Thomas Forss

Automating Text Processing Using Analytics

Automating Text classifications and financial news parsing

Bo Thomas Forss

Born 1984

Studies and degrees

- Bachelor of Software Engineering at Åbo Akademi 2010
- Master of Software Engineering at Åbo Akademi 2011
- Ph.D. in Economics at Åbo Akademi 2017

Automating Text Processing Using Analytics

Automating text classifications and financial news parsing

Thomas Forss

Information Systems
Faculty of Social Sciences, Business and Economics
Åbo Akademi University
Åbo, Finland, 2017

Supervisors

Markku Heikkilä
Docent

Christer Carlsson
Professor Emeritus

Kaj-Mikael Björk
Docent

Information Systems
Faculty of Social Sciences, Business and Economics
Åbo Akademi University
Domkyrkotorget 3, 20500, Åbo
Finland

Reviewers

Professor Rudolf Kruse

Fakultät für Informatik
Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2, D-39106, Magdeburg
Germany

Doctor Yoan Miche

Bells Labs Nokia
Karaportti 3, 02610, Espoo
Finland

Opponent

Professor Moncef Gabbouj

Department of Signal Processing
Tampere University of Technology
P.O. Box 553, 33101, Tampere
Finland

ISBN 978-952-12-3634-1
Painosalama Oy – Turku, Finland 2017

Table of Contents

Abstract

Automating repetitive processes and replacing manual tasks with automated systems is an area of research that will greatly impact and transform our lives during the 21st century. Automation comes in many forms and we are now at the start of an era, after which repetitive non-creative tasks will be handled mainly by machines. In this thesis, two analytics approaches are presented that can be used to automate text processing tasks.

The first is an automation approach using machine learning in which we show how we can improve text classification performance, and how we, through these improvements, can reach practically acceptable performance levels even in certain abstract classification problems. We test the developed methods on problematic web content categories, such as violence, racism, and hate.

The second is an automation approach that uses network analytics to automatically process texts. We use this approach to automate processing of financial news and to automatically extract new information. We show that through automating the process, we can extract company specific sentiment-risks that a person would not identify simply by reading the news articles. Lastly, we show that the risks we have extracted can be used to identify companies that are at higher risk of stock price decrease.

Sammanfattning

Att automatisera repetitiva processer och ersätta manuellt arbete med automatiska system är ett forskningsområde som kommer att ha stor inverkan på vårt samhälle och med stor sannolihet kommer att förändra våra liv under detta århundrade. Automatisering kan göras på många olika sätt. Vi är nu vid början på en era varefter repetitiva icke-kreativa arbetsuppgifter kommer hanteras till största del av maskiner. I denna avhandling presenteras två tillvägagångssätt som kan användas för att automatisera textprocessering.

Det första tillvägagångssättet beskriver en metod för automatisk klassificering av texter till fördefinierade kategorier genom användning av maskininlärning. Vi går igenom hur man kan utveckla textprocesseringsmetoder som kan nå praktisk användbar prestanda även i mera abstrakta och svårhanterliga kategorier, som t.ex. klassificering av våldsamma, rasistiska och hatiska webbsidor.

Det andra tillvägagångssättet beskriver en metod för att automatiskt processera stora mängder nyhetstexter genom nätverksanalytik. Vi använder metoden för att processera finansiella artiklar och skapa ny information. Genom automatisering av processen visar vi att vi kan beräkna företagsspecifika förväntningsrisker som en person inte kunde ha identifierat enbart från att ha läst artiklarna. Slutligen visar vi att det är möjligt att identifiera företag som har en högre risk än medeltalet, och att hög risk korrelerar med ökad risk att aktiepriset för företaget sjunker.

Tiivistelmä

Manuaalisen työn ja toistuvien työprosessien automatisointi ja siihen liittyvä tutkimus tulee mitä todennäköisimmin muuttamaan yhteiskuntamme toisenlaiseksi tämän vuosisadan aikana.

Automatisointia voidaan toteuttaa monilla tavoilla. Elämme nyt sen aikakauden alkuvaihetta, jonka kuluessa koneet tulevat tekemään miltei kaikki toistuvat työprosessit. Tässä väitöskirjassa esitetään kaksi tapaa tekstinkäsittelyn automatisointiin.

Ensimmäisessä tavassa sovelletaan uutta tekstinluokitusta, jossa kehitetty koneoppimismenetelmä luokittelee tekstit ennalta määriteltyihin luokkiin automaattisesti. Menetelmän avulla käyttökelpoinen suorituskyky voidaan saavuttaa jopa abstrakteissa ja vaikeissa tehtävissä, kuten esimerkiksi väkivaltaa ja rasismia sisältävien tekstien luokittelussa.

Toisessa tavassa sovelletaan uutta verkkoanalyysimenetelmää uutistekstien automaattiseen käsittelyyn. Kehitetyllä sentimenttianalyysimenetelmällä analysoidaan rahoitusuutistietokantaa. Tällöin voidaan löytää sellaisia uusia sentimenttejä yrityskohtaisten riskien mittaamiseen, joita tietokannan uutisia lukiessa ei tunnisteta. Lopuksi osoitetaan, että analyysin perustella on mahdollista tunnistaa yrityksiä, joilla on keskimääräistä suurempi riskiarvo, ja että tämä riskiarvo korreloi osakkeen todellisen arvonvähennyksen kanssa.

Acknowledgements

With my thesis now finally finished, I feel that I stand at a rare crossroad and that I have many roads to choose from. Those who know me, know that I am a task oriented person (by choice), who most of all values productivity, self-awareness, and freedom. Therefore, it feels very liberating after this four-and-a-half-year long journey to be able to look back and say that I truly enjoyed most of it, even though I have not known how to proceed at several occasions. At one point, I stopped working on the classification research and focused more on finance, because I was not able to improve the results further. Only to come back to the classifications two months later and test one final idea, the multi-grams, which then turned out better than I hoped. I realized during the first year of my PhD studies that it is possible to learn to enjoy the process of writing, which made the work much easier. Later, when I learned machine learning modelling, I found that the process of creating and testing models is as rewarding to me as playing video games and board games.

I would like to thank the people and organizations that have helped me reach this crossroad. I have had three supervisors during my thesis work: Christer Carlsson, Kaj-Mikael Björk, and Markku Heikkilä. I greatly appreciate all the time and effort you gave me, I have always felt well treated and I have great respect for all of you. I would also like to thank the other researchers that I have worked with: Peter Sarlin and Shuhua Liu. Your help has been essential in getting into the different fields of research. I would like to thank the people that I have had interesting research related discussion with: Magnus Westerlund, Göran Pulkkis, and Xiaolu Wang. I would like to thank all my colleagues and friends at Åbo Akademi and Arcada University of Applied Sciences. I would like to thank Alexander Dockhorn for the insightful comments on my thesis. I would like to thank Liikesivistysrahasto, FSE, and Åbo Akademi for the financial support and the research grants I was awarded over the last two years.

Finally, I would like to thank my family. My wife Sarah is the only one that reads through all the texts that I produce, which means a great deal to me. I would like to thank my parents Håkan and Ylva for always being there and always believing in me, always ready to back me up. I would like to thank my brother Nicklas and sister Annika for being supportive in all matters.

1. Introduction

Information technology is revolutionizing products and has unleashed a new era of competition. Smart, connected products have the potential to shift rivalry, opening up numerous new avenues for differentiation and value-added services (Porter and Heppelmann 2014). This revolution impacts almost every area of every business, and at the center of this revolution lies analytics.

Analytics can be defined in many ways; I have chosen to use a definition by Evans and Lindner (2012) and Davenport and Kim (2013) where analytics is divided into three different main areas. These three areas are descriptive analytics, predictive analytics, and prescriptive analytics. The same kind of division in analytics is used by the respected Informs web site ("Analytics Informs" 2016; “What Analytics Is” 2016). Furthermore, when these three disciplines are combined, we unite them under the term advanced analytics as illustrated in Figure 1 (Gartner Advanced Analytics 2017).

Analytics is used as an umbrella term to cover many types of methods, and is a continuation of operations research in the United Kingdom and management science which has a long history dating back to the Second World War and the early 1950’s (Rau 2005). Analytics is also related to decision support, executive support, online analytical processing, and business intelligence (Porter and Heppelmann 2014). Descriptive analytics consists of methods for understanding and visualizing data, predictive analytics consists of methods that predict future outcomes based on historical data using, for example, machine learning, and prescriptive analytics are methods that try to optimize the best possible outcome in situations when we have several options to choose from. In section 3, we go into more detail about the different analytics methods and define the ones used in our research.

According to Gartner’s hype cycle ("Gartner’s Hype Cycle" 2015), analytics is one of the most important current trends. Furthermore, analytics is together with big data and the Internet of things, one of the fastest growing business areas of this decade (Kar 2016).

Traditionally, companies have focused mostly on the parts of their businesses that they could directly reasonably assume would provide them competitive advantages. Businesses focused mainly on improved products, reduced cost of

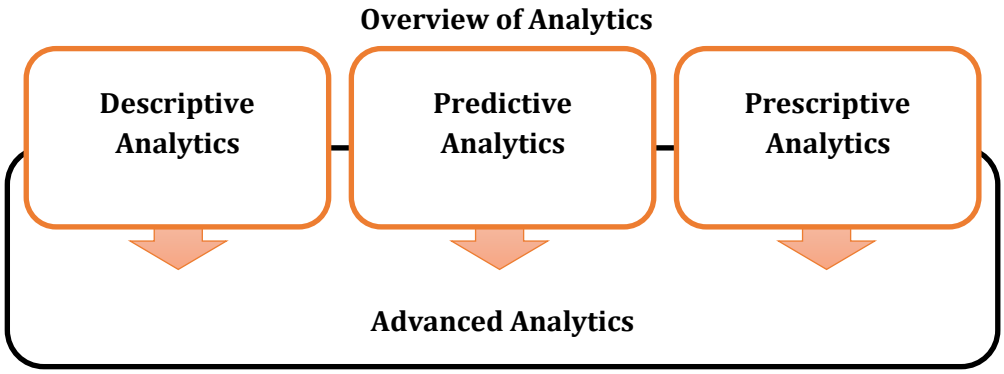

Figure 1 *The different branches of analytics. When methods from several branches are combined, the term advanced analytics is used.*

materials, branding, marketing, and quality. Meanwhile, pioneering companies such as Amazon started using analytical methods to transform and dominate their fields. These pioneering companies started focusing on other areas such as efficiency, availability, easy access, fast shipping, and reduced storage costs. In other words, there was a change in paradigms and companies started competing on analytics. This came at a time when businesses in different fields were increasingly competing on a global market. In global markets, where developing countries have the labor cost advantage, one of the only ways left to compete is through business processes. (Davenport 2006)

1.1. Background

Information systems (IS), as a field of study, combines a wide array of topics (Checkland and Holwell 1997; Alter 1998). Information systems uses information and computation theory (Kullback 1997) to create and analyze links between business and computer science. Analytics can, depending on how it is approached, be considered a sub field of either information systems or computer science. Analytics falls into the domain of information systems when either the focus is business related, or when the outcome of the research has a link to business and/or economics.

Analytics is the process of gathering data and analyzing the data to be able to improve manufacturing processes, optimize supply chains, increase software performance, improve performance of targeted ads, or a multitude of other efficiency improvements (Cooper et al. 2012). Analytics seems to have a compounding effect, where analytical capabilities derive customer value and create new information, which in turn may lead to more business opportunities.

The effectiveness of different algorithms has, during the last decade, improved to the point that new technologies have started emerging. In many cities, we need to look no further than to the roads to be able to see fully automatic cars driving around powered by big data analytics algorithms.

At the same time, the use of analytics has spread to almost every sector of society in the form of practical applications (Davenport and Kim 2013; Raghupathi and Raghupathi 2014; Lee et al. 2014). Some example uses of analytics in industry are decision support systems for bank customers to help make investment decisions (Avriel et al. 2004); optimization of oil refineries that increase profits by helping in capital allocation decisions (Kutz et al. 2014); analytics used in baseball to improve chances of winning games (Lewis 2004); analytics used in health care to manage patient care (Bates et al. 2014).

Analytics is now so widely used that most large and medium-sized companies must use it to be able to keep up with competition (Davenport 2006). In fact, 92% of all enterprises now use analytics to gain marketing insights, and 81% of enterprises rely on analytics to improve their understanding of customers (Columbus 2015). Additionally, 50% of U.S. companies report increased sales due to investments in analytics (Columbus 2015). Davenport and Kim (2013) go even further, they claim that it is now dangerous for companies to not invest in analytical capacity. To corroborate this, researchers have shown that there is a positive correlation between companies performing well and their analytical capabilities (Trkman et al. 2010). In a survey done by the *Harvard Business Review* ("Competing in 2020" 2017), 47% of the responders say their organization's business model will be obsolete by 2020 due to the growing digital economy. Furthermore, they found that there is a significant gap in big data analytics usage and capabilities between companies. As many as 84% of the digital leaders, which are companies that rely on digital technology to deliver products, also use big data analytics, while among the companies that have few products digitally available, only 34% use analytics. Regarding artificial intelligence/machine learning solutions, 51% of the digital leaders report using such solutions, compared to only 7% of the followers. When asked which skills will be the most important in 2020, 69% of the survey responders answered the ability to work with data and analytics ("Competing in 2020" 2017).

Davenport and Kim (2013) write that the financial crisis, which started unfolding in 2008, could have been prevented if companies or government institutions in the financial sector would have had greater analytical capabilities. The argument is that if key people in financial organizations would have had a better understanding of analytics, and if they would have had better tools available,

then the credit default swaps (CDS) used to insure debt holders against default would have been priced differently. The Financial Crisis Inquiry Commission (FCIC) has also written a report on the 2008 financial crisis. The conclusion that the FCIC reached is that the crisis could have been prevented, although the solution was slightly different than what Davenport and Kim (2013) suggested. The FCIC argues that while some financial institutions did sell questionable CDSs, the underlying problem was the mortgage lending policies. It argues that unsustainable lending policies would, regardless, at some point make the housing system unravel. The FCIC concluded that the policies, which could have been more strictly monitored by the Federal Reserve, were left unchecked. (FCIC 2011)

Personally, I would dare suggest that we have much yet to discover in the field of analytics. We have come far in certain areas such as some parts of optimizations (Gabrel et. al. 2014) and self-driving technology (Bojarski et. al. 2016). However, many areas of analytics such as in parts of content classifications and parts of financial analytics can still be improved. For example, the United Kingdom has implemented Internet filtering systems that are supposed to block unwanted pornographic and malicious content. When comparing the world's top 100,000 ranking web sites ("Alexa Top 500" 2016) to different available filters, it has been shown that these default filters block up to 12% of the web sites ("Report On Blocked Sites" 2016). Examples of unfairly censored sites are many, and some businesses are impacted directly because of these filters ("Personal Stories" 2016).

In this thesis, different methods and systems that could be used to help improve performance of content filtering systems are presented. Furthermore, tools and methods are here presented that could help financial experts better understand companies, sectors, and markets. The understanding comes in the form of a new valuation perspective, a new risk perspective, and new knowledge of the interconnectedness between different components in the financial markets.

1.2. Business Relevance

In this part, two fictional characters will be presented. They are meant to offer insights into some of the practical uses of the research presented in the thesis. While these characters are not one-to-one representations of real people, the problems that they face are real-world problems that people working in the industries want to solve. We will refer to these characters throughout the thesis

to highlight the link between research and practical applications. The characters are modelled after real people I have been in contact with during my career as a researcher and the needs of people in the respective industries.

The first character, which I in the future will refer to as Neil, is modelled after an expert from the Data to Intelligence (D2I) project that I will be talking more about in section 1.2.2. Neil represents a person working in a company that provides security solutions and services. The typical person that Neil represents is a technical security expert with a degree in computer science and/or data analysis. He has knowledge of programming and an in-depth knowledge of how the Internet works. He could, for instance, be working at a company that offers security products, such as parental control systems that filter and/or block sensitive data on their customer's devices. Practical applications of such systems could either be preventing people from accessing harmful pages from their work computers, or filtering adult, violent, racist, and hateful content for minors. Part of the job that Neil is performing is evaluating web sites using different means, and deciding whether web sites should be filtered. In practice, this translates into a need to develop systems that can perform filtering automatically. The objective of Neil's work is to categorize websites as accurately as possible, and the most important part is to not categorize a web site as harmful when it really is not.

The second fictional character, which I in the future will refer to as Jenna, is modelled after an investment professional, a person that invests people's savings. Jenna represents a person that works in the finance industry and is a person with a degree in economics and/or data analysis. She could be working at a hedge fund as a money manager or at an institution such as the European Central Bank. Essentially, she has skills that are relevant at any place that keeps track of economic developments and the markets. Part of Jenna's duties is following different economic indicators, analyzing asset values, and finding new indicators that show the health of the economy, specific sectors, and individual companies. Part of her tasks could also be analyzing portfolio risks using different risk models. The objective of Jenna's work is to use different quantitative and qualitative measures to support decisions on investment allocations.

1.2.1. Industry Challenges and Opportunities

At the start of the century, companies started competing by gathering and analyzing data. This data driven approach then evolved into what now is known as "Big Data". Big data is the name used for gathering, creating, and analyzing

massive amounts of data where the dimensions of data themselves lead to new challenges. Many different companies and institutions are facing situations where they have gathered and/or created so much data that they need to come up with new methods to even be able to store and process the data efficiently, let alone make sense of the content. In this part, the discussion will be about different challenges that the relevant industries are facing, the opportunities hidden in these challenges, and the solutions that analytics offer.

The different big data challenges that companies and industries now are facing have aptly been described as the five Vs. The four main challenges are the following: volume, which refers to the scale of the data; variety, which refers to the different forms of data; velocity, which refers to the speed of gathering or creation of data; veracity, which refers to the uncertainty of the quality of the data. The fifth V has been added later and stands for value, which refers to how businesses can gain value out of the big data that is being processed. ("The 5 Vs of Big Data" 2016)

The challenges relating to volume that exist today are many and vary greatly between industries. Essentially, most of the big data challenges revolve around needing the processing power and/or storage capacity of many computers in all parts of gathering, creating, and processing data. Splitting data between different computing entities is already a non-trivial task. For this purpose, many companies such as Google, Amazon, Microsoft, and Oracle have created different big data storage solutions that have become widely used. To solve the problem of processing large volumes of data, different types of new processing solutions had to be developed. Hadoop and Spark are two of the currently popular solutions. Hadoop was developed as a distributed file system that becomes shared between all the computers added to the system (Shvachko et al. 2010). Spark is an engine for large-scale data processing that can run on top of different types of architectures, including Hadoop ("Apache Spark" 2016).

The variety of data is another big data challenge. The different types of data that can be gathered and created are many and they all come with their own challenges. Working with numerical data can vary from analyzing historical bank transaction data to complex tasks such as streaming and processing sensor data from autonomous vehicles or space shuttles. Textual data handling tasks can range from processing customer reviews to filtering spam messages to massive tasks such as keyword indexing the entire Internet for search engine use. To provide an idea of the changing scale of text processing tasks, the Internet consisted of about 350 million web sites in 2011, and in 2016 consisted of over a billion active web sites ("Total Number of Websites" 2016). This is a growth of

about 150 million web pages per year, which would be impossible to keep track of without sophisticated automated systems. Companies today also face many new challenges regarding media content such as audio and video. Most of these challenges stem from having to stream the content to users, at any time and to any location, with varying quality. Compare this to traditional broadcast systems, such as cable TV or radio, which send the same content to all their customers at simultaneously, limited to certain locations, and with a predefined quality.

That takes us to the third type of challenge that big data has brought with it, the velocity. Since the inception of the Internet, there has been a constant need for improvements in the underlying infrastructure to be able to handle the growing data consumption across the globe. This naturally also translates to storage capacity growth needs and data processing problems. With the introduction of streaming services such as YouTube, Netflix, Twitch, and SoundCloud, the magnitude of data transferred over the Internet has increased almost exponentially. Cisco Systems has estimated that IP traffic will surpass 2.3 zettabyte per year by 2020, and video is estimated to take up 82% of the total load with most of the traffic coming at certain peak hours ("The Zettabyte Era" 2016). On another front, security companies are in constant battles against spammers, virus creators, and malicious sites. Kaspersky Labs reported that email spam was over 56% of the total email traffic in the beginning of 2016 ("Spam and Phishing Securelist" 2016). The constant improvement in automated detection of these types of threats is what keeps us from being overloaded by spam emails.

Veracity of data is the fourth and last of the original big data challenges. Veracity refers to the quality of data, the possibility of missing values and having data containing incorrect values, incorrect value types, and unstructured data. The veracity quality challenges generally start appearing once the gathering, creation, and processing is of such a magnitude that it becomes difficult to keep the data in a structured format. Companies started to notice that in certain scenarios, such as when indexing the entire Internet, the SQL databases they were using could no longer be used to process the data efficiently. To solve such challenges, companies had to create new types of databases that are simpler, less structured, and because of that cannot perform the full set of calculations that SQL databases do. These types of databases were given the name NoSQL databases.

Value of data, the fifth V, should be approached more from a business process perspective. Companies need to ascertain that the data that is being processed can be turned into insights, otherwise the data gathering and creation is not

useful. In this thesis, we will see examples of how processing data can be turned into new insights.

Let us examine some statistics to help highlight some of the challenges that companies and professionals are facing related to financial news and automatic classifications. While it is difficult to estimate exactly how many news articles that are published daily, Thomson Reuters has released a few corpuses containing articles that can be used to put the number of articles in perspective. The TRC2 corpus was gathered over a thirteen-month period between 2008 and 2009 and contains over 1.8 million articles (Reuters Corpora 2016). This shows us that experts who would be interested in keeping up with the news are easily overwhelmed by the content published by one single news agency. Add to this the thousands of different news agencies worldwide and the news flow is of such magnitude that even scores of workers would have problems covering everything of interest. This touches upon the concept of having too much information available, also known as information overload. Information overload can be a serious problem both for businesses and for individuals and has been researched extensively (O'Reilly 1980; Edmunds and Morris 2000).

To effectively keep up with news in their work environment, some professionals, such as Jenna that I described earlier, need to come up with strategies to filter and find relevant content. Some professionals simply limit the content that they take in before making decisions. These limits can, for instance, be following only a subsector of sites that post relevant content, limiting content by sectors, limiting by news about certain companies, or even limiting consumed content to certain experts. Alternatives would be to use existing tools to help filter content in different ways or creating tools that could help find the content that is relevant. For this purpose, companies have started to develop products that help with filtering data for consumption for certain niche markets. Here follows a few different examples of such products: AlphaSense.com is a financial search engine; Stocktwits.com is a short messaging platform like Twitter that focuses on financial news only; TipRanks.com is a platform that focuses on keeping track of financial analyst's opinions and predictions.

As the internet grows and new technologies are developed, it affects society in different ways. If we consider the kind of situations that security experts like Neil deal with, we find a set of different challenges. A clear example of the type of challenges that companies can have to face, after introducing new technologies, is the battle that Twitter is fighting against violent extremism and racism on their platform. Twitter reported in August 2016 having banned 360, 000 accounts for promoting terrorism since 2015. A quote taken straight from their blog sums up

the problems they are having: “There is no one “magic algorithm” for identifying terrorist content on the Internet.” ("An Update on Violent Extremism" 2016) They use spam filtering tools that have helped them automatically identify about a third of these accounts. This has left them with the manual work of identifying 240,000 accounts on their platform alone. The problems they are facing are obviously manifold and relating to all five types of big data challenges. Terrorist and violent content can come in the form of text, images, video, and audio. Five hundred million tweets are sent daily on the Twitter platform as of October 2016 ("Twitter Usage Statistics" 2016). Furthermore, people write in different languages and writing can be of a satirical nature, which makes the identification complicated.

Those are a few examples of challenges and opportunities that companies and individual experts are facing daily in their work, and it puts the need of the research we have been conducting somewhat into perspective. The Finnish government also recognized these challenges that the information age has brought with it. The government funded a major research project that aimed to find solutions to these types of challenges. The project participants were industry partners and research institutions from all over Finland. The project was named Data to Intelligence (D2I) (“Data to Intelligence Program”, Tekes project number 340/12) and will be discussed further in the next section.

1.2.2. Data to Intelligence

Four of the six research publications [1, 2, 3, 6] that are part of this thesis are the results of real-world needs identified by Finnish companies. Three of these four publications [1, 2, 3], were developed as part of the D2I project. Publications [4] and [5] have come out of a collaboration with Dr. Peter Sarlin. The last publication [6] was started as a solo project after D2I ended.

D2I was a cooperative project (2012 – 2015), backed by the Finnish government, consisting of 17 research institutions, 27 large enterprises, and 26 small and medium sized companies from various industries in Finland. The research focus in the D2I project was big data and user-centric service development. The aim of the program was to develop intelligent tools and methods to be able to innovate in services and in business models. The project was split into seven different areas: Traffic, Multimedia, Security, Industry, Customer Intelligence, Be-well, and Forest Big Data. In total, the project has resulted in over 310 publications such as conference papers, journal articles, books, and reports. ("Data to Intelligence Program" 2016)

Analytics was in the center of many approaches used in the D2I project. Several research groups used analytics to try and solve different problems that the involved partner companies had identified. These problems have a wide range: improving classifications of web content; recognizing audio data through audio analytics; categorizing social media users based on hobbies and interests; creating new image classification algorithms; using analytics to extract relevant information from incident reports and much more.

Our research team in the D2I project consisted of Dr. Shuhua Liu and myself. Our team was part of and worked in the D2I project in two areas: security and multimedia. For the most part, we worked together with two companies: F-Secure Corporation, which is one of the larger companies in the online security business, and PacketVideo, which is a medium sized media company.

Our research in the project consisted mainly of text-analytics methods. These are key-word-extraction methods, feature-extraction methods, and machine learning algorithms. Together with F-Secure, we worked on using analytical methods to improve text classification results. We also developed analytical methods for searching through social media content, and we worked on extracting text information relevant to social media content with PacketVideo.

1.2.3. Relevance of the Thesis

The research in this thesis is focused on two areas of text analytics: automatic classification and financial news analytics. In automatic classifications, machine learning is used to classify text in web pages. In financial news analytics, we use network theory (Özgür et al. 2008), centrality measures (Stephenson and Zelen 1989), risk measures (Mezei and Sarlin 2017), and visualizations to uncover useful information about companies.

Classification can be done for many purposes and in many different areas. Some of the best known applications are classification of objects in images such as recognizing faces or fingerprints (Marr and Hildreth 1980), classifications used in movie recommendations and targeted ads (Cortes and Vapnik 1995; Specht 1990), and classifications and summarization of text documents (Salton and Buckley 1988). Text classifications can also be used to create parental control systems and have also been used to filter spam messages in messaging services (Androutsopoulos et al. 2000).

Financial news analytics has also been used in practical applications. Among the best known applications, are automated trading systems (Gately 1995),

identification of money laundering transactions (Reuter 2004; Kingdon 2004), and methods for detection of systemic risks (Schwaab et al. 2011).

From a business perspective, there is much to be gained from advances in both areas. By increasing performance of web content classification (Sun et al. 2002), we can improve web filtering techniques. We could also improve malicious web site filtering (Du et al. 2003), and if we are able to achieve high performance we could maybe find totally new business opportunities that previously required humans to perform tasks. One example of this could be automatic link aggregation. Automatic link aggregation would be different from static link aggregation sites, such as Reddit.com, by not needing humans to manually submit content. In 2013, researchers (Frey and Osborne 2013) postulated that in the years leading up to year 2033, up to 47% of all current jobs are at-risk jobs that have a chance of being automated. Improvements in classifications and automatic text-processing methods, such as the ones discussed in this thesis, represent two of the approaches that could be used in automating repetitive text processing tasks.

The business importance of the research presented in financial analytics can provide investors a better understanding of the underlying conditions of different markets, it can provide an increased understanding of the relationships between different companies, and it can help uncover risks that were previously unknown. Understanding risks and the relationships between companies could be crucial in avoiding losses in investments. According to researchers (Dell'Ariccia et al. 2008), a banking crisis can reduce GDP for a country by between 15% to 25%. Being able to prevent or avoid any type of larger crisis through analytical methods would then have a positive effect on the economy.

1.3. State of the Art

Here follows a discussion about the current state-of-the-art methods in the fields automatic classifications and financial analytics. In the field of automatic classifications, we are interested in research in text classifications, as well as any type of classifications done on violent and hateful content. In the field of financial analytics, the research is generally more qualitative, which makes it difficult to directly compare results. Nonetheless, the state-of-the-art comparisons will be done against research that contains elements similar to the research presented in this thesis.

1.3.1. Information Extraction

In every publication presented in this thesis, we have some form of information extraction. The relevant research in information extraction starts with automatic processing of both structured and unstructured text data (Cowie and Lehnert 1996; Soderland 1999; Banko et al. 2007; Aggarwal and Zhai 2012). Another part of information extraction is named entity recognition (Toutanova et al. 2003; Ratinov and Roth 2009). Named entity recognition (NER) is part of one of the publications presented [1] and has, for example, previously been successfully used in extracting and identifying entities in social media (Ritter et al. 2011; Liu et al. 2011). NER has also been used to find disaster related messages in social media (Imran 2013). Researchers have also used extracted information to rank products (Zhang et al. 2010), and extracted tags for categorization from social media (Cantador et al. 2011). We used a similar approach to extract people's hobbies and interests from social media [1]. Extracting company names from news (Rau 1991) can be useful for research in finance. In our publications in financial analytics [4, 5], we extract company names using methods similar to those described by Bullinaria and Levy (2012) in combination with methods from publication [1]. For the automatic classification publications [2, 3, 6], we use extraction methods that are based on popular feature extraction and text mining techniques (Lewis 1992; Aggarwal and Zhai 2012) in combination with the methods from publication [1].

1.3.2. Automatic Classification

Automatic classification is part of the information retrieval domain. In classifications, different types of methods are used depending on the type of data that needs to be classified. Methods used in text classifications differ from methods used in image, video, and audio classifications. The classification contributions in this thesis consist of text classification research, which is why the state of the art discussions focuses mainly on methods used in text classifications.

Automatic text classifications, also known as text categorization and document categorization, became popular in the early 1990s (Sebastiani 2002). It has since developed into subcategories with specific methods that work well for some areas and some categories of text. No single method has yet been shown to work well in all possible text classification tasks. Text classification has because of that, during the last two decades, evolved into a field with a multitude of approaches that work well for a subset of classification tasks.

One of the early and successful word-ranking methods, which is still used in many classification approaches, is one that uses term frequency and inverse document frequency (TF-IDF) (Salton and Buckley 1988; Joachims 1996; Han and Karypis 2000). TF-IDF-based classifications have been shown to be robust and reliable also in recent research (Zhang et al. 2011; Trstenjak et al. 2014; Ko 2012).

From TF-IDF-based classifications emerged n-gram-based classifications (Cavnar et al. 1994). The main contribution from n-gram analysis was that word order became important where it had not been considered before. A study by Fürnkranz (1998) showed that 2-grams and 3-grams could improve results, but that going above that also could decrease performance while classifying the Reuters 20 news groups corpus. Some of the situations that n-gram-based analysis have been shown to work well in are detecting malicious code and viruses (Abou-Assaleh et al. 2004; Reddy and Pujari 2006) and language independent classification of texts using n-grams (Damashek 1995). In another study (Khreisat 2006), n-gram analysis on the Arabic language was shown to outperform previous approaches that had performed well using English. Other studies have successfully used n-grams to identify authors of texts (Kešelj et al. 2003; Houvardas and Stamatatos 2006). In web page classifications, a study based on n-grams and URL information showed improvements over an unigram approach (Kan and Thi 2005).

Soft computing using fuzzy sets and fuzzy logic (Klir and Yuan 1995) is one of the methods that has had some success in text classifications. It has been used together with support vector machines (Wang and Chiang 2007; Abe 2015), neural networks (Pal and Mitra 1992), and genetic algorithms (Yuan et al. 1997). Other approaches in soft computing have also been proposed (Lewis and Gale 1994; Jiang et al. 2011).

Another text processing method gained popularity when Blei et al. (2003) created topic models through latent dirichlet allocation (LDA), and has since become widely used. Further developments in topic models range from unsupervised models (Blei and Lafferty 2006; Blei 2012) to supervised models (Mcauliffe and Blei 2008). Topic model classifications have been successfully used in: tag recommendations (Krestel et al. 2009); web spam filtering (Bíró et al. 2008); and automatic transcriptions (Morchid et al. 2014). Research using LDA has also shown that it can improve performance over cluster-based approaches in information retrieval (Wei and Croft 2006), and a derived implementation DiscLDA has shown potential in reducing the error rate in

classifications compared to features extracted with LDA (Lacoste-Julien et al. 2009).

Sentiment analysis is another research area in information retrieval that has gained traction during the last two decades. Most of the research in the field started out with extraction of sentiment polarity and opinions in different types of reviews (Pang et al. 2002; Turney 2002; Dave et al. 2003; Hu and Liu 2004). From there, the research started covering also other parts of text classifications (Turney and Littman 2003), and further research added new improved methods (Pang and Lee 2005; Liu 2012). SentiWordNet was then developed as a freely available lexical resource for unsupervised sentiment analysis (Baccianella et al. 2010). Both supervised (Go, Bhayani, and Huang 2009), unsupervised (Thelwall et al. 2010), and semi-supervised (Maas et al. 2011) sentiment analysis algorithms have been developed. Sentiment analysis has also been used to classify Twitter tweets and messages on message boards like Yahoo! (Thelwall et al. 2011; Das and Chen 2007; Pak and Paroubek 2010; Kouloumpis et al. 2011).

Other relevant approaches have been researched that do not directly fall under any of the methods mapped so far. Self-organizing maps (SOM) have been used to classify texts (Merkl 1998), and researchers have used SOMs to organize patents into clusters (T. Kohonen et al. 2000). Another study used WordNet text classification with hypernyms instead of using a bag-of-words approach (Scott and Matwin 1998). Culotta and Sorensen (2004) used dependency tree kernels in classifications. Lastly, we have classifications done with features taken from co-occurring words (Figueiredo et al. 2011). Co-occurrences are network methods, which we will go more into detail about in the financial analytics part of the thesis. There are also a couple of relevant studies on feature selection in text classifications (Scott and Matwin 1999; Forman 2003).

In state-of-the-art methods, researchers have shown that combining features from different approaches in automatic classifications can improve performance over features based on any single method. Wallach (2006) combined a bag-of-words approach with n-gram to show increased classification performance. A study by Melville, Gryc, and Lawrence (2009) combined sentiment with text classification, which showed promising results. Others have also combined sentiment analysis with LDA (Lin and He 2009), and sentiment analysis with n-gram classification (Bespalov et al. 2011). More recently, Kusner et al. (2015) built classification models using word embedding's and the k-nearest neighbors algorithm.

Specific research into violent and hateful text content classification is quite limited. This was one of the areas that the companies in the D2I project had identified as problematic classification categories. Thus far, research on violent text content has been done through voting ensemble classification (Guermazi et al. 2007). Violent event extraction from news has also been studied (Piskorski et al. 2007). Basic classification research on textual hate content, which often also covers racism, has been published for web content (Warner and Hirschberg 2012), on twitter content (Kwok and Wang 2013), and on web page comments (Djuric et al. 2015). Specific techniques for web content classifications have also been researched in depth (Lee et al. 2002; Du et al. 2003).

After having mapped the current state of the art research, we can see that there is a clear lack of advanced research into classifying violent and hateful text content. More specifically, there is a lack of research into using combinations of features from different algorithms. This is where three of the publications presented in this thesis fit in the research landscape [2,3, 6]. Our research in this area focuses on combining different methods such as TF-IDF similarity analysis, n-gram analysis, and sentiment features to classify hateful and violent content. We then extend the research to see if majority voting ensembles can increase classification performance (Dietterich 2000). After that we further extend the research into what can be called a multi-gram classification approach, which has some similarities to (Shen et al. 2006) and (Wallach 2006). Finally, we examine the performance of the different models on datasets with imbalanced distributions.

1.3.3. Financial Analytics

Finance is a wide field even when limited to only research in analytics. The relevant state of the art research that will be mapped here is research that was done on text data. In the financial analytics part of the thesis, we will review methods that can be used to automate parsing of financial texts. Thus, we will here review previous research done using network theory, sentiment, and risks. Later we will cover how we can combine all three areas into a single text processing approach. However, mapping the entire state of the art financial research in risks, networks, and sentiment is still too broad. To further limit the scope, we will only consider methods that combine two or more of the areas (sentiment, network theory, and risks), and we are specifically interested in the methods that are applied on text data.

Before the millennium shift and the dot com crash, the majority of quantitative research in finance was done using numerical data gathered from different company metrics, different reported country numbers, and market data. Around the same time analytics started to gain traction ("Analytics Informs" 2016). One of the seemingly eternal debates in economics is whether the market should be seen as efficient or not (Fama et al. 1969; Malkiel and Fama 1970; Jensen 1978; Grossman and Stiglitz 1980; Malkiel 2003). The hypothesis that hidden in financial statements, and other financial texts, is information that has not yet been taken into account by the markets is one of the ideas that has impacted the research using text in finance (Bloomfield 2002).

While the market can still be seen as efficient over a long period of time, as shown by (Malkiel 2005; Tóth and Kertész 2006), the research into the efficient market hypothesis has continued and markets have been shown to be inefficient in many specific ways (Timmermann and Granger 2004; Baker and Wurgler 2007). For example, in 2009 researchers were able to show that market predictions can be done through textual representations (Schumaker and Chen 2009). A year later a study was able to prove that psychology has an impact on the markets, and that investor mood actually can be used to predict the movements in the market based on text (Pak and Paroubek 2010). However, we should also remember that financial research is about more than just market movements. Researchers have used text analytics to predict company revenues (Asur and Huberman 2010), market risk analysis has been done based on text (Groth and Muntermann 2011), and bank risks have been analyzed based on text (Rönnqvist and Sarlin 2014).

In two publications that are part of this thesis [4] and [5], we have built networks from news content. These networks have a basis in network theory (Davis et al. 1979; Ahuja et al. 1993) and in financial theory (Allen and Babus 2008). More specifically, in our research we use co-occurrence networks (Lund and Burgess 1995; Veling and Van Der Weerd 1999; Leydesdorff and Vaughan 2006) and weighted networks (Newman 2004). To be able to quantify nodes in a network, researchers have developed different types of centrality measures (Stephenson and Zelen 1989; Borgatti 2005; Estrada and Rodríguez-Velázquez 2005; Brandes and Fleischer 2005; Bonacich 2007). Using network theory, the following studies have been conducted: in one study, researchers built co-occurrences of people identified in the Reuters news corpus (Özgür et al. 2008); in another series of studies, researchers build a ranking of company co-occurrences based on social media content (Jin et al. 2009); researchers have from texts built networks of banks (Boss et al. 2004; Rönnqvist and Sarlin 2015).

When it comes to sentiment research in finance, the roots are the same as the sentiment research presented in automatic classifications. However, the applications of sentiment analysis is slightly different, as sentiment in finance does not necessarily have the same tonality as in, for example, classifications of positive and negative texts (Loughran and McDonald 2011). Nonetheless, news and investor sentiment have been shown to have an impact on markets (Brown and Cliff 2004; Baker and Wurgler 2007; Tetlock 2007). Researchers have shown that sentiment in finance can be structured into several categories, such as market wide sentiment, industry wide sentiment, and company sentiment (Mitra and Mitra 2011). This is further explored in our research. Studies have quantitatively shown that sentiment measures can predict markets (Schmeling 2009). In the fourth publication presented in this thesis [4], we build upon ideas on network theory and sentiment to create different company sentiment rankings. This is an avenue in finance that has not been widely explored before.

Risks in economics and financial markets have been widely studied and are at this point fairly well defined (Jorion 1997; Campbell et al. 1997). There are several studies that combine risks and networks, such as, for example, banking risks (Garratt et al. 2011) and systemic risks in financial systems (Eisenberg and Noe 2001). There are also books that cover most of the subject (Lando 2009; Allen and Gale 2009). Other studies that uses text data to measure risks consists of (Battiston et al. 2012; Duca and Peltonen 2013; Mezei and Sarlin 2017). In the fifth publication [5], we further develop the network and sentiment research from the fourth [4] publication to quantitatively measure the networks.

2. Research Methodology and Research Questions

In this chapter, the publications that the thesis consist of are presented. Following that we briefly review the different research methodologies used in information systems. Based on these methodologies follows a table that shows which methodologies we follow in the different publications. At the end of the chapter, the delimitations of the thesis are defined, and lastly the research questions that the thesis addresses are formulated.

2.1. Outline of Thesis Publications

[1] Forss T., Liu S., and Bjork K. M. (2014). Extracting People's Hobby and Interest Information from Social Media Content. Presented at the Terminology and Knowledge Engineering Conference, TKE 2014, 19 – 21 June 2014, Berlin, Germany.

We developed methods for extracting hobbies and interests from social media profiles through key word analysis using named entities, term weighting, stop words, and regular expressions. My part of the work was performing all the technical work and writing half of the article as first author.

[2] Liu S. and Forss T. (2014). Web Content Classification based on Topic and Sentiment Analysis of Text. Presented at the International Conference on Knowledge Discovery and Information Retrieval, KDIR 2014, 21-24 October 2014, Rome, Italy.

We did our first classification work on the problematic hate and violence categories using a naïve Bayes algorithm with similarity and sentiment features individually and combined. My part of this work was performing the software development work and feature extraction.

[3] Liu S. and Forss T. (2014). Combining N-gram based Similarity Analysis with Sentiment Analysis in Web Content Classification. Presented at the International Conference on Knowledge Discovery and Information Retrieval, KDIR 2014, 21-24 October 2014, Rome, Italy.

We extended our previous classification work to n-gram models for all categories in the dataset using a naïve Bayes classifier combining n-gram similarity and sentiment features. My part of this work was performing software development work and feature extraction.

[4] Forss T. and Sarlin P. (2016). From News to Company Networks: Co-occurrence, sentiment, and information centrality. Presented at the IEEE Symposium Series on Computational Intelligence, SSCI 2016, 6-9 Dec 2016, Athens, Greece.

We gathered a financial news dataset and used network analytics to create networks and rankings out of components in the dataset. My part of the research was the technical work and writing the article as first author.

[5] Forss T. and Sarlin P. (2017). News-sentiment networks as a company risk indicator. Under review at Journal of Network Theory in Finance.

We continue our work on networks by applying risk analysis to the networks and extracting risks for individual companies. We then show that high risk increases chance of stock price decrease. My part of the research was the software development and writing the article as first author.

[6] Forss T. (2018). Feature Enrichment through Multi-gram Models. Proceedings of the 51st Annual Hawaii International Conference on System Sciences. Paper accepted and presented during the conference, 3–7 Dec 2018, Big Island, Hawaii.

Introducing a new multi-gram classification model that enriches the feature set. This is done by combining different order n-gram models with sentiment analysis into one model. I did the both the research and the article on my own.

2.1.1. Other Relevant Publications

In section 2.1, I selected the six most relevant publications to be part of the thesis. Below follows a list the other relevant publications in analytics that I have been part of, but chose to not include in the thesis. The first two research articles were part of our automatic classification research. In these articles, the focus was on improving performance on imbalanced datasets, however, the methods tested did not perform as we hoped and were replaced by publication [6]. In the third publication, we extract text tags related to images, which is not included in the thesis as it does not have any direct link to either of the automation approaches.

Liu, S., & Forss, T. (2015). Text Classification Models for Web Content Filtering and Online Safety. Presented at the International Conference on Data Mining Workshop, ICDMW 2015, pp. 961-968, 14-17 November 2015, Atlantic City, USA.

Liu, S., & Forss, T. (2015). New classification models for detecting Hate and Violence web content. Presented at the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2015, Vol. 1, pp. 487-495, 12-14 November 2015, Lisbon, Portugal.

Liu, S., & Forss, T. (2015). Automatic tag extraction from social media for visual labeling. Presented at the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2015, Vol. 1, pp. 504-510, 12-14 November 2015, Lisbon, Portugal.

2.2. Research Methodology

Research methodologies in Information Systems (IS) are based on six categories introduced by (Jarvinen 2000): mathematical approaches, conceptual analytical approaches, theory-testing approaches, theory-creating approaches, artifact building approaches, and artifact evaluating approaches. All publications in this thesis are worked out as Mathematical approaches, Artifact building approaches, and/or artifact evaluation approaches.

(Von Alan et al. 2004) defines the used of design science in IS as "a purposeful IT artifact created to address an important organizational problem." They define design science as the following seven steps: 1) Provide a viable artifact; 2) Develop a technology-based solution; 3) Rigorous demonstration of artifact via evaluation; 4) Provide clear and verifiable contributions; 5) The research must rely on rigorous methods; 6) Satisfy laws in the problem environment; 7) The research must be effectively presented to technology-oriented and management-oriented audiences. Based on their definition, the publications included in the thesis can be labelled as applications (implementations) of design science. (Von Alan et al. 2004)

Iivari (2007) argues that Information Systems is an applied science and that this is now widely accepted. Iivari continues by saying that IS as a design science should be based on a sound typology of IT artifacts, especially research consisting of IT applications. He goes on to define seven categories that IS applications should fall into: to automate, to augment, to mediate, to inform, to entertain, to artisticize, and to accompany. Applications using design science in IS research can be a combination of the seven different artifacts. The publications in this thesis mainly belong to the categories to automate, to augment, and to inform. Finally, he suggests defining design research in IS as constructive research and that we should follow constructive methods. Constructive research

generally means testing the research contribution analytically, with some predetermined criteria. These criteria can range from surveys sent to participants to statistical benchmarks or comparisons of results from different methods on the same datasets. Our quantitative tests are compared to baselines and benchmarks that we define. (Iivari 2007)

There are also a set of guidelines for when research in IS design science can be considered novel contributions. These guidelines specify that the research has to either be research that has not been done before, or the research should improve upon existing results in some way (Von Alan et al. 2004; Hevner and Chatterjee 2010). As part of action design research (ADR), a new artifact, the ensemble artifact, was defined (Sein et al. 2011). The ensemble artifact also accounts for the organizational domain. ADR consists of two steps: problem formulation and building, intervention, and evaluation (Sein et al. 2011). The problem formulation step can come from a practical problem perceived by either the researchers or an industry. The second step is a continued building and evaluation process based on step one. Using that definition, both the automation tasks that will be presented are considered action design research. The text classification research problem was identified by our security industry partner, and the financial news automation research problem was identified by us, the researchers.

The state of the art in Information Systems research methodology moves the discussion to what Grover and Lyytinen (2015) calls the mid-range script, and how most innovations in the field currently are done by applying mid-level modifications of theories from other fields onto Information Systems problem formulations. Grover and Lyytinen (2015) suggests two approaches that can move IS research forward and enable ground breaking research studies. They call the two approaches working on the left edge and working on the right edge. Working on the right edge is defined as building new innovative IS theories through thought experiments. Working on the left edge is defined as identification of patterns, observations, and descriptions. The left edge refers to observing the world as-is, will-be, and situations that not necessarily yet have theories that match them. In my opinion, the research that we have been conducting in financial news analytics falls into the left edge category. (Grover and Lyytinen 2015)

If we change perspective, we can also split research methodologies in IS into deductive and inductive research. Deductive research is used to test an existing theory using, for example, some form of Information System. Usually, it is done through some form of quantitative evaluation method. Inductive research is

generally used to test theory with some form of qualitative evaluation method (Strauss, et al., 1990; Glaser, 1992). In practice, we can also combine both qualitative and quantitative approaches into what can be labelled as mixed method research (Bryman 2006).

Research in IS can also be considered either prescriptive or descriptive. In prescriptive research, we identify and recommend solutions. This can be done by either showing which methods improve results or by pointing out where further research into the subject could improve results. Descriptive research, on the other hand, describes characteristics without trying to answer questions of why the results are showing what they do. We take a prescriptive approach in all our research, trying to improve results and find further improvements that can be implemented in the future. (March and Smith 1995)

2.2.1. Methodology in Publications

The research methodologies previously described are here specified for each publication and listed in Table 1. The Research methodology column describes which type of research was followed. The Artifact column describes what the research accomplished. The Evaluation method column describes which kind of result validation we used in the publication.

2.2.2. Mathematical Methods

Researchers often use quantitative mathematical methods to be able to evaluate research in Information Systems, especially research in analytics. There are many different mathematical methods that can be used to solve problems (McLeod and Schell 2001). The evaluation methods need to be robust enough to satisfy the research methodologies chosen in the research. In this section, we will first review different relevant mathematical evaluation methods, then in section 2.2.2.5 there is a summary explaining which of the methods that are used in our research. Through discussing both the methods used and some not used, we explain why the mathematical methods were chosen in the research papers.

Paper	Research Methodology	Artifact	Evaluation method
1	Constructive ADR, deductive research, prescriptive research	Social media extraction information system	Survey with user feedback
2	Constructive ADR, deductive research, prescriptive research	Text classification models for hate and violence in text	Cross-validation with labelled dataset
3	Constructive ADR, deductive research, prescriptive research	Text n-gram classification models for 20 category types	Cross-validation with labelled dataset
4	Constructive ADR, prescriptive, quantitative, and qualitative research	Two different rankings of companies: one absolute and one normalized	Qualitative and quantitative analysis of the rankings and networks
5	Constructive ADR, prescriptive, quantitative, and qualitative research	Extracting individual, direct, and indirect sentiment risks for individual companies	Quantitative analysis of network risks and individual risk compared to benchmark
6	Constructive ADR, deductive research, prescriptive research	Ensemble and multi-gram classification models	Cross-validation with labelled dataset compared to baseline

Table 1*. Research methodologies, artifacts, and evaluation method for each publication.*

2.2.2.1. Machine Learning

Machine learning algorithms are data driven approaches that focus on the predictive side of analytics. This means that machine learning algorithms, such as support vector machines (SVM) (Hearst et al. 1998), artificial neural networks (ANN) (Schalkoff 1997), decision trees (DT) (Quinlan, 1986), k-nearest neighbors (k-NN) (Larose 2005), and naïve Bayes (NB) (McCallum et al. 1998), which we will be using, do not have any starting proposition that they try to verify. Instead, the algorithms are used to find patterns in data based on features that describe the different data points. These patterns can then be used to predict where new data points belong. If we use the same data to test different approaches and machine learning algorithms on, we can statistically compare performance and generalize conclusions.

2.2.2.1.1. Naïve Bayes

The naïve Bayes (NB) algorithms used in automatic classifications are based on Bayes' theorem. Naïve algorithms assume that each feature used is independent of other features. There are several different types of distributions that naïve algorithms can use: Gaussian, Multinomial, and Bernoulli. Gaussian distribution is the same as normal distribution and is used when dealing with theoretically infinite data. Multinomial algorithms are used when there is a finite number of categories and a finite number of instances to be classified. Bernoulli distribution is used when feature vectors are binary, which means values can either be 0 or 1. In our experiments, we use the NB classifier as a baseline due to its simplicity. We have a limited number of categories and instances, which means we use the multinomial distribution. NB satisfies the equation defined in (1) when we are performing binary classifications, and the multinomial distribution is defined in equation (2). C_k is the class, p is the probability, k represents the multinomials (can be ignored for binary classifications), and f is the feature set: (Russell and Norvig 2002)

$$p(f_1, \dots, f_n) \propto p(c)p(f_1|c) \dots p(f_n|c) \quad (1)$$

$$\log\big(p(C_k)\big) + \sum_{i=1}^{n} f_i \log(p_{ki}) \quad (2)$$

2.2.2.1.2. Decision Trees

Decision trees (DT) used in classification problems are called classification trees where binary trees are built. Trees are built to contain decision rules where

moving from one node to the next always includes a decision as in an if-else statement. Decision trees are simple in nature, and can be trained and applied fast. (Russell and Norvig 2002)

2.2.2.1.3. k-Nearest Neighbors

k-nearest neighbors (k-NN) is a classification and regression algorithm that uses instance-based learning. This means that training data is stored, and unclassified data points are compared to the data points in the training set. This means that to classify instances using k-NN we need to have a distance measure to compare data points. The distance function we will use is Euclidean distance, which is used for non-categorical attributes as shown in equation (3), where *x, y,* and *z* represent features of the data point, and where the distance *d* satisfies the restrictions (4)-(6): (Larose 2005)

$$d_{Euclidean}(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \quad (3)$$

$$d(x, y) \geq 0, and\ d(x, y) = 0, \text{if } x = y \quad (4)$$

$$d(x, y) = d(y, x) \quad (5)$$

$$d(x, z) \leq d(x, y) + d(y, z) \quad (6)$$

2.2.2.1.4. Support Vector Machines

Support vector machines (SVM) are models that use hyperplanes for classifications, clustering, and regressions. Here we will only cover classifications. SVM models can be either non-probabilistic linear classifiers or non-linear classifiers using kernel functions. For non-kernel versions of SVM, a hyperplane is a subspace of one dimension less than the space that the task is defined in. The algorithms try to find the hyperplane that offers the largest margin between classes or clusters, which leads to lower generalization errors. In our research, we use the linear SVM model.

In the case that classes in the dataset are linearly separable, two hyperplanes that are parallel can be selected to maximize margin (known as hard-margin) and separate classes as far as possible from each other. Such hyperplanes are represented as in equation (7) and (8), where w is the normal vector to the hyperplane, x is from the dataset $\chi = \{x_1, \dots, x_i\}$, and b is a scalar value that determines the offset (bias) of the hyperplane. Expressing these as a linear

function $f(x)$, we can minimize the optimization problem that arises as in equations (9) - (11), where χ is the dataset and y_i indicates the class that x_i belongs to. In the case that the points are not linearly separable (known as soft-margin), we need a hinge loss function, and the minimization problem then changes to how it is represented in equation (12), where λ determines the margin size (Boser et al. 1992; Cortes and Vapnik 1995; Schölkopf et al. 2000):

$$w \cdot x - b = \quad 1 \tag{7}$$

$$w \cdot x - b = -1 \tag{8}$$

$$f(x) = (w \cdot x) + b, \quad w, x \in R^N, b \in R \tag{9}$$

$$\min\{|f(x)| : x \in \chi\} = 1, \quad \text{where } \chi = \{x_1, \dots, x_i\} \tag{10}$$

$$(x_1, y_1), \dots, (x_i, y_i) \in \chi \times \{\pm 1\} \tag{11}$$

$$\left[\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(w \cdot x - b))\right] + \lambda\|w\|^2 \tag{12}$$

If we are dealing with large datasets and/or sparse data, there is also the possibility of using a non-linear sub-gradient descent approach or a coordinate descent approach. These approaches can reduce training times depending on the implementation of the algorithm used, the number of features, and the number of instances used in the model (Shalev-Shwartz et al. 2010). However, neither of these approaches are used in our experiments as we do not have problems with training times. The calculation for a sub-gradient descent approach is as follows, where f is a convex function of $\vec{w}$ and b:

$$f(\vec{w}, b) = [\frac{1}{n}\sum_{i=1}^{n} \max\big(0, 1 - y_i(wx_i + b)\big)] + \lambda\|w\|^2 \tag{13}$$

A coordinate descent approach can be appropriate if a large set of features are used in the training. Equation (14) shows the formula for the coordinate descent approach, where c_i is the coefficent that is adjusted iteratively and projected onto the nearest vector that satisfies equation (15). The iterative process is repeated until close-to-optimal coefficients are found: (Hsieh et al. 2008)

$$f(c_1 \dots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (x_i x_j) y_j c_j \tag{14}$$

The function is maximized and subject the following two restrictions for all i:

$$\sum_{i=1}^{n} c_i y_i = 0, \text{ and } \quad 0 \le c_i \le \frac{1}{2n\lambda} \tag{15}$$

2.2.2.1.5. Artificial Neural Networks

Artificial neural networks (ANN) can be split into the following groups: dynamic neural networks, static neural networks, memory networks, and a number of smaller groups (Gupta et al. 2004; Weston et al. 2014). The type of neural networks that we use are from the dynamic group. Out of these, we use multi-level feedforward neural networks with back propagation (Hornik et. al. 1989). Recurrent neural networks (RNN) (Schuster and Paliwal 1997) is another of the well-known neural networks that has been used in, for example, handwriting recognition, speech recognition, and time series prediction. Long short-term memory (LSTM) is a form of RNN that more recently has shown promise also in text classifications (Hochreiter and Schmidhuber 1997).

Feedforward neural networks are those where connections between neurons do not form directed cycles, which is the main difference to the RNNs that can process data in cycles. As the connections do not form circles, the information only flows in one direction: from input nodes to hidden nodes and finally to the output nodes. The networks can consist of one or many different layers of perceptrons. A perceptron is a function $f(x)$ that is used to decide whether input belongs to a specific class or not, and can mathematically be described as in equation (16), where w is the weight, x is the input value, and b is the bias. A perceptron's output is binary, and it takes input in the form of vectors of numbers. In multi-layer perceptron networks (MLP), a back-propagation algorithm is used to recalibrate weights during training. This means that there is a weight update phase added to the network to help improve classification performance. Training a two-class classification with MLPs is usually done by minimizing a criterion $Q(f_\theta(.),.)$, which usually is either mean square error criterion or cross-entropy criterion, over the training data as in equation (17) using gradient descent until reaching a local optimum. Where θ is the vector parameters (w, b) and x_l is the l^{th} example of a training set $(x_l, y_l)_{l=1..L}$ with $(x_l, y_l) \in \mathbb{R}^n \times \{-1, 1\}$: (Collobert and Bengio 2004)

$$f(x) = \begin{cases} 1 \text{ if } w \cdot x + b > 0 \\ 0 \text{ otherwise} \end{cases} \tag{16}$$

$$\theta \longrightarrow \frac{1}{L} \sum_{l=1}^{L} Q(f_\theta(x_l), y_l) \tag{17}$$

Each perceptron contains an activation function (usually a sigmoid) that decides the output of that neuron. The two general formulas to choose from in MLP activation functions are the following (Russell and Norvig 2002; Haykin and Simon 2004):

$$f(x) = \tanh(x) \quad \text{and} \quad f(x) = (1 + e^{-x})^{-1} \tag{18}$$

The first uses a hyperbolic tangent with an output from -1 to 1, and the second is a logistic function with an output that ranges from 0 to 1. In order to increase performance through back propagation, we need to measure and minimize the error $\epsilon(n)$ in equation (19). Changes in weight based on the error for the learning algorithm can then be found through a gradient descent algorithm (20), where η is step size, w_{ji} is the synaptic weight connecting neuron *i* to neuron *j*, and y_i is the class (Haykin and Simon 2004):

$$\varepsilon(n) = \frac{1}{2}\sum_j e_j^2(n) \tag{19}$$

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n) \tag{20}$$

2.2.2.1.6. Ensemble Classifications

Ensemble classifications are statistical machine learning methods that combine several classification algorithms to increase the performance over single classifiers. Some common ensemble classifiers are bagging, boosting, and stacking. The Bayes optimal classifier has been shown to on average outperform all other ensemble classifiers (Hoeting et al. 1999), but is in practice seldom suitable due to computational requirements, and because the output of algorithms are generally a single class value, when the algorithm would need probabilities for all classes to function.

The bagging algorithms are voting algorithms that give equal weights to classifiers. Boosting algorithms train algorithms in series where later algorithms try to reclassify instances that the previous algorithms failed to classify correctly. The stacking algorithm trains a classifier on the combined output of the other models.

We will be using the majority voting algorithm that gives equal weight to all classification algorithms ensembled. Majority voting can be mathematically represented as in equation (21), where $y_k(x)$ is the classification of classifier number k and $g(y, c)$ in equation (22) is an indicator function: (Rokach 2010)

$$class(x) = arg_{c_i \in dom(y)} \max\left(\sum_k g\left(y_k(x), c_i\right)\right) \tag{21}$$

$$g(y, c) = \begin{cases} 1, & when\ y = c \\ 0, & when\ y \neq c \end{cases} \tag{22}$$

2.2.2.2. Fuzzy Logic and Evolutionary Computing

Soft computing was introduced by Zadeh (1994) and is a field that has evolved and matured during the last twenty years. Soft computing consists of different machine learning algorithms, but also covers methods using fuzzy logic and evolutionary computation. The aim of soft computing is to be able to model complex and dynamic situations such as, for example, human behavior or adaptive systems (Björk 2009). Soft computing models have a wide space that they can model. Soft computing models approaches problems by adding a scale of preciseness. Other methods in classifications and clustering normally say that instances belong to one class or the other. Fuzzy methods, on the other hand, use degree of belonging, which means that instead of saying an instance belongs to class A or class B, they determine a degree of belonging to all available classes. Each instance that is being clustered or classified is given a value between zero and one representing belonging to each cluster or class, and is normally said to belong to the one that has the highest degree of belonging (Klir and Yuan 1995).

Evolutionary computation is another type of method that comes from soft computing and can be used to build models and, for example, test classifications. Conceptually, evolutionary algorithms can be seen as the Darwinian process of evolution where better performing algorithms continue evolving and worse algorithms are discarded. However, both technically and in practice, the process is better described as following a trial and error approach (Eiben and Smith 2003). Neither fuzzy methods nor evolutionary algorithms have been used in the scope of our research.

Genetic algorithms are the most widely used type of algorithm in evolutionary computing. They work by defining a fitness function for the problem that compares performance between different simulations. After each simulation, the worst performing algorithms are discarded. From the remaining algorithms, new ones are then created through a process called crossover, where two algorithms are combined into one new algorithm that again is sent for evaluation. The process is then repeated until a best performing algorithm is found.

2.2.2.3. Self-Organizing Maps

Self-organizing maps (SOM) are unsupervised algorithms for clustering different types of data (Kohonen and Somervuo 1998). Unsupervised algorithms are suited for grouping of data into clusters that are not predetermined. This means that SOMs are not the best choice in situations where we would be interested in

labelling data into predetermined categories. This makes SOMs inappropriate for the supervised classification tasks researched in this thesis.

2.2.2.4. Network Measures

Network measures are quantitative methods for measuring and analyzing nodes and edges in networks. Nodes are also known as vertices and edges are also known as links. There are several different types of network measures that are applicable in different situations, depending on the network and what we are interested in analyzing. In some situations, it can be appropriate to find the shortest path between two nodes. In other situations, it could be appropriate to search for the longest path. In many situations, such as in our financial news research, we are interested in measuring how different nodes are connected to each other, as well as how information propagates through the different links in the networks at the same time. In these situations, the shortest path and longest path algorithms are not suitable.

Borgatti (2005) showed that not all types of centrality measures are suitable for all types of networks. His work has also been further extended to test more measures (Amrit and ter Maat 2016). There are several different algorithms that can be used to measure information flow between nodes, the ones that we examine are degree centrality, closeness centrality, betweenness centrality, and eigenvector centrality. Next, we will review these four different types of network measures in more detail.

2.2.2.4.1. Degree Centrality

Degree centrality is simply the number of links a node in the network has. In directed networks, there are two different degree measures for each node: in-degree is the edges coming in to a node and out-degree is the edges going out from a node. This is the simplest centrality measure. In our research, degree centrality would simply provide us an absolute order of media attention, which we are not especially interested in. Degree centrality for a node in a network is mathematically represented as in (23), where v is the node in question and $C_D(v)$ is the degree centrality value (Friedkin 1991):

$$C_D(v) = \deg(v) \tag{23}$$

2.2.2.4.2. Eigenvector Centrality

Eigenvector centrality measures influence of nodes in a network and is a more complex version of degree centrality. The links between nodes, also known as edges, is what Eigen centrality measures. Edges to higher scoring nodes are given greater influence than edges to lower scoring nodes. The relative eigenvector value for a node can be calculated using equation (24), for a graph $G := (V, E)$ with $|V|$ vertices and the adjacency matrix $A = (a_{v,t})$, where $M(v)$ is the neighbors of node v and λ is a constant: (Bonacich 2007)

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t \tag{24}$$

PageRank is related to eigenvector centrality, but has an added scaling factor. PageRank is the original algorithm behind Googles search engine and is calculated as in (25), where *i* and *j* are nodes in the network and $L(j) = \sum_j a_{ji}$ is the number of neighbours to the node *j*: (Page et al. 1999)

$$x_i = \sum_j a_{ji} \frac{x_j}{L(j)} + \frac{1-\alpha}{N} \tag{25}$$

2.2.2.4.3. Closeness Centrality

Closeness centrality measures the distance between nodes. Between all nodes in a network there is a shortest path. Closeness centrality is measured as the average shortest path from one node to all other nodes in that network. The assumption that closeness centrality follows is that information is transferred along only the shortest path (Brandes and Fleischer 2005), which also disqualifies closeness centrality from being used in our research, as we are interested in measuring information spreading in all directions at the same time. Mathematically, we can represent closeness centrality for a node as in equation (26), where $d(j, i)$ is the distance between two nodes j and i in the network (Brandes and Fleischer 2005):

$$C(i) = \frac{1}{\sum_j d(j,i)} \tag{26}$$

Information centrality is another closeness measure that was defined by (Stephenson and Zelen 1989). Information centrality calculates the harmonic mean of edges instead of the average shortest path. This allows information to flow through each node in a network simultaneously. Information centrality is thus better suited to model flow through multiple paths throughout a network than the standard closeness measure. Information centrality for a node in a

network is calculated as in equation (27), where the pseudo adjacency matrix A is defined as in (28), $S(i)$ is the strength of node i, w is the edge weight, and B is the matrix. We will be going more into detail into the method in section 4.3.5. (Stephenson and Zelen 1989):

$$C(i) = \frac{n}{nA_{ii} + \sum_{j=1}^{n} A_{jj} - 2\sum_{j=1}^{n} A_{ij}} \tag{27}$$

$$A = B^{-1}, B_{ij} = \begin{cases} 1 + S(i), & if\ i = j \\ 1 - w_{ij,} & \text{else} \end{cases} \tag{28}$$

2.2.2.4.4. Betweenness Centrality

Betweenness centrality is a measure that quantifies the number of times a node is found among the shortest path between other nodes. Betweenness has been used in studying human communication in social networks. Nodes that often are found in the shortest path between other nodes are given a higher betweenness value. Betweenness centrality has the same limitation that closeness centrality has, it does not model multiple paths simultaneously (Brandes and Fleischer 2005). The formula for calculating betweenness centrality for a node is as in equation (29), where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of paths that pass through the node v (Brandes and Fleischer 2005):

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{29}$$

2.2.2.4.5. RiskRank

RiskRank is a network measure that can measure risks. The model has some similarities to information centrality in the sense that it also accounts for multiple flows through a network. The calculation of the RiskRank measure RR is a combination of the Choquet integral and the Shapley index $v(c_i)$ by Tarashev et al. (2010), and is defined in equation (30). It has a limitation, which is that interlinkages $I(c_i, c_j)$ between nodes are limited to pairs of nodes. $v(c)x_c$ is the individual node risk. We will be using RiskRank in our last experiment and we will discuss the method in more detail in section 3.3.2.1 (Mezei and Sarlin 2017; Tarashev et al. 2010)

$$RR(x_1, \dots, x_n, x_c) =$$

$$v(c)x_c + \sum_{i=1}^{n} (v(c_i) - \frac{1}{2} \sum_{j \neq i} I(c_i, c_j))x_i \qquad (30)$$

$$+ \sum_{i}^{n} \sum_{j \neq i}^{n} I(c_i, c_j) \prod (x_i, x_j)$$

2.2.2.5. Summary of Methods

In our automatic classification research, we begin by comparing different pre-processing approaches using the NB machine learning algorithm (section 2.2.2.1.1), mainly because the algorithm is fast for testing and training, and because we are interested in defining a baseline performance that can be used to compare the relative performance between approaches. By using NB, we save some time on training models and have the possibility of testing many approaches before we start optimizing the classifications. We then extend our methods and compare the performance between the individual classification algorithms DT, SVM, ANN, and k-NN (sections 2.2.2.1.2 – 2.2.2.1.5). The last mathematical method we work with in classifications is the majority voting ensemble (section 2.2.2.1.6). While ensemble classifications increase the computational requirements both for training and prediction, they can in some cases increase performance.

The human cognitive abilities are limited, and there is a limit to the amount of information that we can process without getting overloaded and losing focus. Centrality measures can be used as a means of reducing the noise by pointing us to the most important nodes in networks. In our financial news research, we start by using information centrality (section 2.2.2.4.3) to analyze the networks quantitatively and qualitatively, as it allows information to flow through multiple paths. In the last part of the financial research, we change to using RiskRank (section 2.2.2.4.5) as our evaluation approach, which allows us to statistically compare different risk thresholds.

Table 2 shows an overview of the mathematical methods that were used in the different publications.

Paper	Method used	Result
1	Key word extraction based on TF-IDF weighting, NER	No quantitative data output, evaluation done through survey
2	Machine learning algorithm: naïve Bayes	The classification models offer improved performance over the baseline
3	Machine learning algorithms: naïve Bayes	The classification model is compared to previous models and shows improvements
4	Information centrality measure used to rank companies	Quantitative measures in the form of ranked information centrality news flow, no baseline comparison available
5	Individual and aggregated risk measures through RiskRank	Individual and aggregated risks evaluated against benchmarks
6	Machine learning algorithms: SVM, NN, DT, k-NN, ensemble voting	The classification models are compared to the previous results and show improvements

Table 2. *Mathematical methods used in the different publications.*

2.3. Research Context and Analytical Methods

The research in this thesis is limited to two areas of analytics: automatic classifications and financial analytics. The research is based on analyzing and processing text and structural content. In this thesis, feature extraction is done based on text content and structural content. The dataset used in text classifications was provided by a security industry partner and is labelled into 20 categories, where one web page is labelled as belonging to only one category, and the total number of labelled pages is roughly 79,000. The focus of the research in this area is constructing methods for classifications of violent and hateful text content.

For the financial analytics part of the research, a dataset of roughly 18,300 articles were gathered and labelled in two distinct ways. First, the author of the article labels the article as either short or long. The label represents the author's expectations for the targeted company, index, or commodity. A short label could suggest that the author of the article has a negative sentiment toward the discussed components. A long label could suggest that the author has a positive sentiment toward to the components. Second, the articles are split and labelled into 7 subsectors. These sectors are technology, health care, consumer related, transportation, finance, energy, and others. In our research, we are mainly interested in examining different network effects that can be found and extracted from news.

2.4. The Research Questions

Four research questions are answered in this thesis. The point is to first broadly offer generalizable answers, and then narrow the focus with each successive research question until we have an understanding of the contributions. The first two research questions are general questions and are answered through both automation approaches. The first one was chosen to show that there is more than one approach in analytics to automating processing tasks. The second question was chosen to show how sentiment can be used in different ways to improve performance, to gain insights, and to show the similarities between the two automation approaches.

The third research question is specifically aimed at the first automation approach to provide in-depth knowledge of the methods used. The question was chosen to show that the state-of-the-art classification approaches can be used on hate and violence texts and to show how our methods go beyond the state of the

art. The fourth question is specifically aimed at the second automation approach, and aims to provide in-depth knowledge of the methods used. The question was chosen because the second automation approach is exploratory. By answering the fourth research question, we can show that the approach is valid. Without answering the fourth question, the approach could be questioned as it has not been done before.

The research questions are the following:

1. How can we use analytics to automate text processing tasks?
2. In which ways can sentiment analysis be useful when automating processing tasks?
3. What can be done to improve unigram classification performance for hate and violence texts?
4. Can risks extracted from sentiment networks predict company stock price movements?

3. Analytics

We have so far reviewed the state of the art in the relevant fields, the mathematical models used in the research, and the research questions that will be answered in the thesis. We now move on to define the different parts of analytics, how these parts are used in our research, how different types of methods can be useful to Neil and Jenna, and last in the chapter, we will go into more detail about the methods used in our research.

3.1. Overview

Analytics is a wide field that contains statistical analysis, data handling, visualization of data, exploratory modelling, predictive modelling based on historical data, recommendations based on different types optimizations, simulations, and more (Kohavi et al. 2002). Not only is analytics a wide field with many different areas of research, within each subfield there are also different processing methods used for different types of data. The methods used need to be modified to fit the right type of data: numerical data, textual data, image data, structural data, video data, and audio data to name a few. Furthermore, when the datasets that are analyzed increase in size, velocity, and/or complexity, we also need to consider the big data dimensions that were discussed earlier, which in turn can require that we use yet another set of analytical methods (Russom et al. 2011).

3.1.1. Descriptive Analytics

Descriptive analytics is the first of three major branches of analytics. Descriptive analytics comprises a set of methods focusing on analyzing and visualizing data. Most data gathered today are of such proportion and/or variety that they are difficult and sometimes even impossible for a human to understand the data without support from algorithms and visualization. Statistical methods and algorithms that help us understand, organize, and handle data are labelled as descriptive analytics (Evans and Lindner 2012). Some examples of visualization are histograms, diagrams, and graphs. Today, many companies have descriptive analytics capabilities, which help them make sense of the data they already have or already are creating. Descriptive analytics is quite common and companies that provide these types of services number in the thousands. Google analytics,

IMB Watson Analytics, and Microsoft Cortana Intelligence Suit are example products from some of the largest providers.

Generally, when taking analytics into use, the first part is to try and understand the available data through visualization and statistical analysis. In this thesis, both the research in classifications and financial news analytics falls partly under the descriptive analytics, mainly because we are exploring the available data using visualization techniques (graphs, plots, figures), to uncover new information and find trends. Descriptive methods can be useful for both Neil and Jenna as a means of understanding trends and obtaining an overview of the data.

3.1.2. Predictive Analytics

Predictive analytics is the second major branch of analytics. Predictive methods create models from historical data, also known as training data, and apply these models on new data to predict labels or behavior of new data. In other words, predictive analytics uses models built on historical data to predict how new data will behave. (Siegel 2013)

Once we understand the data, we can use different predictive models or algorithms to process the data into meaningful information. This can include building regression models, clustering data, creating classification models, or building models for optimization or simulation (Rasmussen and Williams 2006).

There are many algorithms that can be used in predictive analytics. The machine learning algorithms naïve Bayes, artificial neural networks, support vector machines, decision trees, and k-nearest neighbors were already briefly explained in section 2.2, and are part of the set of algorithms that are widely used in predictive analytics. Big-data-analytics methods, such as extreme learning machines (ELM), also fall under the predictive analytics branch (Huang et al. 2006).

The automatic classifications work presented in this thesis is part of the predictive analytics domain. We use historical textual data to train machine learning algorithms and test their predictive performance on sub sets of data that were not included in the learning process. We first use the training set to evaluate the performance of the classifications in what is known as cross-validation, then we test the performance using balanced sets (Browne 2000). When we have all our models built, we further test the best performing models using imbalanced test sets to see how the performance holds up. For Neil, predictive models can help reduce manual workload in web site classifications.

For Jenna, and in the finance industry in general, predictive models have higher uncertainty due to ever changing economic conditions, and can be used to help with investment decisions.

As a guiding rule, a model in automatic text classifications needs an F-measure of about 0.9 to be considered successful. The 0.9 F-measure performance was decided upon as an acceptable level after consulting our security industry partner for performance requirements. Predicting stock price movements, on the other hand, can be successful already at a hit rate of 0.54 (Hellström 1998).

3.1.3. Prescriptive Analytics

Prescriptive analytics is the third major branch of analytics and builds partly on the previous two parts of analytics. Prescriptive analytics focuses on choosing the best possible future outcome based on a set of predefined criteria. The predefined criteria are extracted and formatted through descriptive analytics. Based on the extracted data we can determine different outcomes. Prescriptive systems chooses the best possible outcome, often based on optimizations or simulations built on top of predictive models (Bell and Raiffa 1988). Recommender systems and decision support systems are examples of prescriptive systems. These kinds of systems suggest the action to take based on available information. For example, companies can use such systems to help determine if an investment is sound. The outcome of research presented in the thesis, both in classifications and financial news analytics, can be used as the basis for prescriptive analytics systems (Haas et al. 2011). Haas et al. (2011) argue that people developing analytics systems should strive to create prescriptive systems to take full advantage of the methods.

For Neil and Jenna, prescriptive methods are the easiest type of system to use as at that level the methods are sophisticated enough to generate recommendations without themselves having to analyze the data further. However, from a model development perspective, these methods take the longest to create as they often build upon the previous branches of analytics.

3.1.4. Advanced Analytics

The work in this thesis falls, per definition, under advanced analytics. The definition that Gartner provides is the following: “Autonomous or semi-autonomous examination of data or content using sophisticated techniques and

tools, typically beyond those of traditional business intelligence, to discover deeper insights, make predictions, or generate recommendations. Advanced analytics techniques include methods such as data and text mining, machine learning, pattern matching, forecasting, visualization, semantic analysis, sentiment analysis, network and cluster analysis, multivariate statistics, graph analysis, simulation, complex event processing, neural networks." Both our text classification research and the financial news research are combinations of the different mentioned methods, which is why the research in this thesis is considered advanced analytics. ("Advanced Analytics" 2017)

The optimal solution for Neil would be an autonomous, predictive classification system that has high precision so that the predictions can be trusted without any additional work required, while at the same time maintaining a high coverage, which here means maintaining a high F-measure. Essentially, being able to conclude "if the classification system says the web page is violent, then it is violent." The optimal solution for Jenna would be tools based on methods that have been shown to have statistically predictive power in some regard, for instance, in predicting stock movements. These tools would become optimal when they contain prescriptive functionality that optimizes the outcome. In the case of stock predictions, a tool would be optimal when it can tell Jenna the following: "investing in these stocks at this time will offer us the highest return on investment of the known alternatives."

3.2. Text Classifications

Knowing the type of data we will be working on is also important, because different feature-extraction methods are used on different types of data. For example, an algorithm for classification of text content will have a different approach than an image classification algorithm (Wilcock 2009; Javidi 2002). The work presented in this thesis is limited to methods in text analytics and numerical methods, which means that the models and methods presented are suitable for text content and structural content.

3.2.1. Automatic Classifications

Classification, also known as categorization, is the process of assigning predefined labels to data. One classic example of text classification is dividing documents by language, region, and/or subject (Jajuga et al. 2012). Classification for a human is considered an easy task, it is an ability that we have been

practicing our entire lives. However, it is a complicated task for a computer to perform.

In order to perform automatic classifications, whether we have text data or some other form of data, we need to first create classification models. These models can then be used to perform classifications of unlabeled data. Classification of unlabeled data falls under predictive analytics and is often referred to as predicting classes. The classifiers covered in this thesis are based on the machine learning algorithms NB, DT, SVM, ANN, and k-NN, the mathematical inner workings of which were covered in section 2.2.2.1. The different machine learning algorithms perform differently on different types of data and datasets. Therefore, it is common in research to compare performance between different types of machine learning algorithms. This is also why the results of several algorithms will be compared in this thesis.

To create classification models, we first need to extract features from the training sets. The features can at the start be textual or numerical. However, most machine learning approaches can only process numerical values. This means that the feature sets need to be converted into numbers to be usable in comparisons between the algorithms.

There are many different algorithms used to extract features in text classifications. Among the most widely used text-feature-extraction methods, are approaches such as TF-IDF weighting (Luhn 1958), cosine similarity analysis (Markov and Larose 2007), topic modelling (Papadimitriou et al. 1998), and sentiment analysis (Pang and Lee 2008). We will be going through all in greater detail in this chapter, except for topic modelling.

When performing topic similarity analysis, there are several different approaches that can be used. The vector space model that cosine similarity uses is simply one of the available options. Some other alternatives include using Wikipedia knowledge as a way of enriching available text data and generate features (Gabrilovich and Markovitch 2006) and corpus-based semantic similarity using SVD (Landauer et al. 1998).

Once we have extracted features for the labelled datasets, we feed these features into models that are based on the different machine learning algorithms. The algorithms then follow their respective approaches to generate a model for the input data. As our research does not include improving the machine learning algorithms themselves, there is no gain in re-implementing the machine learning algorithms. Because of that, all classifications and predictions in this thesis are

done using the free data science platform Rapid Miner ("Data Science Platform" 2016).

Once we have built models using the training data, we can use the classifiers on labelled or unlabeled data. To be able to evaluate a classifiers performance, we need to have labelled data that was not part of the training input data. This means that we need to split the data we work on into two parts: an in-sample part and an out of sample part. Henceforth, the two parts will be referred to as the training set and the testing set. Furthermore, to make the results more robust and to ascertain we are not generating unstable models, we use tenfold stratified cross-validation when developing the models. Cross-validation splits between five and ten have been shown to be sufficient in averaging variance (Arlot and Celisse 2010). This means that we split data in ten parts during training and use one tenth of the data as a validation set at the time. This is then repeated for each of the ten runs so that each instance appears once in the testing set and nine times in the training set. The cross-validated result is then calculated as the average performance for each of the ten classification runs. To test the performance once we have built the models, we then do further tests first on balanced data and later on imbalanced data.

3.2.1.1. TF-IDF and Cosine Similarity

TF-IDF and cosine similarity are text-feature-extraction algorithms. TF-IDF weighting means multiplying term frequency (TF) in a text with inverse document frequency (IDF). First, we count the number of the occurrences of each term in the document to generate the term frequency $tf_{t,d}$, as shown in equation (31), where $f_{t,d}$ is the term count (Luhn 1958). Second, dividing the total number of documents available by the number of documents that the term is found in, and calculating the logarithm of the result provides us the inverse document frequency idf_t, as shown in equation (32). The formula for TF-IDF can then be calculated as the multiplication in equation (33): (Salton and Buckley 1988)

$$tf_{t,d} = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{31}$$

$$idf_t = \log\left(\frac{N}{1 + n_t}\right) \tag{32}$$

$$tfidf_{t,d} = tf_{t,d} * idf_t \tag{33}$$

Cosine similarity is a way of calculating the statistical similarity between two vectors, generally measured in a positive value space between zero and one (Markov and Larose 2007). A value of 0 means the vectors are at a 90-degree angle. When processing text, a value of 0 means that the two texts that are compared are not at all similar. A cosine similarity value of one means that the two vectors that are compared have an angle of zero. If we are talking about comparing texts, then the compared texts contain all the same words at an angle of zero, however, the words do not have to be in the same order in the text. We use this method to calculate the similarity between a single text and all texts in a training category, to extract features that are useful for text classification. After converting both the text that is being classified into a vector using equation (33) and converting the category texts into a vector for each category using equation (33), we calculate the similarity between the categories and the single text that we want to predict. The mathematical formula for calculating cosine similarity is defined in (34), where d_i is the document vector for web page i, and c_j is the category vector j (Markov and Larose 2007):

$$sim(d_i, c_j) = \frac{d_i \cdot c_j}{\|d_i\|\|c_j\|} \tag{34}$$

3.2.1.2. Word-Based Analysis vs. N-gram-Based Analysis

When we are performing text analysis with TF-IDF and calculate cosine similarity, we need to decide whether we want to be using a word-based approach or an n-gram approach that considers word order. While n-gram approaches can increase classification performance, we also need to consider that extracting n-grams can increase computation time manifold due to the increase in calculation complexity.

A word-based approach, also known as a unigram approach, means that we extract a list of single words. The approach described in 3.2.1.1 was such an approach. This means that when calculating TF-IDF values, we count the term frequency of each word individually, and when calculating cosine similarity, we compare a vector of unigrams from the web page to a collection of unigrams from each category, so that we have one cosine similarity feature for each category.

In an n-gram approach, we extract features from n-grams, which means we take word combinations of n number of words as they appear in the texts. The higher numbered n-grams that we extract, the stricter the word order in the text becomes. For example, if we use two-gram feature extraction we extract words

in pairs, which means that when calculating cosine similarity, the text order will be important on a pair-wise level. Texts with the same words in different order will then be dissimilar and does not containing the same n-grams. Here should also be noted that we use a stop-word list where we remove n-grams that start or end with any of the words in the stop word list, the stop words are prepositions and other common words.

The formula used in cosine similarity calculations does not change between unigram and n-gram models. The difference between unigram and n-gram models is instead during the term frequency $tf_{t,d}$ extraction and the inverse document frequency idf_t extraction. For unigrams, we considered only single words $t = W_1$. For n-grams, we take into account n subsequent words, expressed as t in equation (35), where n is the order of n-gram we are extracting, W is the words in the text, and i is the position of the starting term in the text: (Cavnar et al. 1994)

$$t = t_i(W_i \dots W_{i+n}) \tag{35}$$

3.2.2. Sentiment Analysis

Sentiment analysis, also known as opinion mining, is the process of extracting sentiment from a written text (Pang and Lee 2008). Sentiment can be expressed in different ways. Normally, sentiment is expressed as either negative/neutral/positive or in a numerical range. In our financial analytics research, we have sentiment defined as the author's own self-reported expectations of a company's performance, either positive or negative. In our automatic classification research, we use and extend the model presented by Thelwall et al. (2010), where sentiment is defined by a numerical range from -5 to 5, where -5 means a strongly negative word, and +5 is a highly positive word.

In sentiment analysis we try to find the general opinion in a text, for example, in a paragraph or a sentence. When writing a text, we generally convey a mood or opinion. Sentiment analysis is the process of trying to identify and quantify that mood. While many texts contain a general mood or opinion, there are specific areas such as financial news that can contain information that is positive or negative towards a company, but does not necessary convey a general tone in the text. This means that sentences and words can have different sentiment depending on the context (Loughran and McDonald 2011).

3.2.2.1. Supervised vs. Unsupervised vs. Semi-supervised

Sentiment analysis is further divided into three different types: supervised sentiment analysis, unsupervised sentiment analysis, and semi-supervised sentiment analysis.

Supervised models means that we have a labelled dataset, in which elements have common attributes (Mohri, Rostamizadeh, and Talwalkar 2012). In the case of sentiment analysis, a supervised model would need labelled data where a label represents a group of positive, negative, or neutral texts or words. To be able to create a supervised model, we also need a machine learning algorithm to train on the data. An example of supervised sentiment analysis is found in Rapid Miner ("Data Science Platform" 2016). Another example of supervised sentiment model was developed by Socher et al. (2013).

Unsupervised models, on the other hand, do not need a dataset for training purposes, but still needs work when set up or created. Once unsupervised models are created they can be used on any text without reconfiguration. This is because most unsupervised models use predefined dictionaries of negative and positive words. However, note that someone must create these dictionaries. They can also contain other more advanced features, such as negating words and boosting words that change sentiment values of other words. The SentiStrength model that we use has ten components for determining the sentiment value of a text on a sentence, paragraph, and document level. The ten components are the following: (Thelwall et al. 2011)

- A sentiment word list containing polarity that is made by human experts
- A spelling correction algorithm
- A list of booster words that strengthen or weaken sentiment of words
- A list of idioms that overrides sentiment in common English phrases
- A list of negating words that invert sentiment
- Repeated letters in words increase sentiment value for the word by 1
- A list of emoticons with polarities is used to convert emoticons to sentiment
- Exclamation marks give a sentence a minimum sentiment of 2 unless negative
- Repeated punctuation boosts the strength of preceding sentiment words by 1
- Negative sentiment is ignored in questions

Semi-supervised sentiment analysis is the third type of model which combine the use of unsupervised models with the structural use of supervised models (Zhu

and Goldberg 2009). Generally, semi-supervised models use a small set of labelled data and a large set of unlabeled data. Studies by Goldberg and Zhu (2006) and Sindhwani and Melville (2008) are examples of research in semi-supervised models for sentiment analysis are.

3.3. Financial News Analytics

News analytics can be interpreted as the process of extracting and processing relevant data from any news source. Part of the process is trying to uncover new relevant information from collections of news that previously were unavailable.

Network analytics, also known as social network data analytics (Aggarwal 2011), can be seen as the process of extracting and processing links between different nodes in a network. A node in a network can be anything measurable. Nodes can be human beings posting about their lives on Facebook, different banking entities communicating with each other through financial systems, or satellites orbiting earth communicating with each other. Links between different nodes in networks are representations of how the nodes are related to each other. Two banks can, for instance, be connected by transactions sent from customers. Researchers and companies have through network analytics been able to develop new powerful products. An example of such a product is Googles PageRank (Page et al. 1999) that was the basis for Googles search business.

Combining those two areas of analytics (news and networks) brings us the field of financial news analytics. Our focus in this research field is different types of networks that appear in news, and automatically processing news articles.

3.3.1. Interconnectedness and Co-occurrence

Relationships between entities do not exist in isolation (T. Ritter 2000). Take companies, for example. One of the first types of relationships that come to mind is business relations, such as relationships to suppliers and customers. However, a relationship can be anything that link entities together. If we study market movements, we can see that stock prices of competitors and sectors move together. If one company reports a positive quarterly result, then we can expect that the company's direct competitors also are affected by the reported result somehow. This indirect link between these companies can be interpreted as a relationship. This type of effect where changes to one entity affects other entities

is a type of economic interconnectedness and the spread of the effects, in our case the spread of news, can be calculated through different network measures.

There are several ways that we can represent economic networks. The type of network that we have been researching is known as co-occurrence network (Veling and Van Der Weerd 1999). In these types of networks, we first define a type of entity that we try to identify. We then search for all entities of that type in a collection of data and register the co-occurring entities as networks. To explain the concept in more practical terms, let us consider that the entities we will be searching for are companies in the form of company names and company tickers (e.g. Apple Inc. and AAPL), and the data that we are searching is financial news articles. The co-occurrences that we identify become representations of which companies are mentioned together in financial news (Veling and Van Der Weerd 1999). This type of co-occurrence relation *R,* in a text *t,* can be defined as the number of company matches $M \subset N$. The relation can be described as in equation (36), where ∧ is the exterior product: (Rönnqvist and Sarlin 2015)

$$R_t = \{r | r \in M_t \times M_t \ \wedge \ r_i < r_2\} \tag{36}$$

If we extract the co-occurrence relations for all articles in our collection over a given period of time, and combine these co-occurrences into a matrix formation, we develop a structure known as a co-occurrence matrix (Leydesdorff and Vaughan 2006). The information contained in such a structure represents how many times each entity was mentioned together with the other entities in the matrix. Co-occurrence matrices can be interpreted and visualized as networks. By mapping entities that are co-occurring, we define a network structure that has the potential of uncovering information that was previously unknown.

If we continue with the example of companies co-occurring in articles, and create networks from these co-occurrences, we could find indirect links between companies that we previously thought were unrelated. This could happen, for example, if a company A and a company B are not directly co-occurring, but both companies are mentioned in combination with a company C. Company C could, for example, be a supplier or a competitor to both company A and B. Extracting relationships could be valuable in many parts of economics, especially in situations where network effects apply. Risk analysis and/or identifying undervalued or overvalued assets are two of the possible areas that could benefit from uncovering such information. For instance, mapping relationships between banks has been used in risk analysis to predict crises (Rönnqvist and Sarlin 2014), and in the next section we will cover a way of identifying company

sentiment risks through relationships in texts. Risks that we later show can predict when company stock prices are at increased risk of falling.

With the rise of social networks, researchers identified a need to further explore network theory and measuring different nodes in networks. One of the goals many researchers have had is to try and identify which nodes are the most central in a network, depending on either connections or information that flow through the connections (Borgatti 2005). The different mathematical centrality measures that are of relevance were already discussed in section 2.2.2. Out of these we will start by using information centrality in our financial news research.

3.3.2. Quantitative Risks

Different types of risks have for decades been part of that which businesses, institutions, and governments need to plan for and prepare for. Failing to identify a risk can in the worst case for a business mean bankruptcy and/or accidents. Much of the quantitative risk research has been well documented (Haimes 2015; McNeil, Frey, and Embrechts 2015). The risk research that has gained popularity recently is research into systemic risks such as early warning models, cyclical risks research, and cross-sectional risk research. Many of these have come as a response to the regulation created after the great recession of 2008-2009 (Bisias et al. 2012). The motivation behind much of the research in this field is that the last crisis could have been prevented and that similar situations should not happen again.

Cross-sectional systemic risk and cyclical risks are the two distinct tracks along which systemic risk research develops (Borio 2011). Cross-sectional research analyzes collections of data and further examines specific points in time. Cyclical risks are recurring risks that stem from business cycles, also known as boom-and-bust cycles. The research in early-warning models has generally been done through combining different data sources into models. These models then produce probabilistic output (Scheffer et al. 2009). The output of such models can, for example, be some form of crisis probability indicator (Bussiere and Fratzscher 2006). A large part of the cyclical systemic risk studies that are conducted consists of network effects and how risks spread through different systems (Cabrales et al. 2014).

3.3.2.1. RiskRank

The two ways of quantitatively measuring risks (cross-sectional and cyclical risk) have individually been extensively studied. However, different ways of combining these tracks have not been widely studied. To be able to combine individual-entity risks in a network, with network-wide risk, Mezei and Sarlin (2017) introduced a model called RiskRank. In our risk research, we will be extending the use of RiskRank to components of the financial news networks that were discussed in the previous section. We use sentiment in the networks to quantify the individual, direct, and indirect links to determine whether risks can be measured based on news sentiment. A previous study into the effects of news sentiment on company stock prices has shown that negative sentiment affects volatility more than positive sentiment (Ho et al. 2013).

RiskRank measures both cross-sectional systemic risk and cyclical risks, and outputs a combined result. The main goal of the model is to provide a measure of systemic risk, and at the same time estimate the vulnerability in individual components in the network. The RiskRank equation in (37) consists of three parts: an individual risk component, a direct neighbor risk component, and one indirect market-wide effect component: (Mezei and Sarlin 2017)

$$RR(x_1, \dots, x_n, x_c) = r_{own} + r_{direct} + r_{indirect} \tag{37}$$

$$r_{own} = v(c)x_c \tag{38}$$

$$r_{direct} = \sum_{i=1}^{n}(v(c_i) - \frac{1}{2}\sum_{j \neq i} I(c_i, c_j))x_i \tag{39}$$

$$r_{indirect} = \sum_{i}^{n}\sum_{j \neq i}^{n} I(c_i, c_j) \prod(x_i, x_j) \tag{40}$$

Where the variable $I(c_i, c_j)$ is the interlinkage between nodes and x_i, x_j are the nodes being compared. In this context, v represents the Shapley index, which in our case translates to the risk a company transfers to other companies that it is connected to. $v(c)x_c$ in equation (38) is the individual risk inputted into the model for single components, which in our case translates to individual company sentiment risk and will be discussed further in section 4.3.5. In equation (39), we calculate the risk transferred from direct neighbors in the networks, which in our case means risk transferred from other companies that are mentioned together with the company that we are analyzing. In equation (40), we calculate system-wide risk, which in our case can be seen as an overall sentiment risk in the market. (Mezei and Sarlin 2017)

3.4. Summary and Relevance

In section 2.4, the research questions that will be answered in the thesis were defined. To show their relevance, the questions will now be connected to the methods presented in this chapter.

The first research question: ”How can we use analytics to automate text processing tasks?” is a general question that is answered throughout sections 4.2 and 4.3. We answer it by going through two approaches to automation: a machine learning approach as defined in sections 3.2.1.1 – 3.2.2, where we use TF-IDF weighing, cosine similarity, and sentiment analysis to extract features that are used in automatic classifications; a network risk extraction approach as defined in sections 3.3.2 and 3.3.2.1, where we use co-occurrence, sentiment, and network analytics to automate news processing.

The second research question: “In which ways can sentiment analysis be useful when automating processing tasks?” is answered in sections 4.2.3 – 4.2.8 and 4.3.3 – 4.3.6. We answer it by going through what other researchers have done and by showing how we use sentiment analysis in both automation approaches.

The third research question: “What can be done to improve unigram classification performance for hate and violence texts?” is answered in sections 4.2.3 – 4.2.8. We answer it by combining sentiment analysis features with different unigram and n-gram models, and then going beyond the state of the art by extending to multi-gram analysis. The method definitions are found in 3.2.1.2 and continued in section 4.2.7.

The fourth research question: “Can risks extracted from sentiment networks predict company stock price movements?” is answered in sections 4.3.4 - 4.3.6. We answer it by extracting company risks based on sentiment and co-occurrence networks, and by statistically comparing subsets of data points against a baseline. The methods are defined in section 3.3.2.1 and continued in section 4.3.6.

4. Tools and Results

In this chapter, we review the research contributions and the different tools and information systems that these contributions could be further developed into. The characters Jenna and Neil who were presented earlier will be referenced throughout the chapter to show the practical uses of the tools.

4.1. Text Extraction

In all methods presented in this chapter, we use text-extraction methods. The text-extraction methods we use can be defined as different types of key word extractions and a more in-depth analysis is found in the first research paper [1], which was listed among contributions in section 2.1 and is found in its entirety in the appendix with the other papers. We use pre-processing steps including tokenization of texts through regular expressions, segmentation of texts through paragraph and sentence parsing, stop word removal through dictionaries of English words, named entity recognition, and machine translation. The extraction steps consist of TF-IDF weighting as was discussed in section 3.2.1.1 and functionality for modifying the weights to either increase or decrease importance of words of specific categories.

4.1.1. Key Word Extraction

Key word extraction is a central concept in text analytics. One approach to extracting key words from texts is using regular expressions to match certain pre-defined criteria. We use such an approach in our financial research when extracting material on companies from texts. Key word extraction becomes more complex when we do not have predefined expressions that we are searching for. In these cases, we can use methods such as TF-IDF weighting (Salton and Buckley 1988) to find key words and we can combine these approaches with named entity recognition (X. Liu et al. 2011) and/or dictionary-based approaches, depending on the context of the extraction. In our classification research, we extend the TF-IDF key word extraction work from [1] in several ways (n-grams, two different IDF extraction approaches, and multi-grams). We have also researched extractions that use different topic models, however, these approaches were not included in the thesis as they did not prove fruitful in the format they were used.

4.2. Automatic Classification Results

In the first part of the thesis, two hypothetical people Jenna and Neil were introduced. In this part of the thesis, the contributions that we have made in automatic classifications are discussed, as well as the possible uses of these contributions. In this section, we will refer to Neil on occasion to provide an overview of the practical uses and how the contributions could positively affect his workday and workflow.

We first use tenfold stratified cross-validation when developing the models in sections 4.2.1 – 4.2.7 and report results using balanced test sets. In section 4.2.8, we further test the models we have built on imbalanced test sets and discuss our findings. We find that the F-measure performance drops significantly when we move to imbalanced test sets.

4.2.1. Violence and Hate Content Classification

Some content categories are easier to classify than others. Adult content is one of the categories that has been successfully categorized using both text features and image features. Other categories such as violence, racism, and hate speech often contain more abstract concepts that algorithms find difficult to classify. In these categories, we also find borderline cases, which humans also find difficult to categorize.

If we start analyzing language use, we realize how nuanced languages are. First, different people use different words to describe the same situations. This step is generally not a problem for computers to handle, as long as the grammar is understandable. However, humans also tend to write indirectly and reference back to recent events and/or common knowledge that is not necessarily found in the current text. Algorithms in general have problems understanding indirect references as they cannot yet understand context the way that humans do. To be able to understand concepts not directly stated, we humans have knowledge gathered from experiences that we can draw upon. This type of context knowledge is generally not available to the current state of the art classification algorithms. There are some algorithms, such as RNNs (Schuster and Paliwal 1997), that have memory and show promise. However, we are still some way from having algorithms that can effectively link contexts the way humans do.

Image classification can be used as an example of illustrating the problems that algorithms face when classifying content. Let us entertain the thought that we are using a computer algorithm to classify images as violent or non-violent.

Consider the following scenario: we have a picture of two humans standing next to each other. Current algorithms should be able to classify this picture as non-violent. Consider now that in the next picture one of the people in the photo is pointing a knife towards the other person. The current algorithms would be able to label this picture as violent with some level of certainty. However, a human would not decide whether it is violent only based on that. Imagine that the person holding the knife is the famous chef Jamie Oliver. This changes the context, and the current generation of classification algorithms do not account for such factors, no matter what type of data used. This brings the discussion to datasets, and more specifically, the data that we have used to perform our automatic classification research on.

4.2.2. Dataset

The dataset used in our automatic classification research, papers [2], [3] and [6], consists of text gathered from web pages labelled into 20 categories using a single labelling system. The labelled dataset contains a total of 79,063 web pages split unevenly over the 20 categories. For different parts of our research, we have experimented with different categories, different sized categories, and different balances between true and false labelled pages in binary classifications. One of the real-world problems identified by the security industry that automatic classifications face are highly imbalanced data skews. No matter which content category we choose to categorize, the category is only a small subset of the whole textual content found on the Internet. Because of this, we also perform experiment validations in section 4.2.8. with skewed datasets (5% positive and 95% negative instances).

Each page in the dataset follows a structure based on HTML tags. From the web sites, the content of 31 different HTML tags was gathered. These 31 elements are, for example, the URL of the page, links to other pages, the whole page content, each paragraph found in the page, and metadata such as search keywords.

The categories in the dataset have been manually labelled and the sizes of the different categories vary from 400 web pages to about 6,800 web pages. Descriptions and sizes of the 20 categories can be found in Table 3. As the categories were labelled using a single label annotation, some of the categories ended up having related and overlapping content. This is a constraint that can reduce performance, although, in practice it might not matter if a page is labelled as being part of, for example, the “cigars” category or the “cigarette” category, as they both will be handled in the same manner. The categories in Table 3 that we

are conducting our research on is the violence category (17) and the racism and hate categories (8, 12, and 13). Henceforth, these three categories together will be referred to as the hate categories.

4.2.3. Baseline Classifications

In this section, we start developing automatic classification models. The final models in this section will serve as the basis for the tool that the security industry expert Neil could use to reduce his workload. To be practically useful, the tool requires high performance. As a guiding rule, for these types of problems performance can be considered practically acceptable at an F-measure of 0.9 or

Category	Description	Labelled Pages
1	Adult	6,801
2	Beer	5,913
3	Casino and gambling	3,651
4	Cigars	1,939
5	Cigarette	3,845
6	Cults	3,282
7	Dating	4,703
8	Hate, anti-Semitism	3,479
9	Prescription drugs	5,397
10	Occult	5,105
11	Marijuana	6,042
12	Racism, white supremacy	400
13	Racism, againt minorities	4,667
14	Religion	5,438
15	Sports betting	2,820
16	Spirits and liquor	3,671
17	Violence	1,919
18	Unknown	3,432
19	Wine	4,095
20	Weapons	2,464
Total		79,063

Table 3. *List of the 20 categories in the classification dataset. Taken from publication [6].*

higher. If we can achieve this, we further want to maximize either precision or recall in the models depending on the classification problem. As high performance is easier to achieve on balanced datasets, we will also be testing highly imbalanced datasets.

Previously, it was mentioned that in the publications included in the thesis, we have experimented with different sized datasets, different categories, and different feature sets. However, to obtain generalizable results between our research publications all experiment results in the thesis are re-run using the same datasets for each extension of the baseline system. Furthermore, feature selection is not included in the thesis experiments. Due to re-running experiments without feature selection, performance of the early extensions is slightly lower than as reported in the research publications where we used feature selection. The reason for not performing feature selection in the thesis extensions is that when this was tested, it was shown that we would have to do separate feature selections for each extension to find the optimal performance.

In our baseline models for violence and hate content classification, we only account for text content and create classification models consisting of text similarity features and unsupervised sentiment features. The similarity measures used are calculated using cosine similarity, as defined in 3.2.1.1, where we compare the text content of one web page to the text content of an entire category of pages. The unsupervised sentiment features are extracted using a modified version of SentiStrength (Thelwall et al. 2010), as defined in 3.2.2.1. We can represent the full similarity feature set for one text through equation (41), where d is the web page, c is the category, and i is the n-gram order:

$$y_{sim}(i) = \{sim(d_i, c_1), \dots, \mathrm{sim}(d_i, c_{20})\} \tag{41}$$

We create vectors of unigrams from words in each page that we want to classify, as well as one vector of words from each category that we are trying to identify. To decide which words are the most important, and to choose which ones should be included in the limited sized word vectors, we use the TF-IDF weighting method described in 3.2.1.1 together with a generalized IDF dictionary from (Radev et al. 2004). This approach will be referred to as the unigram approach. We then compare similarity between each category and the text from each web page, by comparing the word vector from the page against the 20 different category vectors. Through this process, we convert each web page into 20 features where the feature values take a number between 0 and 1, where one feature represents the page similarity to one category. A similarity feature value of 0 means the web page is dissimilar to pages in the category, and would

essentially mean that none of the words found in the web page were found in the combined TF-IDF values of the category we compare it to. A similarity feature value close to one would mean that almost all words in the web page are relevant to the category we compare it to. We then experiment with different limits on the sizes of the word vectors to see the effect the limits have on classification performance. The different sized word vectors that we test for unigram are 500, 10,000, and 15,000 top weighted words (both for the categories and the individual pages).

In our research leading up to building the baseline classifications, which can be found in publication [2], we tried different classifications using different parts of web content found in web pages. The following have been tried: 1) using all textual web content that is retained after stripping out HTML tags (referred to as the full-text content); 2) using only part of the content such as the URL, keywords, and meta text content; 3) using combinations of full-text content and meta content, which means lengthening the text beyond those in the original web page. By lengthening the text, we can emphasize some parts of the text and change weights in the calculations. We found that using the third approach where we include all the text in the web site and emphasize some parts, by adding them a second time to the text, performs better for violence and hate classifications.

Out of the 31 HTML text types that are available in the dataset, we use the uniform resource locator (URL), the full text content of the page, the meta-text content, and the keywords as input to our classification system. To define a baseline performance, we use binary naïve Bayes classifiers, which was previously discussed in section 2.2.2.1.1. The positive instances used are all the labelled instances in the chosen category and the negative instances used are spread evenly over the other 19 categories in the dataset, but randomized within each category. When building the baseline models, we use close to balanced datasets. For example, the violence category contains 1915 usable positive samples after pre-processing. Notice that pre-processing (stop word removal and TF-IDF weighting) slightly reduced the number of positive instances. To have a balanced set to input into the machine learning algorithm, we split the negative samples evenly among all other categories, and randomly pick 1/19 of the number of positive samples from each of the other categories.

In binary classifications, the results are measured through the following four counts: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). True positive and true negative are the instances that were correctly predicted. False positives, also known as false flags, are instances that

were labelled as negative, but classified by the algorithm as positive. False negatives are instances that are labelled as positive but classified as negative. From these values, we can calculate the performance of the classifier if we have labelled data. The performance measures calculated are accuracy, F-measure, precision, and recall. The formulas for the different calculations are as follows (Powers 2011):

$$accuracy = (TP + TN) / (TP + FP + TN + FN) \tag{42}$$

$$precision = \frac{TP}{TP+FP} \tag{43}$$

$$recall = \frac{TP}{TP+FN} \tag{44}$$

$$F\text{-}measure = (2 * \frac{TP}{TP+FP} * \frac{TP}{TP+FN}) / (\frac{TP}{TP+FP} + \frac{TP}{TP+FN}) \tag{45}$$

The baseline classification results are found in Table 4. We can see that the violence category 17 performance and the minority hate category 12 performance is generally lower than the performance of the other two categories. The overall performance of the unigram baseline is quite poor.

We continue defining the baseline performance by extracting unsupervised sentiment features and classifying the same datasets using unsupervised sentiment features only. The base version of SentiStrength, which was described in section 3.2.2, outputs sentiment features as the number of words found containing a sentiment value. The program uses a scale of -5 to +5, where a negative value (-1 to -5) means the word has a negative sentiment, and a positive value (+1 to +5) means the word has a positive sentiment. A value of -5 represents the words with the most negative sentiment and +5 represents the words with the most positive sentiment. For each web page, we count the number of sentiment words (-5 to +5) and group them by value strength (words that have value 0 are neutral and not counted). Furthermore, the program also

Unigram Similarity-Based Classification Performance				
Category	Accuracy	Precision	Recall	F-measure
8	80.83%	0.78	0.86	0.82
12	69.51%	0.64	0.88	0.74
13	74.95%	0.71	0.85	0.77
17	65.62%	0.62	0.78	0.69

Table 4. *The baseline centroid based unigram classification performance for violence and hate was achieved using all 20 similarity features. Method taken from publication [2], experiments were re-run to match the dataset and features of later extensions.*

outputs a page-wide sentiment value, which is calculated as the highest and the lowest sentiment value found in the text.

Here we recognize an opportunity to extend the unsupervised sentiment feature extraction, by further developing the algorithm to fit our needs. In publication [2], we define two new sentiment features that we calculate and output together with the existing features. These features are named NewScale1 and NewScale2. NewScale1 is calculated as a sum of sentiment values in a text, normalized by the total number of sentiment words. NewScale2 is calculated as a count of positive minus negative sentiment words, where only the polarity (positive or negative) of the word is considered. This means that a page containing more negative words should end up with a negative number for the NewScale2 value, and pages containing more positive words end up with a positive number. Pages containing the same amount of negative and positive words can still be either negative or positive for the NewScale1 feature, depending on the sentiment strength of the words in the text, but would end up as zero for the NewScale2 feature.

To better understand how the sentiment is represented in different categories in our dataset, we plot the different average sentiment scores for 8 categories from our dataset of 20 categories, these are categories that our industry partners identified as problematic categories. The plot is done using different compression rates, which means that we compress pages by taking a subset of the highest ranked TF-IDF words. Figure 2 shows the average category sentiment for the NewScale1 feature when using compression between 10% and 100% of highest weighted TF-IDF words as input to the sentiment analysis.

From Figure 2, we see that there is a clear split between sentiment polarities in different categories. We can see that the three hate categories, the violence category, and the religion category all have an average sentiment always below -1, which also gradually decreases when more of the TF-IDF words are included. On the other side of the spectrum, we see the categories unknown, cults, and occults start around a neutral sentiment and move toward a positive average sentiment when we include more of the textual data.

After we developed two new sentiment features, we have 13 sentiment features that we can use as our sentiment classification baseline. We use all 13 sentiment features in our generalizing re-run, while in publication [2] we ended up using eight of the thirteen features. After testing different compression levels, we end

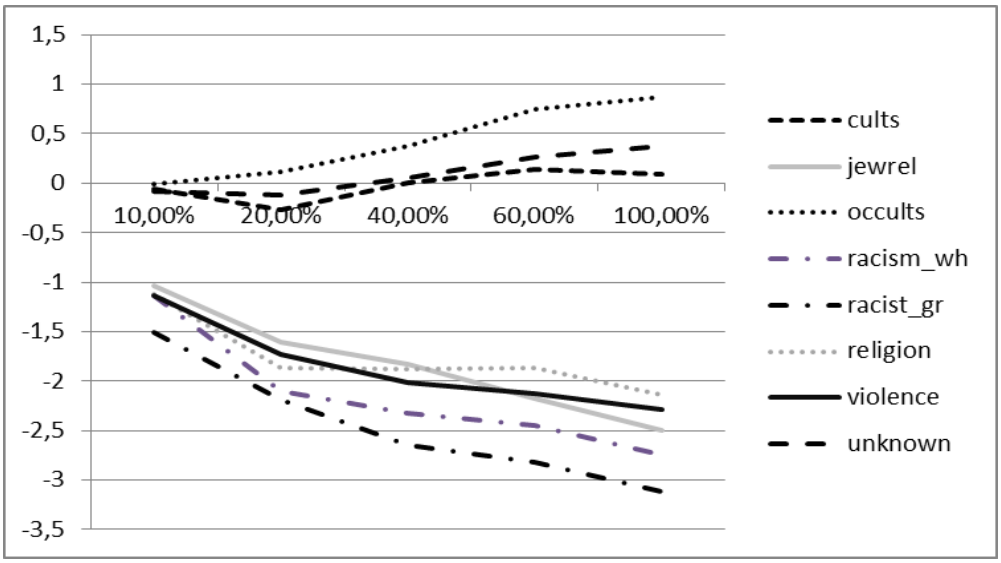

Figure 2. *Visualization of average sentiment polarization between 8 categories starting from 10% of top weighted words going up to 100% weighted words. Some categories become more positive when including more top weighted words and others become more negative. Figure taken from publication [2].*

up using top 30% of the TF-IDF weighted words from each page as the input to the sentiment feature extractions. We tested using higher compression rates, but it did not affect performance in any meaningful way above 30%. The set of 13 sentiment features is represented in equation (46) as y_{sent}, where y represents the individual sentiment features. $y_1 - y_{10}$ are the counts of sentiment strengths (-5 to +5), y_{11} is the page total value, y_{12} is NewScale1 and y_{13} is NewScale2:

$$y_{sent} = \{y_1, \dots, y_{13}\} \tag{46}$$

Table 5 shows the performance measures for the different sentiment classifications. We see that the performance is lower than the unigram classifications. Based on the results in publication [2], we find that classifications using only similarity features seems to perform slightly better than sentiment features on violence and hate content also when incorporating feature selection.

4.2.4. Combining Similarity and Sentiment Features

After having defined the baseline performance of both similarity classifications and sentiment classifications for hate and violence, we start applying state-of-the-art methods on the data to see if that can improve performance. We start by

Unsupervised Sentiment Classification Performance				
Category	Accuracy	Precision	Recall	F-measure
8	52.93%	0.51	0.99	0.68
12	52.44%	0.51	1.00	0.67
13	68.09%	0.64	0.81	0.72
17	61.46%	0.86	0.27	0.41

Table 5. *Baseline unsupervised sentiment classification performance for violence and hate. Method taken from publication [2], experiments were re-run to match the dataset and features of later extension models.*

using an approach containing some similarities to the one done by Melville et al. (2009) and Bespalov et al. (2011). In this approach, we take features of both the similarity and sentiment approaches, and run a classification using a combination of these features. Here we partly answer the second and third research questions by showing how sentiment features can improve classifications when using a naïve Bayes classifier.

The combination of features is simple as we already have the feature extraction in place. We combine the sentiment features y_{sent} with the unigram features y_{sim} as represented in equation (47), where i is the page. We then run the features through the same naïve Bayes classifier we used before to see comparable results. As shown in Table 6, this offers us an overall improvement in F-measure results over the sentiment results and for category 13 over similarity features. Performance is still quite poor. We have now answered the first part of the research questions two and three.

$$y_{tot}(i) = \{y_{sim}(i), y_{sent}(i)\} \tag{47}$$

Combined Similarity + Sentiment Classification Performance (Naïve Bayes)				
Category	Accuracy	Precision	Recall	F-measure
8	71.10%	0.65	0.92	0.76
12	58.54%	0.54	0.98	0.70
13	75.16%	0.70	0.87	0.78
17	64.06%	0.32	0.88	0.47

Table 6. *Classification results from combining unigram cosine similarity features with unsupervised sentiment features. Method taken from publication [2], experiments were re-run to match the dataset and features of later extension models.*

In cases where it is not clear that both precision and recall is improved, it might be better to use the F-measure of the classification to see if there was an overall improvement. Accuracy and F-measure both express an overall classification performance, but accuracy becomes less useful when the sets we use are imbalanced, which we will see later. Systemic risk models and early warning models are examples of when maximizing recall would be of interest. We would in such models rather flag all possible shock situations than miss one. On the other hand, in parental control system or other filtering systems, we would not want to filter out legitimate sites due to errors in the classification, because filtering legitimate sites erodes user confidence in the system. If not accidentally filtering sites is the goal, then we would rather maximize the precision. Creating algorithms that have both a high precision and a high recall in real-world scenarios is uncommon.

4.2.5. Combining N-gram and Sentiment Features

Next, we extend our experiments to an n-gram-based approach for the cosine similarity feature extraction. In this part of the research, we will cover one-gram, tri-gram and five-gram similarity analysis and then combine the n-grams with the same sentiment features that we used with the unigram classifications. To do n-gram analysis, we start by creating IDF dictionaries for the different n-grams, as in publication [3], because the dictionary we have been using so far is unigram-based and cannot be applied to the higher order n-grams. The unigram and the one-gram approaches are identical, except that we create a new word-based IDF dictionary from our dataset for the one-gram approach, and used an existing dictionary for the unigram approach. The theory behind using n-grams is that taking word order into account in texts can improve classification performance (Khreisat 2006; Bespalov et al. 2011). As we limit the analysis to one-grams, tri-grams, and five-grams it means that we decide that either one, three, or five-word combinations found in the texts are of interest to us. We are also interested in testing the performance of the different n-grams against each other. In theory, an n-gram approach can lead to either better or worse performance. In texts where there are few tri-grams or five-grams, there is a possibility that the performance will go down due to no matches. Previous studies using n-grams have shown that using higher than tri-grams will not necessarily increase classification performance (Fürnkranz 1998). Our approach here has some similarities to the approach used by (Bespalov et al. 2011).

Here we will partly answer the second and third research questions by showing that combining sentiment features with n-gram features can improve

performance with the naïve Bayes classifier, however, we also show that it is not always the case.

Changing to higher n-gram analysis also raises the computational requirements due to the manifold increased sizes of dictionaries and the increased number of category TF-IDF values that need to be calculated and compared. In the unigram models, we worked with different top weighted TF-IDF words and ended up using top 15,000 weighted words, as our experiments showed that performance was only marginally increased beyond that point. Our research in [3] found that we needed to increase the number of weighted words per category above the 15,000 when performing n-gram classifications, because the n-gram words in a category can be over two million, while most unigrams categories contained only around 100,000 words. We tested a couple of different sizes and ended up using top 100,000 TF-IDF weighted category words for tri-grams, and top 120,000 TF-IDF weighted category words for the five-gram analysis. This was done to scale the number of words per category with the order of n-grams.

Table 7 shows the performance of the one-gram classification that uses IDF calculated based on our dataset. Table 8 shows the tri-gram classification results. Table 9 shows the performance of the five-gram classification. The classification performance when adding sentiment feature increases for category 13 over using only similarity features. Unigram classification still has the best performance for category 8, while category 12 has the best performance using

One-gram Similarity Classification Performance (Naïve Bayes)				
Category	Accuracy	Precision	Recall	F-measure
8	77.83%	0.75	0.82	0.79
12	80.49%	0.77	0.85	0.81
13	74.73%	0.73	0.79	0.76
17	71.35%	0.81	0.56	0.66
Combined One-gram + Sentiment Performance (Naïve Bayes)				
Category	Accuracy	Precision	Recall	F-measure
8	70.53%	0.64	0.92	0.76
12	68.29%	0.61	1.00	0.75
13	76.23%	0.74	0.82	0.78
17	67.19%	0.88	0.39	0.54

Table 7. *One-gram classification performance using an IDF dictionary created from the dataset. Method taken from publication [3], experiments were extended to cover one-grams with our own IDF-dictionary for comparison.*

Tri-gram Similarity Classification Performance (Naïve Bayes)				
Category	Accuracy	Precision	Recall	F-measure
8	68.96%	0.63	0.92	0.75
12	73.17%	0.65	0.98	0.78
13	70.56%	0.64	0.95	0.76
17	67.45%	0.62	0.92	0.74
Combined Tri-gram + Sentiment Performance (Naïve Bayes)				
Category	Accuracy	Precision	Recall	F-measure
8	67.81%	0.62	0.94	0.74
12	64.63%	0.58	1.00	0.73
13	71.84%	0.65	0.94	0.77
17	79.17%	0.94	0.62	0.75

Table 8. Tri-gram classification performance using the IDF dictionary created from the dataset. Method taken from publication [3], experiments were re-run to match the dataset and features of later extension models.

one-grams. Category 17 has the best performance using five-grams with sentiment, and for category 13 the performance is even between models. Contrary to the study by Fürnkranz (1998), the five-gram performance seems to so far be better than the tri-gram classifications on average.

Five-gram Similarity Classification Performance (Naïve Bayes)				
Category	Accuracy	Precision	Recall	F-measure
8	73.25%	0.66	0.93	0.78
12	73.17%	0.65	0.98	0.78
13	68.42%	0.62	0.94	0.75
17	69.27%	0.63	0.92	0.75
Combined Five-gram + Sentiment Performance (Naïve Bayes)				
Category	Accuracy	Precision	Recall	F-measure
8	71.67%	0.65	0.95	0.77
12	64.63%	0.58	1.00	0.73
13	69.91%	0.64	0.92	0.75
17	83.33%	0.94	0.71	0.81

Table 9. *Five-gram classification performance using IDF dictionary developed from the dataset. Method taken from publication [3], but experiments were re-run to match the dataset and features of later extension models.*

The performance of the models is at this point still quite far from practically usable as we defined the practically usable performance threshold as an F-measure of 0.9 or higher. The highest reached so far is 0.82 for category 8.

4.2.6. Extending to Other Machine Learning Algorithms

To see if we can further raise performance of the automatic classifications, we extend our research to other machine learning algorithms. By using more sophisticated algorithms, such as SVM and ANN, we should be able to raise the performance, however, using them also significantly increases the training times.

We now continue the research by testing decision trees, *k*-nearest neighbors, support vector machines, and feedforward artificial neural networks, which were introduced in sections 2.2.2.1.2 to 2.2.2.1.5, on each of the four categories for the unigram, one-gram, tri-gram, and five-gram features. Each set of n-gram features are combined with sentiment features. The ANNs have the overall best performance as can be seen from Tables 10 – 13. Using ANNs, we are able to break over 90% accuracy and 0.9 F-measure for all four categories using both tri-gram and five-gram features. Tri-gram and five-gram classification performance is now mixed when comparing F-measures, better when using tri-grams for category 17 and worse in majority of algorithms for category 8. When comparing accuracy, we have similar mixed results between classes.

Comparing runs with sentiment features against runs without, we find that the performance here is also mixed. For most classifiers tested, category performance 13 increases with sentiment features, for other categories the performance is mixed. This could be because category 13 had better performance with sentiment features in section 4.2.3 than the other categories. Sentiment features seem to have an overall negative effect when using k-NN.

Using research from paper [6], we partly answer research questions two and three by showing that sentiment features can improve performance when using SVM, ANN, and DT. However, when using k-NN performance did not improve. Furthermore, changing from NB to other algorithms improve the results with or without sentiment features.

Category 8	Unigram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	79.54%	0.83	0.74	0.78
SVM	83.83%	0.84	0.84	0.84
ANN	87.70%	0.86	0.90	0.88
K-NN	78.97%	0.76	0.84	0.80
Category 8	One-gram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	76.25%	0.64	0.85	0.73
SVM	84.69%	0.87	0.82	0.84
ANN	87.98%	0.90	0.85	0.88
K-NN	78.54%	0.75	0.85	0.80
Category 8	Tri-gram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	88.84%	0.89	0.88	0.89
SVM	89.70%	0.91	0.88	0.89
ANN	92.85%	0.91	0.95	0.93
K-NN	82.69%	0.80	0.87	0.83
Category 8	Fivegram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	90.99%	0.86	0.96	0.90
SVM	89.99%	0.95	0.84	0.89
ANN	92.85%	0.95	0.90	0.93
K-NN	81.83%	0.79	0.87	0.83

Table 10. *Classifications for category 8 (anti-Semitism) using the combined similarity and sentiment features with other machine learning algorithms. Results from publication [6].*

Category 12	Unigram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	68.29%	0.61	0.98	0.75
SVM	80.49%	0.75	0.90	0.82
ANN	86.59%	0.85	0.88	0.86
K-NN	68.49%	0.67	0.72	0.70
Category 12	One-gram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	85.37%	0.78	0.98	0.87
SVM	90.24%	0.86	0.95	0.90
ANN	87.80%	0.89	0.85	0.87
K-NN	69.53%	0.68	0.74	0.71
Category 12	Tri-gram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	86.59%	0.87	0.85	0.86
SVM	81.71%	0.96	0.65	0.78
ANN	93.90%	0.95	0.93	0.94
K-NN	74.48%	0.74	0.75	0.75
Category 12	Fivegram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	86.59%	0.87	0.85	0.86
SVM	82.93%	1.00	0.65	0.79
ANN	93.90%	0.97	0.90	0.94
K-NN	73.70%	0.76	0.69	0.72

Table 11. *Classifications for category 12 (white supremacy) using the combined similarity and sentiment features with other machine learning algorithms. Results taken from publication [6].*

Category 13	Unigram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	71.52%	0.66	0.87	0.75
SVM	80.51%	0.79	0.82	0.81
ANN	83.30%	0.82	0.85	0.84
K-NN	80.09%	0.78	0.84	0.81
Category 13	One-gram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	73.98%	0.55	0.88	0.68
SVM	83.30%	0.84	0.83	0.83
ANN	87.15%	0.86	0.89	0.87
K-NN	78.05%	0.75	0.84	0.79
Category 13	Tri-gram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	85.97%	0.86	0.86	0.86
SVM	82.55%	0.92	0.71	0.80
ANN	89.51%	0.88	0.91	0.90
K-NN	89.40%	0.88	0.91	0.90
Category 13	Fivegram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	83.73%	0.93	0.73	0.82
SVM	81.48%	0.96	0.66	0.78
ANN	91.22%	0.91	0.92	0.91
K-NN	81.37%	0.80	0.83	0.82

Table 12. *Classifications for category 13 (racism against minority groups) using the combined similarity and sentiment features with other machine learning algorithms. Results taken from publication [6].*

After having tested the performance of individual classifiers, the next step is to try an ensemble of classifications on different n-grams to see if a majority voting algorithm can offer further performance increases. Ensemble classifications combine the results of different classifications. The ensemble method used in our experiments is the majority voting algorithm (Rokach 2010). We try three different voting ensemble combinations on the different n-grams: DT/SVM/ANN/k-NN, DT/SVM/ANN, and SVM/ANN. Tables 14 – 17 show the ensemble performances for the different n-gram classifications.

Comparing them to the best performing ANN results we find that overall ANN performs best on balanced data. The ensemble performance using DT/SVM/ANN is better than ANN alone on category 8 using five-grams, but not the other

Category 17	Unigram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	60.94%	0.98	0.22	0.36
SVM	82.29%	0.93	0.70	0.80
ANN	83.33%	0.85	0.80	0.83
K-NN	68.49%	0.67	0.72	0.70
Category 17	One-gram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	64.32%	0.95	0.30	0.45
SVM	82.81%	0.95	0.69	0.82
ANN	88.02%	0.89	0.87	0.88
K-NN	69.53%	0.68	0.74	0.71
Category 17	Tri-gram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	89.06%	0.95	0.83	0.88
SVM	86.20%	0.99	0.73	0.84
ANN	91.67%	0.89	0.95	0.92
K-NN	74.48%	0.74	0.75	0.75
Category 17	Fivegram Performance Other Algorithms			
ML-algorithm	Accuracy	Precision	Recall	F-measure
DT	86.20%	0.93	0.79	0.85
SVM	86.20%	0.99	0.73	0.84
ANN	90.36%	0.85	0.98	0.91
K-NN	73.70%	0.76	0.69	0.72

Table 13. *Classifications for category 17 (violence) using the combined similarity and sentiment features with other machine learning algorithms. Results taken from publication [6].*

categories. Furthermore, the ensemble models improve precision in several cases. For category 12, we are able to achieve a 100% precision on the balanced test set. Using ensembles, we also find that tri-gram and five-gram performance is mixed, but better than the performance of unigram and one-gram classifications.

Category 8	Unigram Ensemble Classification Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM/ANN	84.98%	0.80	0.93	0.86
DT/SVM/ANN	85.98%	0.89	0.82	0.85
DT/SVM/ANN/k-NN	86.98%	0.84	0.91	0.87
Category 8	One-gram Ensemble Classification Performance			
SVM/ANN	87.70%	0.85	0.92	0.88
DT/SVM/ANN	85.84%	0.89	0.82	0.85
DT/SVM/ANN/k-NN	87.98%	0.86	0.90	0.88
Category 8	Tri-gram Ensemble Classification Performance			
SVM/ANN	90.84%	0.88	0.95	0.91
DT/SVM/ANN	92.70%	0.94	0.91	0.93
DT/SVM/ANN/k-NN	92.56%	0.91	0.95	0.93
Category 8	Five-gram Ensemble Classification Performance			
SVM/ANN	91.99%	0.92	0.92	0.92
DT/SVM/ANN	93.13%	0.98	0.88	0.93
DT/SVM/ANN/k-NN	92.85%	0.95	0.91	0.93

Table 14. *Ensemble classification performance for category 08 (anti-Semitism). Results taken from publication [6].*

Category 12	Unigram Ensemble Classification Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM/ANN	85.37%	0.85	0.85	0.85
DT/SVM/ANN	79.27%	0.73	0.90	0.81
DT/SVM/ANN/k-NN	86.59%	0.82	0.93	0.87
Category 12	One-gram Ensemble Classification Performance			
SVM/ANN	90.24%	0.94	0.85	0.89
DT/SVM/ANN	90.24%	0.86	0.95	0.90
DT/SVM/ANN/k-NN	91.46%	0.92	0.90	0.91
Category 12	Tri-gram Ensemble Classification Performance			
SVM/ANN	90.84%	0.88	0.95	0.91
DT/SVM/ANN	91.46%	0.97	0.85	0.91
DT/SVM/ANN/k-NN	87.80%	0.97	0.78	0.86
Category 12	Five-gram Ensemble Classification Performance			
SVM/ANN	91.99%	0.92	0.92	0.92
DT/SVM/ANN	90.24%	0.97	0.83	0.89
DT/SVM/ANN/k-NN	90.24%	1.00	0.80	0.89

Table 15. *Ensemble classification performance for category 12 (white supremacy). Results taken from publication [6].*

Category 13	Unigram Ensemble Classification Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM/ANN	81.58%	0.87	0.74	0.80
DT/SVM/ANN	81.91%	0.80	0.85	0.82
DT/SVM/ANN/k-NN	83.73%	0.84	0.84	0.84
Category 13	One-gram Ensemble Classification Performance			
SVM/ANN	85.44%	0.90	0.79	0.84
DT/SVM/ANN	85.22%	0.88	0.81	0.85
DT/SVM/ANN/k-NN	84.58%	0.92	0.76	0.83
Category 13	Tri-gram Ensemble Classification Performance			
SVM/ANN	83.08%	0.96	0.69	0.80
DT/SVM/ANN	88.22%	0.90	0.86	0.88
DT/SVM/ANN/k-NN	87.47%	0.94	0.80	0.86
Category 13	Five-gram Ensemble Classification Performance			
SVM/ANN	81.48%	0.97	0.65	0.78
DT/SVM/ANN	86.19%	0.95	0.76	0.85
DT/SVM/ANN/k-NN	85.65%	0.96	0.74	0.84

Table 16. *Ensemble classification performance for category 13 (racism against minorities). Results taken from publication [6].*

Category 17	Unigram Ensemble Classification Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM/ANN	83.07%	0.96	0.69	0.80
DT/SVM/ANN	81.51%	0.82	0.81	0.81
DT/SVM/ANN/k-NN	76.56%	0.95	0.55	0.70
Category 17	One-gram Ensemble Classification Performance			
SVM/ANN	82.81%	0.96	0.68	0.80
DT/SVM/ANN	82.55%	0.96	0.68	0.80
DT/SVM/ANN/k-NN	77.86%	0.95	0.59	0.72
Category 17	Tri-gram Ensemble Classification Performance			
SVM/ANN	86.20%	0.99	0.73	0.84
DT/SVM/ANN	90.10%	0.99	0.81	0.89
DT/SVM/ANN/k-NN	88.54%	1.00	0.77	0.87
Category 17	Five-gram Ensemble Classification Performance			
SVM/ANN	84.11%	1.00	0.68	0.81
DT/SVM/ANN	88.02%	0.97	0.79	0.87
DT/SVM/ANN/k-NN	86.46%	0.99	0.74	0.84

Table 17. *Ensemble classification performance for category 17 (Violence). Results taken from publication [6].*

4.2.6.1. Parameters of the Models

Here follows a list of parameters used to run the machine learning algorithms in 4.2.6. These parameters are listed for experiment reproducibility.

SVM (Linear)

- Kernel cache: 200
- Complexity constant: 0
- Convergence epsilon: 0.001
- Max iterations: 100,000

ANN

- Training cycles: 500
- Learning rate: 0.3
- Momentum: 0.2
- Error epsilon: 0.00001
- Decay: false
- Hidden layers: 1
- Hidden nodes: 19 for n-grams, 42 and 49 for multi-grams

DT

- Criterion: gain ratio
- Maximal depth: 20
- Confidence: 0.25
- Minimal gain: 0.1
- Minimal leaf size: 2
- Minimal size for split: 4
- Number of prepruning alternatives: 3

k-NN

- Number of neighbors k: 1
- Measure type: Mixed Euclidean Distance

4.2.7. Extending Models to Multi-gram Analysis

In section 3.4, one more extension to the methods was mentioned, which we now will cover. This last extension is a full aggregation of all the steps presented so far, aggregating features into an approach that we call a multi-gram classification. In the previous classifications, we have extracted features for

specific sized n-grams and combined these features with unsupervised sentiment features. Using 33 features (sentiment + similarity) we ran classifications using several other machine learning algorithms and ensemble classifications. In this section, we partly answer the third research question by showing that multi-gram extraction improves balanced results over the other models tested so far.

In section 4.2.5, I mentioned that we could be missing vital information by choosing to only do the experiments on one type of n-gram at the time. Therefore, in this section we will perform experiments to try to validate that claim. This is done by combining all the n-gram features we have together with unsupervised sentiment features, as done in research paper [6]. This becomes a multi-gram classification with a up to 93 features. Thus, we increase the variety of features that are used in the algorithm instead of using feature selection that reduces the already limited number of features, as was done in the earlier research papers [2] and [3] with limited success. The generalized multi-gram feature set for a web page can be represented as in equation (48), where *n* is the n-gram used, y_{sim} is from equation (41), and y_{sent} is from equation (46). We test two combinations to stay consistent with the different n-gram models used so far in our experiment: one including features from unigram, one-gram, tri-gram, and five-gram as in equation (49), and one that also includes sentiment features as in equation (50):

$$y_{mg(n)} = \{y_{sim(1)}, \cdots, y_{sim(n)}\} \tag{48}$$

$$y_{mg_1} = \{y_{sim(uni)}, y_{sim(1)}, y_{sim(3)}, y_{sim(5)}\} \tag{49}$$

$$y_{mg_2} = \{y_{sim(uni)}, y_{sim(1)}, y_{sim(3)}, y_{sim(5)}, y_{sent}\} \tag{50}$$

Using up to 93 features we then repeat our classification experiments to see if the classification performance can be further improved. This multi-gram approach has some similarities to the models used by Shen et al. (2006) and Wallach (2006).

Table 18 shows a summary of the best performing classification models that we have tested using multi-gram classifications. We can see that the multi-gram extension to the model with 93 features provides us an overall increase for most algorithms that were tested, with ANNs having F-measures above 0.93 for each of the categories and outperforming all other algorithms. Furthermore, we can see that the ensemble of DT/SVM/ANN/k-NN achieves a precision of 100% for the violence category while maintaining an F-measure of 0.92. We also tested

running the experiments without sentiment features in publication [6]. We found that it varies between categories and algorithms whether adding sentiment features to multi-gram models increase performance or not. This seems to suggest that feature selection algorithms could further improve the results, and that there is further room for fine-tuning the algorithms.

The performance difference between previous models and the multi-gram models could be explained by either underfitting, overfitting, or a combination

Category 8	Multi-gram + Sentiment Classification Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	92.56%	0.95	0.90	0.92
ANN	95.28%	0.96	0.95	0.95
SVM/ANN	93.85%	0.93	0.95	0.94
DT/SVM/ANN	94.56%	0.96	0.93	0.94
DT/SVM/ANN/k-NN	94.42%	0.94	0.95	0.94
Category 12	Multi-gram Classification Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	91.46%	0.92	0.90	0.91
ANN	96.34%	0.95	0.98	0.96
SVM/ANN	92.68%	0.97	0.88	0.92
DT/SVM/ANN	93.90%	0.91	0.98	0.94
DT/SVM/ANN/k-NN	95.12%	0.97	0.93	0.95
Category 13	Multi-gram Classification Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	86.72%	0.94	0.78	0.85
ANN	93.04%	0.92	0.94	0.93
SVM/ANN	89.83%	0.96	0.83	0.89
DT/SVM/ANN	91.33%	0.92	0.90	0.91
DT/SVM/ANN/k-NN	90.58%	0.94	0.87	0.90
Category 17	Multi-gram Classification Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	90.89%	1.00	0.82	0.90
ANN	94.79%	0.95	0.94	0.95
SVM/ANN	90.89%	1.00	0.82	0.90
DT/SVM/ANN	93.75%	0.99	0.88	0.93
DT/SVM/ANN/k-NN	92.45%	1.00	0.85	0.92

Table 18. *Multi-gram classification performances for all four categories using 80 similarity features and 13 sentiment features. Results taken from publication [6].*

of both. Underfitting means that a model and its features does not represent the data fully and have reduced predictive performance. In our case, it could be that the previous models (unigram and n-gram) were underfitting the data and that the multi-gram models simply are a better representation of the data. On the other hand, it can also be that the multi-gram models overfit the data, giving models that show excellent performance with the data they are trained on, but the performance not translating well to future data.

4.2.8. Imbalanced Testing

In a final experiment on automatic text classifications, we will test the performance of the models on test sets that better represents the natural skew between classes in real-world scenarios. We will here see that performance drops significantly, for all models, on highly imbalanced data, with multi-gram ensemble performance holding up better than other models. Here we also answer the last part of research question three.

As the number of violent and hateful web pages on the Internet are few compared to the total number of web pages, we should account for that in our experiments to see whether the results can be applied in practice. To do this, we test the models that we have developed on imbalanced labelled test sets. Here we need to note that the imbalance skews the dynamic of the datasets. Understanding the performance is no longer as clear as when we used the negative under-sampling approach to have roughly 50% positive and 50% negative distributions in sections 4.2.3 – 4.2.7.

Accuracy, precision, and recall are not necessarily descriptive measures of performance for imbalanced datasets (Jeni et al. 2013). We use test sets with imbalances close to 5% positive and 95% negative instances, which gives us a skew of 20 (Jeni et al. 2013). A classifier classifying a dataset with 95% negative samples would be able to have a 95% classification accuracy simply by predicting all pages as negative, which is why we should use the F-measure instead to understand the performance. The imbalance was chosen to reflect an even spread between the 20 categories in the training set, mainly because we do not know the real-world category distributions. We do know that the 20 categories we have used do not represent all types of text content on the Internet, and because of that it is likely that even when tested with an imbalance factor of 20, our tests will not be equivalent to real-world performance.

We only show tests for the best performing models. This test is performed by training the algorithms using balanced datasets with the same parameters as we defined through using cross-validation. The performance tests for the four categories are done using imbalanced test sets consisting of instances that were not included in the training. Using research paper [6], we provide the final part of the answer to research question three by showing that multi-gram models combined with sentiment, on average, does not improve classification performance. We also find that the voting ensemble using ANN/SVM improves our results over individual classifiers at high class imbalances.

Tables 19 and 20 show the performance of our multi-gram algorithms when run on the imbalanced test sets. Looking at the F-measures, we can see that performance for all categories drop. The four categories have an F-measure performance of between 0.71 to 0.83 when not including sentiment features, and a range of 0.69 to 0.82 when including sentiment features. We compare this range to that of tri-grams and five-grams with sentiment features, which were the best performing among the other models tested. The tri-gram models have an F-measure range of 0.44 to 0.68. The five-gram models have an F-measure range of 0.57 - 0.68, both significantly lower than the multi-gram models.

Using the 95-to-5 imbalance, the multi-gram models without sentiment features outperformed those with sentiment features, however, when testing other sized imbalances, we noticed that this was not always the case. In Tables 19 and 20, we can see that the ensemble classification that uses ANN/SVM outperform the other classifiers, except for category 17 where the ensemble classification using ANN/SVM/DT/k-NN provide the best overall results.

If we want to compare these results to those of other models that have been applied on similar problems, we can examine a problem with similar context (web pages) and similar imbalances between classes. While the results should not be directly compared without running on the exact same data, we can at least determine whether these multi-gram models perform in the same ranges as other models. We find one study that uses a similar 95-to-5 imbalance on web pages, one from the web spam challenge (Erdélyi et al. 2011). However, they report performance results using the area under the receiver operating characteristic (AUROC). To compare performance against their results, we need to know our AUROC performance as well. The best performing models presented on the web spam filtering challenge have shown an average AUROC of 0.892. The average AUROC for our multi-gram models without sentiment features is 0.950, and 0.952 with sentiment features, which equals higher performance. The conclusion

Multi-gram with Sentiment Imbalanced Performance				
Category 8	5.1 % / 94.9 % Imbalanced Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	95.60 %	0.53	0.88	0.66
ANN	95.43 %	0.52	0.92	0.66
SVM/ANN	93.48 %	0.42	0.94	0.58
DT/SVM/ANN	96.10 %	0.56	0.90	0.69
DT/SVM/ANN/k-NN	94.35 %	0.46	0.94	0.62
Category 12	5.3 % / 94.7 % Imbalanced Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	96.16 %	0.57	0.93	0.71
ANN	97.07 %	0.64	0.98	0.77
SVM/ANN	97.90 %	0.73	0.93	0.82
DT/SVM/ANN	95.61 %	0.54	0.96	0.69
DT/SVM/ANN/k-NN	96.61 %	0.61	0.93	0.73
Category 13	5.1 % / 94.9 % Imbalanced Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	94.25 %	0.45	0.82	0.58
ANN	93.17 %	0.41	0.92	0.57
SVM/ANN	96.82 %	0.64	0.80	0.71
DT/SVM/ANN	96.16 %	0.57	0.84	0.68
DT/SVM/ANN/k-NN	96.85 %	0.64	0.80	0.71
Category 17	4.9 % / 95.1 % Imbalanced Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	96.85 %	0.62	0.83	0.71
ANN	79.70 %	0.18	0.99	0.31
SVM/ANN	96.95 %	0.63	0.83	0.72
DT/SVM/ANN	95.70 %	0.52	0.88	0.65
DT/SVM/ANN/k-NN	96.57 %	0.59	0.86	0.70

Table 19. *Testing the performance of the best performing multi-gram models with 80 similarity features on natural sized datasets. Results taken from publication [6].*

Multi-gram Imbalanced Performance				
Category 8	5.1 % / 94.9 % Imbalanced Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	95.81 %	0.54	0.88	0.67
ANN	96.72 %	0.61	0.90	0.73
SVM/ANN	94.50 %	0.47	0.93	0.62
DT/SVM/ANN	96.50 %	0.49	0.92	0.64
DT/SVM/ANN/k-NN	95.06 %	0.49	0.92	0.64
Category 12	5.3 % / 94.7 % Imbalanced Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	96.89 %	0.63	0.91	0.75
ANN	97.16 %	0.96	0.65	0.77
SVM/ANN	98.08 %	0.76	0.91	0.83
DT/SVM/ANN	96.43 %	0.59	0.95	0.73
DT/SVM/ANN/k-NN	97.26 %	0.95	0.66	0.78
Category 13	5.1 % / 94.9 % Imbalanced Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	94.28 %	0.45	0.81	0.58
ANN	94.41 %	0.46	0.91	0.61
SVM/ANN	96.95 %	0.63	0.80	0.71
DT/SVM/ANN	95.18 %	0.50	0.86	0.63
DT/SVM/ANN/k-NN	96.60 %	0.61	0.85	0.71
Category 17	4.9 % / 95.1 % Imbalanced Performance			
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM	95.94 %	0.54	0.86	0.66
ANN	94.64 %	0.46	0.95	0.62
SVM/ANN	97.38 %	0.67	0.85	0.75
DT/SVM/ANN	96.53 %	0.58	0.89	0.70
DT/SVM/ANN/k-NN	97.40 %	0.67	0.87	0.76

Table 20. *Testing the performance of the best performing multi-gram models with 80 similarity features and 13 sentiment features on natural sized datasets. Results taken from publication [6].*

we can draw from this comparison is that the multi-gram models seem promising and should be tested on other datasets as well.

While the performance does not quite meet the requirements for practical use that were defined earlier (F-measure of 0.9), the performance of the classification can still be acceptable in practice. The reason is that we generally

are not concerned about sub-categories in practical applications. If we take parental control systems, for example, we will want to filter all hate categories. In this practical application, we do not care whether we have classified them into the correct hate sub-group. As we are not concerned whether we misclassify one hate or violent category as another hate category, hate content is either filtered out correctly or it is not filtered.

Using category 12, we can show that the classifications have practically acceptable performance also on the imbalanced dataset, if we relax the classification conditions to account for overlap. Let us further examine the 16 false positives web pages that we have in the best performing category 12 NN/SVM ensemble classification, which showed an F-measure of 0.83. As can be seen in Table 21, 13 of those 16 false positives belong to another hate category (category 13), which means that in a practical system we want these filtered out, and they should not be considered false positives. If we exclude these 13 false positives from the result classification and recalculate performance, as shown in Table 21, we reach an F-measure of 0.93.

4.2.9. Uses and Practical Relevance

In the beginning of the chapter, I mentioned Neil the security expert. Here we will discuss the practical uses this research could have in his line of work. As of October 2016, the Internet consisted of over 4.81 billion web pages ("WorldWideWebSize" 2016). The pages are so many that the human mind can

List of False Positive Classifications Category 12				
Category	16 False Positives			
Casino (3)	1			
Racism (13)	13			
Sports betting (15)	2			
Recalculated Imbalanced Performance Category 12				
ML-algorithm	Accuracy	Precision	Recall	F-measure
SVM/ANN	99.26 %	0.94	0.91	0.93

Table 21. *Recalculated performance for category 12 when category limitations are relaxed to fit a real-world scenario. In a parental control application, the pages would either be filtered or not, meaning the 13 false positive web pages from category 13 are in practical terms not false positives as they should be filtered. We then only have 3 false positives in the classification if the ones from category 13 are excluded. Results from publication [6].*

barely grasp the number's significance, let alone manually working through and classifying all of them.

Automatic systems must be developed to be able to keep up with the growing web. For professionals such as Neil, the optimal solution would be systems with performance at such a level that the results can be relied on without manual work. The research that was presented on automatic classifications is an approach that comes close to achieving that. Using our methods, practically acceptable performance was presented for all four categories with balanced datasets, and one example was shown of how that performance can be practically acceptable using highly imbalanced datasets. Still, these numbers were achieved in an experimental setting and might not fully translate to real world performance. We saw performance drops when we changed distributions and can expect further changes in performances in real-world settings.

Regardless, the automatic classification performance shown in our experiments could already in their current state be used in many applications. The first most obvious practical use could be to replace existing filtering systems, in parental control systems and/or other web site filtering systems, in cases where our algorithms provide better overall performance than what is currently in use. These algorithms could reduce problems for Internet users by reducing the number of sites that become incorrectly blocked by Internet Service Providers.

Other possible uses would be in other types of text classifications such as event detection, spam detection, or automatization of manual text processing tasks. Even in situations where our model performance is not directly usable in practice, the algorithms could be used to reduce the manual workload needed by functioning as a filter. In situations where we want to minimize false positives, the manual workload would be focused on going through positive label predictions, and in situations where we minimize false negatives, the manual workload would be focused on going through negative label predictions.

4.2.10. Reflections on Classifications

The automatization of classifications that was presented shows potential and could be adapted into a practically useful tool used by people like Neil, who otherwise would need to process large amounts of text through other solutions. Google Safe-browsing is an example of a competing product, however, the functionality is limited to web site URLs only, which means it cannot be used for general text classification ("Google Safe Browsing" 2017). ANNs performed well

on the balanced training data, however, we saw a significant performance drop when tested on imbalanced data. The ensemble performance and specifically the ensemble vote using SVM/ANN did well and had a lower performance drop on imbalanced data.

I have two specific insights that I would like to offer regarding the performance. The first insight was gained from the experiments done in research papers [2] and [3]. Sampling the negative training data evenly between the different 19 categories had the largest positive impact on performance of everything we did in our classifications, compared to fully randomized negative sampling. Even though we used binary classifiers, it was useful to extract features for many other categories. Creating more sub-categories could further increase performance. The second insight is one gained from publication [6], which is that multi-gram classifications stabilize results while increasing performance across the board. Going from unigram to multi-gram classifications increased our pre-processing and training times, however, when the difference is between having practically usable results and not having them, the multi-gram route becomes the obvious choice.

4.3. Financial Analytics Results

In this part of the chapter, we will review the financial news analytics contributions that are part of the thesis. We will also discuss practical tools that the contributions could be turned into that could directly affect Jenna's work positively. People in Jenna's line of work, such as money managers and institutional investors, are interested in tools that can help them decide on when to re-allocate portfolio holdings. Professional investors rely more on quantitative models and are less likely to invest based on what they see in news (Barber and Odean 2008). They need to keep up to date on a wide array of companies, both the ones they currently have holdings in and potential other companies. Any tools that can help them do that can be of use. Optimal tools should be able to recommend an action based on the data so that Jenna will not need to put time into analyzing the data herself.

4.3.1. Dataset

The dataset used in our financial news research was gathered at the start of the project by scraping web pages of the Internet. We gathered a set of roughly 18,300 financial news articles written by about 3,600 unique authors from the

news site Seeking Alpha ("Stock Market Insights" 2016). The articles in the dataset are labelled by the authors themselves as either positive or negative. The label can be interpreted as a self-reported sentiment value. About 75% of the articles in the dataset were labelled as positive, and about 25% were labelled as negative. The articles gathered were published between the start of 2006 and the end of the second quarter of 2016. The content of each article is an analysis made by the author of the article about companies, commodities, indexes, sectors, or a combination of components. We treat the positive and negative sentiment values given to the articles as a reflection of the author's expectations. The basic assumption made regarding author sentiment is that when an author writes an article about Apple Inc, and provides it a positive label, he or she predicts that Apple will perform better than the average company in the market, either in the immediate future or over the long term. This assumption also means that a negative label is an expectation of worse than average performance in the stock market.

We can split financial news into different categories depending on the requirements that come with the type of news. Authors have different perspectives and different incentives to why they are publishing news content. There can also be biases in news depending on how authors are positioned relative to what they are writing about. For example, a gold proponent will most likely write positive articles about gold when he or she has the opportunity to voice opinions.

If we split financial news into categories based on the requirements, the first type of news that needs mentioning is reports mandated by different governments or bodies of authority. Most governments require publicly traded companies to annually submit a set of financial reports that explain the state of the business. Misrepresenting information in such reports is illegal. The second type of financial news articles are reports published by registered analysts. These analysts have a mandate to stay objective and can also be held accountable for information that they publish, although that rarely happens. Registered analysts post suggestions and expectations regarding company performances based on the numbers that the companies present. Many analysts use scales of sell, hold, and buy to rate companies. The third type is the main source of financial news. Here we have journalists and industry professionals that publish articles, for example, at Reuters or Bloomberg's. These types of articles can contain opinions and the authors are held accountable by the company's internal ethics guidelines and policies. Lastly, we have crowd sourced news, such as the dataset that we have gathered. Crowd sourced news are news sites where anyone can post their

own analysis, opinions, and ideas. The only requirement to post content to such sites is that the post follows the guidelines of the site.

The intent behind posts on crowd sourced news sites is not vetted, and because of that it would be fully possible for a person to intentionally deceive or manipulate others through their articles. Yet, the popularity of these types of sites have steadily grown over the last decade. Grown to the extent that many of the sites now are considered widely successful. We find this dynamic interesting and that is part of the reason why we chose to focus our research on this type of financial data. Furthermore, comparatively little research has been done on crowd sourced news, especially in comparison to the research done on the standard financial news datasets, such as the different Reuters news corpuses.

4.3.2. Economic Measures and Equity Valuations

Publicly traded markets, and the price movements of indexes and individual components in these markets, are a central part of most of the economies on the globe. While reading news, we are bombarded by different opinions, views, and conclusions. When we try to turn the news into investable or tradable actions, we are just as likely to lose money as we are of increasing our returns, unless we are following a rigorous approach when interpreting news. Therefore, many money managers, risk managers, and successful investors turn to different market measures to understand the markets and the investment environment.

There are thousands of different economic measures available to choose from depending on what we are searching for. There are differences in measures depending on strategy, time frame, data types, seasonality, and many other factors. All measures aim to offer more information or new information about a specific part of the economy, for example, to help managers have more information available when the time comes to make decisions.

Different economic measures are useful in different situations. Some of the best-known metrics used to value publicly traded companies are the following: discounted cash flow analysis (DCF), free cash flow to equity (FCFE), price-to-earnings ratio (P/E), price-earnings-growth ratio (PEG), enterprise value to EBITDA multiple (EBITDA/EV), price-to-book ratio (P/B), revenue multiples, and sales multiples (Damodaran 2012). These are relative measures and whether a company is considered overvalued or undervalued depends on what we are comparing the methods to. A company can be considered undervalued relative to its peers but overvalued relative to the broader market (Henry et al.

2010). Furthermore, if we are talking about size of companies, then the most widely used measure is market capitalization. It is the total value of the shares outstanding of a publicly traded company.

In our research into financial news analytics, we are interested in developing new measures that could be of practical use and that could help automate news processing tasks. We are also interested in defining new measures that could be further developed into tools used inside the finance industry. This research is part exploratory and part quantitative. In this part, we combine our previous text extraction and text analytics knowledge with different types of network analytics to explore the relationships between companies in news. We introduce two new ways of measuring companies: methods for ranking companies based on news flow and a way of measuring sentiment-based company risks. Worth noting here is that we are not trying to replace traditional valuation approaches, instead we are developing new measures to complement the existing methods or provide new insights into trends, investment periods, and risks. The markets can stay irrational for long periods of time (Shiller 2015), which means that valuations do not always return to the expected valuation in the short term. Hence, we will cover developing methods that can help Jenna and her peers decide when to move in or out of positions.

4.3.3. Sentiment in Finance

Sentiment in finance differs somewhat from sentiment in violence and hate classifications. Sentiment refers to the tone of a text and through sentiment analysis we try to find either emotionally loaded words or a general tonality in sentences, paragraphs, or entire texts. Sentiment in finance is more complex for two reasons. First, the tonality and meaning of some words have been shown to be different in finance (Loughran and McDonald 2011). This means that we need additional knowledge in the form of different dictionaries or training sets when working on financial sentiment analysis. Second, sentiment in finance is not useful unless we know the specifics of what the sentiment is referring to. Knowing that a text is positive or negative does not add much value in finance, unless we know, for instance, which company or commodity the text is referring to. Essentially, for sentiment analysis in finance to be deemed useful, we need more information than general sentiment analysis offers.

In the dataset that we gathered, we have those necessary components. We have the articles labelled with an author reported sentiment value, and from the texts we use our text-extraction methods from [1] and [4] to link the sentiment to

companies that are mentioned. The documents in the dataset follow a general structure where companies mentioned are followed by the company's trading ticker in parentheses, for example, "Apple Inc. (AAPL)." That makes it easy for us to reliably extract all the companies tagged by the authors in the texts using regular expressions. Furthermore, we use a list of the company names found in the Standard & Poor 500 index during May 2016 to search for companies that the authors have mentioned but not appropriately tagged in the texts. To identify such companies, we search for full company name matches or partial matches in cases where the name contains several words. For example, in the case of "American Airlines Group Inc.", we accept a match of "American Airlines" with or without "Group" and "Inc." found in the text. The number of non-tagged companies that we find are marginal to the total number of identified company mentioning's in the texts. This means that thanks to the structure of the articles, the identification of companies in the articles is highly accurate.

4.3.4. Sentiment-based Co-occurrence Networks

Once we have extracted the different company occurrences for each article in the dataset, we want to extract new information by examining relationships between different companies. We are also interested in finding out whether these relationships are different if we group the dataset into subsets based on sentiment polarization.

Before we start examining the relationships between companies, we choose to limit the scope of our research to the companies found in the Standard & Poor 500 index. Thus far we have extracted companies that are mentioned in the news articles, and now we continue building networks out of the identified companies. We do that, by creating co-occurrence networks of companies as described in section 3.3.1, the same approach as in research paper [4]. A co-occurrence is here defined as two companies mentioned in the same article. The basic idea is that if two companies often are mentioned together, then these two companies have formed some type of relationship. When positive or negative situations affect one of the companies, these situations will also influence the other company. We can find companies that are linked together through news by creating co-occurrence networks. Competitors and strategic partners are often mentioned together, however, it is worth noting that the two relationships are opposite.

We start by building one co-occurrence network for each quarter, consisting of all the S&P 500 companies mentioned in the dataset starting from Q1 2011 and ending in Q2 2016. We name the networks that consist of all articles in a quarter

the mixed sentiment networks. As we do not have more information about the dynamic of the relationships between companies in the networks, we decide to create undirected weighted networks. With more information, we could have created directed graphs. Each S&P 500 company becomes a node in the networks and we measure the edges between nodes as the number of times the companies that the nodes represent were mentioned together with the other companies during the quarter. Two companies that occurred in articles but were never mentioned together in the same article would not have an edge between their respective nodes in the network. Figure 3 shows an example of the whole network structure for Q1 2011. The size of nodes in a network reflects the total number of co-occurrences that company had during the quarter. The length of the edges in the networks means nothing, the algorithm used places nodes to be easily visualized. From Figure 3, we find that ranking and/or filtering nodes will be necessary for the visualizations to be useful as it is not understandable in its current format.

From there, we continue by splitting the dataset into two parts treated as subsets: the articles that have a positive sentiment label and the articles that have a negative sentiment label. We repeat the process of constructing co-occurrence networks of companies for both subsets. We then have three different types of networks: mixed sentiment networks that consist of all articles for each quarter; positive sentiment networks that consist of all the articles with positive labels, which translates to about 75% of articles for each quarter; negative sentiment networks that consist of negative sentiment articles for each quarter, which translates to about 25% of articles each quarter.

Here we partly answer research question two by showing how sentiment can be used to extract co-occurrence networks. Creating the mixed sentiment networks is the starting point to answering the fourth research question.

4.3.5. Company Sentiment Rankings from News

Now that we have converted the news articles into networks of nodes and edges with weights between nodes, we have enough information to start measuring and comparing the components in different ways. Here we come to the first measures used to rank companies by news flow, which could be of use to people like Jenna that are working in the finance industry. The methods were introduced in research paper [4].

Cluttered graph example

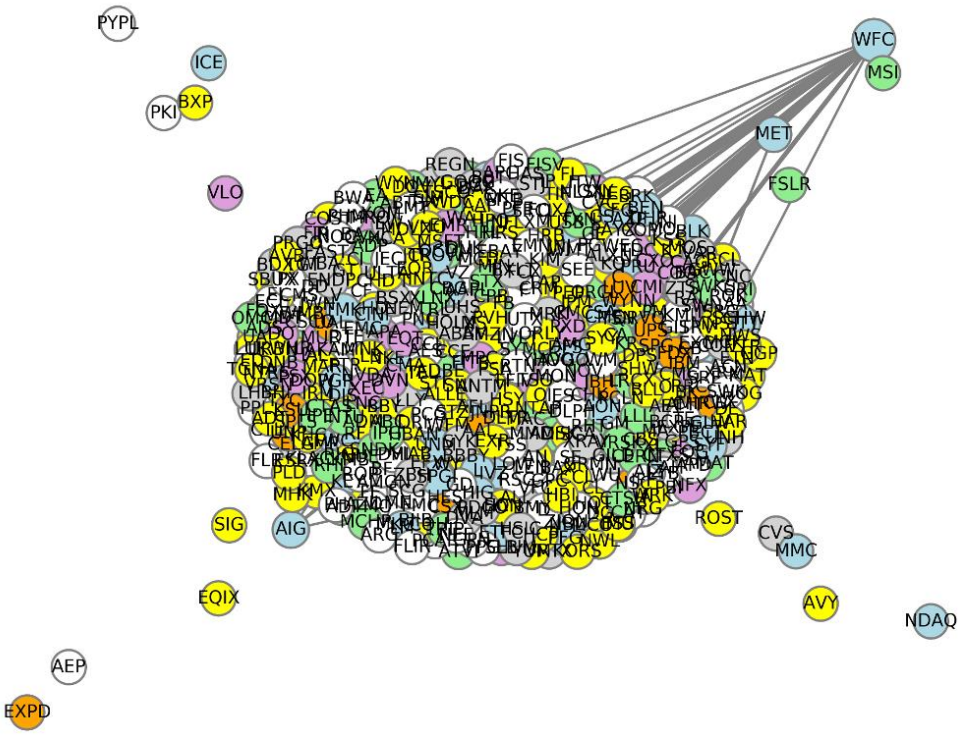

Figure 3. *Cluttered graph showing all S&P 500 components from Q1 2011. The figure illustrates why we need to filter and reduce the number of components for the visualizations to add any value.*

As mentioned in section 2.2.2.4, different centrality measures are normally used to quantitatively measure network components. We use the information centrality measure as defined in section 2.2.2.4.3, because we are interested in measuring the total flow through nodes by letting the flow take walks (traverse multiple paths), rather than just calculating the shortest path between nodes. In other words, by calculating information centrality we can compare the total flow of news through the different companies, and with the help of that we can find companies that have a large news presence.

Information centrality is a relative measure where the sizes of the values depend on the information that is processed. By using information centrality, we can both rank the nodes in the networks by value and determine which components to filter out to make visualizations techniques useful. However, information centrality calculations in our networks are only useful if the graphs are fully connected. Nodes need to have at least one edge that connect them to the rest of

the nodes or their information centrality values will not be comparable to each other. This means that the information centrality measure would not be useful in situations where we have several sub-networks inside a quarterly network. To pre-emptively remedy that problem, we need to first add a Laplace smoothing (Chen and Goodman 1996) to each connection between the nodes in the network. Laplace smoothing is an additive smoothing technique where we give each data point the same added smoothing value (in our case we test adding values 0.1 and 1). This allows us to connect all nodes and compare them to each other, instead of only comparing inside sub-networks. The value that we give the smoothing can be seen as representing unknown relationships between different components in the networks (Rönnqvist and Sarlin 2015). Essentially, a higher smoothing value would mean that we assume a higher degree of uncertainty due to unknown factors, such as missing data, and a low smoothing value would assume that the networks are less affected by unknown factors.

We define the formula for calculating node values as a slightly modified version of the formula used by Rönnqvist and Sarlin (2015). Here we need to consider different types of networks as we earlier split the dataset into three types of networks (mixed, positive, negative). The formula for information centrality (51) is the same as in section 2.2.2.4.3, with the modification that n is decided by the nodes from the different networks sets: positive network nodes (52), negative network nodes (53), or mixed network nodes (54):

$$I(i) = \frac{n}{nC_{ij} + \sum_{j=1}^{n} C_{jj} - 2\sum_{j=1}^{n} C_{ij}} \tag{51}$$

$$n = n_{positive} \tag{52}$$

$$n = n_{negative} \tag{53}$$

$$n = n_{negative} + n_{positive} \tag{54}$$

To gain an understanding of how the smoothing parameter affects the centrality calculations, we try two different experiments: first we calculate information centrality in all networks using a Laplace smoothing of 0.1, and based on these measures we calculate the average rank of companies over all quarters for all three networks. Second, we calculate information centrality using a higher Laplace smoothing of one and repeat the calculation of the average rank using this smoothing. Table 22 shows the effect that changing smoothing has on the average basic ranking of the S&P 500 index companies. Due to the relatively low average edge count in all networks (1.76 mixed, 1.64 positive, 1.49 negative), we

decide to continue using Laplace smoothing of 0.1 for the rest of our experiments.

With the help of the information centrality measure, we can now filter which nodes to visualize. If we want to filter the nodes based on centrality, we have the option of examining companies in each network with either the highest or the lowest centrality values. However, analyzing nodes with the lowest centrality values does not make sense in this context, because there are several companies that are not found co-occurring in any given quarter. Visualizing and comparing the nodes with the lowest centrality values, would simply be an exercise in analyzing companies that were not mentioned. As that does not interest us in the context of this research, we continue by analyzing nodes based on the highest centrality values.

Studying the first ranking in Table 22 tells us that it consists mostly of large corporations with consumer oriented products, where Apple Inc. places first in all rankings, and Netflix Inc. is the only company to move up many positions between rankings, coming into top 5 of the negative ranking. The knowledge that the ranking contains mostly larger companies provides us an incentive to further develop the ranking methods. From here on, I will refer to the ranking with Laplace smoothing 0.1 as the absolute ranking. We now become interested in finding companies that have a high news flow relative to their size. To do that, we need to normalize the information centrality calculations.

To better understand the rankings, we try two different normalizations. The first normalization $J(i)$, which is defined in equation (55), divides the quarterly centrality values by market capitalization m. The average top 25 components that use this single normalization are shown in Table 23, to the left. In the second normalization, we start by normalizing information centrality to values between 0 and 1, this is done because the centrality measure is not linear. Because of the nonlinearity, we expect ending up favoring small companies in the single-normalized ranking. After the centrality normalization, we apply the same market capitalization normalization as before. The second-double-normalized calculations are represented in equation (56), where I_{min} is the lowest information centrality value and I_{max} is the largest value in the network. In Table 23, to the right, the average top 25 components in the second-double-normalized ranking can be seen.

$$J(i) = (I(i)) * \frac{1}{m} \tag{55}$$

	Average rank top 25					
	Laplace 0.1			Laplace 1.0		
Pos.	*Positive*	*Mixed*	*Neg.*	*Positive*	*Mixed*	*Neg.*
1	AAPL	AAPL	AAPL	AAPL	AAPL	AAPL
2	MSFT	GOOG	AMZN	MSFT	GOOG	GOOG
3	GOOG	MSFT	GOOG	GOOG	MSFT	AMZN
4	AMZN	AMZN	MSFT	AMZN	AMZN	MSFT
5	INTC	INTC	NFLX	INTC	INTC	NFLX
6	FB	FB	WMT	FB	FB	WMT
7	IBM	WMT	FB	IBM	IBM	FB
8	WMT	IBM	INTC	WMT	WMT	INTC
9	BAC	NFLX	IBM	BAC	NFLX	IBM
10	GS	BAC	YHOO	HPQ	BAC	YHOO
11	HPQ	YHOO	MS	WFC	HPQ	MS
12	WFC	GS	VZ	CSCO	CSCO	VZ
13	CSCO	HPQ	TGT	GS	YHOO	TGT
14	YHOO	CSCO	HPQ	JPM	GS	HPQ
15	JPM	MS	ORCL	YHOO	ORCL	ORCL
16	KO	ORCL	CSCO	ORCL	MS	CSCO
17	ORCL	JPM	T	IP	JPM	T
18	IP	WFC	GS	KO	VZ	CRM
19	BRK	IP	CRM	C	WFC	GS
20	NFLX	KO	JPM	NFLX	IP	JPM
21	MS	VZ	QCOM	GM	KO	QCOM
22	C	QCOM	DIS	MS	T	DIS
23	GM	C	IP	JNJ	QCOM	BBY
24	JNJ	T	BBY	VZ	C	COST
25	QCOM	GM	COST	BRK	GM	MCD

Table 22. *Top 25 average rank for all companies by ticker from the S&P 500 list over the 2011 – Q2 2016 period. Split into two sets of Laplace smoothing of 0.1 and 1. Differences in rank between especially negative and positive networks are evident. The relative rank of companies does not significantly change using different smoothing coefficients. Results taken from publication [4].*

$$J(i) = \left(\frac{I(i) - I_{min}(i)}{I_{min}(i) + I_{max}(i)}\right) * \frac{1}{m} \tag{56}$$

Studying the normalized rankings in Table 23, we find that the first normalization, as expected, consists of mainly small companies and that the second double normalization consists of a mix of different-sized companies. Henceforth, I will refer to the double-normalized ranking as the normalized ranking, and will not continue analyzing the single normalized ranking. Netflix Inc. places first in all the normalized averaged rankings, which makes sense as it was already present in the absolute rankings and has a much lower market capitalization than for example Apple Inc. By using information centrality to rank companies, we have here partly answered research question two. We have

	Average rank top 25 normalized					
	Norm. by market cap			Double norm.		
Pos.	*Positive*	*Mixed*	*Neg.*	*Positive*	*Mixed*	*Neg.*
1	QRVO	QRVO	FSLR	NFLX	NFLX	NFLX
2	WRK	WRK	DNB	NVDA	BBY	BBY
3	URI	FSLR	PKI	BBY	NVDA	NVDA
4	FSLR	URI	OI	IP	IP	YHOO
5	DNB	DNB	GT	QRVO	YHOO	IP
6	PKI	PKI	LM	GPS	GPS	CMG
7	PBI	PBI	NFX	DNB	DNB	DO
8	OI	OI	HBI	YHOO	SPLS	SPLS
9	AIZ	AIZ	URBN	JNPR	JNPR	AVGO
10	AVY	AVY	PHM	FSLR	AVGO	M
11	ETFC	FLIR	SEE	SPLS	CMG	CRM
12	FLIR	ETFC	MLM	SEE	FSLR	JNPR
13	TE	HAR	SNA	AVGO	SEE	MLM
14	HAR	TE	CVC	DPS	QRVO	DNB
15	PDCO	PDCO	TSS	UA	DO	TGT
16	LEG	LEG	GAS	EA	UA	RIG
17	GT	GT	FTR	MU	EA	KSS
18	LM	LM	FL	HPQ	MU	AMZN
19	PBCT	PBCT	NDAQ	GT	M	GPS
20	NFX	ZION	AN	HRS	GT	HPQ
21	URBN	NFX	ENDP	DO	HPQ	TWC
22	ZION	URBN	TSO	AA	CHK	MS
23	HBI	HBI	LVLT	CHK	COH	COH
24	PHM	TGNA	DO	CMG	CRM	SEE
25	TGNA	PHM	SWKS	HRB	DPS	CA

Table 23. *Top 25 average rank for all companies from the S&P 500 over the 2011 – Q2 2016 period when normalized. We normalize by market cap to the left. To the right, we first normalize information centrality values between 0 and 1 and then we normalize by market cap. Results from publication [4].*

shown that it is possible to automatically rank companies based on news flow and sentiment.

As we determined earlier in Figure 3, we need to filter visualizations to be able to visually interpret the networks. Using the absolute ranking, we limit components in graphs and diagrams to the average top 25 nodes. Visualizations are here used as a way of exploring the data and giving us clues to what can be further quantitatively researched. One way of visualizing trends is to take cross-sectional snapshots of the quarterly networks and comparing networks in succession. We are especially interested in finding quarters where the visualizations show trends that would move in opposition to our expectations,

or quarters where trends seem to be heading in different directions in the different network types.

Using graphs, we identify a unique scenario in our data in three consecutive quarters from Q4 2013 to Q1 2014. The divergence between the quarters is visualized in Figures 4 - 6. At the start of Q4 2013, the oil commodity prices started falling and at the same time the information centrality values of energy companies started to go up, meaning that energy companies started to gain more news coverage. The interesting pattern here is that energy companies move into the top 25 of the positive absolute ranking, but not into the top 25 of the negative absolute ranking.

We do a few tests and comparisons to have a better understanding of what the news ranking we have created represents. First, we are interested to see if there is a difference between the representations of top ranked companies in the three different absolute networks. We count the times that companies are found as top ranked and show the counts in Table 24. We find that Apple Inc. takes the top spot in 59 quarters of the absolute networks. In the normalized ranking, we find that Netflix Inc. is the most frequent company in first place with a count of 19, and NVidia places second with a count of 9.

Next, we plot information centrality for the absolute ranking to better understand how information centrality moves quarter-to-quarter in the

Graph of absolute ranking top 25 components Q4 2013

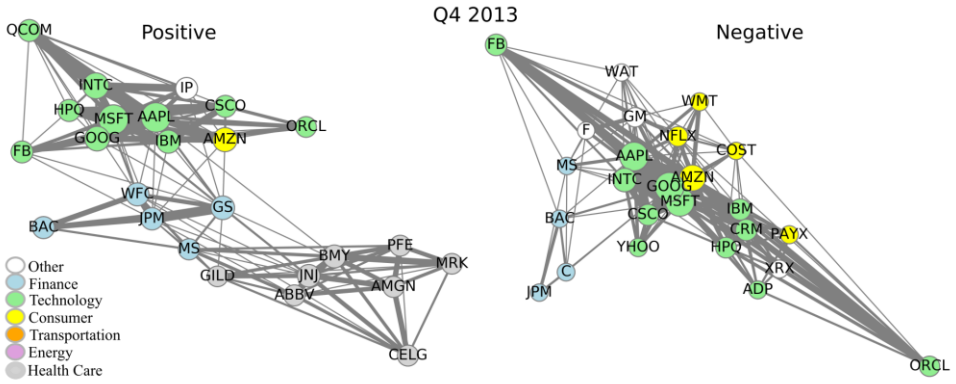

Figure 4. *Quarter 4 2013. Co-occurrence networks for absolute top 25 positive and negative nodes. Health care sector companies are represented in the top 25 positive nodes, but not in the negative, indicating that news flow increased for that sector. Thicker edges represent more co-occurrences between two entities and larger sized nodes indicate larger total sum of company occurrences. Length of edges in the networks are not representing any added value. Figure from publication [4].*

Graph of absolute ranking top 25 components Q1 2014

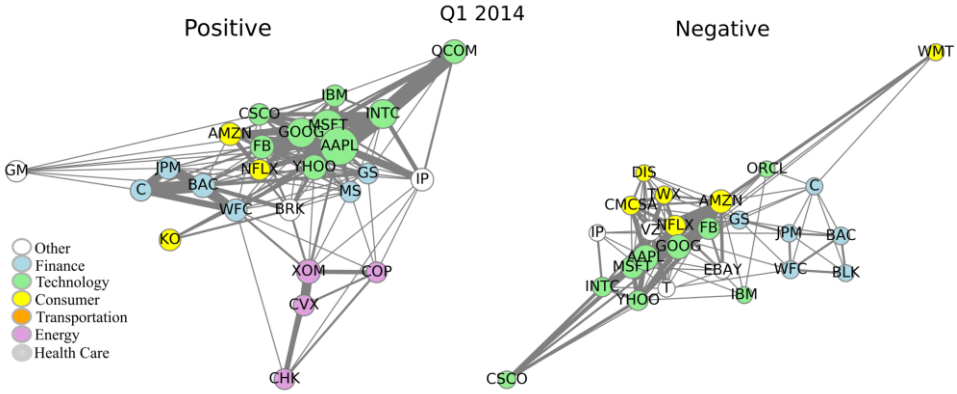

Figure 5. *Quarter 1 2014. Co-occurrence networks for top 25 positive and negative nodes. 4 nodes from energy sector jumps into top 25 of positive but not the negative network as a response to falling oil prices while health care sector falls from top 25. Our default assumption would have been that reduced performance of the energy sector due to lower oil prices would be more reflected in negative sentiment. Figure from publication [4].*

different rankings. We plot the results of the average top 25 components information centrality for the absolute network in Figures 7 - 9. From the plots, we can see that the different types of networks generally move in unison up and down, except for some isolated quarters when the information centrality seems to be moving at a steeper angle in either the negative or the positive figures. Two examples can be seen in Q1 2012 and Q2 2015.

Graph of absolute ranking top 25 components Q2 2014

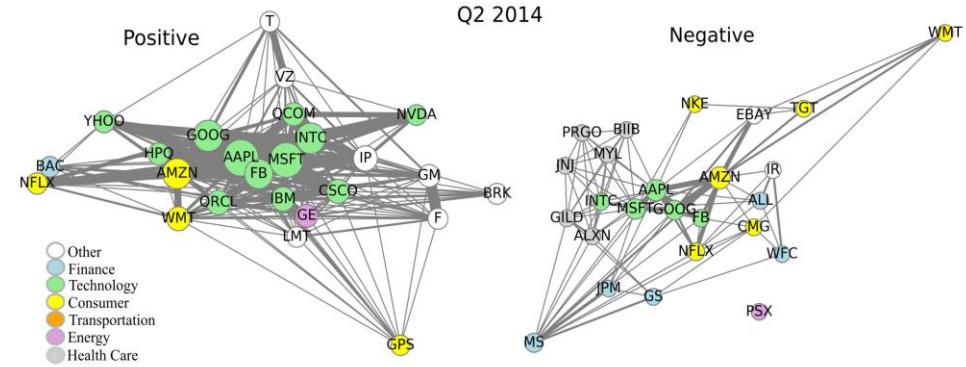

Figure 6. *Quarter 2 2014. Co-occurrence network of top 25 positive and negative companies. Oil related energy companies fall from the top 25 representations again and health care sector companies enter the negative top 25. PSX shown as alone in the network because the Laplace smoothing links of 0.1 are not drawn to reduce noise. This means that Phillips 66 (PSX) is only loosely connected to the larger network. Figure from publication [4].*

Quarterly Top Performing Absolute Components				
Company	**Positive**	**Mixed**	**Negative**	**Total**
AAPL	21	22	16	59
AMZN	0	0	4	4
GOOG	0	0	1	1
BAC	1	0	0	1
NFLX	0	0	1	1
Total	**22**	**22**	**22**	

Quarterly Top Performing Normalized Components				
Company	**Positive**	**Mixed**	**Negative**	**Total**
NVDA	4	4	1	9
NFLX	3	7	9	19
QRVO	3	1	0	4
FSLR	2	0	0	2
MCO	1	0	0	1
DO	1	1	2	4
GT	1	1	0	2
DNB	1	1	0	2
IP	1	0	0	1
SPLS	1	2	1	4
SEE	1	2	1	4
BBY	1	0	1	2
XLNX	0	1	1	2
AES	0	0	1	1
WAT	0	0	1	1
DRI	0	0	1	1
PMH	0	0	1	1
Total	**20**	**20**	**20**	

Table 24. *Companies with highest information flow per network. Results partly taken from publication [4], and partly extended for the thesis.*

In Figures 10 and 11, we plot the mixed network information centrality values for the top 25 companies in the absolute and the normalized ranking. This offers us an insight into how the two rankings differ from each other. We can in Figure 10 see Apple's dominance, while other companies change positions between quarters. We can see that as expected, the news flow values in Figure 11 are slightly lower on average, and we see how different companies place in the top at different times in the normalized ranking.

This far we have created two news rankings that could be of use in finding trends. If we try to apply them to Jenna's work, we can at this point state that the measures are not yet especially useful as she needs quantitative measures that statistically can be shown to help make decisions. At best, the methods would currently be able to help her recognize new investment companies, but she would still need other methods to analyze the companies.

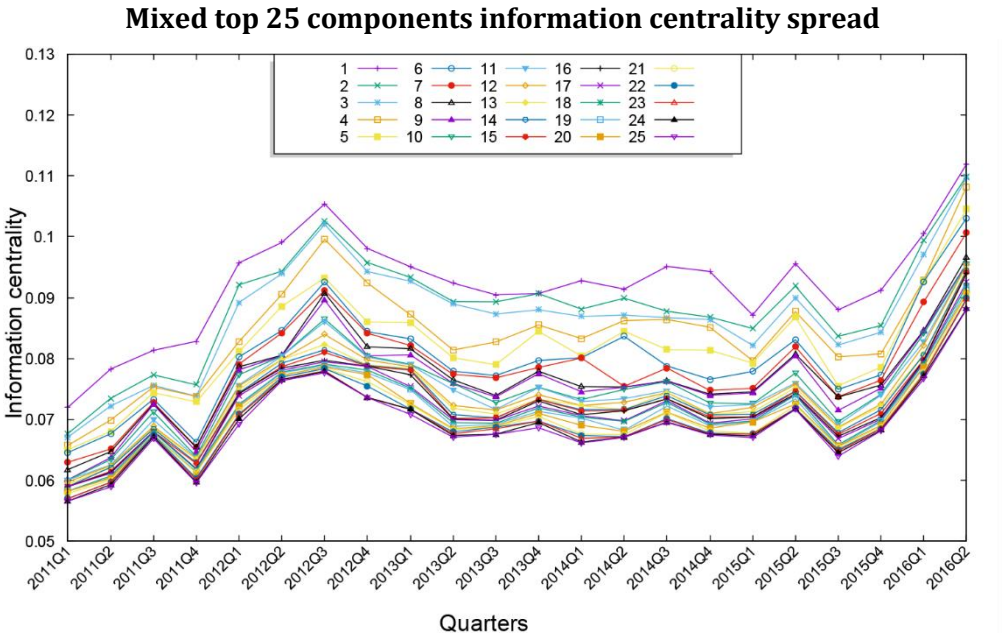

Figure 7. Line graph of information centrality for the average top 25 components in the mixed absolute network for each quarter.

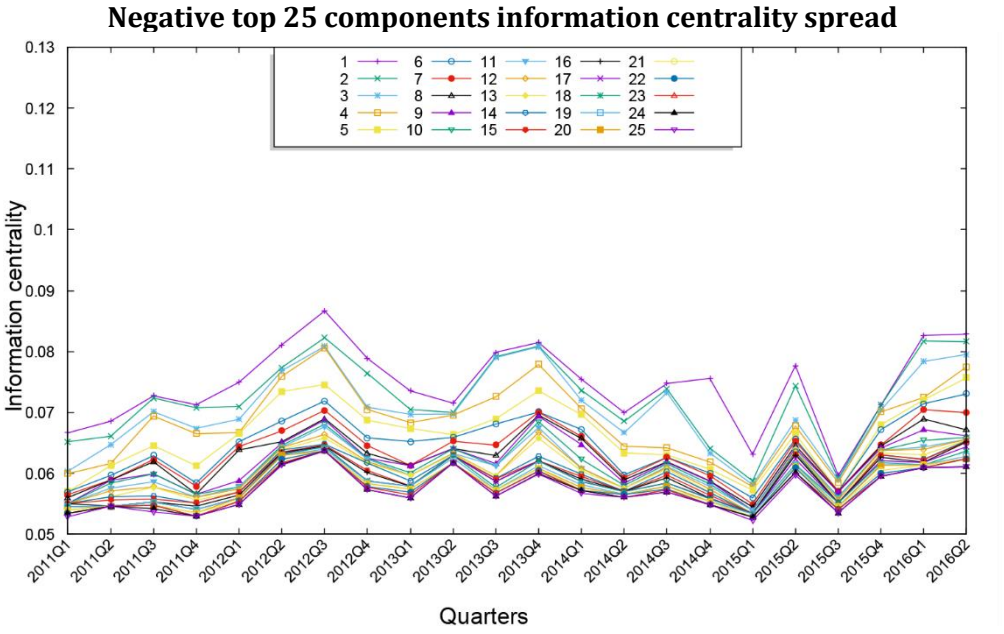

Figure 8. Line graph of information centrality for the average top 25 components in the positive absolute network for each quarter.

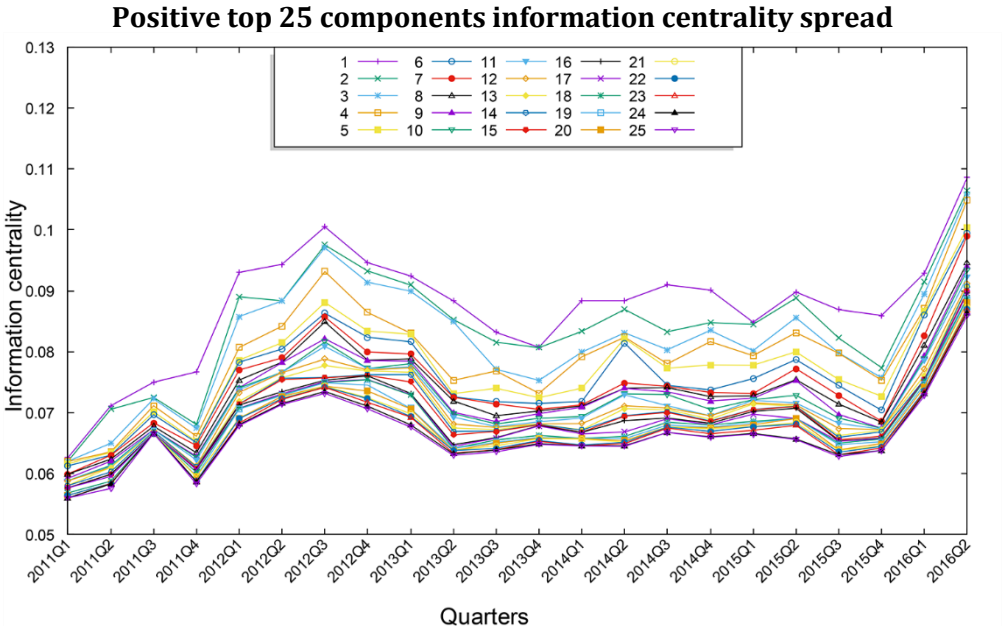

Figure 9. *Line graph of information centrality for top 25 negative firms for each quarter.*

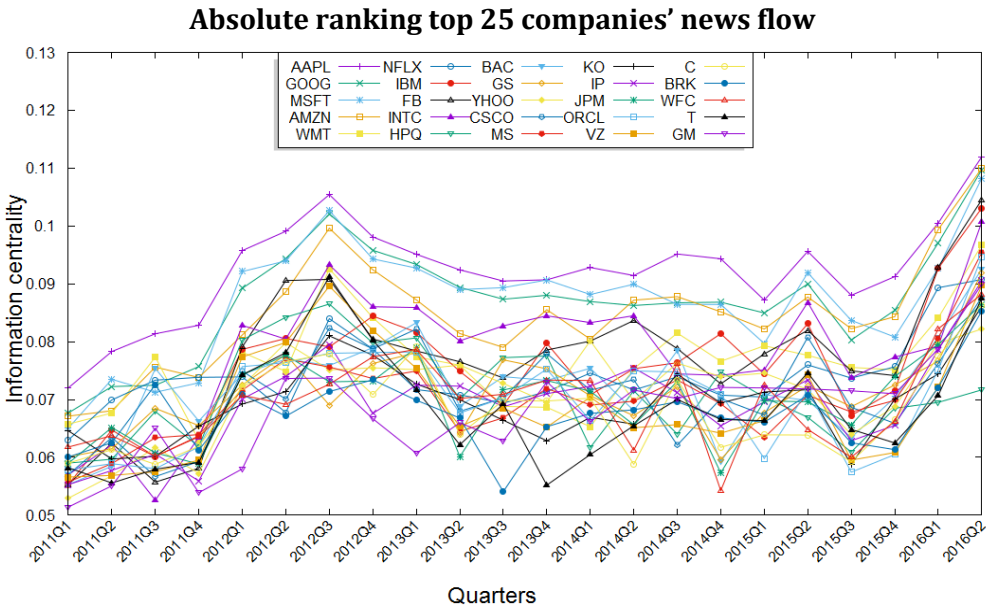

Figure 10. *Line graph of information centrality for the average top 25 companies in the absolute ranking. Figure taken from publication [5].*

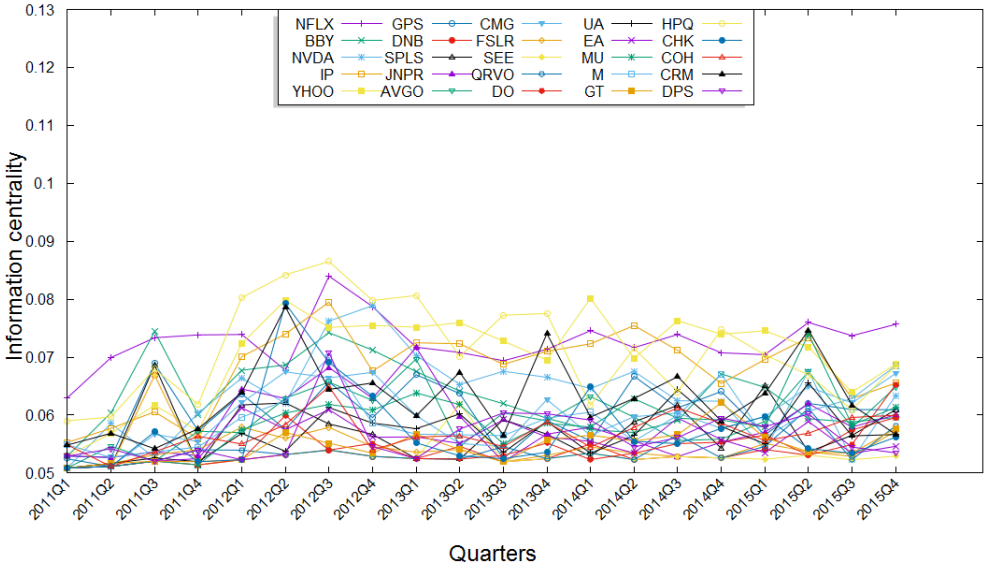

Figure 11. *Line graph of information centrality for the average top 25 companies in the normalized ranking. Notice that the graph ends after Q4 2015 as we lack market capitalization data for the last two quarters. Figure taken from publication [5].*

4.3.6. Company Risks from News

As a next step, we continue development of news-processing methods based on the same networks. The main problem that we so far identified in our ranking methods is that we have not been able verify that company co-occurrences, and in extension the rankings, contain any statistically significant information. Hence, we will now focus on quantitative measurements. The way we will achieve this is by using text-analytics methods to assess company sentiment risk from news. The method presented here is part of research paper [5].

Together with the self-reported sentiment for news articles, we use the mixed co-occurrence networks to develop a method that can indicate if a company is at higher risk of stock price decrease than the average company. Furthermore, statistically we will show that our methods for automatically parsing news are extracting useful information.

The risk model we use is based on RiskRank (Mezei and Sarlin 2017), as discussed in section 3.3.2.1. The general-purpose nature of the algorithm allows us to adapt it to our needs. As sentiment has previously been shown to move markets (Bollen et al. 2011; Checkley et al. 2017), we are interested in demonstrating that sentiment also can be used as a valuation indicator. In order to do that, we continue our work with co-occurrence networks. We need to first calculate individual sentiment values for each company. The assumption here is that if we aggregate mentionings of companies per quarter, we can from these mentionings extract additional information. We will use the ratio of number of negative mentionings $s_{negative}$ to positive sentiment mentionings $s_{positive}$ for a company, during the quarter, as the sentiment input. Thus, we define a relative sentiment variable s_{rel} as in equation (57):

$$s_{rel} = \frac{s_{negative}}{s_{negative}+s_{positive}} \tag{57}$$

We then use the relative sentiment value as the individual risk for companies each quarter and feed it into the risk algorithm together with the co-occurrence links that we have previously extracted for each quarter. As an output from the risk algorithm, we have three risk measures for each company each quarter: individual risk, direct risk, and indirect risk. The individual risk output is always equal to s_{rel} for the company, the direct risk is the sentiment risk that is transferred from directly linked neighbors in the network up to a cardinality of 2. Lastly, the indirect risk is a network-wide risk value, which we interpret as the current market-wide sentiment risk. When we aggregate the risk values together, the algorithm binds quarterly risk for a company to a value between

zero and one where the individual risk is always the relative measure we calculated in (57), and the other risk components are added to the individual risk, but bound at an upper value of 1, and a lower value of 0.

We have now partly answered research question four by showing how we can extract sentiment-based risks from news networks. Next, we will answer the remaining part of the question by showing how high-risk values are correlated to stock price movements. This also answers the last part of research questions one and two by showing how sentiment can be used to automate risk extraction from news.

Figure 12 illustrates our approach by plotting an example of the quarterly risk extracted for Apple Inc., and show it next to the company's stock price for the same period. Next, we move to statistically showing that stock price is at higher chance of decreasing 11-70 days after the risk value was measured, when aggregated risk reaches the maximum value of 1. Examples of aggregated risk

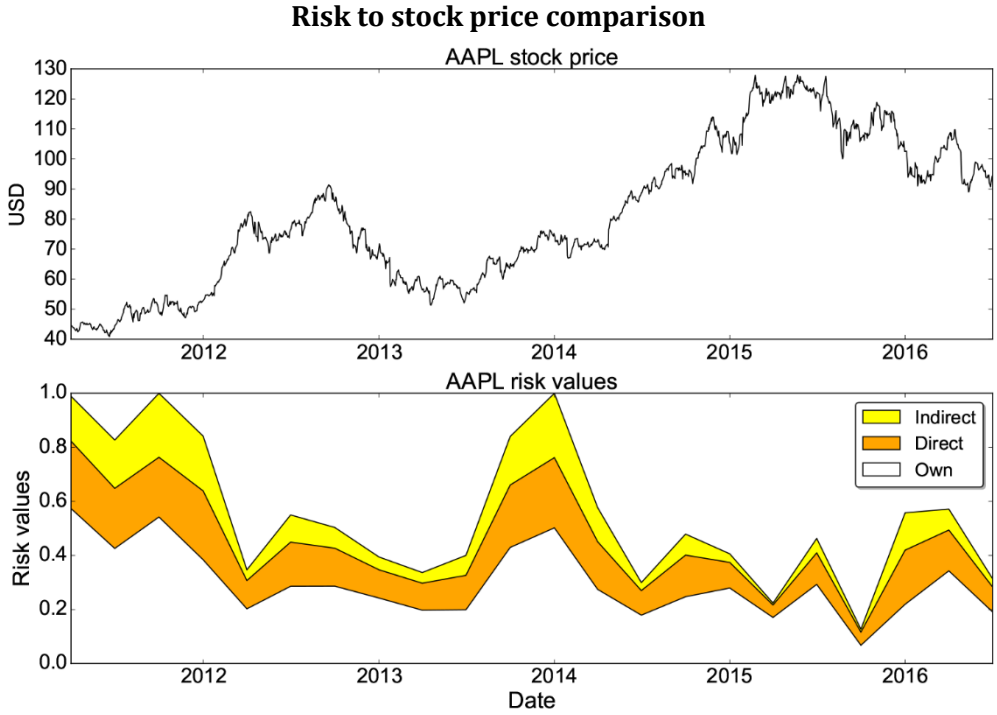

Figure 12. *Two subplots comparing Apple Inc. stock price to sentiment risk during the analyzed period 2011 – Q2 2016. The upper risk line represents aggregated risk, the lowest risk line represents the individual risk, and the middle risk line aggregates individual and direct risk. Figure taken from publication [5].*

reaching one can be seen in Figure 12 when the tops reach the upper bound of the lower graph during 2011 and 2014.

To statistically show that our approach is valid, we compare aggregated risk at risk threshold one against a benchmark, which consists of all the available quarterly data points in our analysis. We also compare aggregated risk against individual risk, mainly to determine whether the network effects add value over simply using the s_{rel} individual risks alone. Due to performance constraints introduced by the risk algorithm, we are only able to run the calculations on 50 components at a time. However, as we have developed two rankings that we find interesting, the absolute ranking and the normalized ranking, we decide to run our risk analysis on the components that place on average in the top 50 in either ranking. Twelve companies are found in the top 50 of both rankings, leaving us with 88 unique companies and 22 quarters of data. Here we are forced to discard many quarters due to not having available stock price data. We have 1864 data points in total for the benchmark comparisons. The reason that we choose the top performing components in the rankings is that they have a demonstrably higher news flow each quarter, which should translate into more reliable results than picking random components.

Figure 13 shows the number of data points for aggregated risk and individual risk at threshold intervals of 0.1. We can see that aggregating risk values considerably increases the number of data points for higher thresholds. For instance, at risk threshold 1, the individual data points number only 66, which is equal to about 3.5% of the total number of data points, while there are 176 aggregated-risk data points, which is equal to about 9.4% of the total quarterly data points.

Table 25 compares aggregated risk against the benchmark for different ranges of delays. The benchmark performance is used as a comparison instead of a 50% comparison, because of the upward bias that the stock market has shown since its inception. Comparing subsets of data points to all available data points provides us a way of proving statistical significance. In Table 25, we can see that the highest standard deviation intervals that we measured are found between 21-50-day delays, where all ranges are more than 10 standard deviations above the benchmark. While further examining the data, we find that we have a 13.1 percentage point higher risk of stock price decrease at a delay of 28 days, when the aggregated risk measure has reached one for a company.

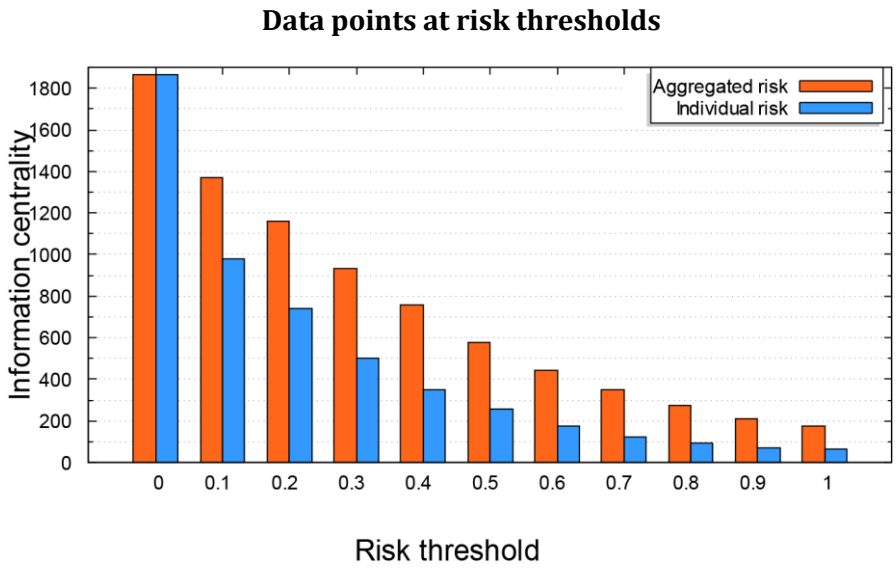

Figure 13. *Comparison of data points available at different risk thresholds with intervals of 0.1. Aggregated risks greatly increase data points for all higher thresholds. Results taken from publication [5].*

Table 26 further compares the aggregated measures against the individual measures, and finds that the aggregated risks outperform individual risks for the period of 11-40 days. Lastly, as the sets are of different sizes, we calculate the standard errors for the comparisons. For the periods of 11-20 and 61-70-day delays, which show the least statistical outperformance over the benchmark, we calculate the probability error to 0.09 for both ranges. For the individual subsets, we have a higher probability error of 0.14 as the sets have fewer data points. The probability error between aggregated risk and individual risk, at threshold 1, is 0.46 for the range between 11-50-day delays. We can conclude that all errors are lower than the standard deviation differences.

Research using sentiment analysis in finance has shown different time horizons for the predictive power, ranging from minutes to days (Bollen et al. 2011; Checkley et al. 2017). These studies have been conducted on other types of data, such as tweets, and formatted differently. Our findings here lead us to believe that different data types such as, for example full length articles and micro blogs, show predictive potential during different time horizons, and that depending on the methods used when aggregating and formatting data, we change when predictive power materializes. Self-reported author sentiment could also be

changing the predictive time horizon compared to sentiment extracted from the text content.

Finally, to see what could be further improved, let us compare the sentiment-risk methods described so far against the optimal solution that a finance professional like Jenna would use to support her work. As defined in section 3.1.4, optimal tools should contain prescriptive elements to reduce the work Jenna has to perform to a minimum. They should, for example, be able to say whether an asset should be added or removed from a portfolio. Our sentiment-risk models fall short of this goal as our work so far has not included portfolio strategies. This means that our methods could currently only be used as support to other investment strategies, such as those discussed in 4.3.2. For example, if we

Stock price decrease at risk threshold 1						
Days delay	**Agg. dec. %**	**Comp. dec. %**	**Abs. diff.**	**Rel. diff. %**	**St. dev.**	**St.dev. diff.**
3 to 90	46.93	42.11	4.82	11.44	2.82	1.71
3 to 45	50.74	43.71	7.03	16.09	3.09	2.28
45 to 90	43.28	40.58	2.70	6.65	1.29	2.09
3 to 10	50.21	45.29	4.92	10.86	5.36	0.92
11 to 20	52.67	46.22	6.45	13.95	1.52	**4.24**
21 to 30	51.59	42.00	9.59	22.83	0.72	**13.41**
31 to 40	50.11	42.75	7.36	17.22	0.56	**13.22**
41 to 50	47.33	41.52	5.81	13.98	0.56	**10.41**
51 to 60	47.95	41.80	6.16	14.73	0.88	**6.96**
61 to 70	41.99	40.14	1.85	4.61	0.66	**2.82**
71 to 80	41.31	40.17	1.14	2.83	1.29	0.88
81 to 90	39.83	39.73	0.10	0.26	1.22	0.08
Average	47.00	42.17	4.83	11.29	1.66	4.92

Table 25. *Average stock price decrease for different periods of time after news risk has reached the maximum value of 1.0. We compare aggregated risk at threshold 1 against the baseline risk of all data points. Results taken from publication [5].*

Aggregated vs. individual stock price decrease at risk threshold 1.0									
Days delay	**Agg. % decr.**	**Ind. % dec.**	**Comp. % decr.**	**Agg. diff.**	**Ind. diff.**	**Comp. st.d.**	**Agg.st. d.diff.**	**Ind.st. d.diff.**	**Agg. st.d. outperf.**
3 to 90	46.93	44.11	42.11	4.82	2.00	2.82	1.71	0.71	1.00
3 to 45	50.74	45.42	43.71	7.03	1.71	3.09	2.28	0.55	1.72
45 to 90	43.28	42.86	40.58	2.70	2.28	1.29	2.09	1.77	0.33
3 to 10	50.21	41.86	45.29	4.92	-3.44	5.36	0.92	-0.64	1.56
11 to 20	52.67	48.33	46.22	6.45	2.11	1.52	4.24	1.39	**2.85**
21 to 30	51.59	45.91	42.00	9.59	3.91	0.72	13.41	5.46	**7.95**
31 to 40	50.11	44.85	42.75	7.36	2.10	0.56	13.22	3.77	**9.46**
41 to 50	47.33	45.91	41.52	5.81	4.39	0.56	10.41	7.87	**2.55**
51 to 60	47.95	48.03	41.80	6.16	6.23	0.88	6.96	7.05	-0.09
61 to 70	41.99	41.06	40.14	1.85	0.92	0.66	2.82	1.40	1.42
71 to 80	41.31	41.52	40.17	1.14	1.34	1.29	0.88	1.04	-0.16
81 to 90	39.83	39.09	39.73	0.10	-0.64	1.22	0.08	-0.52	0.61
Average	47.00	44.08	42.17	4.83	1.91	1.66	4.919	2.487	2.432

Table 26. *Average stock price decrease for different periods of time after news risk has reached the maximum value of 1. We compare aggregated risk against individual risk. Results taken from publication [5].*

through a book-to-market valuation analysis find that Walt Disney Co. fit wll to a portfolio, we could complement that analysis with our sentiment-risk analysis. This could help us determine whether now is an excellent time to invest or whether we should wait until the sentiment risk is reduced.

4.3.7. Uses and Practical Relevance

In sections 4.3.4 – 4.3.6, we saw how it is possible to combine text-analytics methods with network-analytics methods to automate news processing and develop new ways of extracting data from financial news. These new measures

that we developed can be of practical use to financial industry portfolio managers, such as the fictional person Jenna and to traders, investors, and risk managers. We defined two ways of ranking companies based on news flow which could, for example, be of use to find companies whose stock prices are highly news driven. Building on these methods, we were able to define an algorithm for extracting sentiment risk for companies. These risk values could be of use to risk managers or portfolio managers when determining whether to invest in a new company or whether it would be a good idea to rotate out certain companies from a portfolio of stocks. Portfolio management is largely about minimizing risks and our methods have demonstrated how we automatically can extract measurable sentiment risks. These tools could, for example, be further developed to create new portfolio management strategies. They could also be used by investors and traders to develop investment strategies, and maybe further developed into machine learning based automatic investment strategies.

5. Conclusion

At the start of the thesis, the goal was set to develop new methods using text analytics that could be used within the security and/or the financial industry to automate different types of text processing tasks. Two fictional people, Neil and Jenna, have been used as examples of industry experts to show how the methods developed could be used by similar people in real world scenarios.

In section 1.1, statistics were provided on the differences between leaders and followers in the digital economy. A majority of survey responders said they believed that current business models would be obsolete by 2020. This further showed that the leaders in the digital economy have a strong presence in analytics and that a large number of companies are lagging in that regard, especially when analyzing the use of machine learning. In this thesis, two automation approaches that provide various new insights into these relevant points have been presented. Furthermore, we have seen how both approaches can serve as the foundations of systems solving practical problems.

In the security industry, the battles against threats are many and continuous. One of the problems is keeping up to date with the expanding Internet. No economic way of manually keeping track of all the web sites available exists, and because of this there is a definite need for automatic systems that tag, filter, and keep track of security threats. We have developed new advanced analytics models for automatic classification, which show promise. These models use machine learning, ensemble learning, and state-of-the-art techniques by combining features extracted through sentiment analysis and multi-gram similarity extraction. Using these methods, we showed how we can, under certain conditions, reach performance that is applicable on highly imbalanced data.

In the finance industry, there is an ever-present need for finding new methods, new tools, and new measures that can provide an edge over the competition, because there is always someone else on the other side of asset trades. In this thesis, an approach was presented as to how we can automate news parsing by using advanced-analytics methods that combine text analytics and network analytics techniques. Furthermore, it was shown how we, through text-analytics and network-analytics methods, can develop ways of extracting new information from financial news. We statistically showed that automated parsing of news can contain predictive power. Finally, we showed how the predictive power manifests using our automated news processing as a higher chance of stock price decrease at high sentiment risk. This could be used by industry professionals to

complement relative valuation approaches by adding information telling us when it is beneficial to enter or exit positions.

5.1. Answering the Research Questions

Research Question 1: How can we use analytics to automate text processing tasks?

There are several different ways of automating text processing. Both methods presented in the thesis start with a generalized approach of extracting texts from a data source. The fundamentals for the text extractions are defined in research paper [1]: segmentation, tokenization, stop word removal, key word matching, and TF-IDF weighting. We then continue by converting the extracted texts into numerical measures using pre-processing steps. When using machine learning, we convert text into features and when performing news parsing, we convert texts to networks [2-6]. Lastly, we can apply different mathematical methods to the numerical measures to complete the automation of the processing tasks. In automatic classifications, we can use different machine learning algorithms such as the ones we tested (NB, DT, k-NN, SVM, ANN) or other similar algorithms. The machine learning algorithms can be applied either alone or in combination through ensemble classifications [2], [3], and [6]. In text parsing, we showed that we can also use centrality measures and/or network risk measures to extract information from networks. To achieve this, we use the networks created out of the texts. While we in our experiments used co-occurrence networks, there are also other network alternatives (Aggarwal 2011). Depending on the type of flow we want to measure, we choose either a network measure such as information centrality or eigenvector centrality, or a risk measure such as RiskRank.

Research Question 2: In which ways can sentiment analysis be useful when automating processing tasks?

In automatic classifications of violent and hateful content, we have seen that classification performance can be improved when sentiment features are added to the existing feature sets [2], [3]. However, this is not the case for all types of machine learning algorithms and all categories. The machine learning algorithms that in some cases showed improved F-measures when adding sentiment features were NB, SVM, ANN, and DT. When tested, we found that k-NN in most cases had reduced performance with sentiment features [6]. When using k-NN algorithms, attribute normalization could maybe improve the performance, as

the variance in the sentiment values is higher than the other features that are bound between 0 and 1.

Researchers in finance have shown that sentiment can be used to predict movements of indexes in stock markets (Bollen, Mao, and Zeng 2011). Using network analytics and sentiment, we created new ways of ranking companies. Furthermore, we showed that it is possible to build and compare different types of networks using sentiment polarizations and co-occurrences and we can through network theory find interesting trends in the data [4]. We can also extract sentiment-based risks from news. These sentiment risks represent three different risk components in the networks: individual company risks, direct company risks, indirect company risks [5].

Research Question 3: What can be done to improve unigram classification performance for hate and violence texts?

Sentiment features can improve classifications of NB classifiers when combined with unigram or n-gram features, but it depends on the category [2], [3]. We also found that performance can be improved when using sentiment features and ANN, SVM, and DT, but generally not when using k-NN [6]. When we do ensemble classification of different algorithms, it seems to be on a case-by-case basis whether sentiment features will improve classification performance on balanced data. Furthermore, we showed that multi-gram classifications that combine unigram, one-gram, tri-gram, five-gram, and sentiment features increases F-measure performance in most cases to levels above other approaches tested [6]. However, whether sentiment features will increase performance when using multi-grams varies and needs to be tested for each category.

While ANNs performed best on balanced sets, they seem to overfit, and did not perform as well on the imbalanced testing sets that better represents a real-world setting. While testing on imbalanced data, we found that ensemble voting using SVM/ANN gave the best results for all categories except category 8, where the ensemble using SVM/ANN/DT performed better [6]. We also found that as concerns multi-gram classifications, sentiment features improve performance in some cases, but not the majority of those tested. We were not able to find a pattern to when they add value to multi-gram classification. As the ensemble algorithm of SVM/ANN did not perform exceptionally well on balanced datasets, this shows us that it is useful to test different ensemble combinations on different data skews and not only stick to the best performing for balanced sets.

Research Question 4: Can risks extracted from sentiment networks predict company stock price movements?

We can make comparisons between subsets of quarters and determine whether high risk indicates higher chance of stock price decrease if we limit data points by risk threshold. Comparisons of stock price decrease probability at high risk values were performed using top 50 performing components in both the absolute and the normalized rankings [4, 5]. By taking a subset of data points at risk threshold of 1, which was the highest possible risk using our model, and comparing these quarters to a baseline, which consisted of all analyzed data points, we statistically showed that companies that register a risk value of one are at higher chance of stock price decline for a period of 11-70 days after a quarterly risk measurement [5]. At the same time, we managed to determine that aggregated network effects, using our dataset, on average had a higher chance of stock price decrease than individual sentiment risk, for the period of 11-50 days after a quarterly risk measurement [5]. Additionally, aggregating the three risk components in our comparisons significantly increased the number of quarters with high-risk values, going from 3.5% of the data points to 9.4% at risk threshold of one [5]. Finally, we found that the companies whose sentiment risks we analyzed in our experiment are 13.1 percentage points more likely to have reduced stock price at a delay of 28 days after a quarterly risk measurement of one [5]. However, we are only able to determine the probabilities for directional moves, not the magnitude.

5.2. Limitations

The automatic classification research has been tested to date only on hateful and violent text content. The methods can be applied on any type of text content, but the performance tests of the models have so far been limited to these categories. For the methods to be used on other types of text, a similar type of training set of text data would be needed. Furthermore, the classification research that was performed has been limited to binary classifications. The models would have to be modified to accept more than two labels. The models will not perform well in situations where the text content of categories regularly change, without also updating the models. The classification models will also only perform well in situations where the labelled training instances offer an accurate representation of future data that will be used.

Our approach to automating news processing has been limited to crowd sourced financial news data. For the methods to be applicable on other types of data, we need to be able to extract labels like the sentiment labels, negative and positive found in our dataset. To create networks out of a text dataset, we also need predefined entities that we search for in the text. The risk measures that we extracted are currently performance constrained, and because of that we cannot use more than 50 components in our current models at once even though we had a total of 500 components that we were ranking.

5.3. Future Research

Future work in automatic classifications will be divided into three different areas. As we were able to show that text classifications can be improved by including different types of feature extractions, we will consider comparing and possibly combining our approach with other relevant approaches, such as the approach presented by (Kusner et al. 2015). The second research avenue will be considering feature selection algorithms on the ensembles and multi-gram classifications that were presented. The third avenue will be considering more mathematical approaches, such as evolutionary computing, recurring neural networks, LSTMs, other ensemble algorithms, and feature normalization. The classification research will also be further extended to other types of datasets to see whether the performance is generalizable to other types of categories.

In the financial news research, the research will continue in four directions. First, I am interested in portfolio allocation strategies using the sentiment risk as signals. Second, I will consider creating automated trading strategies using the sentiment risk. Third, I am interested in extending the sentiment measurements from quarterly measurements into rolling averages with different time periods. Fourth, the risk research so far was performed on top performing components in the rankings due to performance constraints. I will develop a risk algorithm that can support more simultaneous components, and research whether the normalized and absolute rankings contain different predictive power.

References

Abe, Shigeo. 2015. "Fuzzy Support Vector Machines for Multilabel Classification". *Pattern Recognition* 48 (6): 2110–17.

Abou-Assaleh, Tony, Nick Cercone, Vlado Keselj, and Ray Sweidan. 2004. "N-Gram-Based Detection of New Malicious Code." In *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, 2:41–42. IEEE.

"Advanced Analytics" 2017. Accessed April 12, 2017. http://www.gartner.com/it-glossary/advanced-analytics.

Aggarwal, Charu C. 2011. "An Introduction to Social Network Data Analytics." In *Social Network Data Analytics*, edited by Charu C. Aggarwal, 1–15. Springer US.

Aggarwal, Charu C., and ChengXiang Zhai. 2012. *Mining Text Data*. Springer Science & Business Media.

Ahuja, Ravindra, Thomas Magnanti, and James Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.

"Alexa Top 500" 2016. Accessed September 30, 2016. http://www.alexa.com/topsites.

Allen, Franklin, and Ana Babus. 2008. "Networks in Finance." SSRN Scholarly Paper ID 1094883. Rochester, NY: Social Science Research Network.

Allen, Franklin, and Douglas Gale. 2009. *Understanding Financial Crises*. Oxford University Press.

Alter, Steven. 1998. *Information Systems*. 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Amrit, Chintan, and Joanne ter Maat. 2016. "Understanding Information Centrality Metric: A Simulation Approach." Accessed November 7. http://www.academia.edu/download/44090983/Understanding_Information_Centrality_Met20160325-29849-1lrje7e.pdf.

"Analytics Informs." 2016. Accessed September 30, 2016. https://www.informs.org/Community/Analytics.

"An Update on Violent Extremism" 2016. *Twitter Blogs*. Accessed October 28, 2016. https://blog.twitter.com/2016/an-update-on-our-efforts-to-combat-violent-extremism.

Androutsopoulos, Ion, John Koutsias, Konstantinos V. Chandrinos, George Paliouras, and Constantine D. Spyropoulos. 2000. "An Evaluation of Naive Bayesian Anti-Spam Filtering." *arXiv Preprint cs/0006013*.

"Apache Spark" 2016. Accessed October 27, 2016. http://spark.apache.org/.

Arlot, Sylvain and Celisse Alain. 2010. "A survey of cross-validation procedures for model selection". *Statistics surveys 4*: 40 - 79.

Asur, S., and B. A. Huberman. 2010. "Predicting the Future with Social Media." In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 1:492–99.

Avriel, Mordecai, Hanna Pri-Zan, Ronit Meiri, and Avi Peretz. 2004. "Opti-Money at Bank Hapoalim: A Model-Based Investment Decision-Support System for Individual Customers." *Interfaces* 34 (1): 39–50.

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. 2016. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." In LREC, vol. 10, pp. 2200-2204. 2010.

Baker, Malcolm, and Jeffrey Wurgler. 2007. "Investor Sentiment in the Stock Market." *The Journal of Economic Perspectives* 21 (2): 129–51.

Banko, Michele, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. "Open Information Extraction from the Web." In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, 2670–2676. Morgan Kaufmann Publishers Inc.

Barber, Brad M., and Terrance Odean. 2008. "All That Glitters: The Effect of Attention and News on the Buying Behavior of Individual and Institutional Investors." *Review of Financial Studies* 21 (2): 785–818.

Bates, David W., Suchi Saria, Lucila Ohno-Machado, Anand Shah, and Gabriel Escobar. 2014. "Big Data in Health Care: Using Analytics to Identify and Manage High-Risk and High-Cost Patients." *Health Affairs* 33 (7): 1123–1131.

Battiston, Stefano, Michelangelo Puliga, Rahul Kaushik, Paolo Tasca, and Guido Caldarelli. 2012. "Debtrank: Too Central to Fail? Financial Networks, the Fed and Systemic Risk." Scientific reports 2 (2012): 541.

Bell, David E., and Howard Raiffa. 1988. *Decision Making: Descriptive, Normative, and Prescriptive Interactions*. Cambridge University Press.

Bespalov, Dmitriy, Bing Bai, Yanjun Qi, and Ali Shokoufandeh. 2011. "Sentiment Classification Based on Supervised Latent N-Gram Analysis." In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 375–382. CIKM '11. New York, NY, USA: ACM.

Bíró, István, Jácint Szabó, and András A. Benczúr. 2008. "Latent Dirichlet Allocation in Web Spam Filtering." In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, 29–32. AIRWeb '08. New York, NY, USA: ACM.

Bisias, Dimitrios, Mark D. Flood, Andrew W. Lo, and Stavros Valavanis. 2012. "A Survey of Systemic Risk Analytics." SSRN Scholarly Paper ID 1983602. Rochester, NY: Social Science Research Network.

Björk, Kaj-Mikael. 2009. "An Analytical Solution to a Fuzzy Economic Order Quantity Problem." *International Journal of Approximate Reasoning* 50 (3): 485–493.

Blei, David M. 2012. "Probabilistic Topic Models." *Communications of the ACM* 55 (4): 77–84.

Blei, David M., and John D. Lafferty. 2006. "Dynamic Topic Models." In *Proceedings of the 23rd International Conference on Machine Learning*, 113–120. ACM.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. "Latent Dirichlet Allocation." *Journal of Machine Learning Research* 3 (Jan): 993–1022.

Bloomfield, Robert J. 2002. “The ‘Incomplete Revelation Hypothesis’ and Financial Reporting.” *Accounting Horizons* 16 (3): 233–43.

Bojarski, M, Del Testa, D, Dworakowski, D, Firner, B, Flepp, B, Goyal, P, Jackel, L.D, Monfort, M, Muller, U, Zhang, J, Zhang, X, Zhao, J, and Zieba, K. "End to end learning for self-driving cars." arXiv preprint arXiv:1604.07316 (2016).

Bollen, Johan, Huina Mao, and Xiaojun Zeng. 2011. “Twitter Mood Predicts the Stock Market.” *Journal of Computational Science* 2 (1): 1–8.

Bonacich, Phillip. 2007. “Some Unique Properties of Eigenvector Centrality.” *Social Networks* 29 (4): 555–64.

Borgatti, Stephen P. 2005. “Centrality and Network Flow.” *Social Networks* 27 (1): 55–71.

Borio, Claudio. 2011. “Implementing a Macroprudential Framework: Blending Boldness and Realism.” *Capitalism and Society* 6 (1).

Boser, Bernhard, Guyon, Isabelle, Vapnik Vladimir. 1992. "A training algorithm for optimal margin classifier." In *Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152). ACM.*

Boss, Michael, Helmut Elsinger, Martin Summer, and Stefan Thurner 4. 2004. “Network Topology of the Interbank Market.” *Quantitative Finance* 4 (6): 677–84.

Brandes, Ulrik, and Daniel Fleischer. 2005. “Centrality Measures Based on Current Flow.” In *STACS 2005*, edited by Volker Diekert and Bruno Durand, 533–44. Lecture Notes in Computer Science 3404. Springer Berlin Heidelberg.

Brown, Gregory W., and Michael T. Cliff. 2004. “Investor Sentiment and the near-Term Stock Market.” *Journal of Empirical Finance* 11 (1): 1–27.

Browne, Michael W. 2000. “Cross-Validation Methods.” *Journal of Mathematical Psychology* 44 (1): 108–132.

Bryman, Alan. 2006. “Integrating Quantitative and Qualitative Research: How Is It Done?” *Qualitative Research* 6 (1): 97–113.

Bullinaria, John A., and Joseph P. Levy. 2012. “Extracting Semantic Representations from Word Co-Occurrence Statistics: Stop-Lists, Stemming, and SVD.” *Behavior Research Methods* 44 (3): 890–907.

Bussiere, Matthieu, and Marcel Fratzscher. 2006. “Towards a New Early Warning System of Financial Crises.” *Journal of International Money and Finance* 25 (6): 953–73.

Cabrales, Antonio, Piero Gottardi, and Fernando Vega-Redondo. 2014. “Risk-Sharing and Contagion in Networks.” SSRN Scholarly Paper ID 2425558. Rochester, NY: Social Science Research Network.

Campbell, John Y., Andrew Wen-Chuan Lo, Archie Craig MacKinlay, and others. 1997. *The Econometrics of Financial Markets*. Vol. 2. princeton University press Princeton, NJ.

Cantador, Iván, Ioannis Konstas, and Joemon M. Jose. 2011. “Categorising Social Tags to Improve Folksonomy-Based Recommendations.” *Web Semantics: Science, Services and Agents on the World Wide Web* 9 (1): 1–15.

Cavnar, William B., John M. Trenkle, and others. 1994. “N-Gram-Based Text Categorization.” *Ann Arbor MI* 48113 (2): 161–175.

Checkland, Peter, and Sue Holwell. 1997. *Information, Systems and Information Systems : Making Sense of the Field*. Wiley.

Checkley, M. S., D. Añón Higón, and H. Alles. 2017. "The Hasty Wisdom of the Mob: How Market Sentiment Predicts Stock Market Behavior." *Expert Systems with Applications* 77: 256–263.

Chen, Stanley F., and Joshua Goodman. 1996. "An Empirical Study of Smoothing Techniques for Language Modeling." In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, 310–318. ACL '96. Stroudsburg, PA, USA: Association for Computational Linguistics.

Collobert, Ronan, and Samy Bengio. 2004. "Links between Perceptrons, MLPs and SVMs." In *Proceedings of the Twenty-First International Conference on Machine Learning*, 23. ACM.

Columbus, Louis. 2015. "81% of Enterprises Are Relying On Analytics To Gain Greater Customer Insights." *Forbes*. Accessed September 30, 2016. http://www.forbes.com/sites/louiscolumbus/2015/07/26/81-of-enterprises-are-relying-on-analytics-to-gain-greater-customer-insights/.

"Competing in 2020" 2017. *Harvard Business Review*. April 25. Accessed May 31, 2017 https://hbr.org/sponsored/2017/04/competing-in-2020-winners-and-losers-in-the-digital-economy.

Cooper, Adam, and others. 2012. "What Is Analytics? Definition and Essential Characteristics." *CETIS Analytics Series* 1 (5): 1–10.

Cortes, Corinna, and Vladimir Vapnik. 1995. "Support-Vector Networks." *Machine Learning* 20 (3): 273–297.

Cowie, Jim, and Wendy Lehnert. 1996. "Information Extraction." *Commun. ACM* 39 (1): 80–91.

Creswell, J. W. and Clark V. L. P. 2007. "Designing and Conducting Mixed Methods Research." Wiley Online Library.

Culotta, Aron, and Jeffrey Sorensen. 2004. "Dependency Tree Kernels for Relation Extraction." In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*. ACL '04. Stroudsburg, PA, USA: Association for Computational Linguistics.

Damashek, Marc. 1995. "Gauging Similarity with N-Grams: Language-Independent Categorization of Text." *Science* 267 (5199): 843.

Damodaran, Aswath. 2012. *Investment Valuation: Tools and Techniques for Determining the Value of Any Asset*. Vol. 666. John Wiley & Sons.

Das, Sanjiv R., and Mike Y. Chen. 2007. "Yahoo! For Amazon: Sentiment Extraction from Small Talk on the Web." *Management Science* 53 (9).

"Data Science Platform." 2016. *RapidMiner*. Accessed September 21, 2016. https://rapidminer.com/.

"Data to Intelligence Program." 2016. *D2I*. Accessed September 30, 2016. http://www.datatointelligence.fi/.

Dave, Kushal, Steve Lawrence, and David M. Pennock. 2003. "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews." In *Proceedings of the 12th International Conference on World Wide Web*, 519–528. WWW '03. New York, NY, USA: ACM.

Davenport, Thomas H. 2006. “Competing on Analytics.” *Harvard Business Review* 84 (1): 98.

Davenport, Thomas H., and Jinho Kim. 2013. *Keeping up with the Quants: Your Guide to Understanding and Using Analytics*. Harvard Business Review Press.

Davis, L. S., S. A. Johns, and J. K. Aggarwal. 1979. “Texture Analysis Using Generalized Co-Occurrence Matrices.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1 (3): 251–59.

Dell’Ariccia, Giovanni, Enrica Detragiache, and Raghuram Rajan. 2008. “The Real Effect of Banking Crises.” *Journal of Financial Intermediation*, Financial Contracting and Financial System Architecture, 17 (1): 89–112.

Dietterich, Thomas G. 2000. “Ensemble Methods in Machine Learning.” In *International Workshop on Multiple Classifier Systems*, 1–15. Springer.

Djuric, Nemanja, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. “Hate Speech Detection with Comment Embeddings.” In *Proceedings of the 24th International Conference on World Wide Web*, 29–30. WWW ’15 Companion. New York, NY, USA: ACM.

Du, Rongbo, Reihaneh Safavi-Naini, and Willy Susilo. 2003. “Web Filtering Using Text Classification.” In Networks, 2003. ICON2003. The 11th IEEE International Conference on, pp. 325-330. IEEE, 2003.

Duca, Marco Lo, and Tuomas A. Peltonen. 2013. “Assessing Systemic Risks and Predicting Systemic Events.” *Journal of Banking & Finance* 37 (7): 2183–2195.

Edmunds, Angela, and Anne Morris. 2000. “The Problem of Information Overload in Business Organisations: A Review of the Literature.” *International Journal of Information Management* 20 (1): 17–28.

Eiben, Agoston E., and James E. Smith. 2003. *Introduction to Evolutionary Computing*. Vol. 53. Springer.

Eisenberg, Larry, and Thomas H. Noe. 2001. “Systemic Risk in Financial Systems.” *Management Science* 47 (2): 236–49.

Erdélyi, Miklós, András Garzó, and András A. Benczúr. 2011. “Web Spam Classification: A Few Features Worth More.” In *Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality*, 27–34. ACM.

Estrada, Ernesto, and Juan A. Rodríguez-Velázquez. 2005. “Subgraph Centrality in Complex Networks.” *Physical Review E* 71 (5): 56103.

Evans, James R., and Carl H. Lindner. 2012. “Business Analytics: The next Frontier for Decision Sciences.” *Decision Line* 43 (2): 4–6.

Fama, Eugene F., Lawrence Fisher, Michael C. Jensen, and Richard Roll. 1969. “The Adjustment of Stock Prices to New Information.” *International Economic Review* 10 (1): 1–21.

Figueiredo, Fábio, Leonardo Rocha, Thierson Couto, Thiago Salles, Marcos André Gonçalves, and Wagner Meira Jr. 2011. “Word Co-Occurrence Features for Text Classification.” *Information Systems* 36 (5): 843–58.

Forman, George. 2003. “An Extensive Empirical Study of Feature Selection Metrics for Text Classification.” *Journal of Machine Learning Research* 3 (Mar): 1289–1305.

Frey, Carl Benedikt, and Michael A. Osborne. 2013. "The Future of Employment: How Susceptible Are Jobs to Computerisation." *Technological Forecasting and Social Change, 114, 254-280.*

Friedkin, Noah E. 1991. "Theoretical Foundations for Centrality Measures." *American Journal of Sociology* 96 (6): 1478–1504.

Fürnkranz, Johannes. 1998. "A Study Using N-Gram Features for Text Categorization." *Austrian Research Institute for Artifical Intelligence* 3 (1998): 1–10.

Gabrel, Viginie, Cécile, Murat, and Aurélie Thiele. "Recent advances in robust optimization: An overview. *European journal of operational research,* 235(3), 471-483.

Gabrilovich, Evgeniy, and Shaul Markovitch. 2006. "Overcoming the Brittleness Bottleneck Using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge." In *AAAI*, 6:1301–1306.

Garratt, Rodney, Lavan Mahadeva, and Katsiaryna Svirydzenka. 2011. "Mapping Systemic Risk in the International Banking Network." SSRN Scholarly Paper ID 1786571. Rochester, NY: Social Science Research Network.

"Gartner's Hype Cycle." 2015. Accessed September 30, 2016. http://www.gartner.com/newsroom/id/3114217.

Gately, Edward. 1995. *Networks for Financial Forecasting*. John Wiley & Sons, Inc.

"Get the Report : Financial Crisis Inquiry Commission." 2016. Accessed September 30, 2016. http://fcic.law.stanford.edu/report.

Glaser, Barney G. 1992. *Emergence vs Forcing: Basics of Grounded Theory Analysis*. Sociology Press.

Go, Alec, Bhayani, Richa, and Huang, Lei. 2016. "Twitter Sentiment Classification Using Distant Supervision." CS224N Project Report, Stanford, 1(12).

Goldberg, Andrew B., and Xiaojin Zhu. 2006. "Seeing Stars When There Aren't Many Stars: Graph-Based Semi-Supervised Learning for Sentiment Categorization." In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA: 45–52.

"Google Safe Browsing." 2017. *Google Developers*. Accessed May 30, 2017. https://developers.google.com/safe-browsing/.

Grossman, Sanford J., and Joseph E. Stiglitz. 1980. "On the Impossibility of Informationally Efficient Markets." *The American Economic Review* 70 (3): 393–408.

Groth, Sven S., and Jan Muntermann. 2011. "An Intraday Market Risk Management Approach Based on Textual Analysis." *Decision Support Systems*, Enterprise Risk and Security Management: Data, Text and Web Mining, 50 (4)

Grover, Varun, and Kalle Lyytinen. 2015. "New State of Play in Information Systems Research: The Push to the Edges." *Mis Quarterly* 39 (2): 271–296.

Guermazi, Radhouane, Mohamed Hammami, and Abdelmajid Ben Hamadou. 2007. "Combining Classifiers for Web Violent Content Detection and

Filtering." In *International Conference on Computational Science*, 773–780. Springer.

Gupta, Madan, Liang Jin, and Noriyasu Homma. 2004. *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*. John Wiley & Sons.

Haas, Peter J., Paul P. Maglio, Patricia G. Selinger, and Wang-chiew Tan. 2011. *Data Is Dead … Without What-If Models*. PVLDB 4, no. 12: 1486-1489.

Haimes, Yacov Y. 2015. *Risk Modeling, Assessment, and Management*. John Wiley & Sons.

Han, Eui-Hong (Sam), and George Karypis. 2000. "Centroid-Based Document Classification: Analysis and Experimental Results." In *Principles of Data Mining and Knowledge Discovery*, edited by Djamel A. Zighed, Jan Komorowski, and Jan Żytkow, 424–31. Lecture Notes in Computer Science 1910. Springer Berlin Heidelberg.

Haykin, Simon, and Neural Network. 2004. "A Comprehensive Foundation." *Neural Networks* 2 (2004): 41.

Hearst, Marti A., Susan T. Dumais, Edgar Osman, John Platt, and Bernhard Scholkopf. 1998. "Support Vector Machines." *IEEE Intelligent Systems and Their Applications* 13 (4): 18–28.

Hellström, Thomas. 1998. "A Random Walk through the Stock Market." PhD diss., Univ. Umeå.

Henry, Elaine, Thomas R. Robinson, John D. Stowe, and others. 2010. *Equity Asset Valuation*. Vol. 27. John Wiley & Sons.

Hevner, Alan, and Samir Chatterjee. 2010. *Design Research in Information Systems: Theory and Practice*. Springer Science & Business Media.

Ho, Kin-Yip, Yanlin Shi, and Zhaoyong Zhang. 2013. "How Does News Sentiment Impact Asset Volatility? Evidence from Long Memory and Regime-Switching Approaches." *The North American Journal of Economics and Finance* 26 (December): 436–56.

Hoeting, Jennifer A., David Madigan, Adrian E. Raftery, and Chris T. Volinsky. 1999. "Bayesian Model Averaging: A Tutorial." *Statistical Science*, 382–401.

Hochreiter, Sepp, and Schmidhuber, Jürgen. 1997. "Long short-term memory". *Neural computation,* 9(8), 1735-1780.

Hornik, Kurt, Stinchcombe, Maxwell, and White, Halbert. "Multilayer feedforward networks are universal approximators". *Neural networks*, 2(5), 359-366.

Houvardas, John, and Efstathios Stamatatos. 2006. "N-Gram Feature Selection for Authorship Identification." In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, 77–86. Springer.

Hsieh, Cho-Jui, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and Sellamanickam Sundararajan. 2008. "A Dual Coordinate Descent Method for Large-Scale Linear SVM." In *Proceedings of the 25th International Conference on Machine Learning*, 408–415. ACM.

Hu, Minqing, and Bing Liu. 2004. "Mining and Summarizing Customer Reviews." In *Proceedings of the Tenth ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, 168–177. KDD '04. New York, NY, USA: ACM.

Huang, Guang-Bin, Qin-Yu Zhu, and Chee-Kheong Siew. 2006. "Extreme learning machine: theory and applications." *Neurocomputing* 70.1, 489 - 501.

Iivari, Juhani. 2007. "A Paradigmatic Analysis of Information Systems as a Design Science." *Scandinavian Journal of Information Systems* 19 (2): 5.

Imran, Muhammad. 2013. "Extracting Information Nuggets from Disaster-Related Messages in Social Media." Proc. of ISCRAM, Baden-Baden, Germany (2013).

Ishibuchi, Hisao, Tadahiko Murata, and I. B Türkşen. 1997. "Single-Objective and Two-Objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems." *Fuzzy Sets and Systems* 89 (2): 135–50.

Jajuga, Krzystof, Andrzej Sokolowski, and Hans-Hermann Bock. 2012. *Classification, Clustering, and Data Analysis: Recent Advances and Applications*. Springer Science & Business Media.

Jarvinen, P. H. 2000. "Research Questions Guiding Selection of an Appropriate Research Method." *ECIS 2000 Proceedings*, 26.

Javidi, Bahram. 2002. *Image Recognition and Classification: Algorithms, Systems, and Applications*. CRC Press.

Jeni, László A., Jeffrey F. Cohn, and Fernando De La Torre. 2013. "Facing Imbalanced data–Recommendations for the Use of Performance Metrics." In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, 245–251. IEEE.

Jensen, Michael C. 1978. "Some Anomalous Evidence Regarding Market Efficiency." SSRN Scholarly Paper ID 244159. Rochester, NY: Social Science Research Network.

Jiang, J. Y., R. J. Liou, and S. J. Lee. 2011. "A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification." *IEEE Transactions on Knowledge and Data Engineering* 23 (3): 335–49.

Jin, Yingzi, Yutaka Matsuo, and Mitsuru Ishizuka. 2009. "Ranking Companies on the Web Using Social Network Mining." In *Web Mining Applications in E-Commerce and E-Services*, edited by I.-Hsien Ting and Hui-Ju Wu, 137–52. Studies in Computational Intelligence 172. Springer Berlin Heidelberg.

Joachims, Thorsten. 1996. "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization." No. CMU-CS-96-118. Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

Jorion, Philippe. 1997. *Value at Risk*. McGraw-Hill, New York.

Kan, Min-Yen, and Hoang Oanh Nguyen Thi. 2005. "Fast Webpage Classification Using URL Features." In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 325–326. CIKM '05. New York, NY, USA: ACM.

Kar, Saroj. 2016. "Gartner: Business Intelligence and Analytics Are Fastest Growing Software Market | CloudTimes." Accessed September 30, 2016. http://cloudtimes.org/2013/02/26/gartner-business-intelligence-and-analytics-are-fastest-growing-software-market/.

Kešelj, Vlado, Fuchun Peng, Nick Cercone, and Calvin Thomas. 2003. "N-Gram-Based Author Profiles for Authorship Attribution." In *Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING*, 3:255–264.

Khreisat, Laila. 2006. "Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study." *DMIN* 2006: 78–82.

Kingdon, Jason. 2004. "AI Fights Money Laundering." *IEEE Intelligent Systems* 19 (3): 87–89.

Klir, George, and Bo Yuan. 1995. *Fuzzy Sets and Fuzzy Logic*. Vol. 4. Prentice hall New Jersey.

Ko, Youngjoong. 2012. "A Study of Term Weighting Schemes Using Class Information for Text Classification." In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1029–1030. SIGIR '12. New York, NY, USA: ACM.

Kohavi, Ron, Neal J. Rothleder, and Evangelos Simoudis. 2002. "Emerging Trends in Business Analytics." *Communications of the ACM* 45 (8): 45–48.

Kohonen, T., S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, and A. Saarela. 2000. "Self Organization of a Massive Document Collection." *IEEE Transactions on Neural Networks* 11 (3): 574–85.

Kohonen, Teuvo, and Panu Somervuo. 1998. "Self-Organizing Maps of Symbol Strings." *Neurocomputing* 21 (1–3): 19–30.

Kouloumpis, Efthymios, Theresa Wilson, and Johanna D. Moore. 2011. "Twitter Sentiment Analysis: The Good the Bad and the Omg!" *Icwsm* 11: 538–541.

Krestel, Ralf, Peter Fankhauser, and Wolfgang Nejdl. 2009. "Latent Dirichlet Allocation for Tag Recommendation." In *Proceedings of the Third ACM Conference on Recommender Systems*, 61–68. ACM.

Kullback, Solomon. 1997. *Information Theory and Statistics*. Courier Corporation.

Kusner, Matt, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. "From Word Embeddings To Document Distances." In *Proceedings of The 32nd International Conference on Machine Learning*, 957–966.

Kutz, Ted, Mark Davis, Robert Creek, Nick Kenaston, Craig Stenstrom, and Margery Connor. 2014. "Optimizing Chevron's Refineries." *Interfaces* 44 (1): 39–54.

Kwok, Irene, and Yuzhou Wang. 2013. "Locate the Hate: Detecting Tweets Against Blacks." In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 1621–1622. AAAI'13. Bellevue, Washington: AAAI Press.

Lacoste-Julien, Simon, Fei Sha, and Michael I. Jordan. 2009. "DiscLDA: Discriminative Learning for Dimensionality Reduction and Classification." In *Advances in Neural Information Processing Systems*, 897–904.

Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. 1998. "An Introduction to Latent Semantic Analysis." *Discourse Processes* 25 (2–3): 259–284.

Lando, David. 2009. *Credit Risk Modeling: Theory and Applications*. Princeton University Press.

Larose, Daniel T. 2005. “K-Nearest Neighbor Algorithm.” *Discovering Knowledge in Data: An Introduction to Data Mining*, 90–106.

Lee, Jay, Hung-An Kao, and Shanhu Yang. 2014. “Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment.” *Procedia CIRP* 16: 3–8.

Lee, P. Y., S. C. Hui, and A. C. M. Fong. 2002. “Neural Networks for Web Content Filtering.” *IEEE Intelligent Systems* 17 (5): 48–57.

Lewis, David D. 1992. “Feature Selection and Feature Extraction for Text Categorization.” In *Proceedings of the Workshop on Speech and Natural Language*, 212–217. HLT ’91. Stroudsburg, PA, USA: Association for Computational Linguistics.

Lewis, David D., and William A. Gale. 1994. “A Sequential Algorithm for Training Text Classifiers.” In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 3–12. SIGIR ’94. New York, NY, USA: Springer-Verlag New York, Inc.

Lewis, Michael. 2004. *Moneyball: The Art of Winning an Unfair Game*. WW Norton & Company.

Leydesdorff, Loet, and Liwen Vaughan. 2006. “Co-Occurrence Matrices and Their Applications in Information Science: Extending ACA to the Web Environment.” *Journal of the American Society for Information Science and Technology* 57 (12): 1616–28.

Lin, Chenghua, and Yulan He. 2009. “Joint Sentiment/Topic Model for Sentiment Analysis.” In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 375–384. CIKM ’09. New York, NY, USA: ACM.

Liu, Bing. 2012. “Sentiment Analysis and Opinion Mining.” *Synthesis Lectures on Human Language Technologies* 5 (1): 1–167.

Liu, Xiaohua, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. “Recognizing Named Entities in Tweets.” In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, 359–367. HLT ’11. Stroudsburg, PA, USA: Association for Computational Linguistics.

Loughran, Tim, and Bill McDonald. 2011. “When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks.” *The Journal of Finance* 66 (1): 35–65.

Luhn, Hans Peter. 1958. “The Automatic Creation of Literature Abstracts.” *IBM Journal of Research and Development* 2 (2): 159–165.

Lund, Kevin, and Curt Burgess. 1996. “Producing High-Dimensional Semantic Spaces from Lexical Co-Occurrence.” *Behavior Research Methods, Instruments, & Computers* 28 (2): 203–8.

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. “Learning Word Vectors for Sentiment Analysis.” In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, 142–150. HLT ’11. Stroudsburg, PA, USA: Association for Computational Linguistics.

Malkiel, B. G., & Fama, E. F. 1970. “Efficient Capital Markets: A Review of Theory and Empirical Work.” The journal of Finance 25, no. 2 (1970): 383-417.
Malkiel, B. G. 2003. “The Efficient Market Hypothesis and Its Critics.” *The Journal of Economic Perspectives* 17 (1): 59–82.
Malkiel, B. G. 2005. “Reflections on the Efficient Market Hypothesis: 30 Years Later.” *Financial Review* 40 (1): 1–9.
March, Salvatore T., and Gerald F. Smith. 1995. “Design and Natural Science Research on Information Technology.” *Decision Support Systems* 15 (4): 251–66.
Markov, Zdravko, and Daniel T. Larose. 2007. *Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage*. John Wiley & Sons.
Marr, David, and Ellen Hildreth. 1980. “Theory of Edge Detection.” *Proceedings of the Royal Society of London B: Biological Sciences* 207 (1167): 187–217.
Mcauliffe, Jon D., and David M. Blei. 2008. “Supervised Topic Models.” In *Advances in Neural Information Processing Systems 20*, edited by J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, 121–128. Curran Associates, Inc.
McCallum, Andrew, Kamal Nigam, and others. 1998. “A Comparison of Event Models for Naive Bayes Text Classification.” In *AAAI-98 Workshop on Learning for Text Categorization*, 752:41–48. Citeseer.
McLeod, Raymond, and George Schell. 2001. “Management Information Systems 8/e.” *Upper Saddle River, NJ: Prentice Hall, 1998.*
McNeil, Alexander J., Rüdiger Frey, and Paul Embrechts. 2015. *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton University Press.
Melville, Prem, Wojciech Gryc, and Richard D. Lawrence. 2009. “Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification.” In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1275–1284. KDD ’09. New York, NY, USA: ACM.
Merkl, Dieter. 1998. “Text Classification with Self-Organizing Maps: Some Lessons Learned.” *Neurocomputing* 21 (1–3): 61–77.
Mezei, József, and Peter Sarlin. 2017. “RiskRank: Measuring Interconnected Risk.” Economic Modelling 2017.
Mitra, Gautam, and Leela Mitra. 2011. *The Handbook of News Analytics in Finance*. John Wiley & Sons.
Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of Machine Learning*. MIT press.
Morchid, Mohamed, Richard Dufour, and Georges Linarès. 2014. “A LDA-Based Topic Classification Approach from Highly Imperfect Automatic Transcriptions.” In *LREC*. Reykjavik, Iceland.
Newman, Mark EJ. 2004. “Analysis of Weighted Networks.” *Physical Review E* 70 (5): 56131.
O’Reilly, Charles A. 1980. “Individuals and Information Overload in Organizations: Is More Necessarily Better?” *Academy of Management Journal* 23 (4): 684–96.

Özgür, Arzucan, Burak Cetin, and Haluk Bingol. 2008. "Co-Occurrence Network of Reuters News." *International Journal of Modern Physics C* 19 (5): 689–702.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. "The PageRank Citation Ranking: Bringing Order to the Web." Stanford InfoLab, 1999.

Pak, Alexander, and Patrick Paroubek. 2010. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." In *LREc*, 10:1320–1326.

Pal, S. K., and S. Mitra. 1992. "Multilayer Perceptron, Fuzzy Sets, and Classification." *IEEE Transactions on Neural Networks* 3 (5): 683–97.

Pang, Bo, and Lillian Lee. 2005. "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales." In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 115–124. ACL '05. Stroudsburg, PA, USA: Association for Computational Linguistics.

Pang Bo, and Lillian Lee. 2008. "Opinion Mining and Sentiment Analysis." *Found. Trends Inf. Retr.* 2 (1–2): 1–135.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 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*, 79–86. EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics.

Papadimitriou, Christos H., Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. 1998. "Latent Semantic Indexing: A Probabilistic Analysis." In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 159–168. ACM.

"Personal Stories." 2016. *Report Mobile and Internet Service Providers Blocking Sites*. Accessed September 30, 2016. https://www.blocked.org.uk/personal-stories.

Piskorski, Jakub, Hristo Tanev, and Pinar Oezden Wennerberg. 2007. "Extracting Violent Events From On-Line News for Ontology Population." In *Business Information Systems*, edited by Witold Abramowicz, 287–300. Lecture Notes in Computer Science 4439. Springer Berlin Heidelberg.

Porter, Michael E., and James E. Heppelmann. 2014. "How Smart, Connected Products Are Transforming Competition." *Harvard Business Review* 92 (11): 64–88.

Powers, David Martin. 2011. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation." Flinders University

Quinlan, J. R. 1986. "Induction of Decision Trees." *Machine Learning* 1 (1): 81–106.

Radev, Dragomir R., Hongyan Jing, Ma\lgorzata Styś, and Daniel Tam. 2004. "Centroid-Based Summarization of Multiple Documents." *Information Processing & Management* 40 (6): 919–938.

Raghupathi, Wullianallur, and Viju Raghupathi. 2014. "Big Data Analytics in Healthcare: Promise and Potential." *Health Information Science and Systems* 2 (1): 1.

Rasmussen, C. E. and Williams C. 2006. “Gaussian Processes for Machine Learning.”

Ratinov, Lev, and Dan Roth. 2009. “Design Challenges and Misconceptions in Named Entity Recognition.” In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, 147–155. CoNLL ’09. Stroudsburg, PA, USA: Association for Computational Linguistics.

Rau, Erik P. 2005. “Combat Science: The Emergence of Operational Research in World War II.” *Endeavour* 29 (4): 156–161.

Rau, L. F. 1991. “Extracting Company Names from Text.” In *, Seventh IEEE Conference on Artificial Intelligence Applications, 1991. Proceedings*, i:29–32.

Reddy, D. Krishna Sandeep, and Arun K. Pujari. 2006. “N-Gram Analysis for Computer Virus Detection.” *Journal in Computer Virology* 2 (3): 231–39.

“Report On Blocked Sites” 2016. *Report Mobile and Internet Service Providers Blocking Sites*. Accessed September 30. https://www.blocked.org.uk/.

Reuter, Peter. 2004. *Chasing Dirty Money: The Fight against Money Laundering*. Peterson Institute.

“Reuters Corpora” 2016. Accessed October 28, 2016. http://trec.nist.gov/data/reuters/reuters.html.

Ritter, Alan, Sam Clark, Mausam, and Oren Etzioni. 2011. “Named Entity Recognition in Tweets: An Experimental Study.” In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1524–1534. EMNLP ’11. Stroudsburg, PA, USA: Association for Computational Linguistics.

Ritter, Thomas. 2000. “A Framework for Analyzing Interconnectedness of Relationships.” *Industrial Marketing Management* 29 (4): 317–26.

Rokach, Lior. 2010. “Ensemble-Based Classifiers.” *Artificial Intelligence Review* 33 (1): 1–39.

Rönnqvist S., and Sarlin P. 2015. “Bank Networks from Text: Interrelations, Centrality and Determinants.” *Quantitative Finance* 15 (10): 1619–35.

Rönnqvist, S., and Sarlin. P. 2014. “From Text to Bank Interrelation Maps.” In *2014 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*, 48–54.

Russell, S. J., and Norvig P. 2002. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, Egnlewood Cliffs, 25, 27.

Russom, Philip, and others. 2011. “Big Data Analytics.” *TDWI Best Practices Report, Fourth Quarter*, 1–35.

Salton, Gerard, and Christopher Buckley. 1988. “Term-Weighting Approaches in Automatic Text Retrieval.” *Information Processing & Management* 24 (5): 513–23.

Schalkoff, Robert J. 1997. *Artificial Neural Networks*. 1st ed. McGraw-Hill Higher Education.

Scheffer, Marten, Jordi Bascompte, William A. Brock, Victor Brovkin, Stephen R. Carpenter, Vasilis Dakos, Hermann Held, Egbert H. van Nes, Max Rietkerk, and George Sugihara. 2009. “Early-Warning Signals for Critical Transitions.” *Nature* 461 (7260): 53–59.

Schmeling, Maik. 2009. "Investor Sentiment and Stock Returns: Some International Evidence." *Journal of Empirical Finance* 16 (3): 394–408.

Schölkopf, Bernhard, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. 2000. "New Support Vector Algorithms." *Neural Computation* 12 (5): 1207–1245.

Schumaker, Robert P., and Hsinchun Chen. 2009. "Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFin Text System." *ACM Transactions on Information Systems (TOIS)* 27 (2): 12.

Schuster, Mike, and Kuldip K. Paliwal. 1997. "Bidirectional Recurrent Neural Networks." *IEEE Transactions on Signal Processing* 45 (11): 2673–2681.

Schwaab, Bernd, Siem Jan Koopman, and Andre Lucas. 2011. "Systemic Risk Diagnostics: Coincident Indicators and Early Warning Signals." SSRN Scholarly Paper ID 1802346. Rochester, NY: Social Science Research Network.

Scott, Sam, and Stan Matwin. 1998. "Text Classification Using WordNet Hypernyms." In Use of WordNet in natural language processing systems: Proceedings of the conference (pp. 38-44).

Scott S. and Matwin S. 1999. "Feature Engineering for Text Classification." In *Proceedings of ICML-99, 16th International Conference on Machine Learning*, 379–388. Morgan Kaufmann Publishers.

Sebastiani, Fabrizio. 2002. "Machine Learning in Automated Text Categorization." *ACM Comput. Surv.* 34 (1): 1–47.

Sein, Maung, Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren. 2011. "Action Design Research." MIS quarterly, 37-56.

Shalev-Shwartz, Shai, Yoram Singer, Nathan Srebro, and Andrew Cotter. 2010. "Pegasos: Primal Estimated Sub-Gradient Solver for SVM." *Mathematical Programming* 127 (1): 3–30.

Shen, Dou, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2006. "Text Classification Improved through Multigram Models." In , 672–81. ACM.

Shiller, Robert J. 2015. *Irrational Exuberance*. Princeton university press.

Shvachko, Konstantin, Hairong Kuang, Sanjay Radia, and Robert Chansler. 2010. "The Hadoop Distributed File System." In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 1–10. IEEE.

Siegel, Eric. 2013. *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die*. John Wiley & Sons.

Sindhwani, V., and P. Melville. 2008. "Document-Word Co-Regularization for Semi-Supervised Sentiment Analysis." In *2008 Eighth IEEE International Conference on Data Mining*, 1025–30.

Socher, R., Perelygin A., Wu J.Y., Chuang J.,Manning C. D., Ng A., and Potts C.. 2016. "Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank." In Proceedings of the conference on empirical methods in natural language processing (EMNLP), vol. 1631, p. 1642. 2013.

Soderland, S.. 1999 "Learning Information Extraction Rules for Semi-Structured and Free Text." *Machine Learning* 34 (1–3): 233–72.

"Spam and Phishing Securelist." 2016. Accessed October 27, 2016. https://securelist.com/analysis/quarterly-spam-reports/74682/spam-and-phishing-in-q1-2016/.

Specht, D. F. 1990. "Probabilistic Neural Networks." *Neural Networks* 3 (1): 109–118.

Stephenson, K., and Zelen M. 1989. "Rethinking Centrality: Methods and Examples." *Social Networks* 11 (1): 1–37.

"Stock Market Insights." 2016. *Seeking Alpha*. Accessed October 17, 2016. http://seekingalpha.com/.

Strauss, Anselm, Juliet Corbin, and others. 1990. *Basics of Qualitative Research*. Vol. 15. Newbury Park, CA: Sage.

Sun, Aixin, Ee-Peng Lim, and Wee-Keong Ng. 2002. "Web Classification Using Support Vector Machine." In *Proceedings of the 4th International Workshop on Web Information and Data Management*, 96–99. ACM.

Tarashev, Nikola A., Claudio EV Borio, and Kostas Tsatsaronis. 2010. "Attributing Systemic Risk to Individual Institutions." BIS Working Paper No. 308.

Tetlock, Paul C. 2007. "Giving Content to Investor Sentiment: The Role of Media in the Stock Market." *The Journal of Finance* 62 (3): 1139–68.

"The 5 Vs of Big Data." 2016. *Watson Health Perspectives*. September 17, 2016. https://www.ibm.com/blogs/watson-health/the-5-vs-of-big-data/.

"The Zettabyte Era." 2016. *Cisco*. Accessed October 27, 2016. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html.

Thelwall, M., Buckley K., and Paltoglou G. 2011. "Sentiment in Twitter Events." *Journal of the American Society for Information Science and Technology* 62 (2): 406–18.

Thelwall, M., Buckley K., Paltoglou G., Cai D., and Kappas A. 2010. "Sentiment Strength Detection in Short Informal Text." *Journal of the American Society for Information Science and Technology* 61 (12): 2544–58.

Timmermann, Allan, and Clive W. J. Granger. 2004. "Efficient Market Hypothesis and Forecasting." *International Journal of Forecasting* 20 (1): 15–27.

"Total Number of Websites." 2016. Accessed October 27. http://www.internetlivestats.com/total-number-of-websites/#sources.

Tóth, Bence, and János Kertész. 2006. "Increasing Market Efficiency: Evolution of Cross-Correlations of Stock Returns." *Physica A: Statistical Mechanics and Its Applications* 360 (2): 505–515.

Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network." In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, 173–180. NAACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics.

Trkman, Peter, Kevin McCormack, Marcos Paulo Valadares De Oliveira, and Marcelo Bronzo Ladeira. 2010. "The Impact of Business Analytics on Supply Chain Performance." *Decision Support Systems* 49 (3): 318–327.

Trstenjak, Bruno, Sasa Mikac, and Dzenana Donko. 2014. "KNN with TF-IDF Based Framework for Text Categorization." *Procedia Engineering*, 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013, 69 (January): 1356–64.

Turney, Peter D. 2002. "Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews." In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 417–424. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics.

Turney, Peter D., and Michael L. Littman. 2003. "Measuring Praise and Criticism: Inference of Semantic Orientation from Association." *ACM Trans. Inf. Syst.* 21 (4): 315–346.

"Twitter Usage Statistics" 2016. Accessed October 28, 2016. http://www.internetlivestats.com/twitter-statistics/.

Vapnik, Vladimir. 2017. *Statistical Learning Theory*. Vol. 1. New York: Wiley.

Veling, Anne, and Peter Van Der Weerd. 1999. "Conceptual Grouping in Word Co-Occurrence Networks." In *Proceedings of the 16th International Joint Conference on Artificial Intelligence-Volume 2*, 694–699. Morgan Kaufmann Publishers Inc.

Von Alan, R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. "Design Science in Information Systems Research." *MIS Quarterly* 28 (1): 75–105.

Wallach, Hanna M. 2006. "Topic Modeling: Beyond Bag-of-Words." In *Proceedings of the 23rd International Conference on Machine Learning*, 977–984. ACM.

Wang, Tai-Yue, and Huei-Min Chiang. 2007. "Fuzzy Support Vector Machine for Multi-Class Text Categorization." *Information Processing & Management* 43 (4): 914–29.

Warner, William, and Julia Hirschberg. 2012. "Detecting Hate Speech on the World Wide Web." In *Proceedings of the Second Workshop on Language in Social Media*, 19–26. LSM '12. Stroudsburg, PA, USA: Association for Computational Linguistics.

Wei, Xing, and W. Bruce Croft. 2006. "LDA-Based Document Models for Ad-Hoc Retrieval." In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 178–185. SIGIR '06. New York, NY, USA: ACM.

Weston, Jason, Sumit Chopra, and Antoine Bordes. 2014. "Memory Networks." *arXiv Preprint arXiv:1410.3916*. https://arxiv.org/abs/1410.3916.

"What Analytics Is" 2016. Accessed September 30, 2016. https://www.informs.org/Sites/Getting-Started-With-Analytics/What-Analytics-Is.

Wilcock, Graham. 2009. "Introduction to Linguistic Annotation and Text Analytics." *Synthesis Lectures on Human Language Technologies* 2 (1): 1–159.

"WorldWideWebSize" 2016. Accessed October 17, 2016. http://www.worldwidewebsize.com/.

Yuan, Yufei, and Huijun Zhuang. 1996. “A Genetic Algorithm for Generating Fuzzy Classification Rules.” *Fuzzy Sets and Systems* 84 (1): 1–19.

Zadeh, Lofti A. 1994. “Fuzzy Logic, Neural Networks, and Soft Computing.” *Communications of the ACM* 37 (3): 77–85.

Zhang, Lei, Bing Liu, Suk Hwan Lim, and Eamonn O’Brien-Strain. 2010. “Extracting and Ranking Product Features in Opinion Documents.” In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 1462–1470. COLING ’10. Stroudsburg, PA, USA: Association for Computational Linguistics.

Zhang, Wen, Taketoshi Yoshida, and Xijin Tang. 2011. “A Comparative Study of TF*IDF, LSI and Multi-Words for Text Classification.” *Expert Systems with Applications* 38 (3): 2758–65.

Zhu, Xiaojin, and Andrew B. Goldberg. 2009. “Introduction to Semi-Supervised Learning.” *Synthesis Lectures on Artificial Intelligence and Machine Learning* 3 (1): 1–130.

Thomas Forss

Automating Text Processing Using Analytics

AutomatingText classifications and financial news parsing

In this thesis, two text processing tasks are automated using analytics. This is done through machine learning and network analytics. The two approaches presented are automatic text classifications and financial news parsing.

I denna avhandling presenteras två tillvägagångssätt som kan användas för att automatisera textprocessering: automatisk textklassificering och automatisk processering av nyhetstexter. Automatiseringen genomförs med maskininlärning och nätverksanalytik.

ISBN 978-952-12-3634-1