch-simulation.

First a feature-vector is built based on the *old* news articles. The labeling of the documents happens during the creation of the feature vector based on the data in the *Financialdata* table. Then a grid-search for the right parameters is performed, based on the description in the guide of the LIBSVM by Hsu et al. (2003). After having the right parameters, the classifier is used to classify *new* articles. The performance of the classifier can be distinguished using a 10-fold cross validation. To do so, the *new* articles also have to be labeled, which is possible, due to the prior downloaded financial data. In a real-world simulation one has to wait until an article has a certain age in order to download exchange rates to evaluate the results. Then the results, the parameters and the number of articles in each category are saved in the "*Predication-params*" table. Furthermore the timestamp of the article including the predicted label is saved to the "*Trading*" table, which can be used to easily verify the profitability of the classification.

6.10. Summary

In this chapter the actual implementation is described in more detail, as well as the problems and workflow of the different modules are outlined. To get meaningful data, articles have to be downloaded from websites, where the content has to be parsed. This is not always easy, as different obstacles like a broken syntax make an automated parsing hard. After the article is in the database it is filtered using a clustering method. Two different groups of clusters are created: one from the financial domain, another from the non-financial domain. The clusters inside each group are labeled according to thematic content and a new article can be matched using a similarity calculation to validate if the article is in the financial domain or not. Then, using a new NLP framework from the Apache Foundation, the articles can be tokenized and the POS tags can be found. The underlaying data to an article is written back to a MySQL database, which makes it easy to gain in-detail insights and to get powerful results with relatively simple queries. Having the article's tokens and POS tags in the database, then a feature vector with a dictionary from one, or many POS tags can be created. Furthermore an offset and a duration is introduced during labeling-time, which should enable a validation if articles are reflected in the market prices around 20 minutes after an articles release. The classification runs as batch process and all possibilities of feature vector combinations, as well as offsets and durations for the prediction-time are validated. The performance, including all used parameters is written back to the database for further use.

7. Results and Discussion

The last chapter explained in detail how the prototype was built, while this chapter presents the results gained from running the prototype in a simulated environment. Basically it is assumed that the crawling module works as expected and downloads article-content without any further problems. Also the text processing with the OpenNLP Framework is expected to work within the known performance boundaries to extract features such as POS tags or named entities from news articles. Thus, the only modules where the results are of interest are the filtering module and the prediction module.

First a general overview is given, which data was downloaded and used for the different modules. Then the topic filtering will be examined and insights will be explained. Based on the results of the filtering module, the Forex prediction results will be presented and discussed.

7.1. General Overview

In total, the crawler successfully downloaded 208.989 articles, which were released between January 2012 and Feburary 2012. These around 209 thousand articles represent the whole corpus, which contain articles with mixed topics. Not every article is from the financial domain, or inside trading hours, thus the corpus has to be filtered in various ways. After excluding such news articles which are released outside of the trading hours of interest (see 5.5.5), 52.121 news articles remain. Furthermore, only considering articles which were marked as "Forex related" by the filtering module, then 4.867 articles remain, 4.131 from January 2012 and another 736 articles for a later market simulation, from February 2012. The next section explains the outcome of the filtering module in more granular detail, while the section 7.3 provides more details about the actual economic news sentiment prediction.

7.2. Topic Filtering

In general, the filtering is a main contributer for good results. If textual data is used which has not much, or nothing to do with the financial domain, then noise is introduced into the system. Therefore the purpose of the filtering is to avoid having articles in the training or test set which are not from the financial domain. The workflow of the filtering explained in section 5.5.5.

Financially interesting news topics change over the time and the missing an annotated corpus makes it impossible to have hard evidence about the performance of such a filtering module. Basically, the *top* macroeconomic topics, which drive the exchange rate between the US Dollar and the Euro from January and February 2012 are well known to an economically interested person. But what are "economic and financial interesting" news which can influence a currencypair in general? It may range from very large corporations stock values to a countries economic crisis news. So, a real, hard ground truth measure is missing and is difficult to define, but a general impression can be gained based on random samples which are manually labeled and compared to the outcome of the filtering module. Therefore the overall performance will be based on 200 samples between January 2012 and February 2012. 50% of those samples are randomly drawn from articles which were flagged as "financially related" from the filtering module, and 50% randomly drawn from articles which were flagged as "financially unrelated". Those 200 articles will then be manually labeled, so the domain and impact will be manually identified. After that the outcome of the filtering will be compared to the manual assigned flag ("important"/"unimportant"). The confusion matrix (table 7.1) summarizes the outcome.

		Predicted Category	
		Related	Unrelated
Actual Category	Related	**49**	**8**
	Unrelated	**51**	**92**

Table 7.1.: This confusion matrix shows the following: in the left column the number of articles are listed that are automatically labeled as "related" to the financial domain. In the second column the number of articles are listed that are automatically labeled as "unrelated" to the financial domain. In the first row are the *true positives* and *false negatives*. In the second row are the *false positives* and *true negatives*.

The full tables to this comprehensive summary are listed in the appendix A.1 and A.2. Based on the numbers the precision, recall, accuracy and F1 values are caculated and shown in table 7.2.

Precision	**49,00%**
Recall	**85,96%**
Specificity	**92,00%**
F1	**62,42%**
Accuracy	**70,50%**

Table 7.2.: Precision, Recall, Specificity, F1 and Accuracy for Filtering

Discussion

Due to the fact that there are not many false negatives in the confusion matrix, it appears that the filtering module is not removing articles which would be actually important. Therefore, information is not lost. But, due to the many false positives, noise is introduces on a large scale. Thus, the filter is still far from perfect, because many articles are not from the financial domain, but were still considered for the further classification task. A reason could eventually be found in the corpus itself: It seems that most of the articles are actually not important to the financial domain. A manual approach to filter the articles would be probably the best way, but is undesired due to the large amount of data on a continuous basis. A stronger filter without introducing more false negatives is desired. Usually one immediately thinks of a keywords list for white-listing/black-listing articles, but it is undesired as well due to the fact that such a list has to be manually created and maintained.

Summarized can be concluded that an achieved accuracy of 70.5% is very good for an attempt to filter topics continuously without the need of a manual intervention. The filtering module is not removing articles with important information from the corpus, but the filtering could be stronger, because not all unrelated articles were removed. Even more, due to the weakness of the filter, a lot of noise is still introduced during learning and prediction for the news sentiment classification.

7.3. Forex Prediction

This section presents the results from predictions made on the time series representing the exchange rates US Dollar - Euro from January 2012 based on news news articles from the same period. As already stated in section 5.5.5, only articles during the main trading hours are considered. The results were gained by using a batch label, training and classification method (see section 6.9). Thus, the following two main points were tested and are incorporated in the results:

- **Time:** What the best offset and duration is, after an article release, to perform the prediction, is tried be answered within the batch run. It could be advantageous to wait a certain time after an article release (offset) and predict market movements for a certain period. See section 5.10 for theoretical details.

- **Features:** Which type of words as feature-set performs best, which was also tested in the batch run. The feature set included all possible combinations of "adjectives", "nouns", "verbs", "adverbs". See section 6.9 for further details.

Not all results can be used to draw a conclusion. For example such results will be excluded which perform lower than a "always hold" recommendation. In related work often only such results are excluded which are worse than a random classifier (33% accuracy), but it was advised that also such results should be excluded which perform worse than a classifier which gives always a "hold" recommendation, similar to a "do nothing" prediction (Bacher & Stuckenschmidt, 2012). Furthermore a threshold is introduced for the maximum rate for unbalanced data. Results will be excluded when there are more than 75% of all articles in one category. In other words in one category (up/down/flat) must be less than $\frac{3}{4}$ of all articles for learning. To present full results very comprehensive, several pages long tables are necessary, which will be included only in the appendix A.3. Thus, in this chapter only the summarized, graphical representation of the comprehensive tables will be presented and discussed. First the feature sets will be compared, which follows a more complex comparison between the offset, duration, feature-set and performance.

7.3.1. Duration and Offset Performance

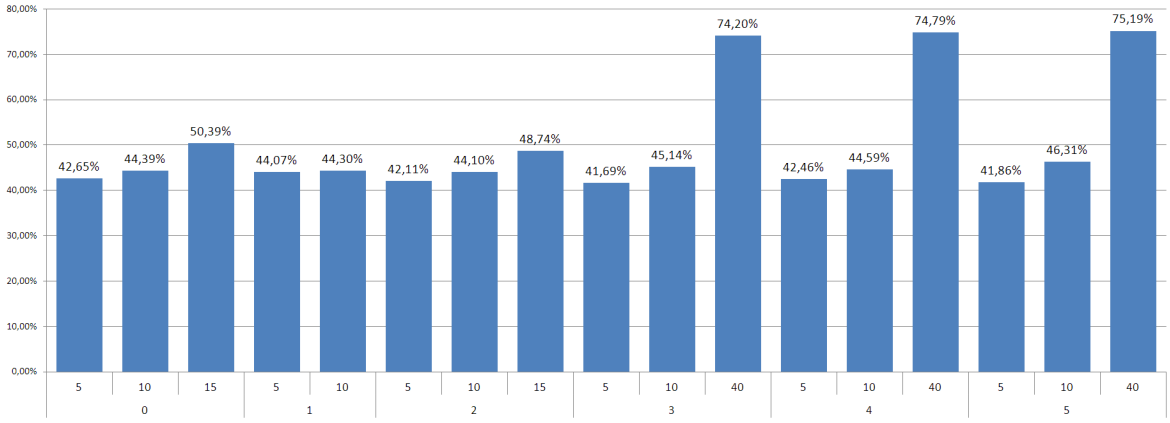


Figure 7.1.: This shows the performance, grouped by the duration and the offset. The offset was tested for 0 to 5 minutes (x-axis outer group) and the duration from 5 to 40 minutes (x-axis inner group). The performance shown is the maximum performance achieved for the group (best feature set).

This section will briefly try to explain the connection between offset and duration in more detail. The diagram in figure 7.1 shows the maximum performance achieved, grouped by the minutes duration used for labeling and prediction (inner labels x-axis) and the minutes of offset (outer x-axis) between the articles release date and the labeling and prediction.

Discussion

It seems that a high performance is achieved, when there is a certain offset between the release date of the article and the actual prediction. Furthermore, the unevenly distributed duration time should not lead to any confusion: All results were dropped which have either a lower accuracy than an accuracy which could be achieved with an "all hold" recommendation, or which have an accuracy below a random classifier (33%). From the results can be seen that the performance is best at the upper level of the range that was tested.

7.3.2. Feature Set vs. Performance

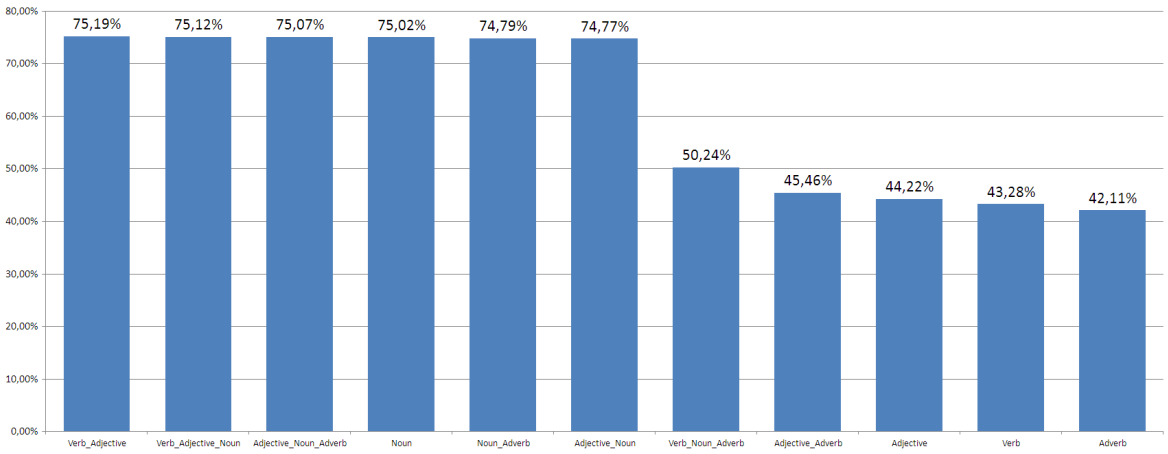


Figure 7.2.: This shows the maximum performance of each feature set. Along the x-axis the feature-types are shown which were used to achieve the performance.

One of the main questions was, which kinds of features should be used to make predictions on the Foreign Exchange market. This was tested in a batch run where all possible combinations of feature-sets were tested with a training set from January 2012 and a evaluated on a test set from February 2012. In the diagram 7.2, the features sets are depicted, ordered by their maximum achieved performance.

The table 7.3 shows how many results are beyond the 75% hold-all recommendation threshold in each of the different types of feature sets. What can be seen is that using only such words as features, which were POS tagged as adverbs leads to significantly less results, due to the fact that probably an "all hold" recommendation would achieve a better result than the actual performance of the prediction. But the picture is incomplete without looking at the feature set and putting all three variables, the duration, the offset and the features, into proper perspective. To gain more insights about the features, they will be described next, whereas the whole overall performance will be described thereafter.

Table 7.3.: The number of results from each of the different sets of feature types.

Feature Types	Number of Results
Adjective	12
Adjective_Adverb	12
Adjective_Noun	13
Adjective_Noun_Adverb	15
Adverb	7
Noun	14
Noun_Adverb	13
Verb	10
Verb_Adjective	15
Verb_Adjective_Noun	15
Verb_Noun_Adverb	13
Total	139

Discussion of the Feature Set

No clear evidence can be found which features should be used to make predictions. Clearly, almost every of the top performing feature sets contain adjectives, but a hard evidence, in the sense of "the more features the better" cannot be found and reasoned with hard facts. It seems that adjectives carry a lot of information, which is no surprise. In contrast, it was not expected that verbs would be within the main contributers for good results. What seems to get through by looking at the table 7.3 is that the more features are in the training set, the more predictions are actually performing better than a classifier which would always give "hold" recommendations.

Taking only the features and the performance into consideration is not enough to get the whole picture. What was also tested was the time for which the prediction is most effective, starting from the offset, between the article's release time and the time where the prediction is in place to the duration of the prediction. To get the total context, the performance will be put in a perspective to the features and the time of the impact.

7.3.3. Overall Performance

To get a better overview, in the two diagrams[1] 7.3 and 7.4, the two most important variables during the batch run are compared: the duration and the different kinds of features which are used in comparison to their impact on the performance. The duration also represents the time the market fully reflects the news, due to the fact that it was inherently labeled with exchange rate movements duration (see section 5.5.7 for further details). The kinds of feature show which set of feature carried the information that results in the performance. The results in these two graphics also represent the overall performance, which is the most interesting and which was measured during the batch run. The best performance could be measured with an accuracy of 75.19% with a feature set containing only verbs and adjectives with 40 minutes duration used for the linear regression labeling process. Furthermore, what cannot be shown in this graphic additionally, but can be seen in the table in appendix A.3, the offset is 5 minutes from each articles release time. Two things seem to be obvious, when looking at the results: Firstly, the longer the duration, the higher the accuracy. Secondly, the less features are used the worse the prediction performance.

Discussion of the Overall Performance

Looking at the resultset, it seems evident that a larger "duration-time" used for labeling and prediction leads to a greater performance. Unfortunately that seems to saturate at around 40 minutes duration and 5 minutes offset, which is the also the maximum scale that was tested, since after that mostly an "all hold" recommendation was given. It was expected that the highest performance will be achieved with a 20 minutes duration and an immediate impact after the release of the article. This would reflect what was seen on the Stock market: the news is reflected in the market price within 20 minutes. In other words, not having the highest accuracy for a duration of 20 minutes and an offset of 0 minutes is interesting and yields to the conclusion that due to the large operational area of the Forex market, the market needs more time to fully reflect the news. Even more, eventually because of the high amount of information running into the economic

[1] Sidenote: although a bar-chart would be scientifically correct in that context, the area-chart was chosen over the bar-chart to improve readability of the data.

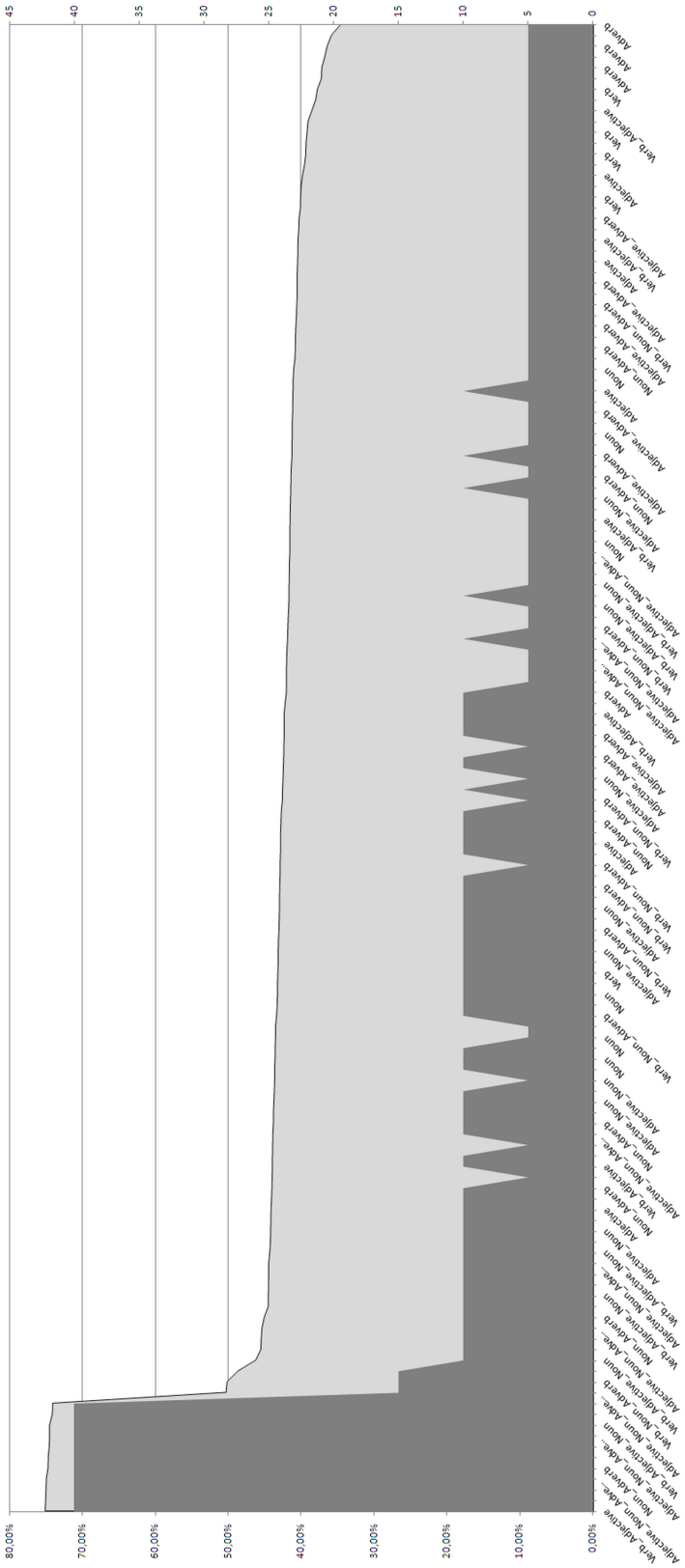


Figure 7.3.: This shows the accuracy (light-gray) versus the amount of minutes taken for the linear regression (dark-gray), in comparison to the used features (labels along the x-axis). Along the x-axis the feature-types are shown which were used to achieve the performance. What can be seen, and is the most important, is that the more values are taken into account for the linear regression, the higher the accuracy. This is further discussed in 7.3.3.

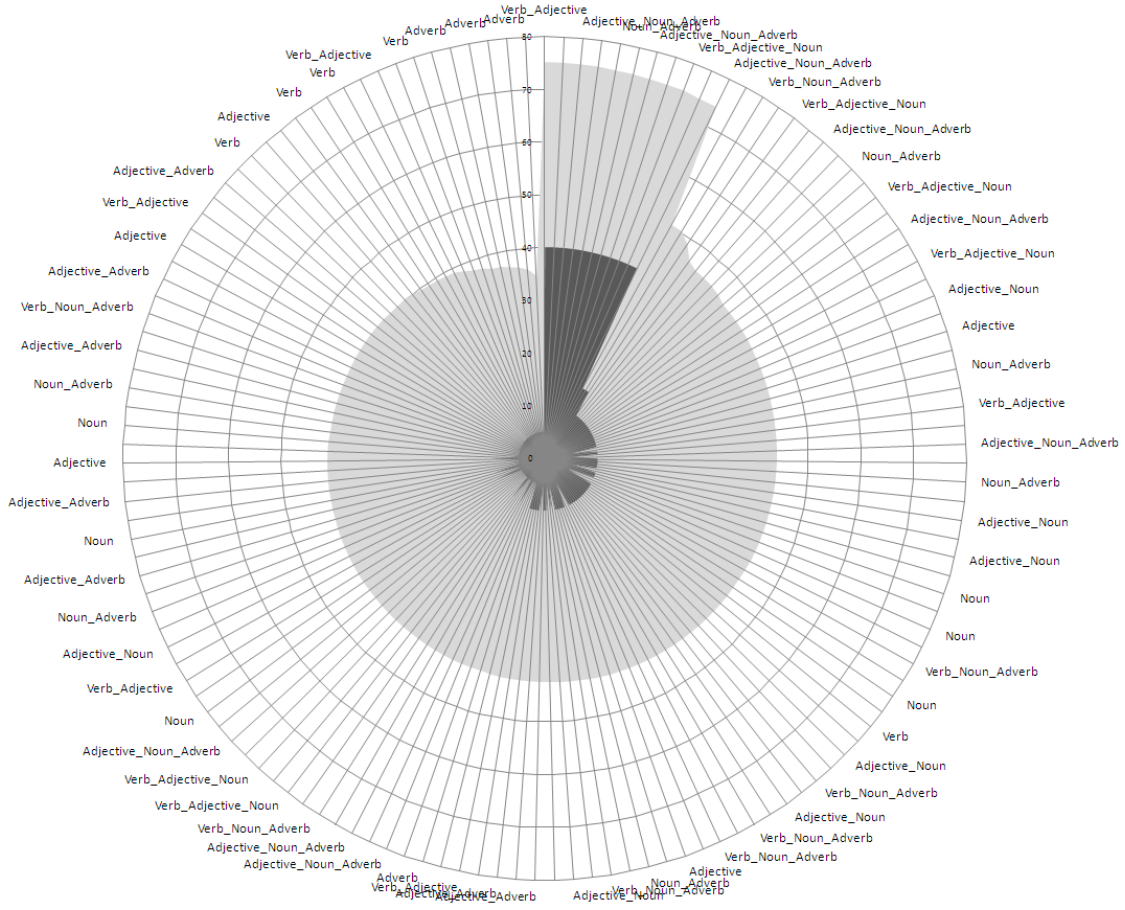


Figure 7.4.: This shows the comparison between the accuracy (light-gray) versus the duration (dark-gray) for the feature types which were used to achieve the accuracy. Labeled are the types of words which were used as features for the prediction task. It shows that, while groups of features increase the performance, also the duration increases the performance. More details are given in 7.3.3.

reflection the Forex market presents, it is possible that news which is released in the time of the prediction influences the market in an opposite way. At any rate, whatever causes the effect would be pure speculation at this point. From the results it can be seen that, on average, after 40 minutes it seems that the market has fully incorporated the information in the news and after that the news will also not lead to more impact.

What is also surprising is that within the top five performing parameter sets, the duration is the driving part, not the feature set: The highest performance was achieved with a mix between verbs and adjectives, the third highest performance was achieved with nouns and adverbs. There are several possible reasons for that. Having only articles in the article set that were released during the main European trading hours, it seems to be possible that also only market driving news regarding the European economy was released. This makes it reasonable that either nouns or adjectives are enough to predict with a relatively high accuracy the next exchange rate movements. Another reason is that actually nouns and adjectives are carrying a lot of information, which could lead to a *good* performance. Anything further along is also again pure speculation and will not be reasoned therefore.

This section presented the results and insights which were gained from the results of the prediction made on the Forex market. In the next section the research questions, which were stated in section 5.6, will be addressed.

7.4. Research Questions

Having the Filtering and the Forex sentiment prediction discussed, leads to the last part: Answering the questions, which were asked during stating the initial idea (see 5.6).

- *Is it possible to determine "interesting" news from the news corpus on an automated basis on the fly?*

  This question could be answered party with yes and partly with no: It seems that there are almost no "interesting" news lost. So interesting news articles are not accidentally dropped by the filter, but the filtering module also does not remove every financially uninteresting news article. Theoretically, it could be achieved with a stronger filter. So the question could be answered with: yes, determining "interesting" news from a mixed corpus is possible with a stronger filter.

- *Across all combinations which are possible with a bag-of-words approach, which is the best feature set?*

  It seems that adjectives and nouns carry a lot of information, as expected. Also, and very surprisingly, verbs in combination with adjectives seem to be a good choice, although it is highly indicated that it is an outlier due to the fact that articles in a short period of time were tested. Avoided should be adjectives, verbs and adverbs alone, since they miss the context. Without any doubt a logical good choice is the combination between nouns, adjectives and verbs.

- *Is there a measurable delay between the news release and the impact until the news is fully reflected by the market?*

  In related work, it is stated that news is usually reflected in the market price within 20 minutes. No evidence could be found that news is reflected within the first 20 minutes after an articles release date. Instead, it seems that the Forex market takes around 40 minutes to fully reflect the news, or eventually it takes even more time to incorporate all information in a news article into the market price. Unfortunately that can not be tested very easily, since a lot of information is continuously flowing into the price point of the exchange rates. In the Stock market this can be separated more clearly, since one Stock represents only the value of one company, where there is no continuous news flow within seconds. Having around 4100 different articles from one month detected as financially interesting, only from the main European trading hours (released in half a day), means around 4 articles per minute are contributing to the exchange rates.

- *What if the polarity regarding an entity changes over the time?*

  Unfortunately this could not be tested, since the lack of an appropriate amount of articles. It will be addressed in the chapter outlook (9).

8. Lessons Learned

Having the idea for such a system is a first good step in the right direction. Actually doing research, reading trough the theory, planning, prototyping and solving each of the underlaying problems takes the whole work to another level. Scientific research, on the one hand, requires to work very accurately and does not leave a lot room for mistakes. Still having fun programming, luckily, leaves still a lot room for creativity. Bringing both, the personal passion for programming and thrive for "new things" and the scientific work together into this thesis was for sure one of the biggest challenges. The lessons learned while doing so is what is going to be shared here in the following order: First the lessons regarding the literature will be shared. This immediately influenced the programming part, which is content of the second section. Lastly it will be shared what was learned regarding the evaluation.

8.1. Literature

Financial markets were, since they exist, topic of interest for research. How researcher are targeting the topic is different: One part is just researching how, when and which information has an impact on the market, the other big part of research is to use information to make predictions on the market. In general, how predictions are tried to be made is pretty equal for the Stock and for the Forex market. The majority is, except of some systems, using supervised learning methods. What is interesting, even with very noisy and "bad" data from social networks and forum posts in slang language, it is possible to achieve pretty good results using support vector machines.

A lot of different systems were used to make predictions. On the one hand, as already mentioned, are the supervised learning systems. With these systems, the main point is to try to get as good data as possible, for both, training and classification. So a big portion of time is spend on creating a great feature vector within prediction systems using supervised learning methods. In the beginning it were the Naïve Bayesian networks, which delivered outstanding results. After a while they made the way for Support Vector Machines, while there are a vast amount of different kernels available now and it is still not sure which one is *the best*, although a simple and fast linear kernel seems to be the recommended choice. On the other hand to supervised learning,

there is great progress within unsupervised learning methods: lexical classifier and clustering methods are used to try to classify sentiment and content in a way that it can be used to predict the outcome.

Unfortunately, after having read through the theory, making predictions on financial markets in a very challenging issue, but not a hopeless cause. The literature offers a lot of information about the Stock market, about the impact of news, about the workflow, about indicated insider trading, and so on. The Stock market is great playground for research, since it is regulated and one can focus on a single Stock and see how information is impacting the market price. The Forex market, on the other hand, is a highly volatile market, especially the currency pairs Euro - US Dollar. It is not easy to make simple analytics on that market about information impact due to the vast amount of different factors flowing into the market price. There are some big players, plenty of smaller and thousands of "mini" traders on the market, and everyone has its own way of trading, and even more, everyone influences the market due to the trading activity. That the Forex market is not yet sufficiently researched shows eventually the importance to spend more time doing analytics on the Forex market, rather than on the Stock market.

8.2. Programming

During writing such a prototype a lot of different kinds of "worlds" are touched: Crawling requires knowledge of multimedia information systems, HTML, Javascript. Saving this crawled information requires fundamental knowledge of systems and libraries like Lucene, or SQL. Having the need for continuous filtered data leads to creative ideas like clustering inspired filtering, which requires understanding the principle of unsupervised learning. Learning about machine learning and features requires a tremendous amount of research what was already done in related work, to find out it is fundamentally not enough. A wise saying is "know your data" and, for example, an evidence which features should used best could never be found. Furthermore, the Forex market is not sufficiently studied regarding the impact of news, which requires the testing and evaluation of different kinds of parameters, like when and how news sentiment impacts the market.

Bringing all this together lets one stumble across several challenges which have to be solved: Firstly, "things" that are created by humans tend often to be not perfect, like broken HTML Syntax, weather it was intended to be broken or not. It is exactly this, what is not expected in advance and turns out to be a huge challenge afterwards. Parsers are not working as expected, crawlers were never meant to parse this kind of syntax and have to be adjusted or have to be self written. Secondly, always when

someone thinks it is working as expected, a new challenge[1] waits to be solved, if it is the way data is saved, how the whole process can be accelerated, or which library to use.

In theory making predictions, in general, seems to be a not very challenging task: One just has to get labeled training samples, learn a classifier and evaluate everything on a test set. As mentioned before, the first step is: There is no such labeled training set, no such corpus, which means, such a corpus has to be created by hand. Then there is an arbitrary amount of uncertainties: When does news impact the market? Which news impacts the market? For how long does news impact the market? The list goes on: Which part of the article is the decisive factor for the impact? Can that be modeled with software? Furthermore, due to the fact that the Forex market is so incredible fast moving, a system has to be created that works in parallel: An article has to be dispatched in as few minutes as possible. This includes the time for crawling, filtering, the creation of a feature vector, the classification and then the opening a position on the market. The idea of modules was born. Each of the functions of the program was moved into a separate module, where the communication is only done through the database. This requires a, even though simplistic, locking module to avoid race conditions. That meant, all of the modules can run on separate machines, not interfering, even they are using 100% of the available CPU time.

Having finally a crawler created, having all libraries and having put together everything, from the filtering using clustering, text processing using a maximum entropy framework, labeling using piecewise linear regression on exchange rate movements, and the final classification task, using a support vector machine, unfortunately, is just the beginning of the actual research. A tremendous amount of time flows into scientific work, which will never be enough. Having spend this time, leads to the last part, the evaluation.

8.3. Evaluation

The goal of the thesis was primarily to overcome gaps in the research of the Forex market and to pave the way for making predictions on the fly on the Forex market in the future. While for private development a quick forward movement is possible, it is time and resource consuming to gather all results in matter appropriate for scientific research. On the other hand, having such facts together means having hard evidence if the outcome is then as good as expected and one can prove that the methods used are working within their boundaries.

One part focused on the filtering, due to the problem of having a mixed news corpus.

[1] http://www.huffingtonpost.com/donna-labermeier/problems-or-challenges-wh_b_5112932.html

The overall accuracy of about only 70% is, although not bad, surprising, since it was actually expected that the filtering would work very well, far better than 70%. First intermediate results showed great results. Having additional noise introduced in the system obviously reduced the overall performance. Nevertheless, the filtering reduced the noise and, at least, did not remove a lot of information which is actually important.

Very diverse results were achieved with the prediction module. It was not expected *at all* that there is a low prediction performance within the first 20 to 30 minutes after an article release. This leaves a lot of open research questions for future work, where the focus should be laid on the interaction between a better filtering and the effects on the prediction performance. All in all can be concluded that evaluation is not only a necessary part, but leads to interesting and surprising results and is the most exciting part of the work.

9. Conclusion and Outlook

Due to the fact that the amount of available news is tremendous, it is inevitable that an automatic solution is needed and required to screen through the documents. Traders simply cannot handle this huge amount of data continuously throughout the day. Automatic methods, such as filtering, could lead to alert services, which inform traders about interesting articles. Furthermore powerful machine learning and text processing libraries can extract features from the articles with an impressive performance. Having those features, it is a next logical step to rationalize the trader and use an automatic trading system. The first steps to automate the whole workflow was successfully tried as a first prototype within this thesis. The following section will summarize again the different chapters of the thesis, while in the section thereafter the limitations are outlined and ideas for future improvements are given.

9.1. Summary

Chapter 1 starts with a brief introduction into the topic of Foreign Exchange market prediction. Several motivations are outlines, why predictions should be made on news rather than historical exchange rate movements.

Chapter 2 introduces the reader fundamentally into the Foreign Exchange market. It is explained how the Forex market emerged historically, how the workflow is and who are the main participants. Furthermore some important concepts, how to make an order, what is a PIP and how to use a leverage is explained. On top of that, the different ways of how news can impact the Forex market is researched.

In chapter 3 the theoretical methods for a sentiment analysis are explained. First the question is answered what sentiment is, then different application and goals of sentiment analysis are demonstrated. To do any sentiment analysis task, it is imperative to have a news corpus to do the tasks on. Therefore all necessary steps, from getting a news corpus to processing it in an appropriate way are explained next. This includes the downloading, tokenization, POS tagging and named entity recognition. At the end the methods for sentiment prediction are explained, which includes all machine learning tasks from creating a feature vector to performing an actual classification.

Chapter 4 related work is studied. From the work presented, the ideas are derived which are defined in the following chapter. This includes the features used to build the feature vector as well as using support vector machines for the actual classification task.

A news system, based on the findings of the related work, is modeled in chapter 5. First the main idea, based on the findings of the related work is outlines, which is then modeled into more detail for every aspect of the system. Articles should be downloaded continuous basis. A new filtering approach is presented, which should filter the articles using a clustering method, on the fly. The news are labeled automatically based on historical exchange rate movements and, furthermore, different textual features are extracted from the articles. Finally a batch testing method is introduced to test all different kinds of feature combinations on a distinct test set. All of the functional requirements should be implemented as modules which are running as daemons and are only interconnected via a database.

In chapter 6 the actual prototype is built. The different modules are described in more detail and how they are working together as a daemon. Furthermore the local class usage of the different parts of the prototype is described.

In chapter 7, the evaluation of the results is done. Two main functions were evaluated. Firstly the filtering method, which tries to discard articles which are not from the financial domain. While the filtering is not having a lot of false positives (not removing a lot of articles which are from the financial domain), it could be stronger, because, unfortunately, it does not remove a large part which are absolutely uninteresting. In the second part, the actual Forex prediction based on news sentiment is evaluated. Several things were considered, where the most important outcome is the connection between the time the market fully reflects the news and the performance. Also different feature sets for the feature vector were tested, where the most surprising outcome is that a combination of verbs and adjectives is ranging on the top of the results.

In the next section some limitations of the work in this thesis will be outlined and ideas for a future work will be given.

9.2. Outlook

Having the idea and the prototype for a system which could potentially make predictions on the Forex market is just the first step. One of the main contributors for a good prediction system the underlaying data it is trained with. Unfortunately in

this thesis the underlaying data was a mixed corpus of articles from various topics, full of duplicates and uninteresting content. Therefore it is beneficial to remove the filtering module and use financially interesting data instead. If that is not possible, one of the big contributions in a future work could be a more robust filtering module. Different unsupervised learning methods could be incorporated, thesauri and eventually other rule based classifier could add more performance to the filtering module.

Having just news data which actually *really* shows an impact on the Foreign Exchange market, it could be beneficial to re-run the part of the prototype, which tests the different feature sets, to get the set with the most impact. Furthermore, it could be beneficial to have data from about a one to two year period, instead of two months. Having that, then the testing method could be changed, that a new classifier is always generated from a previous months new corpus and compared to a classifier which is older, to test if sentiment really changes over the time.

One last improvement, which is not be incorporated in this thesis, is a market simulation. The amount of articles in this thesis makes it impossible to run a even remotely realistic market simulation due to the high amount of articles which would be taken under consideration. On average a market order would be placed every four minutes, although the orders would be held for about 40 minutes which makes it highly suspicious for a even simulated market simulation. Therefore, and referencing back to the first paragraph of this section, it is imperative to have good underlaying data.

A. Data Medium

The DVD attached is structured in the following way:

- `/thesis/thesis.pdf`: contains a PDF version of this document.
- `/thesis/references/`: contains all the references mentioned in the Bibliography, as far as available in digital form.
- `/data/db.sql`: contains a SQL dump of the database. The dump was created with mysqldump and can be re-inserted into a new database with `mysql -u username -p databasename < db.sql`.
- `/data/results.xlsx`: an Excel sheet with the values which were gained out of the database for the results.
- `/source/`: The source code to the project, including all libraries necessary to compile and start the BaseController.

Bibliography

Aggarwal, C. C. & Zhai, C. (2012). A survey of text classification algorithms. In *Mining text data* (pp. 163–222). Springer.

Alisneaky. (2011). *Kernel machines*. Retrieved March 9, 2014, from http://en.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png

Andersen, T. G., Bollerslev, T., Diebold, F. X., & Vega, C. (2007). Real-time price discovery in global stock, bond and foreign exchange markets. *Journal of International Economics*, *73*(2), 251–277.

Bacher, S. & Stuckenschmidt, H. (2012). *Mining unstructured financial news to forecast intraday stock price movements* (Doctoral dissertation).

Baillie, R. T. & McMahon, P. C. (1990). *The foreign exchange market* [Theory and econometric evidence]. Cambridge University Press. Retrieved from http://www.cambridge.org/gb/knowledge/isbn/item1139761/?site_locale=en_GB

Barber, B., Lee, Y.-T., Liu, Y.-J., & Odean, T. (2010). *Do day traders rationally learn about their ability*. Working Paper (September), Graduate School of Business, Columbia University.

Bauwens, L., Omrane, W. B., & Giot, P. (2005). News announcements, market activity and volatility in the euro/dollar foreign exchange market. *Journal of International Money and Finance*, *24*(7), 1108–1125. doi:10.1016/j.jimonfin.2005.08.008

BIS. (2010, April). *Triennial central bank survey - foreign exchange and derivatives market activity in april 2010*. Bank for International Settlements. Retrieved from http://www.bis.org/publ/rpfx10.pdf

Bjønnes, G. H. & Rime, D. (2005). Dealer behavior and trading systems in foreign exchange markets. *Journal of Financial Economics*, *75*(3), 571–605. doi:10.1016/j.jfineco.2004.08.001

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152). ACM.

Chaboud, A., Hjalmarsson, E., Vega, C., & Chiquoine, B. (2009). Rise of the machines: algorithmic trading in the foreign exchange market. *FRB International Finance Discussion Paper*, (980).

Chan, E. (2012, March 23). *High-frequency trading in the foreign exchange market*. Retrieved from http://epchan.blogspot.co.at/2012/03/high-frequency-trading-in-foreign.html

Chang, C.-C. & Lin, C.-J. (2011, May). Libsvm: a library for support vector machines. *ACM Trans. Intell. Syst. Technol. 2*(3), 27:1–27:27. doi:10.1145/1961189.1961199

Chinchor, N., Brown, E., Ferro, L., & Robinson, P. (1999). 1999 named entity recognition task definition. *MITRE and SAIC*.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

DailyFX. (2011, December 12). *Forex education: what time of day should i trade*. Retrieved from http://www.dailyfx.com/forex/education/trading_tips/daily_trading_lesson/2011/12/12/When_is_the_Best_Time_of_Day_to_Trade_Forex.html

DeGennaro, R. P. & Shrieves, R. E. (1997). Public information releases, private information arrival and volatility in the foreign exchange market. *Journal of Empirical Finance*, *4*(4), 295–315. doi:http://dx.doi.org/10.1016/S0927-5398(97)00012-1

Devitt, A. & Ahmad, K. (2007). Sentiment polarity identification in financial news: a cohesion-based approach. In *Proceedings of the 45th annual meeting of the association of computational linguistics*. doi:10.1.1.143.7157

Dictionaries, O. (2013, March 9). *Definition of rational expectations hypothesis in oxford dictionaries (british & world english)*. Retrieved from http://oxforddictionaries.com/definition/english/rational+expectations+hypothesis

Dragut, E., Wang, H., Yu, C., Sistla, P., & Meng, W. (2012). Polarity consistency checking for sentiment dictionaries. In *Proceedings of the 50th annual meeting of the association for computational linguistics: long papers - volume 1* (pp. 997–1005). ACL 12. Jeju Island, Korea: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2390524.2390658

Evans, C., Pappas, K., & Xhafa, F. (2013). Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange

speculation. *Mathematical and Computer Modelling*, pages. doi:10.1016/j.mcm.2013.02.002

Fama, E. F. (1970). Efficient capital markets: a review of theory and empirical work*. *The Journal of Finance*, *25*(2), 383–417. doi:10.1111/j.1540-6261.1970.tb00518.x

Fama, E. F. (1991). Efficient capital markets: ii. *The Journal of Finance*, *46*(5), 1575–1617. doi:10.1111/j.1540-6261.1991.tb04636.x

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008, June). Liblinear: a library for large linear classification. *J. Mach. Learn. Res. 9*, 1871–1874. Retrieved from http://dl.acm.org/citation.cfm?id=1390681.1442794

Fawcett, T. & Provost, F. (1999). Activity monitoring: noticing interesting changes in behavior. In *Proceedings of the fifth acm sigkdd international conference on knowledge discovery and data mining* (pp. 53–62). KDD '99. San Diego, California, USA: ACM. doi:10.1145/312129.312195

Forex Magnates. (2011). *Us forex brokers profitability report for q4 2011 shows steep drop in number of us accounts amid increase in traders' profitability.* Retrieved October 22, 2012, from http://forexmagnates.com/forex-brokers-profitability-report-q4-2011-shows-steep-drop-number-accounts-increase-traders-profitability/

Fox, C. (1989, September). A stop list for general text. *SIGIR Forum*, *24*(1-2), 19–21. doi:10.1145/378881.378888

Fung, G. P. C., Yu, J. X., & Lu, H. (2005). The predicting power of textual information on financial markets. *IEEE Intelligent Informatics Bulletin*, *5*(1), 1–10.

Fung, G. P. C., Yu, J., & Lam, W. (2003). Stock prediction: integrating text mining approach using real-time news. In *Computational intelligence for financial engineering, 2003. proceedings. 2003 ieee international conference on* (pp. 395–402). doi:10.1109/CIFER.2003.1196287

Fung, G., Yu, J., & Lam, W. (2002). News sensitive stock trend prediction. In M.-S. Chen, P. Yu, & B. Liu (Eds.), *Advances in knowledge discovery and data mining* (Vol. 2336, pp. 481–493). Lecture Notes in Computer Science. Springer Berlin Heidelberg. doi:10.1007/3-540-47887-6_48

Geisser, S. (1993). *Predictive inference.* CRC Press.

Ghahramani, Z. (2004). Unsupervised learning. In O. Bousquet, U. Luxburg, & G. Rätsch (Eds.), *Advanced lectures on machine learning* (Vol. 3176, pp. 72–112).

Lecture Notes in Computer Science. Springer Berlin Heidelberg. doi:10.1007/978-3-540-28650-9_5

Giancarlo, G. (2002). *International finance and open-economy macroeconomics*. Springer Science+Business Media. Retrieved from http://www.springer.com/economics/macroeconomics/book/978-3-540-43459-7

Gidófalvi, G. & Elkan, C. (2001). Using news articles to predict stock price movements. *Department of Computer Science and Engineering, University of California, San Diego*.

Godbole, N., Srinivasaiah, M., & Skiena, S. (2007). Large-scale sentiment analysis for news and blogs. In *Proceedings of the international conference on weblogs and social media (icwsm)* (Vol. 2).

Goetz, B. (2000). The lucene search engine: powerful, flexible, and free. *JavaWorld. Available http://www. javaworld. com/javaworld/jw-09-2000/jw-0915-lucene. html*.

Hagenau, M., Liebmann, M., Hedwig, M., & Neumann, D. (2012). Automated news reading: stock price prediction based on financial news using context-specific features. In *System science (hicss), 2012 45th hawaii international conference on* (pp. 1040–1049). doi:10.1109/HICSS.2012.129

High Scaleability. (2011, June 27). *Spot rate definition*. Retrieved from http://highscalability.com/blog/2011/6/27/tripadvisor-architecture-40m-visitors-200m-dynamic-page-view.htm

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification.

IBTimesFx. (2011). *5 differences between forex trading and stock trading*. Retrieved from http://au.ibtimes.com/articles/136256/20110420/5-differences-between-forex-trading-and-stock-trading-forex-tips.htm

Informed Traders. (2008, April 16). *Fractional pips - an explanation of pips and fractional pips*. Retrieved from http://www.informedtrades.com/22740-fractional-pips-explanation-pips-fractional-pips.html#post23281

Investopedia. (2013, September 24). *Volatility definition*. Retrieved from http://www.investopedia.com/terms/v/volatility.asp

Investopedia. (2014, January 21). *Algorithmic trading*. Retrieved from http://www.investopedia.com/terms/a/algorithmictrading.asp

James, D. (2012, June 12). *What is the difference between web-crawling and web-scraping?* Retrieved from http://stackoverflow.com/questions/4327392/what-is-the-difference-between-web-crawling-and-web-scraping

Jin, X., Li, Y., Mah, T., & Tong, J. (2007). Sensitive webpage classification for content advertising. In *Proceedings of the 1st international workshop on data mining and audience intelligence for advertising* (pp. 28–33). ADKDD '07. San Jose, California: ACM. doi:10.1145/1348599.1348604

Jindal, N. & Liu, B. (2006). Mining comparative sentences and relations. In *Proceedings of the national conference on artificial intelligence* (Vol. 21, *2*, p. 1331). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Joachims, T. (2006). Training linear svms in linear time. In *Proceedings of the 12th acm sigkdd international conference on knowledge discovery and data mining* (pp. 217–226). ACM.

Kazienko, P. & Adamski, M. (2007). Adrosa—adaptive personalization of web advertising. *Information Sciences*, *177*(11), 2269–2295. doi:10.1016/j.ins.2007.01.002

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. (pp. 1137–1143). Morgan Kaufmann.

Koppel, M. & Shtrimberg, I. (2006). Good news or bad news? let the market decide. In J. G. Shanahan, Y. Qu, & J. Wiebe (Eds.), *Computing attitude and affect in text: theory and applications* (Vol. 20, pp. 297–301). The Information Retrieval Series. Springer Netherlands. doi:10.1007/1-4020-4102-0_22

Laakkonen, H. & Lanne, M. (2010). Asymmetric news effects on exchange rate volatility: good vs. bad news in good vs. bad times. *Studies in Nonlinear Dynamics & Econometrics*, *14*(1).

Lai, K., Yu, L., & Wang, S. (2005). A neural network and web-based decision support system for forex forecasting and trading. In Y. Shi, W. Xu, & Z. Chen (Eds.), *Data mining and knowledge management* (Vol. 3327, pp. 243–253). Lecture Notes in Computer Science. Springer Berlin Heidelberg. doi:10.1007/978-3-540-30537-8_27

Lauber, M., Gütl, C., & Liu, W. (2012). *Stock market sentiment analysis* (Doctoral dissertation).

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000a). Language models for financial news recommendation. In *Proceedings of the ninth international conference on information and knowledge management* (pp. 389–396). CIKM '00. McLean, Virginia, USA: ACM. doi:10.1145/354756.354845

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000b). Mining of concurrent text and time series. In *Kdd-2000 workshop on text mining* (pp. 37–44). Citeseer.

Learn to trade the Market. (2014, January 25). *Low frequency vs. high frequency forex trading*. Retrieved from http://www.learntotradethemarket.com/forex-articles/low-frequency-vs-high-frequency-forex-trading

Li, X., Wang, C., Dong, J., Wang, F., Deng, X., & Zhu, S. (2011). Improving stock market prediction by integrating both market news and stock prices. In A. Hameurlain, S. Liddle, K.-D. Schewe, & X. Zhou (Eds.), *Database and expert systems applications* (Vol. 6861, pp. 279–293). Lecture Notes in Computer Science. Springer Berlin Heidelberg. doi:10.1007/978-3-642-23091-2_24

Lin, M.-C., Lee, A. J. T., Kao, R.-T., & Chen, K.-T. (2008, October). Stock price movement prediction using representative prototypes of financial reports. *ACM Trans. Manage. Inf. Syst. 2*(3), 19:1–19:18. doi:10.1145/2019618.2019625

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993, June). Building a large annotated corpus of english: the penn treebank. *Comput. Linguist. 19*(2), 313–330. Retrieved from http://dl.acm.org/citation.cfm?id=972470.972475

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., ... Schasberger, B. (1994). The penn treebank: annotating predicate argument structure. In *Proceedings of the workshop on human language technology* (pp. 114–119). HLT '94. Plainsboro, NJ: Association for Computational Linguistics. doi:10.3115/1075812.1075835

Marketwatch. (2010, September 1). *Daily foreign-exchange turnover hits $4 trillion*. Retrieved from http://articles.marketwatch.com/2010-09-01/industries/30787868_1_share-japanese-yen-turnoverl

Marshall, A., Musayev, T., Pinto, H., & Tang, L. (2012). Impact of news announcements on the foreign exchange implied volatility. *Journal of International Financial Markets, Institutions and Money*, *22*(4), 719–737. doi:http://dx.doi.org/10.1016/j.intfin.2012.04.006

Mejova, Y. (2009). Sentiment analysis: an overview. Retrieved from http://www.cs.uiowa.edu/~ymejova/publications/CompsYelenaMejova.pdf

Melvin, M. & Yin, X. (2000). Public information arrival, exchange rate volatility, and quote frequency. *The Economic Journal*, *110*(465), 644–661.

Miah, F., Hassan, M. K., & Rahman, A. (2004). An empirical examination of rational expectations hypothesis in the foreign exchange market. *Cameron University*. Retrieved from http://www.cameron.edu/uploads/QD/dy/QDdyBJPfYT4PO5-Sg1Gkvg / Vol - 11 - An - Emperical - Examination - Of - Rational - Expectations - Hypothesis.pdf

Mitkov, R. (2003). *The oxford handbook of computational linguistics*. Oxford Handbooks in Linguistics. OUP Oxford. Retrieved from http://books.google.at/books?id=yl6AnaKtVAkC

Mittermayer, M.-A. [M.-A.] & Knolmayer, G. [G.F.]. (2006). Newscats: a news categorization and trading system. In *Data mining, 2006. icdm '06. sixth international conference on* (pp. 1002–1007). doi:10.1109/ICDM.2006.115

Mittermayer, M.-A. [Marc-André] & Knolmayer, G. [Gerhard]. (2006). *Text mining systems for market response to news: a survey*. Institut für Wirtschaftsinformatik der Universität Bern.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. MIT Press.

Niederhoffer, V. (1971, April). The analysis of world events and stock prices. *The Journal of Business*, *44*(2), 193–219.

Osiński, S., Stefanowski, J., & Weiss, D. (2004). Lingo: search results clustering algorithm based on singular value decomposition. In *Intelligent information processing and web mining* (pp. 359–368). Springer.

Pal, S. (2011a, August 4). *An uima sentence annotator using opennlp*. Retrieved from http://sujitpal.blogspot.co.at/2011/04/uima-sentence-annotator-using-opennlp.html

Pal, S. (2011b). *Computing document similarity using lucene term vectors*. Retrieved April 12, 2014, from http://sujitpal.blogspot.ch/2011/10/computing-document-similarity-using.html

Pang, B. & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, *2*(1-2), 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the acl-02 conference on empirical methods in natural language processing - volume 10* (pp. 79–86). EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1118693.1118704

Pareek, S. K., Saha, A. K., & Ghosh, B. K. (2011). Information and communication technology in foreign exchange management. In *Communication and industrial application (iccia), 2011 international conference on* (pp. 1–4). IEEE.

Peramunetilleke, D. & Wong, R. K. (2002, January). Currency exchange rate forecasting from news headlines. *Aust. Comput. Sci. Commun. 24*(2), 131–139. doi:10.1145/563932.563921

Peter Buch. (2008). *Graphic showing the maximum separating hyperplane and the margin.* Retrieved March 7, 2014, from http://en.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png

Premanode, B., Vonprasert, J., & Toumazou, C. (2013). Prediction of exchange rates using averaging intrinsic mode function and multiclass support vector regression. *Artificial Intelligence Research, 2013, Vol. 2, No. 2*, *2*(2), 47–81. doi:10.5430/air.v2n2p47

Princeton WordNet 3.1. (2012). *Definition of sentiment.* Retrieved from http://wordnetweb.princeton.edu/perl/webwn?s=sentiment

Probe-meteo.com. (2013). *Scheme of a the 2 kinds of basic candlestick chart.* Retrieved March 27, 2014, from http://en.wikipedia.org/wiki/File:Candlestick_chart_scheme_03-en.svg

Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. doi:10.1023/A:1022643204877

Ratnaparkhi, A. et al. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing* (Vol. 1, pp. 133–142). Philadelphia, PA.

Ratnaparkhi, A. (1998). *Maximum entropy models for natural language ambiguity resolution* (Doctoral dissertation, University of Pennsylvania).

Robertson, C. S. (2008). *Real time financial information analysis* (Doctoral dissertation, Queensland University of Technology). Retrieved from http://eprints.qut.edu.au/16609/

Robertson, C. S., Geva, S., & Wolff, R. (2006). What types of events provide the strongest evidence that the stock market is affected by company specific news? In *Proceedings of the fifth australasian conference on data mining and analystics - volume 61* (pp. 145–153). AusDM '06. Sydney, Australia: Australian Computer Society, Inc. Retrieved from http://dl.acm.org/citation.cfm?id=1273808.1273828

Robertson, C. S., Geva, S., & Wolff, R. (2007a). Can the content of public news be used to forecast abnormal stock market behaviour? In *Data mining, 2007. icdm 2007. seventh ieee international conference on* (pp. 637–642). doi:10.1109/ICDM.2007.74

Robertson, C. S., Geva, S., & Wolff, R. (2007b). News aware volatility forecasting: is the content of news important? In *Proceedings of the sixth australasian conference on data mining and analytics - volume 70* (pp. 161–170). AusDM '07. Gold Coast, Australia: Australian Computer Society, Inc. Retrieved from http://dl.acm.org/citation.cfm?id=1378245.1378267

Robertson, C. S., Geva, S., & Wolff, R. (2007c). *The intraday effect of public information: empirical evidence of market reaction to asset specific news from the us, uk, and australia*. doi:10.2139/ssrn.970884

Rosenstreich, P. (2005). *Forex revolution: an insider's guide to the real world of foreign exchange trading*. Retrieved from http://www.ftpress.com/store/forex-revolution-an-insiders-guide-to-the-real-world-9780131486904

Sager, M. J. & Taylor, M. P. (2006). Under the microscope: the structure of the foreign exchange market. *International Journal of Finance & Economics*, *11*(1), 81–95. doi:10.1002/ijfe.277

Schulz, A., Spiliopoulou, M., & Winkler, K. (2003). Kursrelevanzprognose von ad-hoc-meldungen: text mining wider die informationsüberlastung im mobile banking. In W. Uhr, W. Esswein, & E. Schoop (Eds.), *Wirtschaftsinformatik 2003/band ii* (pp. 181–200). Physica-Verlag HD. doi:10.1007/978-3-642-57445-0_10

Schumaker, R. P. & Chen, H. (2006). Textual analysis of stock market prediction using financial news. In *Americas conference on information systems*.

Schumaker, R. P. & Chen, H. (2008). Evaluating a news-aware quantitative trader: the effect of momentum and contrarian stock selection strategies. *Journal of the American Society for Information Science and Technology*, *59*(2), 247–255. doi:10.1002/asi.20739

Schumaker, R. P. & Chen, H. (2009, March). Textual analysis of stock market prediction using breaking financial news: the azfin text system. *ACM Trans. Inf. Syst. 27*(2), 12:1–12:19. doi:10.1145/1462198.1462204

Schumaker, R. P. & Chen, H. (2010). A discrete stock price prediction engine based on financial news. *Computer*, *43*(1), 51–56. doi:10.1109/MC.2010.2

Schumaker, R. P., Zhang, Y., Huang, C.-N., & Chen, H. (2012). Evaluating sentiment in financial news articles. *Decision Support Systems*, *53*(3), 458–464. doi:http://dx.doi.org/10.1016/j.dss.2012.03.001

Sebastiani, F. (2002, March). Machine learning in automated text categorization. *ACM Comput. Surv. 34*(1), 1–47. doi:10.1145/505282.505283

Shalev-shwartz, S., Singer, Y., & Srebro, N. (2007). Pegasos: primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on machine learning (icml-07)* (pp. 807–814).

Silva, C. & Ribeiro, B. (2003, July). The importance of stop word removal on recall values in text categorization. In *Neural networks, 2003. proceedings of the international joint conference on* (Vol. 3, 1661–1666 vol.3). doi:10.1109/IJCNN.2003.1223656

Simon, P. (2013). *Too big to ignore: the business case for big data.* John Wiley & Sons.

Singhal, A. (2001). Modern information retrieval: a brief overview. *IEEE Data Eng. Bull. 24*(4), 35–43.

Spiliopoulou, M., Schulz, A., & Winkler, K. (2003). Text mining an der börse: einfluss von ad-hoc-mitteilungen auf die kursentwicklung. *Data Mining und Statistik in Hochschule und Wirtschaft, Shaker, Aachen*, 215–228.

Stack Exchange: Quantitative Finance. (2011, October 23). *What is a medium to low frequency trading strategy and why is it less hyped?* Retrieved from http://quant.stackexchange.com/questions/2218/what-is-a-medium-to-low-frequency-trading-strategy-and-why-is-it-less-hypedl

Taylor, A., Marcus, M., & Santorini, B. (2003). The penn treebank: an overview. In A. Abeillé (Ed.), *Treebanks* (Vol. 20, pp. 5–22). Text, Speech and Language Technology. Springer Netherlands. doi:10.1007/978-94-010-0201-1_1

Teare, K. (2006, March 26). *Is scraping and crawling stealing?* Retrieved from http://www.teare.com/2006/03/26/is-scraping-and-crawling-stealing/

Tetlock, P. C. (2007). Giving content to investor sentiment: the role of media in the stock market. *The Journal of Finance*, *62*(3), 1139–1168. doi:10.1111/j.1540-6261.2007.01232.x

Thomas, J. D. (2003). *News and trading rules* (Doctoral dissertation, Carnegie Mellon University).

Thomas, J. D. & Sycara, K. (2000). Integrating genetic algorithms and text learning for financial prediction. *Data Mining with Evolutionary Algorithms*, 72–75.

Tjong Kim Sang, E. F. & De Meulder, F. (2003). Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on natural language learning at hlt-naacl 2003 - volume 4* (pp. 142–147). CONLL '03. Edmonton, Canada: Association for Computational Linguistics. doi:10.3115/1119176.1119195

Toshchakov, I. (2006). *Beat the odds in forex trading*. John Wiley & Sons, Inc., Hoboken.

Turney, P. & Littman, M. L. (2002). Unsupervised learning of semantic orientation from a hundred-billion-word corpus.

Walker, D. J., Clements, D. E., Darwin, M., & Amtrup, J. W. (2001). Sentence boundary detection: a comparison of paradigms for improving mt quality. In *Proceedings of the mt summit viii*.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 347–354). HLT '05. Vancouver, British Columbia, Canada: Association for Computational Linguistics. doi:10.3115/1220575.1220619

Wolpert, D. H. & Macready, W. G. (1995). *No free lunch theorems for search*. Technical Report SFI-TR-95-02-010, Santa Fe Institute.

Wüthrich, B., Permunetilleke, D., Leung, S., Lam, W., Cho, V., & Zhang, J. (1998). Daily prediction of major stock indices from textual www data. *HKIE Transactions*, *5*(3), 151–156.

Yi, J., Nasukawa, T., Bunescu, R., & Niblack, W. (2003). Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In *Data mining, 2003. icdm 2003. third ieee international conference on* (pp. 427–434). doi:10.1109/ICDM.2003.1250949

Zack Weinberg. (2012). *Graphic showing how a support vector machine would choose a separating hyperplane for two classes of points in 2d.* Retrieved March 7, 2014, from http://en.wikipedia.org/wiki/File:Svm_separating_hyperplanes_(SVG).svg

Zhou, G. & Su, J. (2002). Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 473–480). ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics. doi:10.3115/1073083.1073163

Zhuangi, L., Jing, F., & Zhu, X.-Y. (2006). Movie review mining and summarization. In *Proceedings of the 15th acm international conference on information and knowledge management* (pp. 43–50). CIKM '06. Arlington, Virginia, USA: ACM. doi:10.1145/1183614.1183625

A. Results

A.1. Filtering

The following two listings were used to get the random samples. The first one only gets samples from the database which are marked as "Forex related", the second one only samples which are marked as "Forex unrelated". The table lists the output at the time of the execution of the query.

```
SELECT a.Article_ID, a.Article_Title FROM Article a
JOIN Link l ON l.Link_ID = a.Link_Link_ID
 WHERE
 TIME(a.Article_Created) <> '13:30:00' AND
 TIME(a.Article_Created) >= '09:00:00' AND TIME(a.Article_Created) <=
    '24:00:00' AND
 TIME(a.Article_Created) BETWEEN '08:30:00' AND '19:00:00' AND
 TIME(a.Article_Created) <> '14:00:00' AND
 DATE_FORMAT(a.Article_Created, '\%w') <> 0 AND
 DATE_FORMAT(a.Article_Created, '\%w') <> 6 AND
 DATE(a.Article_Created) BETWEEN '2012-01-01' AND '2012-02-01'
AND l.Link_Status = 104
ORDER BY RAND()
LIMIT 50
```

```
SELECT a.Article_ID, a.Article_Title FROM Article a
JOIN Link l ON l.Link_ID = a.Link_Link_ID
 WHERE
 TIME(a.Article_Created) <> '13:30:00' AND
 TIME(a.Article_Created) >= '09:00:00' AND TIME(a.Article_Created) <=
    '24:00:00' AND
 TIME(a.Article_Created) BETWEEN '08:30:00' AND '19:00:00' AND
 TIME(a.Article_Created) <> '14:00:00' AND
 DATE_FORMAT(a.Article_Created, '\%w') <> 0 AND
 DATE_FORMAT(a.Article_Created, '\%w') <> 6 AND
 DATE(a.Article_Created) BETWEEN '2012-01-01' AND '2012-02-01'
AND l.Link_Status = 15
ORDER BY RAND()
LIMIT 50
```

Article_ID	Article_Title	Is Forex related?
41092	REG - Turner Funds Plc - Restructuring of the Investment Manager & Promoter	no
62690	Mechel Reports 2011 Operational Results	no
35143	Portugal banks' ECB borrowing edges up in December	yes
28043	SE Asia Stocks-Mostly higher but euro zone woes still a drag	yes
52541	PRECIOUS-Gold retreats as U.S. data curbs euro's rise	yes
64475	Market Research Projects Smartphone Sales at 1 Billion Units by 2016	yes
58117	The King of Cringe-worthy Viral Videos Is Back! Daniel Tosh Returns With the Fourth Season Premiere of "Tosh.0" on Tuesday, January 31 at 10:00 P.M. ET/PT	no
12379	Comment on U.S. Bureau of Labor Statistics Employment Situation Report: Kathy Bostjancic, Director of Macroeconomic Analysis, The Conference Board	no
41011	REG - Martin Currie AbsRet - AGM Results	no
43636	Euro rises 1 percent on day against yen	yes
25074	Neptune to Hold Conference Call	no
61232	Pengrowth Announces 2012 Capital Program	no
28144	UPDATE 2-Sodexo keeps FY goals, Rugby World Cup boosts Q1	no

Table A.1.: The articles are chosen by random from the category "Forex related". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label.

Article_ID	Article_Title	Is Forex related?
65574	CORRECTED-BRIEF-HSBC to sell Thailand retail banking firm	yes
59630	M/I Homes, Inc. Announces Fourth Quarter and Year-End Earnings Webcast	yes
61143	Q+A-Can Europe tax financial transactions without UK?	yes
54563	Tetragon Financial Group Limited (TFG) Announces Update on its Share Repurchase Program	yes
53650	Condor Partners with Discover the World Marketing in Greece and Cyprus	no
66444	REG - JZ Capital Ptnrs Ltd - Total Voting Rights	no
44134	Spain clears short-term debt test, bigger hurdle looms	yes
24607	UPDATE 1-China sets 8 trln yuan 2012 loan target -sources	yes
33876	PropertyCasualty360.com Celebrates Achievements, One Year Anniversary	no
48042	FOREX-Euro rallies on Fitch, IMF comments, but risks selling	yes
35295	RPT-Bund futures hit record high, Italian yields rise	yes
24550	African Markets - Factors to watch on Jan 11	yes
67459	The CHA Winter Show Draws Hollywood A-listers	no

Table A.1.: The articles are chosen by random from the category "Forex related". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label.

Article_ID	Article_Title	Is Forex related?
44961	ING Global Equity Dividend and Premium Opportunity Fund and ING International High Dividend Equity Income Fund Declare Monthly Distributions	yes
37153	Capital Market Conferences Continue to Pave the Way for Investor Confidence, Funding Opportunities and Economic	yes
32091	Indian shares drop; Infosys falls the most in 9 mths	yes
25140	RAPALA'S ANNUAL SUMMARY 2011	no
40839	Olympus Appoints Mr. Kevin Tomlinson as Director and Deputy Chair Replacing Mr. Doug Willock	no
65917	SWM ANNOUNCES CONFERENCE CALL TO DISCUSS FOURTH QUARTER 2011 RESULTS	yes
53794	Bank of McKenney Reports Solid Earnings on Double Digit Core Loan Growth and Expanding Margins	yes
39723	New NanoMarkets Report Projects Healthy Growth for Conductive Coatings in New Energy and Electronics Markets	yes
45279	Finning to Acquire Portion of Bucyrus Distribution Business From Caterpillar	yes
53106	ZTE shares log biggest fall in 5 months on slew of rumors	yes
39459	Catastrophe bond market set for busy Q1 - Aon	yes
29860	Zacks Releases Four Powerful "Buy" Stocks: InnerWorkings, A. Schulman, Men's Warehouse and Lincoln Electric Holdings	no

Table A.1.: The articles are chosen by random from the category "Forex related". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label.

Article_ID	Article_Title	Is Forex related?
68955	TEXT-S&P:Excess capital helps reinsurers ride out 2011 catastrophe losses	yes
35611	S&P will not cut Dutch, German rating: senior source	yes
39761	Oplink to Report Second Quarter Fiscal 2012 Results on February 2, 2012	no
48826	ECB sees tentative signs of economy stabilising	yes
50456	The Broe Group Announces that Halliburton Will Invest in Colorado Facility Sand Terminal on 54 Acres Within Great Western Industrial Park	no
55745	TEXT-S&P report looks at prospects for European business services	yes
60271	CIBER to Sell Federal Division; Transaction Allows CIBER to Focus on Core Offerings	yes
12368	The Zacks Analyst Blog Highlights: El Paso, Kinder Morgan, TOTAL S.A., Chesapeake Energy and Devon Energy	no
19770	REG - Lonmin PLC - Annual Information Update	no
20311	ITG Releases December 2011 US Trading Volumes and Reschedules Fourth Quarter 2011 Results Announcement	no
35414	UPDATE 3-Novartis cuts 2,000 US jobs after drug setback	yes
41259	HCL Technologies Q2 2012 Revenues at US$ 1,022 mn, Up 18.7% YoY and 3.7% QoQ in Constant Currency	no

Table A.1.: The articles are chosen by random from the category "Forex related". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label.

Article_ID	Article_Title	Is Forex related?
47505	REG - Source Mrkts Rus2000 - Net Asset Value(s)	no
61247	CORE BioFuel Inc. Engages Osprey Capital Partners Inc. as Investment Banking Advisor	yes
64813	EURO GOVT-Bunds drop on better data; no panic on Greece	yes
53160	UPDATE 1-UK retailers get Christmas boost from discounting	yes
21093	ClearOne to Showcase Its Unified Communications Solutions at IBM Lotusphere 2012	no
45078	REG - Melrose PLC - Notice of Results	no
20446	Kanban Optimizes Content Management Systems for Global Consumer Electronics Product Launch	no
65993	The Stock Radio Interviews Henry Fahman, Chairman and CEO of PHI Group, Inc. About Its Energy Program and Plans for	no
31946	TEXT-S&P asgns 'A+' rtg to Sun Hung Kai Properties' notes	yes
67496	InsideSales.com Awarded Patent as Pioneer of Immediate Response Technology	no
62187	REG - SREI Inf Fin Ltd - Notice of Board Meeting for 3rd Quarter Results	no
32199	UPDATE 3-Peugeot Citroen leaves options open on possible tie-up	yes
34948	REG - Future PLC - Annual Report 2011 including Notice of AGM	yes

Table A.1.: The articles are chosen by random from the category "Forex related". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label.

Article_ID	Article_Title	Is Forex related?
39537	UPDATE 1-Overnight deposits at ECB top half a trillion euros	yes
50018	Edgewater Announces Expiration of Odd-Lot Tender Offer	no
40031	Methode Electronics Power Solutions Group to Exhibit at Applied Power Electronics Conference February 5 to 9	no
33678	REG - Kentz Corporation - Holding(s) in Company	no
66590	REG-INVESCO Lev Hgh Yld: Interim Management Statement	no
60587	UPDATE 1-PetMed Express Q3 results beat estimates; shares up	no
39597	Germany sees 2013 growth of 1.6 pct - govt source	yes
5111	Innovus Pharmaceuticals Announces Free Availability of Real-Time Level 2 Stock Quotes	no
33397	First South Bancorp, Inc. Reports Increase in December 31, 2011 Quarterly and Year End Operating Results	no
52091	REG - O'Key Group SA - Unaudited operating results for 4Q and FY 2011	no
61782	Meredith Significantly Increases Its Digital Scale With Acquisition of Allrecipes.com, World's Top Food Website, From Reader's Digest Association	no
32923	REG - Titon Holdings PLC - 2011 Annual Report & Financial Statements	no

Table A.1.: The articles are chosen by random from the category "Forex related". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label.

Article_ID	Article_Title	Is Forex related?
65980	Breitling Oil and Gas CEO to Present at EMEA Unconventional Gas Exploration and Production Forum	no
24470	UniCredit shares rebound in volatile trade	yes
52931	Morgan Stanley Q4 commodities risk down 13 percent	yes
64904	UPDATE 1-Russian handset sales back to pre-crisis level-MTS	yes
20795	Information Services Group Announces Merger Into Single Go-to-Market Brand	no
59814	InspireMD Provides Corporate Update, Plans for 2012	no
44864	MHRA Issues GMP Certification to AMRI UK Facility	no
66838	Thoratec Schedules Fourth Quarter Conference Call, Webcast	no
53478	Noranda Income Fund Announces a Cash Distribution for the Month of January 2012 of $0.04167 Per Priority Unit	no
66904	Gallup Launches Redesigned Website With More Global Data	no
44868	Zipcar Appoints Frerk-Malte Feller as President of Zipcar Europe	no
30385	Broadband Delivers Strongest Growth Since 2009	no
8338	Global Axcess Corp Refinances and Re-Amortizes Credit Facility	no

Table A.1.: The articles are chosen by random from the category "Forex related". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label.

Article_ID	Article_Title	Is Forex related?
66525	Zacks Industry Outlook Highlights: Eastman Chemical Company, Celanese and The Dow Chemical Company	no
8130	RPM Reports Fiscal 2012 Second-Quarter Results	no
52909	Banks lead 4th day of European share rally	yes
25708	Limited Brands to Present at the ICR Xchange Conference	yes
30443	U.S. Consumers Doing Better at Paying Retail and Bank Credit Card Bills	yes
67381	AT&T Named Top Organization for Multicultural Business Opportunities	no
35332	FOREX-Euro stumbles, Italy auction gives reality check	yes
61193	Greek Nov tourism income drops, fewer Germans come	yes
41086	CombineNet Completes Record-Breaking Year for Financial Performance and Growth in E-Sourcing Software Market Share	no
25866	S&P Capital IQ Issues Twelve Internet Predictions for 2012	yes
38521	REG - Source Tech S&P US - Net Asset Value(s)	yes
48734	Euro Coal-Prices make modest gains on spring cargo orders	yes

Table A.1.: The articles are chosen by random from the category "Forex related". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label.

Article_ID	Article_Title	Is Forex unrelated?
30669	Austen BioInnovation Institute in Akron Announces Creation of First Company, APTO Orthopaedics	yes
35589	EU national regulators to vote on D.Boerse-NYSE January 17	yes
39681	REG - JPMorgan Small Co IT - Net Asset Value(s)	yes
64529	REG - JPMorgan Chin IT - Net Asset Value(s)	yes
48220	Spain favours CaixaBank, Bankia merger - sources	yes
50309	Can Newt Gingrich Beat Mitt Romney in South Carolina Primary?	yes
14702	Syria forces fire at protesters in capital: witness	yes
40251	Lifelock's Clarissa Cerda Honored by Arizona Business Magazine and the Association of Corporate Counsel	yes
53633	X-Rays, the Latest Tool in 'Green' Manufacturing	yes
1147	MTBC Recognized as One of Top 5 Medical Billing Companies	yes
39803	REG - New India In Tst PLC - Net Asset Value(s)	yes
28739	Talks on Greek bond swap to continue Friday - govt source	no
30303	Minister of State Goodyear Applauds the Appointment of Dr. Tilghman	yes

Table A.2.: The articles are chosen by random from the category "Forex unrelated". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label

Article_ID	Article_Title	Is Forex unrelated?
65471	UPDATE 1-Mixed reviews for Obama speech among Davos mighty	yes
26235	Gazelle's 2nd Annual CES RECOMMERCE Survey Reports Consumer Electronics Purchase and Reuse Trends in 2011 and 2012	yes
70620	TV ratings: Fox wins night with "Idol"	yes
38914	CORRECTED-UK gas prices fall on healthy supply, weak economy	no
57615	NetAmerica Completes Installation of IMS SuperCenter and NOC	yes
59248	Deluxemoda.com Protects Consumers Against Fake Prada, Coach, and Fendi	yes
25061	FuelCell Energy Delivers One Billion Kilowatt Hours of Ultra-Clean Power Generation	yes
64687	TEXT-S&P lowers rtgs in SME CLO deals PULS 2006-1 & 2007-1	no
18792	REG-FirstGroup PLC: Director/PDMR Shareholding	yes
3729	REG - Rob GilesPrologic plc - Form 8.3 - Prologic plc	yes
56375	Taco Bell hopes fresh food will whet diner demand	yes
44404	Merkel reserved about German moves after EFSF downgrade	no
29270	Zumbox and Catalog Choice Partner to Digitize Consumer Postal Mailboxes	yes

Table A.2.: The articles are chosen by random from the category "Forex unrelated". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label

Article_ID	Article_Title	Is Forex unrelated?
36819	PAMPERED PASSIONS FINE LINGERIE: Specialty Retailer One of the Largest Bridal Lingerie Stores in the Country	yes
65953	Esther Dyson's hopes for Russia	yes
49113	REG - JPMorganUS Small Cos - Net Asset Value(s)	yes
66775	Pacific Valley Dairy Introduces YoMazing!™ Frozen Yogurt Mix	yes
57270	REG - F&C U.S. Smaller - Net Asset Value(s)	yes
37057	REG - Clarity Commerce - Cancellation of Trading	yes
65486	DIARY-U.S. MEETINGS/WEEK AHEAD	yes
29436	Rio Tinto and CGI Renew and Expand Their IT Outsourcing Contract	yes
53373	Major P&F Industries Shareholder Ups His Stake to 8.1%	yes
62935	Singer Sewing Company to Launch The SINGER 160(TM) Limited Edition Sewing Machine on HSN	yes
48589	Instant view: Industrial output rises on strong	yes
40967	First Preclinical Proof-of-Concept of Mutation-Based Individualized Cancer Vaccine	yes
25258	REG - RBS Mkt Acs Lev F100 - Net Asset Value(s)	yes
27829	New Issue-KfW prices $1.0 bln 2014 FRN	yes

Table A.2.: The articles are chosen by random from the category "Forex unrelated". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label

Article_ID	Article_Title	Is Forex unrelated?
57368	Providence Expands into California	yes
19402	Australia's Spotless names its price at $760 million	yes
8963	John Hancock Funds Taps Chris Mee to Head Wirehouse Distribution Channel; Names Robert Milliman as New Western Divisional Manager	yes
12193	Global Aviation Finishes College Bowl Season with 2-2 Record	yes
64897	UPDATE 2-Mecom to charge for news online in new strategy	yes
24674	EU launches new challenge over payment card fees	no
53296	REG - Scot.Mort Inv Tst - Net Asset Value(s)	yes
39728	REG - Legal & General Grp - Blocklisting Interim Review	yes
29513	REG - BlackRock GroupFiberweb Plc - Disclosure of Short Position- Fiberweb PLC	yes
38545	REG-Fidelity Special Val: Net Asset Value(s)	yes
11088	US Airways Reports Record December Load Factor	yes
3081	REG - Marwyn Capital II Ld - Holding(s) in Company	yes
66712	Introducing Dutch's Spirits: An Original New York Distillery	yes

Table A.2.: The articles are chosen by random from the category "Forex unrelated". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label

Article_ID	Article_Title	Is Forex unrelated?
49510	REG - JPMorgan Elect PLC - Ten Largest Investments	yes
36879	REG - Standard Bank Plc - Doc re MTN Programme issue ISIN XS0558411525	yes
4658	Poll: Americans Embrace "Do-It-Yourself" Lifestyle in 2012	yes
20227	Gameloft and Qualcomm Demonstrate Advanced Mobile Gaming Experiences at CES 2012	yes
30130	Mountain Province Diamonds Announces Proposed Spin-out of Kennady Diamonds	yes
67103	AFGE Statement on Federal Employee Retirement Hearing	no
53143	Athletics-British sprinter Wilson handed four year ban	yes
20021	REG - Lyxor ETF Japan GBP - Net Asset Value(s)	yes
21368	New Mystery Novel Ponders the Pandemonium That Would Ensue if a Secret Biological Daughter of Princess Diana and Prince	yes
56993	BP and EU lobbied U.S. on Iran sanctions - sources	no
4583	Credo Petroleum Reports Record Reserves for Fiscal 2011	yes
31745	ServiceNow Used by Capgemini Application Services for UK Public Sector Client	yes
40715	Varolii Earns Speech Technology Excellence Award; Delivers High Quality Voice Interactions to Positively Impact Health	yes

Table A.2.: The articles are chosen by random from the category "Forex unrelated". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label

Article_ID	Article_Title	Is Forex unrelated?
48432	Ship's course made impact with rocks "inevitable": police divers	yes
57578	REG - The Cayenne Trust - Net Asset Value(s)	yes
61522	Vyatta Captures 'Virtualization Security' Product of the Year	yes
50211	ThinkGeek Announces Release of Exclusive DEXTER(R) Products	yes
21973	Bach to Rock Launches National Franchise Program	yes
38706	Vega-Chi sets 7th Feb launch for high yield MTF	yes
31856	REG - SQS Software Quality - Holding(s) in Company	yes
43335	Li3 Signs Agreement for Intensified Evaporative Technology	yes
20863	Compuware and Savvis Expand APM Partnership	yes
57204	REG-Pacific Assets Tst: Net Asset Value(s)	yes
18571	Achillion Reports Clinical Data on Portfolio of Protease Inhibitors	yes
48751	DIARY - European Market Holidays to December 2012	yes
54289	REG - Berendsen PLCPrudential PLC - Holding(s) in Company	yes
64618	REG-Fins Growth Inc Tst: Doc re Monthly Fact Sheet	yes

Table A.2.: The articles are chosen by random from the category "Forex unrelated". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label

Article_ID	Article_Title	Is Forex unrelated?
14699	Arab League asks for Hamas help with Syria	yes
64218	REG - Myriad Group AG - Form 8.3 - Myriad Group AG	yes
12621	REG-BlackRock Latin Am: Net Asset Value(s)	yes
25711	Schulz Academy Celebrates "National Coach of the Year" Award to Dr. Josef Schulz - the Former Coach of Jozy Altidore	yes
23597	REG - Source Markets 100 - Net Asset Value(s)	yes
62597	TrainSignal Releases VMware View 5 Essentials Training	yes
63032	Sirius XM Radio to Announce Full Year and Fourth Quarter 2011 Results	yes
37404	REG - CareCapital Grp plc - Holding in Company	yes
41281	GM Approves Naked Lime Marketing for Web, Marketing, and Advertising Solutions Under GM's in-Market Retail Turnkey Program	yes
23678	Costa Bingo Celebrates 2012 With a Trip to the Caribbean	yes
37156	LocateADoc.com Reveals Trend: More Moms Want Tummy Tucks, Despite Record Low Birth Rate	yes
39094	Not optimal time to launch Shanghai international board-mayor	yes
56220	Treasuries slip as Greece deal seen	no

Table A.2.: The articles are chosen by random from the category "Forex unrelated". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label

Article_ID	Article_Title	Is Forex unrelated?
26287	REG - Craneware plc - Holding(s) in Company	yes
25640	REG - Lyxor ETF MSCI EmLaD - Net Asset Value(s)	yes
30777	PHOTO ADVISORY – President of National Association of Attorneys General Accepts Petitions From More Than 720,000 Americans Calling for End to Sex Trafficking of Children & Young People	yes
49785	Relaxation Drink Maker BeBevCo Announces New York Distribution	yes
41886	RPost Patents Validated by California Federal Court	yes
22522	Oxi Fresh Cleans Up on Entrepreneur's 2012 Franchise 500	yes
375	Michael A. Marletta Takes Office as New President of Scripps Research Institute	yes

Table A.2.: The articles are chosen by random from the category "Forex unrelated". In this category only such articles are inside, which were automatically labeled as such by the filtering-algorithm. The third column reflects if a manual labeling approach would result in the same label

A.2. Classification

Regression	Offset	Feature	Performance Classifier	Performance all hold
40	5	Verb_Adjective	75,19%	74,99%
40	5	Verb_Adjective_Noun	75,12%	74,99%
40	5	Adjective_Noun_Adverb	75,07%	74,99%
40	5	Noun	75,02%	74,99%
40	4	Noun_Adverb	74,79%	74,55%
40	4	Adjective_Noun	74,77%	74,55%
40	4	Adjective_Noun_Adverb	74,62%	74,55%
40	4	Verb_Adjective	74,57%	74,55%
40	4	Verb_Adjective_Noun	74,57%	74,55%
40	3	Noun	74,20%	74,10%
40	3	Adjective_Noun_Adverb	74,15%	74,10%
15	0	Verb_Adjective_Noun	50,39%	49,04%
15	0	Verb_Noun_Adverb	50,24%	49,04%
15	2	Verb_Adjective	48,74%	48,62%
10	5	Verb_Adjective_Noun	46,31%	42,80%
10	5	Noun_Adverb	45,61%	42,80%
10	5	Adjective_Noun_Adverb	45,56%	42,80%
10	5	Adjective_Adverb	45,46%	42,80%
10	3	Noun_Adverb	45,14%	39,47%
10	5	Adjective_Noun	44,59%	42,80%
10	4	Verb_Adjective_Noun	44,59%	40,69%
10	5	Noun	44,57%	42,80%
10	4	Adjective_Noun_Adverb	44,54%	40,69%
10	5	Verb_Noun_Adverb	44,54%	42,80%
10	0	Verb_Adjective_Noun	44,39%	41,61%
10	1	Verb_Adjective_Noun	44,30%	41,56%

Table A.3.: From the left to right: Number of minutes used for linear regression. Number of minutes offset from the release-time of the article. Used features for classification. The performance of the classifier. And the performance of an "all hold" classifier which would always give "hold" recommendations. Only results are listed which are better than the "all hold" classifier and which are better than a random classifier (33.3%).

Regression	Offset	Feature	Performance Classifier	Performance all hold
10	0	Adjective_Noun	44,27%	41,61%
10	3	Noun	44,25%	39,47%
10	4	Adjective	44,22%	40,69%
10	4	Noun	44,15%	40,69%
10	2	Noun_Adverb	44,10%	39,65%
5	1	Noun_Adverb	44,07%	31,34%
10	0	Verb_Adjective	44,02%	41,61%
10	5	Verb_Adjective	44,02%	42,80%
5	1	Adjective_Noun_Adverb	44,00%	31,34%
10	5	Adjective	43,97%	42,80%
10	0	Noun_Adverb	43,90%	41,61%
10	3	Verb_Adjective_Noun	43,90%	39,47%
10	2	Adjective_Noun	43,82%	39,65%
10	0	Adjective_Noun_Adverb	43,77%	41,61%
5	1	Adjective_Noun	43,72%	31,34%
10	3	Adjective_Noun_Adverb	43,72%	39,47%
10	1	Noun	43,67%	41,56%
10	4	Adjective_Noun	43,65%	40,69%
5	1	Noun	43,60%	31,34%
5	1	Verb_Adjective_Noun	43,60%	31,34%
10	0	Verb_Noun_Adverb	43,45%	41,61%
10	4	Adjective_Adverb	43,38%	40,69%
10	0	Noun	43,33%	41,61%
10	1	Noun_Adverb	43,33%	41,56%
10	5	Verb	43,28%	42,80%
10	0	Adjective	43,23%	41,61%
10	1	Adjective_Noun	43,23%	41,56%

Table A.3.: From the left to right: Number of minutes used for linear regression. Number of minutes offset from the release-time of the article. Used features for classification. The performance of the classifier. And the performance of an "all hold" classifier which would always give "hold" recommendations. Only results are listed which are better than the "all hold" classifier and which are better than a random classifier (33.3%).

Regression	Offset	Feature	Performance Classifier	Performance all hold
10	3	Verb_Adjective	43,20%	39,47%
10	1	Verb_Noun_Adverb	43,13%	41,56%
10	1	Adjective_Adverb	43,10%	41,56%
10	3	Adjective_Noun	43,05%	39,47%
10	2	Verb_Adjective_Noun	43,05%	39,65%
10	4	Verb_Noun_Adverb	43,03%	40,69%
10	1	Verb_Adjective	42,98%	41,56%
5	1	Verb_Noun_Adverb	42,98%	31,34%
10	2	Verb_Noun_Adverb	42,93%	39,65%
10	3	Adjective	42,90%	39,47%
10	2	Noun	42,90%	39,65%
10	4	Noun_Adverb	42,85%	40,69%
10	1	Adjective_Noun_Adverb	42,80%	41,56%
5	0	Verb_Noun_Adverb	42,65%	32,54%
10	3	Verb_Noun_Adverb	42,63%	39,47%
5	0	Adjective_Noun	42,58%	32,54%
10	1	Adjective	42,53%	41,56%
10	3	Adjective_Adverb	42,48%	39,47%
5	4	Noun	42,46%	31,17%
10	0	Adjective_Adverb	42,41%	41,61%
10	2	Adjective_Noun_Adverb	42,41%	39,65%
10	4	Verb_Adjective	42,41%	40,69%
10	3	Verb	42,28%	39,47%
10	0	Adverb	42,11%	41,61%
5	2	Adjective_Noun	42,11%	31,84%
5	4	Adjective_Noun_Adverb	42,11%	31,17%
5	0	Verb_Adjective_Noun	42,08%	32,54%

Table A.3.: From the left to right: Number of minutes used for linear regression. Number of minutes offset from the release-time of the article. Used features for classification. The performance of the classifier. And the performance of an "all hold" classifier which would always give "hold" recommendations. Only results are listed which are better than the "all hold" classifier and which are better than a random classifier (33.3%).

Regression	**Offset**	**Feature**	**Performance Classifier**	**Performance all hold**
5	2	Adjective_Noun_Adverb	42,03%	31,84%
10	2	Verb_Adjective	41,96%	39,65%
5	2	Verb_Noun_Adverb	41,91%	31,84%
5	5	Adjective_Noun_Adverb	41,86%	31,15%
5	5	Verb_Adjective_Noun	41,78%	31,15%
10	2	Verb	41,76%	39,65%
5	4	Verb_Adjective_Noun	41,74%	31,17%
5	4	Adjective_Noun	41,71%	31,17%
5	3	Adjective_Noun_Adverb	41,69%	31,79%
5	5	Adjective	41,64%	31,15%
5	3	Noun	41,64%	31,79%
5	2	Noun_Adverb	41,64%	31,84%
5	3	Verb_Adjective	41,61%	31,79%
5	4	Verb_Noun_Adverb	41,59%	31,17%
5	5	Adjective_Noun	41,54%	31,15%
10	4	Verb	41,51%	40,69%
5	4	Noun_Adverb	41,49%	31,17%
5	3	Noun_Adverb	41,44%	31,79%
10	2	Adjective_Adverb	41,34%	39,65%
5	0	Adjective_Noun_Adverb	41,34%	32,54%
5	5	Noun	41,31%	31,15%
5	3	Adjective_Noun	41,31%	31,79%
5	3	Adjective_Adverb	41,26%	31,79%
5	2	Verb_Adjective_Noun	41,24%	31,84%
10	2	Adjective	41,21%	39,65%
5	4	Adjective_Adverb	41,19%	31,17%
5	0	Noun	41,11%	32,54%

Table A.3.: From the left to right: Number of minutes used for linear regression. Number of minutes offset from the release-time of the article. Used features for classification. The performance of the classifier. And the performance of an "all hold" classifier which would always give "hold" recommendations. Only results are listed which are better than the "all hold" classifier and which are better than a random classifier (33.3%).

Regression	Offset	Feature	Performance Classifier	Performance all hold
5	3	Verb_Adjective_Noun	40,94%	31,79%
5	0	Noun_Adverb	40,89%	32,54%
5	3	Verb_Noun_Adverb	40,87%	31,79%
5	0	Adjective_Adverb	40,82%	32,54%
5	2	Noun	40,74%	31,84%
5	5	Verb_Noun_Adverb	40,69%	31,15%
5	1	Adjective	40,62%	31,34%
5	1	Adjective_Adverb	40,62%	31,34%
5	1	Verb_Adjective	40,62%	31,34%
5	3	Adjective	40,57%	31,79%
5	2	Adjective_Adverb	40,54%	31,84%
5	5	Verb_Adjective	40,52%	31,15%
5	5	Noun_Adverb	40,42%	31,15%
5	5	Adjective_Adverb	40,37%	31,15%
5	2	Adjective	40,19%	31,84%
5	3	Verb	40,17%	31,79%
5	4	Verb_Adjective	40,09%	31,17%
5	4	Adjective	39,90%	31,17%
5	2	Verb_Adjective	39,62%	31,84%
5	1	Verb	39,45%	31,34%
5	0	Adjective	39,42%	32,54%
5	4	Verb	39,27%	31,17%
5	5	Verb	39,15%	31,15%
5	0	Verb_Adjective	38,63%	32,54%
5	4	Adverb	38,11%	31,17%
5	2	Verb	37,86%	31,84%
5	0	Verb	37,31%	32,54%

Table A.3.: From the left to right: Number of minutes used for linear regression. Number of minutes offset from the release-time of the article. Used features for classification. The performance of the classifier. And the performance of an "all hold" classifier which would always give "hold" recommendations. Only results are listed which are better than the "all hold" classifier and which are better than a random classifier (33.3%).

Regression	Offset	Feature	Performance Classifier	Performance all hold
5	2	Adverb	37,26%	31,84%
5	5	Adverb	36,86%	31,15%
5	1	Adverb	36,52%	31,34%
5	3	Adverb	35,94%	31,79%
5	0	Adverb	34,68%	32,54%

Table A.3.: From the left to right: Number of minutes used for linear regression. Number of minutes offset from the release-time of the article. Used features for classification. The performance of the classifier. And the performance of an "all hold" classifier which would always give "hold" recommendations. Only results are listed which are better than the "all hold" classifier and which are better than a random classifier (33.3%).

A.3. Additional Materials

a's	able	about	above	according
accordingly	across	actually	after	afterwards
again	against	ain't	all	allow
allows	almost	alone	along	already
also	although	always	am	among
amongst	an	and	another	any
anybody	anyhow	anyone	anything	anyway
anyways	anywhere	apart	appear	appreciate
appropriate	are	aren't	around	as
aside	ask	asking	associated	at
available	away	awfully	be	became
because	become	becomes	becoming	been
before	beforehand	behind	being	believe
below	beside	besides	best	better
between	beyond	both	brief	but
by	c'mon	c's	came	can

Table A.5.: The stop word list that the relational database engine MyISAM uses for MySQL as described in http://dev.mysql.com/doc/refman/5.6/en/fulltext-stopwords.html.

can't	cannot	cant	cause	causes
certain	certainly	changes	clearly	co
com	come	comes	concerning	consequently
consider	considering	contain	containing	contains
corresponding	could	couldn't	course	currently
definitely	described	despite	did	didn't
different	do	does	doesn't	doing
don't	done	down	downwards	during
each	edu	eg	eight	either
else	elsewhere	enough	entirely	especially
et	etc	even	ever	every
everybody	everyone	everything	everywhere	ex
exactly	example	except	far	few
fifth	first	five	followed	following
follows	for	former	formerly	forth
four	from	further	furthermore	get
gets	getting	given	gives	go
goes	going	gone	got	gotten
greetings	had	hadn't	happens	hardly
has	hasn't	have	haven't	having
he	he's	hello	help	hence
her	here	here's	hereafter	hereby
herein	hereupon	hers	herself	hi
him	himself	his	hither	hopefully
how	howbeit	however	i'd	i'll
i'm	i've	ie	if	ignored
immediate	in	inasmuch	inc	indeed
indicate	indicated	indicates	inner	insofar
instead	into	inward	is	isn't
it	it'd	it'll	it's	its
itself	just	keep	keeps	kept
know	known	knows	last	lately

Table A.5.: The stop word list that the relational database engine MyISAM uses for MySQL as described in http://dev.mysql.com/doc/refman/5.6/en/fulltext-stopwords.html.

later	latter	latterly	least	less
lest	let	let's	like	liked
likely	little	look	looking	looks
ltd	mainly	many	may	maybe
me	mean	meanwhile	merely	might
more	moreover	most	mostly	much
must	my	myself	name	namely
nd	near	nearly	necessary	need
needs	neither	never	nevertheless	new
next	nine	no	nobody	non
none	noone	nor	normally	not
nothing	novel	now	nowhere	obviously
of	off	often	oh	ok
okay	old	on	once	one
ones	only	onto	or	other
others	otherwise	ought	our	ours
ourselves	out	outside	over	overall
own	particular	particularly	per	perhaps
placed	please	plus	possible	presumably
probably	provides	que	quite	qv
rather	rd	re	really	reasonably
regarding	regardless	regards	relatively	respectively
right	said	same	saw	say
saying	says	second	secondly	see
seeing	seem	seemed	seeming	seems
seen	self	selves	sensible	sent
serious	seriously	seven	several	shall
she	should	shouldn't	since	six
so	some	somebody	somehow	someone
something	sometime	sometimes	somewhat	somewhere
soon	sorry	specified	specify	specifying
still	sub	such	sup	sure

Table A.5.: The stop word list that the relational database engine MyISAM uses for MySQL as described in http://dev.mysql.com/doc/refman/5.6/en/fulltext-stopwords.html.

t's	take	taken	tell	tends
th	than	thank	thanks	thanx
that	that's	thats	the	their
theirs	them	themselves	then	thence
there	there's	thereafter	thereby	therefore
therein	theres	thereupon	these	they
they'd	they'll	they're	they've	think
third	this	thorough	thoroughly	those
though	three	through	throughout	thru
thus	to	together	too	took
toward	towards	tried	tries	truly
try	trying	twice	two	un
under	unfortunately	unless	unlikely	until
unto	up	upon	us	use
used	useful	uses	using	usually
value	various	very	via	viz
vs	want	wants	was	wasn't
way	we	we'd	we'll	we're
we've	welcome	well	went	were
weren't	what	what's	whatever	when
whence	whenever	where	where's	whereafter
whereas	whereby	wherein	whereupon	wherever
whether	which	while	whither	who
who's	whoever	whole	whom	whose
why	will	willing	wish	with
within	without	won't	wonder	would
wouldn't	yes	yet	you	you'd
you'll	you're	you've	your	yours
yourself	yourselves	zero		

Table A.5.: The stop word list that the relational database engine MyISAM uses for MySQL as described in http://dev.mysql.com/doc/refman/5.6/en/fulltext-stopwords.html.

Table A.4.: 34 of the 48 Penn Treebank tags describing linguistic context.

POS Tag	**Description**
CC	coordinating conjunction
CD	cardinal number
DT	determiner
EX	existential there
FW	foreign word
IN	preposition/subordinating conjunction
JJ	adjective
JJR	adjective, comparative
JJS	adjective, superlative
LS	list marker
MD	modal
NN	noun, singular or mass
NNS	noun plural
NNP	proper noun, singular
NNPS	proper noun, plural
PDT	predeterminer
POS	possessive ending
PRP	personal pronoun
PRP$	possessive pronoun
RB	adverb
RBR	adverb, comparative
RBS	adverb, superlative
RP	particle
UH	interjection
VB	verb, base form
VBD	verb, past tense
VBG	verb, gerund/present participle
VBN	verb, past participle
VBP	verb, sing. present, non-3d
VBZ	verb, 3rd person sing. present
WDT	wh-determiner
WP	wh-pronoun
WP$	possessive wh-pronoun
WRB	wh-abverb

A.4. Full System Class Overview

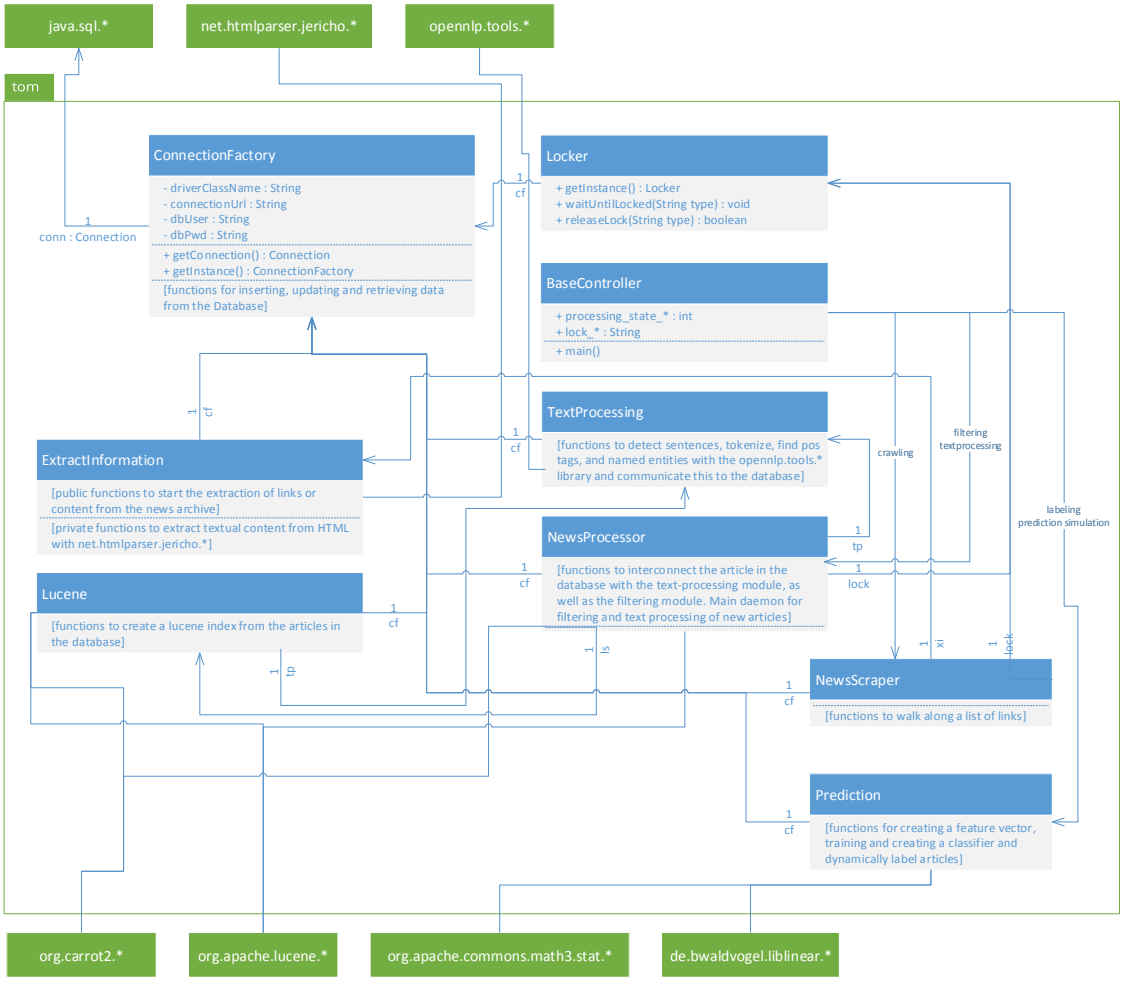


Figure A.1.: An abstracted schematic overview how classes interact with each other. The functionality of each class will be broken down in each of the section within the chapter, because a general overview is not possible due to lack of sufficient space on the page.