WIRTSCHAFTS
UNIVERSITÄT
WIEN VIENNA
UNIVERSITY OF
ECONOMICS
AND BUSINESS

Master Thesis

Dictionaries for Financial Texts

Sergey Yurkevich

Date of Birth: 10.07.1996
Student ID: 01401186

Subject Area: Quantitative Finance

Studienkennzahl: UJ 066 961

Supervisor: Univ.Prof. Dipl.-Ing.Dr.techn. Kurt Hornik

Date of Submission: 01.09.2020

Department of Finance, Accounting and Statistics, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

Contents

List of Figures

List of Tables

"It's written here: 'In the Beginning was the Word!'
Here I stick already! Who can help me? It's absurd,
Impossible, for me to rate the word so highly
I must try to say it differently
If I'm truly inspired by the Spirit. I find
I've written here: 'In the Beginning was the Mind'.
Let me consider that first sentence,
So my pen won't run on in advance!
Is it Mind that works and creates what's ours?
It should say: 'In the beginning was the Power!'
Yet even while I write the words down,
I'm warned: I'm no closer with these I've found.
The Spirit helps me! I have it now, intact.
And firmly write: 'In the Beginning was the Act!"'

Johann Wolfgang von Goethe,
Faust, Part I, Scene III: The Study.
Translation by A. S. Kline.

"Geschrieben steht: Im Anfang war das Wort!
Hier stock' ich schon! Wer hilft mir weiter fort?
Ich kann das Wort so hoch unmöglich schätzen,
Ich muss es anders übersetzen,
Wenn ich vom Geiste recht erleuchtet bin.
Geschrieben steht: Im Anfang war der Sinn.
Bedenke wohl die erste Zeile,
Dass deine Feder sich nicht übereile!
Ist es der Sinn, der alles wirkt und schafft?
Es sollte stehn: Im Anfang war die Kraft!
Doch, auch indem ich dieses niederschreibe,
Schon warnt mich was, dass ich dabei nicht bleibe.
Mir hilft der Geist! Auf einmal seh' ich Rat
Und schreibe getrost: Im Anfang war die Tat!"

Johann Wolfgang von Goethe,
Faust – Der Tragödie erster Teil,
Im Studierzimmer.

Abstract

This thesis deals with dictionaries for textual analysis of financial documents. The first central theme of this work is to analyze the famous and ubiquitous word list for this area created by Loughran and McDonald in 2011. Afterwards we devote our attention to the algorithmic creation of new dictionaries. We compare them with previous lists and evaluate their performance.

Chapter 1

Introduction

"Words can be like X-rays, if you use them properly – they'll go through anything."
A. Huxley, *Brave New World*

This work represents a study of textual analysis by investigation of corresponding dictionaries. As big data supply and computing power increase, the research of text data mining in all areas becomes more and more important. In this thesis we will focus on textual analysis of financial documents, more precisely of 10-K reports. In the related paper [LM11] of 2011 the authors Loughran and McDonald developed new word lists for analyzing such texts. These lists are the main actors of our first natural research question:

> Are the created word lists still up-to-date? Is it possible to reproduce the statistical findings using new data?

A closer study of [LM11] reveals that the dictionary of Loughran and Mcdonald was created "by hand" by examining all candidate words and classifying them as positive or negative. The authors give several reasons why they chose to do so and this leads to the second research question we want to investigate:

> Is it possible to create a new dictionary algorithmically using financial text data such as 10-K reports?

In this context, by "algorithmically" we mean an automatic procedure which takes text data and corresponding labels as input and outputs a dictionary that can be used for exploration of other similar texts.

Finally, we naturally wish to test the resulting word lists. We will not only compare them to the original dictionary but also analyze their performance and significance. In other words, we want to know

> How does the new dictionary relate to the existing word lists? Can we make (better) predictions on firm development by analyzing reports with it?

The thesis is organized as follows: in the first chapter we define the main terms used in the research of textual analysis and introduce concepts we will be working with later. We start with a short introduction into text data mining and provide most important literature sources for further reading.

Chapter 2 is devoted to the summary and analysis of the paper [LM11]. In this section we explain the research design of that work and summarize its most important findings.

In the third and main chapter of the thesis we first demonstrate how the data acquisition and cleansing process was done. Then, in Section 3.3 we discus our results which will help to answer the research questions stated above. We start by reproducing the statistical findings of Loughran and McDonald on new data (2009–2019) and observe similar patterns. However, as we will see, the significance of the new results is evidently lower than in the original work. Afterwards, the summary of two out of several tried approaches for creation of new dictionaries is provided. Both procedures produce word lists which describe the data on which they were developed very well, however they perform miserably on new reports. Moreover, this phenomenon seems to occur for all dictionaries which were created automatically following similar procedures. This justifies the chosen approach in [LM11].

We conclude in the final chapter by summarizing the findings and answering the research questions based on them.

1.1 On Textual Analysis

The growing field of study in text data mining is incredibly important for both: research and industry. In the pioneering work [Hea99] Marti A. Hearst defines it as

> *"the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources."*

Whereby the "written resources" can be any form of text, ranging over publications and articles to tweets and emojis. For practical applications, there has to be just enough data of the specific text resource to perform machine learning techniques on it and then, as a result, discover new facts or figures.

The scientific literature comprises dozens of surveys and introductory texts on textual analysis, of which we shall name just a few. In the recent survey [LM16] Loughran and McDonald not only review existing bibliography on this topic, but also assert to describe the nuances of the method. We use their paper as a reference for most claims about this field of research. The introductory paper [Das14] by Das guarantees an exquisite presentation of text analysis for beginners in the subject. Finally, an extensive survey of older literature is done by Li in [Li10].

From the huge bibliography on the topic it becomes evident that textual analysis is an extremely broad area with many applications. The most prominent directions of this field are readability investigation, measurement of document similarity and sentiment analysis. In this work we will concentrate on the latter, for which the most suitable survey appears to be [KL13] by Kearney and Liu from 2014. Table 3 of this work provides an excellent documentation of most sentiment-related publications which were written before 2013.

Sentiment analysis tries to capture the tone of a (financial) document. Usually this is done by deconstructing the text into a collection of words and simply counting the number of words associated with a particular sentiment. For example, larger proportions of negative words in a document might demonstrate pessimistic tones and vice versa. This technique became known as the *bag-of-words*. At first sight this principle is extremely rough and imprecise, but it turns out to work surprisingly often [LM16; Wei+04]. Moreover, it is clear that in order to apply this procedure it is necessary to have at least one list with negative and/or positive words. In the area of sentiment analysis these kinds of lists are called *dictionaries* and are usually made by researchers for various fields of applications. The choice of a word list is very important and often crucial for the results, consequently in the summarizing Table 3 of [KL13] the authors indicated not only the key findings of previous publications, but also the used dictionary. Moreover, from this table it is evident that most works until 2011 use a dictionary by Harvard, and after this year they mostly change to "L&M". The latter dictionary will play a key role in the present thesis.

Lastly, we mention another critical concept in the bag-of-words approach, namely the *term weighting*. When working with vectors of word counts a non-trivial question is how to normalize them properly. The most simple raw count is clearly tied to document length and therefore heavily biased. On the other hand it allows for the easiest interpretation of coefficients of a linear regression. The usage of proportions solves this problem, but one often wants to account in the word's weight on how unusual the particular term is. Therefore, the most common weighting scheme, the *term frequency-inverse*

document frequency (tf.idf), is more complex:

$$w_{i,j} := \begin{cases} (1 + \log(\mathrm{tf}_{i,j})) \log(N/\mathrm{df}_i) & \text{if } \mathrm{tf}_{i,j} \geq 1, \\ 0 & \text{otherwise,} \end{cases}$$

where N is the total number of documents, df_i is number of documents in which the i-th term appears and $\mathrm{tf}_{i,j}$ the raw count of the i-th word in the j-th document. Jurafsky and Martin write in their book [JM09, p. 771] that term weighting "has an enormous impact on the effectiveness of a retrieval system.". For more details we refer to standard literature like [SB88; ZM98].

1.2 Financial Data

1.2.1 SEC and the EDGAR database

In order to explain the EDGAR database and 10-K forms, we have to introduce the U.S. Securities and Exchange Commission (SEC) first. This institution is an independent agency of the United States federal government, like many other U.S. institutions, for example the CIA, Federal Reserve Board and NASA to name some of the most notable. The SEC is responsible for the control of securities trading in the United States. Its tasks are to check trading for legality and regularity and compliance of stock exchange regulations. To fulfill these tasks, it has been granted extensive legislative, executive and judicial powers, so that it is sometimes referred to as the "fourth power". All companies that want to use the American capital market must undertake the registration process of the SEC. Finally, the SEC ensures that companies publish information which could be important for investors.

EDGAR (or Electronic Data Gathering, Analysis, and Retrieval) is a database operated by the SEC for legally required reports from all registered companies. The institution proudly claims that the "system processes about 3000 filings per day, serves up 3,000 terabytes of data to the public annually, and accommodates 40,000 new filers per year on average". Since the year 2000, data in EDGAR is freely accessible via the internet and can be downloaded in form of `html` files. Since 2004 the database uses so-called CIKs (Central Index Keys), which are public numbers that uniquely identify each participant in the system. For example, the company Alphabet Inc. was assigned 1652044 and Tesla Inc. has the unique number 1318605.

1.2.2 10-K forms

The 10-K form is the name given by the SEC to an annual company report in standardized form. In contrast to the usual annual report, which is often printed in color on glossy paper, the 10-K is simple and strictly uniformed. It contains information on the company's history, its structure, the salaries of the board members, subsidiaries and a standardized annual financial statement. SEC describes the 10-K report by asserting that

> "[The 10-K report] provides audited annual financial statements, a discussion of material risk factors for the company and its business, and a management's discussion and analysis of the company's results of operations for the prior fiscal year."

Companies with assets of more than 10 million USD and more than 2,000 shareholders must file a 10-K report every year. All 10-K reports are freely searchable through the EDGAR database. Table 1.1 explains the structure of such a report. Moreover, the 10-Ks in EDGAR are saved as `txt` files and have a header with practical information about both, the report and the company. At the end of a 10-K file is an appendix with data. Figures 1.1 and 1.2 show the header and the title page out of 173 total pages of the 10-K report of Tesla Inc. from 2019 for the year 2018; the complete file can be downloaded at

www.sec.gov/Archives/edgar/data/1318605/0001564590-19-003165.txt.

Further information about the content of a 10-K report and how to properly read it can be found directly on the SEC website. Other important reports are the quarterly filed 10-Q and the 8-K which is has to be filed if something important and extraordinary happens to the company. However, we will not use these in the present work, so we mention them just for completeness and provide an information source for further investigation: https://www.sec.gov/forms.

```
<SEC-HEADER>0001564590-19-003165.hdr.sgml : 20190219
<ACCEPTANCE-DATETIME>20190219061016
ACCESSION NUMBER:       0001564590-19-003165
CONFORMED SUBMISSION TYPE:  10-K
PUBLIC DOCUMENT COUNT:      162
CONFORMED PERIOD OF REPORT: 20181231
FILED AS OF DATE:       20190219
DATE AS OF CHANGE:      20190219

FILER:

    COMPANY DATA:
        COMPANY CONFORMED NAME:         Tesla, Inc.
        CENTRAL INDEX KEY:          0001318605
        STANDARD INDUSTRIAL CLASSIFICATION: MOTOR VEHICLES & PASSENGER CAR BODIES [3711]
        IRS NUMBER:             912197729
        STATE OF INCORPORATION:         DE
        FISCAL YEAR END:            1231

    FILING VALUES:
        FORM TYPE:      10-K
        SEC ACT:        1934 Act
        SEC FILE NUMBER:    001-34756
        FILM NUMBER:        19613254

    BUSINESS ADDRESS:
        STREET 1:       3500 DEER CREEK RD
        CITY:           PALO ALTO
        STATE:          CA
        ZIP:            94304
        BUSINESS PHONE:     650-681-5000

    MAIL ADDRESS:
        STREET 1:       3500 DEER CREEK RD
        CITY:           PALO ALTO
        STATE:          CA
        ZIP:            94304

    FORMER COMPANY:
        FORMER CONFORMED NAME:  TESLA MOTORS INC
        DATE OF NAME CHANGE:    20050222
</SEC-HEADER>
```

Figure 1.1: Header of a 10-K report of Tesla Inc., 2019.

UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
Washington, D.C. 20549

FORM 10-K

(Mark One)

☒ ANNUAL REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934

For the fiscal year ended December 31, 2018

OR

☐ TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934

For the transition period from _____ to _____

Commission File Number: 001-34756

Tesla, Inc.

(Exact name of registrant as specified in its charter)

Delaware	91-2197729
(State or other jurisdiction of incorporation or organization)	(I.R.S. Employer Identification No.)
3500 Deer Creek Road Palo Alto, California	94304
(Address of principal executive offices)	(Zip Code)

(650) 681-5000

(Registrant's telephone number, including area code)

Securities registered pursuant to Section 12(b) of the Act:

Title of each class	Name of each exchange on which registered
Common Stock, $0.001 par value	The NASDAQ Stock Market LLC

Securities registered pursuant to Section 12(g) of the Act:

None

Figure 1.2: Title page of a 10-K report of Tesla Inc., 2019.

Section	Title	Contains
1	Business	Overview of the company and business environment
1A	Risk Factors	Risk factors, dependence on economic development, risks of business strategy
1B	Unresolved Staff Comments	
2	Properties	Number of factories and offices
3	Legal Proceedings	Pending procedures
4	Mine Safety Disclosures	*only for mining companies*
5	Market for Registrant's Common Equity, Related Stockholder Matters and Issuer Purchases of Equity Securities	Development of the share price
6	Selected Financial Data	Income statement
7	Management's Discussion and Analysis of Financial Condition and Results of Operations	Analysis of the EBIT
7A	Quantitative and Qualitative Disclosures About Market Risk	Numerical risk evaluation
8	Financial Statements and Supplementary Data	Detailed balance sheet
9	Changes in and Disagreements With Accountants on Accounting and Financial Disclosure	
9A	Controls and Procedures	Evaluation of the corporate governance
9B	Other Information	
10	Directors, Executive Officers and Corporate Governance	Management and CVs
11	Executive Compensation	Remuneration of the management
12	Security Ownership of Certain Beneficial Owners and Management and Related Stockholder Matters	Shareholdings of the management
13	Certain Relationships and Related Transactions, and Director Independence	Disclosure of transactions by managers
14	Principal Accounting Fees and Services	Auditing costs
15	Exhibits, Financial Statement Schedules	Next date

Table 1.1: Structure of a 10-K form.

Chapter 2

Procedure of Loughran and Mcdonald

"All textual analysis ultimately stands or falls by the categorization procedures."
T. Loughran and B. Mcdonald, *When is a Liability not a Liability*

This chapter is devoted to the summary of the paper [LM11] which is the main bibliographic source of the present work. "When is a Liability not a Liability" by Tim Loughran and Bill Mcdonald was published in 2011 in The Journal of Finance. The latter is an academic journal of the American Finance Association on financial economic topics. It was established in 1946, publishes six times every year and is considered one of the most respected journals in its field; see [BBS94; OTT05] for more information and analysis. We will explain the research question, its research design and mention most important results of Loughran and Mcdonald's work. Finally, we will mention the natural necessity of the revision of the presented work, which is partly done in the subsequent chapter.

2.1 Synopsis

Loughran and Mcdonald start their work with the observation that a substantial part of finance and accounting research uses textual analysis to examine the sentiment of 10-K reports, press releases, newspaper articles and many more textual sources. They point out that the dictionary that is commonly used for the analysis is the so-called Harvard Psychosociological Dictionary (`H4N`). Yet this is a word list developed for psychology and sociology and given the fact that English words may have many meanings depending on

the context, the authors ask the natural question whether this dictionary translates well into the realm of business.

They manage to provide evidence that the existing list substantially misclassifies words when appraising the sentiment of financial texts. For example, the words *tax, cost, capital, board, liability, foreign* or *vice* are on the Harvard list of negative words, however, there is no reason for them to speak for negative tone in the financial context in general. Therefore, Loughran and Mcdonald create their own dictionary of financial terms: they introduce the lists `Fin-Neg`, `Fin-Pos`, `Fin-Unc` and `Fin-Lit` for negative, positive, uncertain and litigious words in the financial sense, as well as `MW-Strong` and `MW-Weak` for strong and weak modal terms. Then the authors provide statistical evidence that their approach is indeed better than the dictionary used previously. This allows them to "suggest the use of our list to avoid those words in the `H4N` list that might proxy for industry or other unintended effects".

2.2 Data collection and creation of the dictionary

In order to create the word lists, Loughran and Mcdonald first needed to retrieve textual data. They downloaded 10-K reports from the EDGAR for the time period of 1994 to 2008. The firms were required to be listed on NYSE, Amex or NASDAQ with a reported stock price immediately before the filing date of at least 3 USD. Moreover, the companies had to have at least 60 days of trading in the year before and the year after that date and the 10-K document had to include at least 2,000 words. In this way they obtained 50,115 observations consisting of 8,341 unique firms.

The dependent variable of the approach was primary the stock return relative to the 10-K filing date. The file date return was measured as the 4-day holding period excess return over days 0 through 3. In the paper the excess return is defined precisely as "the firm's buy-and-hold stock return minus the CRSP value-weighted buy-and-hold market index return over the 4-day event window". Note that the time period of 4 days is based on Griffin's work in [Gri03, Table II, p. 447]. The regressions included control variables like firm size, book-to-market, share turnover, Fama-French alpha, institutional ownership and dummy variables for NASDAQ listing and industry involvement.

When it comes to term weighting, the authors state that they used one of the most common term weighting schemes with a modification that adjusts

for document length, namely

$$w_{i,j} := \begin{cases} \frac{1+\log(\mathrm{tf}_{i,j})}{1+\log(a_j)} \log(N/\mathrm{df}_i) & \text{if } \mathrm{tf}_{i,j} \geq 1, \\ 0 & \text{otherwise,} \end{cases}$$

where as before N represents the total number of 10-Ks, df_i the number of documents containing at least one occurrence of the i-th word, $\mathrm{tf}_{i,j}$ the raw count of the i-th word in the j-th document, and a_j the average word count in the document. In the empirical results, both the simple proportion of words and the tf.idf weighted measures were examined.

Finally, to create the word lists `Fin-Neg`, `Fin-Pos`, `Fin-Unc` and `Fin-Lit` Loughran and Mcdonald "carefully examine all words occurring in at least 5% of the documents, to consider their most likely usage in financial documents". In other words, they looked individually at each word and decided in which, if any, list to include it. In times of great computer power, established statistical methods and artificial intelligence this approach seems rather inadequate and ineffective. However, the authors defend their strategy, saying that letting the data empirically determine the most impactful words has several drawbacks. This approach would allow to develop only a relatively short list of tonal words. Then managers being aware of this list will systematically avoid those words. Therefore, Loughran and Mcdonald point out the endogeneity problem that would arise and stick to their strategy, creating "a relatively exhaustive list of words that makes avoidance much more challenging".

A known problem in sentiment analysis is the fact that often words make only sense as groups. Even worse, for example, a *no* in front of a word inverts the meaning of it. Loughran and Mcdonald account for this simple negation for `Fin-Pos` words only. In practice, this means if *no*, *not*, *none*, *neither*, *never* or *nobody* occurs within three words preceding a positive word, the word was not counted. Moreover, the authors decided not to stem words to their roots like it is often done in textual analysis. In contrary, when creating the lists they expanded them adding possible inflections. They write that a "problem with stemming is that often a word's meaning changes when common prefixes or suffixes are added" and provide as example the words *odd* and *bitter* and their plural versions *odds* and *bitters*, which have completely different meaning.

The created list of negative words `Fin-Neg` is by far the largest list and incorporates 2,355 words. In contrast, `Fin-Lit` has 904, `Fin-Pos` 354 and `Fin-Unc` only 297 terms.

2.3 Statistical analysis

A large part of the paper [LM11] is devoted to the statistical comparison between the newly created dictionary and Harvard's `H4N`. Since this part is not particularly interesting for our work, we only mention it briefly.

However, arguably the most exciting result of the paper is Figure 2.1. The authors divided the 10-K's into quintiles according to the proportion of negative words. For each category they computed the median of the stock return in the time period of four days after the filing. From the figure it is obvious that the more `Fin-Neg` words appear in a 10-K filing (proportionally) the worse the stock performs in the subsequent time frame. The same cannot be said about words in `H4N`.

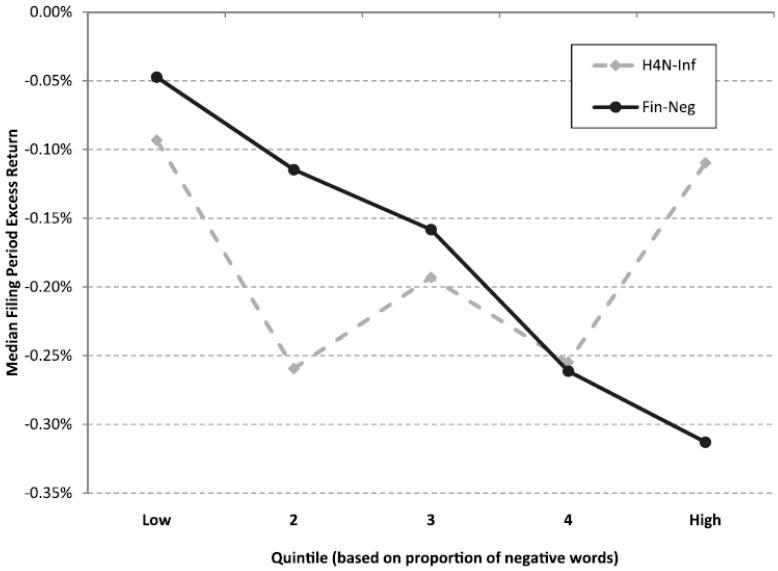

Figure 2.1: Median filing period stock return by quintile for the `H4N` and Financial-Negative `Fin-Neg`.

Moreover, Loughran and Mcdonald performed a linear regression with the stock performance as dependent variable and the proportional count of `Fin-Neg` or `H4N` words as one of the regressor variables. Using the variables visible in Figure 2.2, they could report multiple R^2 values of 2.4% and 2.5% respectively. Admittedly this is low, however, from the regression it can be seen as well that the t-statistic of `Fin-Neg` is -2.64 speaking for the high statistical significance of this variable. In contrary, `H4N` has a t-statistic of just -1.35 which is still negative, but much less significant. It should be noted that the use of tf.idf weights not only improves the R^2 and the statistical significance, but also equalizes it for `Fin-Neg` and `H4N`.

	Proportional Weights	
	(1)	(2)
Word Lists		
H4N-Inf (Harvard-IV-4-Neg with inflections)	−7.422	
	(−1.35)	
Fin-Neg (negative)		−19.538
		(−2.64)
Control Variables		
Log(size)	0.123	0.127
	(2.87)	(2.93)
Log(book-to-market)	0.279	0.280
	(3.35)	(3.45)
Log(share turnover)	−0.284	−0.269
	(−2.46)	(−2.36)
Pre_FFAlpha	−2.500	−3.861
	(−0.06)	(−0.09)
Institutional ownership	0.278	0.261
	(0.93)	(0.86)
NASDAQ dummy	0.073	0.073
	(0.86)	(0.87)
Average R^2	2.44%	2.52%

Figure 2.2: Linear regression with `Fin-Neg` or `H4N`.

2.4 Discussion

The work of Loughran and Mcdonald is a very important milestone for financial textual analysis. The authors designed a dictionary specifically for finance texts and provided statistical evidence that it is more suitable for this kind of content then previous word lists. The data they were using was from the time period 1994–2008, therefore now a natural question arises: "Are the created word lists still up-to-date and how do they perform on new data?" Moreover, as mentioned before, this dictionary was created "by hand" to avoid the endogeneity problem, so another outlook aspect appears: "How do automatically generated word lists compare to `Fin-Neg` and `Fin-Pos`?" We try to answer these questions in the main chapter of the present work.

Chapter 3

Main Part

"My words fly up, my thoughts remain below."
W. Shakespeare, *The Tragedy of Hamlet, Prince of Denmark*

This chapter represents the essence of the present thesis. We start by explaining how the gathering of necessary data was accomplished and demonstrate then the procedure of bringing it into accessible form. We try to be not overly technical but still mention all important steps. Finally, in Section 3.3 we present the main results. All coding was done in the programming language `R` [R C20].

3.1 Acquisition of data

3.1.1 10-K reports

The download procedure of the 10-K reports from the EDGAR database shall now be explained. Note that, of course, there are several ways to acquire the data we need from the database. We follow the advice from https://www.sec.gov/edgar/searchedgar/accessing-edgar-data.htm, where it is communicated that indexes to all public filings since 1994 are available in the following browsable directories:

- /Archives/edgar/daily-index – daily index files through the current year;
- /Archives/edgar/full-index – so-called full indexes, which offer a "bridge" between quarterly and daily indexes.

We are interested in data for the eleven years from 2009 until 2019. For each year and quarter in this time period we download the `index.idx` by visiting

`https://www.sec.gov/Archives/edgar/full-index/i/QTRj/form.idx`,

where `i` stands for the year (looping in the range from 2009 until 2019) and `j` for the quarter (going from 1 to 4). It should be noted that the SEC has a "fair access" policy which is explained detailed in the privacy.htm and is as well summarized on their website:

> "Please use efficient scripting, downloading only what you need and please moderate requests to minimize server load. The SEC reserves the right to limit request rates to preserve fair access for all users."

Therefore, we add a sleep command of 0.5 seconds between each access.

For example, the index file of the first quarter of the year 2019 is available at

https://www.sec.gov/Archives/edgar/full-index/2019/QTR1/form.idx.

Figure 3.1 shows the beginning of the file. Similar to others, it contains nearly

```
Description:           Master Index of EDGAR Dissemination Feed by Form Type
Last Data Received:    March 31, 2019
Comments:              webmaster@sec.gov
Anonymous FTP:         ftp://ftp.sec.gov/edgar/



Form Type   Company Name                                                  CIK         Date Filed  File Name
---------------------------------------------------------------------------------------------------------------------------------------------
1-A         AF 2018 NPL A LLC                                             1756950     2019-02-13  edgar/data/1756950/0001213900-19-002305.txt
1-A         AW Blockchain Mining, Inc.                                    1763626     2019-01-28  edgar/data/1763626/0001477932-19-000272.txt
1-A         Atlanta Hot Wings, Inc.                                       1769642     2019-03-05  edgar/data/1769642/0001615774-19-003654.txt
1-A         BLACK BIRD POTENTIALS INC.                                    1765320     2019-03-21  edgar/data/1765320/0001765320-19-000002.txt
1-A         CB SCIENTIFIC, INC.                                           1022183     2019-03-05  edgar/data/1022183/0001022183-19-000002.txt
1-A         CRL Team 12, Inc.                                             1769999     2019-03-08  edgar/data/1769999/0001615774-19-003791.txt
1-A         Cardone Equity Fund VI, LLC                                   1766343     2019-01-31  edgar/data/1766343/0001477932-19-000312.txt
1-A         Chicago Hogmollies, Inc.                                      1769706     2019-03-05  edgar/data/1769706/0001615774-19-003669.txt
1-A         Circle of Wealth Fund III LLC                                 1762825     2019-02-14  edgar/data/1762825/0001731122-19-000053.txt
1-A         Clikia Corp.                                                  1486452     2019-01-14  edgar/data/1486452/0001486452-19-000004.txt
1-A         Denver Moguls, Inc.                                           1769867     2019-03-07  edgar/data/1769867/0001615774-19-003723.txt
1-A         Dragonize Studio's & Institute, Inc.                          1709247     2019-01-18  edgar/data/1709247/0001709247-19-000001.txt
1-A         Florida Mangos Wild, Inc.                                     1769872     2019-03-07  edgar/data/1769872/0001615774-19-003721.txt
1-A         For The Earth Corp.                                           932265      2019-01-02  edgar/data/932265/0001683168-18-003833.txt
1-A         Fundrise Growth eREIT 2019, LLC                               1768726     2019-03-11  edgar/data/1768726/0001144204-19-013333.txt
1-A         Fundrise Income eREIT 2019, LLC                               1768760     2019-03-11  edgar/data/1768760/0001144204-19-013332.txt
1-A         GULF CHRONIC CARE INC.                                        1762400     2019-03-18  edgar/data/1762400/0001615774-19-004155.txt
1-A         GolfSuites 1. Inc.                                            1765347     2019-01-28  edgar/data/1765347/0001144204-19-002943.txt
1-A         GolfSuites 1. Inc.                                            1765347     2019-01-28  edgar/data/1765347/0001144204-19-002945.txt
1-A         Gravity Storage Inc.                                          1758868     2019-03-25  edgar/data/1758868/0001758868-19-000002.txt
1-A         HCo Cape May LLC                                              1766570     2019-03-04  edgar/data/1766570/0001766570-19-000002.txt
1-A         HempAmericana, Inc.                                           1602929     2019-03-14  edgar/data/1602929/0001683168-19-000661.txt
1-A         Los Angeles Drive, Inc.                                       1770158     2019-03-11  edgar/data/1770158/0001615774-19-003869.txt
1-A         MONOGRAM ORTHOPAEDICS INC                                     1769759     2019-03-13  edgar/data/1769759/0001144204-19-013755.txt
1-A         Money With Meaning Fund, LLC                                  1743113     2019-03-07  edgar/data/1743113/0001213900-19-003728.txt
1-A         NeoVolta Inc.                                                 1748137     2019-01-29  edgar/data/1748137/0001393905-19-000030.txt
```

Figure 3.1: `from.idx` first quarter of 2019.

300,000 lines, of which only 5,000 are links to 10-K reports, so immediately after downloading, we cut out only the important part and save it into a combined `csv` file which we call `firms.csv`. Moreover, since we will be interested in stock prices later, we need the ticker name of each company. We use a file from the SEC website called ticker.txt which has the necessary mapping from CIKs to ticker names. We add those names to our table. With the packages `quantmod` [RU20] and `TTR` [Ulr19] (Technical Trading Rules) we run `stockSymbols()` and obtain from Yahoo! Finance information about the last price of a stock of every company, its market capitalization and the

exchange on which the stock is traded. Now `firms.csv` is in its final form and has 3,827 unique companies and 33,710 rows. Table 3.1[1] shows the first 20 of them.

The columns have self-explanatory names: **CIK** stands for the Central Index Key, **Symbol** is the ticker name, **Name** stands for the full company name, **LastSale** is the price of the last traded stock in Dollars, **MarketCap** is the market capitalization of the company, **EdgarUrl** links to the 10-K report from the corresponding **Year**. Note that in order to access the needed 10-K using the **EdgarUrl**, one needs to append it to the URL `https://www.sec.gov/Archives/edgar/`, so that for example the 10-K report of Nicholas Financial Inc. from 2014 (the first row in `firms.csv`) is availible at

www.sec.gov/Archives/edgar/data/1000045/0001193125-14-237425.txt.

Similar to Loughran and Mcdonald we are only interested in those firms which have a market capitalization of 10Mio USD or more and whose last traded stock price was more than 2 USD. Then 3,204 unique companies remain and the new table consists out of 28,158 rows. Finally, we download every 10-K report using the **EdgarUrl** while creating a folder for each company named according to its CIK.

3.1.2 Stock prices

In order to label the reports, we need stock prices for each company for a reasonably large time period around the filed date of each report. The header of the 10-K form in the EDGAR database contains the date of filing (see Figure 1.1), which we easily extract and define to be the date variable `date`. Then we use the `R` package `quantmod` to run for every `date`

```
try(getSymbols(cikticker, from = date-10,
               to = date+10,warnings = FALSE,
               auto.assign = FALSE),TRUE)
```

Note that the character variable `cikticker` simply denotes the ticker name of the corresponding company. In this way we create for each company a `csv` file `prices.csv` which contains the company's working day stock prices in the time period ± 10 days from each 10-K filing day. For example, as can be seen in Figure 1.1, in 2019 Tesla Inc. filed the 10-K report on the 19th of February, therefore the corresponding time period in `prices.csv` ranges from 11th until 28th of February and is shown in Table 3.2.

[1]The symbol \$ in this and further tables stands for USD and M for millions.

CIK	Symbol	Name	LastSale	MarketCap	Exchange	EdgarUrl	Year
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001193125-09-130987.txt	2009
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001193125-10-138957.txt	2010
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001193125-11-164700.txt	2011
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001193125-12-270341.txt	2012
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001193125-13-259413.txt	2013
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001193125-14-237425.txt	2014
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001193125-15-223218.txt	2015
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001193125-16-620952.txt	2016
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001193125-17-203193.txt	2017
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001193125-18-205637.txt	2018
1000045	NICK	Nicholas Financial, Inc.	6	$47.4M	NASDAQ	data/1000045/0001564590-19-023956.txt	2019
1000209	MFIN	Medallion Financial Corp.	2.85	$70.14M	NASDAQ	data/1000209/0001193125-09-053775.txt	2009
1000209	MFIN	Medallion Financial Corp.	2.85	$70.14M	NASDAQ	data/1000209/0001193125-10-055581.txt	2010
1000209	MFIN	Medallion Financial Corp.	2.85	$70.14M	NASDAQ	data/1000209/0001193125-11-064323.txt	2011
1000209	MFIN	Medallion Financial Corp.	2.85	$70.14M	NASDAQ	data/1000209/0001193125-12-134839.txt	2012
1000209	MFIN	Medallion Financial Corp.	2.85	$70.14M	NASDAQ	data/1000209/0001193125-13-103504.txt	2013
1000209	MFIN	Medallion Financial Corp.	2.85	$70.14M	NASDAQ	data/1000209/0001193125-14-084781.txt	2014
1000209	MFIN	Medallion Financial Corp.	2.85	$70.14M	NASDAQ	data/1000209/0001193125-15-087622.txt	2015
1000209	MFIN	Medallion Financial Corp.	2.85	$70.14M	NASDAQ	data/1000209/0001193125-16-495428.txt	2016
1000209	MFIN	Medallion Financial Corp.	2.85	$70.14M	NASDAQ	data/1000209/0001193125-17-082178.txt	2017

Table 3.1: First 20 rows of `firms.csv`.

	TSLA.Open	TSLA.High	TSLA.Low	TSLA.Close	TSLA.Adjust
11/02/2019	311.600006	318.600006	310.5	312.839996	312.839996
12/02/2019	316.200012	318.190002	309.619995	311.809998	311.809998
13/02/2019	312.350006	312.75	305.570007	308.170013	308.170013
14/02/2019	303.380005	306.769989	301	303.769989	303.769989
15/02/2019	304.5	308	303.899994	307.880005	307.880005
19/02/2019	306.559998	311.540009	305.470001	305.640015	305.640015
20/02/2019	304.410004	306.299988	299	302.559998	302.559998
21/02/2019	301.809998	303.23999	290.5	291.230011	291.230011
22/02/2019	294.48999	296.5	292.100006	294.709991	294.709991
25/02/2019	297.910004	302.899994	297	298.769989	298.769989
26/02/2019	292.220001	302.01001	288.769989	297.859985	297.859985
27/02/2019	301.779999	316.299988	300.549988	314.73999	314.73999
28/02/2019	318.920013	320	310.809998	319.880005	319.880005

Table 3.2: Stock prices TSLA around filed day in 2019.

3.1.3 Dictionaries

All resources of Loughran and McDonald connected to their publications in textual analysis are available on the web page

https://sraf.nd.edu/textual-analysis/resources/.

Note that in the paper [LM11] the authors provide the link

https://afajof.org/supplements.asp

to the internet appendix, however it does not work anymore. It seems that all information was moved to the web site of the University of Notre Dame. We infer that there it is stated that

> "The data compilations provided on this website are for use by individual researchers.",

meaning that we are fully allowed to use the data. We download the file `LoughranMcDonald_SentimentWordLists_2018.xlsx` which contains lists of all categories of words mentioned in [LM11], namely `Negative`, `Positive`, `Uncertainty`, `Litigious`, `Strong Modal`, `Weak Modal` and `Constraining`. For our purposes we are only interested in the `Negative` and `Positive` lists, the first few rows of which are given in Table 3.3. Note that there are 2,355 negative and only 353 positive financial words according to [LM11].

Fin-Neg	Fin-Pos
ABANDON	ABLE
ABANDONED	ABUNDANCE
ABANDONING	ABUNDANT
ABANDONMENT	ACCLAIMED
ABANDONMENTS	ACCOMPLISH
ABANDONS	ACCOMPLISHED
ABDICATED	ACCOMPLISHES
ABDICATES	ACCOMPLISHING
ABDICATING	ACCOMPLISHMENT
ABDICATION	ACCOMPLISHMENTS
ABDICATIONS	ACHIEVE
ABERRANT	ACHIEVED
ABERRATION	ACHIEVEMENT
ABERRATIONAL	ACHIEVEMENTS
⋮ (2,355)	⋮ (353)

Table 3.3: `Negative` and `Positive` words according to [LM11].

3.2 Working with the data

3.2.1 10-K forms

We split each 10-K report into two parts: the header goes into the `year-info.txt` file and the `html` part we save as `year-10K.txt`, where in both cases `year` stands for the year in which the report was filed. We store these files inside the folder associated to the company. Note that since the forms are submitted in `html` format, their file sizes are very high compared to the actual text information they contain. The median size of a 10-K report is 2.5 Megabyte and the average is 3.1 MB. Figure 3.2 reports all file sizes in a boxplot. As we have around 28,000 entries, we have to work with data of more than 85.3 Gigabytes. It becomes evident that it is necessary to clean the 10-K reports and extract only the needed text data deleting the `html` related parts. To do so we use the function `htmlToText()` written by Tony Breyal and published under the CC BY-NC 3.0 License. The code and explanations can be found on the author's blog. After application of the procedure, the size of an average file drops to 0.4 MB, meaning that we reduced 85.3 GB to

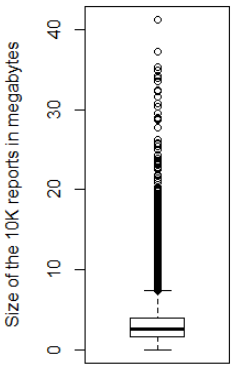

Figure 3.2: Size of the 10-K reports.

approximately 12.2 GB.

Now we are ready for the text utilization process. We will use the `R` package `tm` written by Feinerer and Hornik. The published paper [FHM08] from 2008 explains very well the work style with `tm`, of course we will also follow the official documentation and the introduction vignette by Feinerer from 2019. We wish to create a `TermDocumentMatrix` (TDM in short), however before doing so we need to remove unnecessary white spaces and unwanted characters. Both tasks we will accomplish with the function `tm_map()`, allowing for parallelization, which "can be employed to speed up some of the embarrassingly parallel computations performed in package `tm`" to quote the documentation. However, a problem arises: this function duplicates the input content in RAM and this becomes problematic when working with text data of 12.2 GB on a conventional machine. Therefore, we first split the data into 10 parts according to the year of filing and by putting the 10-Ks from 2009 into the same part as the ones from 2019. In this way the parts become not completely equal in size, however each of them has less than 3 GB in size, which is enough for our purposes. For each of the parts we can first create a TDM and finally combine them. So for each `i` in the range between 0 and 9 we define a pattern and then the directory source:

```
ptrn = paste0("[",i,"]-")
src = DirSource(directory = "./Texts/",
                pattern = ptrn ,encoding = "UTF-8",
                recursive = TRUE)
```

Now we are ready to define the (volatile) corpus.

```
corp = VCorpus(src)
names(corp) = src$filelist
```

For removing white spaces we use the existing function `stripWhitespace()`. In order to remove the unwanted symbols

’ , “, ” , • , – , — , ’ , ” , ® , © , -,

we first define

```
cleaner = content_transformer(function(x,pattern) gsub(pattern,"",x))
```

and then simply invoke

```
symb = "\u0092|\u0093|\u0094|\u0095|\u0096|\u0097|\"|’|®|©|-"
corp = tm_map(tm_map(corp,cleaner,symb),stripWhitespace)
```

Finally, we create the `TermDocumentMatrix` for each part and save them in the list `tdmlist`.

```
tdmlist[[i+1]] = TermDocumentMatrix(corp ,control
                                = list(removePunctuation = TRUE,
                                removeNumbers = TRUE,
                                stopwords = TRUE,
                                stemming = TRUE))
```

In the end, after the looping is done, we combine the elements of `tdmlist` in one large TDM by simply calling

```
tdm_total = c(tdmlist[[1]],tdmlist[[2]],tdmlist[[3]],
            tdmlist[[4]],tdmlist[[5]],tdmlist[[6]],
            tdmlist[[7]],tdmlist[[8]],tdmlist[[9]],tdmlist[[10]])
```

It should be noted that the process just described takes several hours of purely computational time on a regular computer, however it has to be done only once.

We obtain an enormous `TermDocumentMatrix` which has the following attributes:

TermDocumentMatrix (terms: 1785902, documents: 27550)

Non-/sparse entries: 70539916/49131060184

Sparsity: 100%

Maximal term length: 184

Weighting: term frequency (tf)

Obviously, nearly 2Mio unique terms is not realistic and a sparsity level of $0.9985 \approx 100\%$ is not a good indicator as well not to mention the maximal term length. The reason for this is that we started with a very large amount of data (more than 12 GB pure text) and that this data was previously formatted as `html`, therefore some artifacts appeared during conversion. To get rid of these, we allow only those terms that appear in at least 5% of the documents:

```
tdm = removeSparseTerms(tdm_total,0.95)
```

We obtain what we wanted:

TermDocumentMatrix (terms: 5682, documents: 27550)

Non-/sparse entries: 50988393/105550707

Sparsity: 67%

Maximal term length: 26

Weighting: term frequency (tf)

Finally, we create a `TermDocumentMatrix` with a term frequency-inverse document frequency weighting and same terms as `tdm`:

```
tdm_tfidf = weightTfIdf(tdm)
```

3.2.2 Creating labels

In order to create a dictionary and to judge its performance afterwards, we need a file with "labels". That is, a table summarizing how the stock of any company performed after a report was filed. Given the data we acquired in Section 3.1.2, this is not difficult to create. We call the resulting `csv` file `labels.csv` and show the first 20 rows of it in Table 3.4, which fits Table 3.1. In the table the elements in the column **CIK-Year** stand for the CIK of the company and the year the 10-K was filed, separated by a dash. Later we will use this as the unique key for every data point. Clearly, **date** is the filing date and **price** is the (adjusted) price of the stock on this date. Finally, **price_.1** is the price of the stock one day before and **price_i** referrers to the (adjusted) stock price `i` days after the date of filing. From the 27,550 filed dates not all have available stock data. To be more precise, we have to delete 732 ($\approx 2.7\%$) entries because of missing values so that `labels.csv` has 26,818 rows in total.

3.2.3 Fin-Neg and Fin-pos

As is evident from Table 3.3, the word lists provided by Loughran and McDonald are not stemmed. However, in the `TermDocumentMatrix` we created, of course, all words have been first cut down to their root. Therefore, we do the same for the word lists and obtain the much smaller stemmed version. The first several rows are exposed in Table 3.5, which should be compared with Table 3.3. We see that just by stemming we could drop the number of negative words from 2,355 to 882 and similarly the amount of positive words was reduced from 353 to 145. This is a total reduction by 2,475 words and so we have now only 1,027 special words to consider.

CIK-Year	date	price_.1	price	price_1	price_2	price_3	price_4	price_5	price_6
1000045-2009	15/06/2009	4.717504	4.599566	4.591142	4.296298	4.296298	4.313147	4.212057	4.228905
1000045-2010	14/06/2010	7.41322	7.292756	7.403954	7.394688	7.41322	7.524419	7.478085	7.617084
1000045-2011	14/06/2011	11.2959	11.33296	11.27736	11.11056	11.25883	11.11983	11.30516	11.08276
1000045-2012	14/06/2012	12.02148	12.22264	12.34716	12.25138	12.2418	12.27053	12.07896	12.31843
1000045-2013	14/06/2013	14.7894	14.45462	14.75986	14.76971	14.98808	15.12704	14.86897	14.71
1000045-2014	16/06/2014	14.37	14.36	14.37	14.43	14.44	14.46	14.39	14.4
1000045-2015	15/06/2015	12.35	12.42	12.66	12.75	12.96	13.55	13.18	13.22
1000045-2016	14/06/2016	10.49	10.52	10.53	10.55	10.7	10.67	10.54	10.52
1000045-2017	14/06/2017	8.16	7.79	7.74	7.81	7.95	8.24	8.16	8.15
1000045-2018	27/06/2018	8.65	8.66	8.67	9.2	9.47	9.71	9.42	10.19
1000045-2019	28/06/2019	9.15	9.4	9.38	9.04	9.05	9.1	5.49	5.4
1000209-2009	13/03/2009	3.033984	3.185412	3.390922	3.607249	4.039903	3.883066	3.666739	4.701705
1000209-2010	12/03/2010	4.755375	4.749462	4.749462	4.784949	4.743548	4.725803	4.784949	4.997804
1000209-2011	14/03/2011	5.329738	5.278675	5.176548	5.163781	5.163781	5.22761	5.361652	5.406333
1000209-2012	28/03/2012	7.715801	7.667792	7.702085	7.654076	7.750096	7.667792	7.674652	7.660933
1000209-2013	13/03/2013	9.439079	9.547658	9.641758	9.395648	9.410126	9.550026	9.564752	9.454305
1000209-2014	05/03/2014	11.04034	10.91733	11.04034	11.03265	11.12491	11.00958	11.0174	10.93145
1000209-2015	11/03/2015	8.539553	8.456644	8.605879	8.588905	8.487062	8.580418	8.359755	8.139091
1000209-2016	07/03/2016	8.064228	8.064228	7.760438	7.870907	7.90773	7.990582	8.266754	8.358811
1000209-2017	14/03/2017	2.07	2.15	2.09	2.11	2.2	2.35	2.39	2.23

Table 3.4: First 20 rows of `labels.csv`.

Stemmed Fin-Neg	Stemmed Fin-Pos
abandon	abl
abdic	abund
aberr	acclaim
abet	accomplish
abnorm	achiev
abolish	adequ
abrog	advanc
abrupt	advantag
absenc	allianc
absente	assur
abus	attain
accid	attract
accident	beauti
accus	benefici
⋮ (882)	⋮ (145)

Table 3.5: Stemmed `Negative` and `Positive` words according to [LM11].

3.3 Results

We will first analyze the word lists by Loughran and Mcdonald on our 10-K data and then discuss two representative approaches for the creation of new dictionaries. In the first method we try to generate a completely new financial dictionary based on the text data and the labels. After evaluating the weaknesses and strengths of Loughran and Mcdonald's dictionary and the preceding procedure, we will combine them and propose the second approach. Before diving into the study and discussion of these methods, we inspect some simple statistical facts about our data.

3.3.1 Statistics

In Table 3.6 we examine most popular words of the 10-K data which are at the same time part of `Fin-Neg`, `Fin-Pos` or their union in column one, two and three respectively. In each column we first list the words, then their percentage count inside the corresponding group and its cumulative version. Note that there is a difference to the analogous Table in [LM11], because we work with the stemmed version of both, the 10-Ks and the dictionaries.

Table 3.6 reflects the popular law of Zipf, which (heuristically) states that

Fin-Neg			Fin-Pos			Total		
Word	% of Fin-Neg count	% Cumu-lative	Word	% of Fin-Pos count	% Cumu-lative	Word	% of total count	% Cumu-lative
cost	10.22%	10.22%	effect	14.13%	14.13%	cost	7.01%	7.01%
loss	9.35%	19.57%	inform	9.99%	24.12%	loss	6.41%	13.42%
will	6.79%	26.36%	benefit	9.87%	33.99%	will	4.65%	18.07%
content	4.41%	30.77%	gain	5.63%	39.62%	effect	4.44%	22.51%
subject	3.86%	34.63%	posit	4.88%	44.5%	inform	3.14%	25.65%
contract	3.47%	38.1%	profit	3.49%	47.99%	benefit	3.1%	28.75%
limit	3.35%	41.45%	depend	3.16%	51.15%	content	3.02%	31.77%
impair	3.34%	44.79%	improv	3.13%	54.28%	subject	2.65%	34.42%
advers	3.28%	48.07%	assur	2.78%	57.06%	contract	2.38%	36.8%
defer	3.21%	51.28%	except	2.76%	59.82%	limit	2.3%	39.1%
claim	2.17%	53.45%	success	2.49%	62.31%	impair	2.29%	41.39%
liquid	1.51%	54.96%	construct	2.42%	64.73%	advers	2.25%	43.64%
termin	1.34%	56.3%	abl	2.22%	66.95%	defer	2.2%	45.84%
sever	1.18%	57.48%	integr	2.2%	69.15%	gain	1.77%	47.61%
declin	1.13%	58.61%	achiev	1.81%	70.96%	posit	1.53%	49.14%
excess	1.06%	59.67%	advanc	1.72%	72.68%	claim	1.49%	50.63%

Table 3.6: Most common words in Fin-Neg, Fin-Pos and their union.

if elements of a set are ordered by their frequency then often in practice the probability $p(n)$ of the occurrence of the n-th element is inversely proportional to n. The proportionality factor is then given by the N-th harmonic number

$$H_N \coloneqq \sum_{k=1}^{N} \frac{1}{k},$$

where N is the cardinality of the set, so that one often observes $p(n) \approx (nH_N)^{-1}$. In practice, Zipf's law is often observed in connection with textual analysis. We illustrate this law in our case in Figure 3.3, where we plot the relative frequencies from Table 3.6 as circles and the associated density curve $p(x) = (xH_N)^{-1}$, choosing $N = 880$, $N = 145$ and $N = 1,025$ respectively.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Original data	-1.569	-0.021	0.004	0.003	0.029	0.887
Outliers removed	-0.100	-0.017	0.004	0.004	0.026	0.100

Table 3.7: Stock returns with and without outliers.

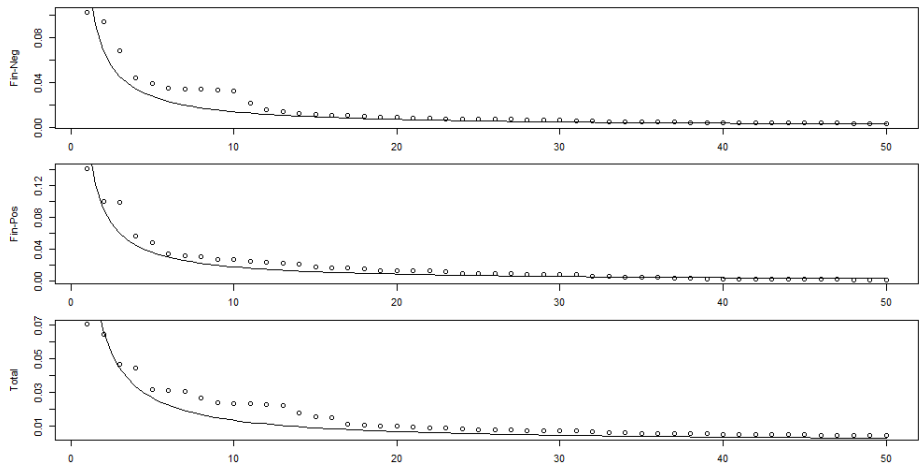

Figure 3.3: Zipf's law for Fin-Neg, Fin-Pos and their union.

Another interesting phenomenon arises when we look at our labels. Following the thoughts of [LM11] we look at the logarithmic returns of the stock price in the time period of one day before the filing to three days after. The boxplot is given in Figure 3.4. It is immediately clear that there are extreme outliers in the data. To be more precise, the first row of Table 3.7 summarizes these log returns. We will get rid of the outliers by looking only at those cases which have a logarithmic return of less than 10% in absolute value in the given time period of 4 days. By doing so, we neglect 2,728 out of 26,818 data points, so roughly 10%. We arrive at 24,090 legitimate observations, whose statistics are summarized in the second row of Table 3.7. Note that both, the median and mean returns are positive, an expected fact, since we are taking into account data in the time period 2009–2019, the time of an economic re-

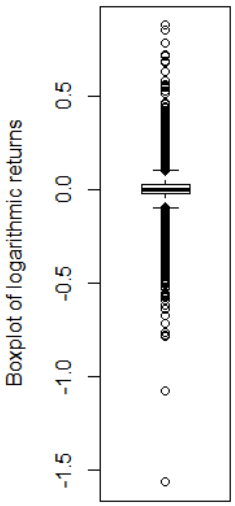

Figure 3.4: Log returns of stocks.

covery and growth.

3.3.2 Neg-Fin and Pos-Fin in 2009 – 2019

We want to analyze how the dictionaries of Loughran and Mcdonald perform on our data. Similarly to [LM11] we will fist make a simple comparison whether more negative (positive) words mean lower (larger) median of logarithmic returns of the corresponding stock price in the time period of one day before filing to three days after (see Figure 2.1). Then we perform and analyze a linear regression model with term frequencies as regressors and the logarithmic return of the stock price in as the dependent variable. Following the procedure of the original paper, we add company data and the Fama-French three-factor model to the regression.

First, let `words` to be the sequence of (stemmed) negative and positive words

```
words_neg = as.character(read.csv("./LM_Words/neg.csv")$x)
words_pos = as.character(read.csv("./LM_Words/pos.csv")$x)
words = c(words_neg,words_pos)
```

Now we can restrict `tdm` to have only terms which also appear in `words_neg`, `words_pos` and `words`

```
LMtdm_neg = tdm[rownames(tdm) %in% words_neg,]
LMtdm_pos = tdm[rownames(tdm) %in% words_pos,]
LMtdm = tdm[rownames(tdm) %in% words,]
```

The new total `TermDocumentMatrix` has only 1025 terms, meaning that two words out of 1027 were never used. A quick comparison shows that these are *happiest* and *happili* which, of course, were classified as positive.

Further, using `labels.csv` we create a vector `y` with the logarithmic stock returns in the time period of our interest. As explained before, we use only those observations in which the logarithmic return of the 4 day period is less than 10% in absolute value. Splitting the 10-Ks in eight quantiles, according to how many words from `Fin-Neg` (respectively `Fin-Pos`) occur and computing for each the median return, we create Figure 3.5. This figure shows that indeed in theory the amount of negative (positive) words in the filed 10-K corresponds to a worse (better) stock performance in the next several days. However, we observe that the fit is far from perfect in the second and very bad in the first case. In fact the straight black line (the best linear fit to the data) explains just 1% of the variance in the top (`Fin-Neg`) and 60%

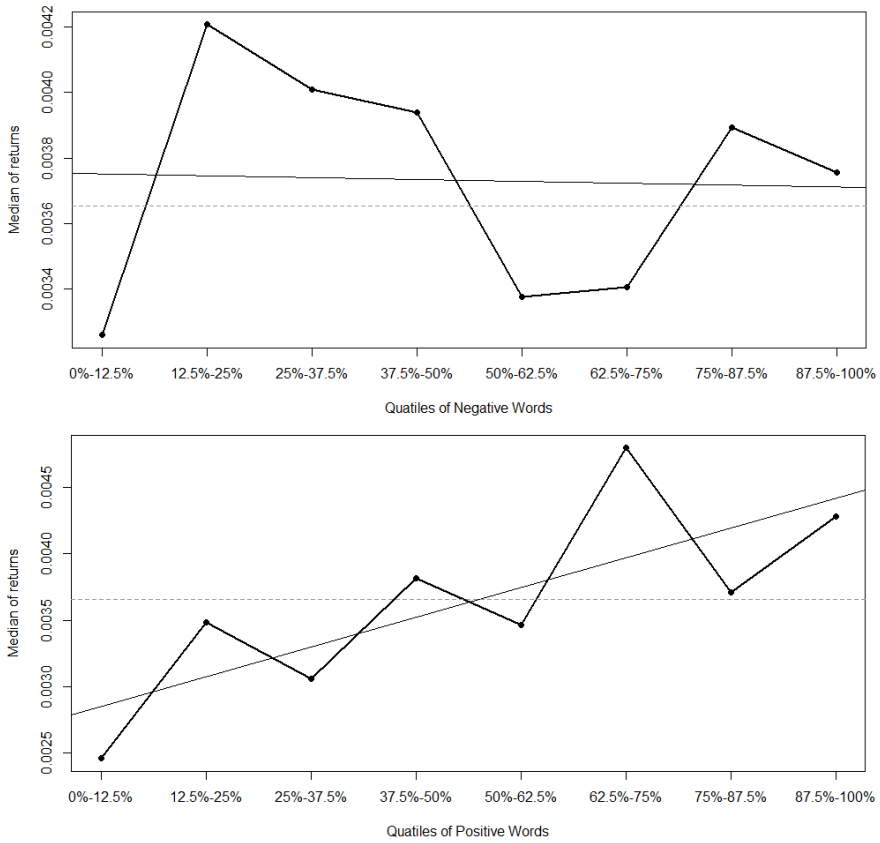

Figure 3.5: `Fin-Neg` (top) and `Fin-Pos` (bottom) quantiles vs. median of stock returns.

in the bottom (`Fin-Pos`) case. Obviously, in the above illustration the fit is highly non-significant (t-statistic of -0.10) but in the case of positive words the correlation is significantly positive with a t-statistic of $+2.92$.

Now we want to analyze the goodness of fit of the prediction of the stock price return using the dictionary. First, we perform a similar analysis as [LM11] in Figure 2.2, then we study the impact of each individual word.

We import the Fama and French data, which was downloaded from the website of Kenneth R. French

```
ff = read.csv("CSVs/FF_Factors_daily.CSV")
ff$X = as.character(as.Date(as.character(ff$X), "%Y%m%d"))
```

The factors are `Mkt.RF` (return of the market portfolio minus risk-free return rate), `SMB` (small minus big market capitalization) and `HML` (high minus low

	Estimate	t-statistic	$\mathbb{P}(>\|t\|)$
(Intercept)	−0.0087	−2.866	0.4%
Mkt.RF	0.0046	17.551	0
SMB	0.0049	11.081	0
HML	0.0034	7.363	0
MarketCap	0.0004	3.514	0
ExchangeNASDAQ	0.0031	1.915	5.5%
ExchangeNYSE	0.0030	1.839	6.6%
Freq_Neg	−0.0343	−1.252	21.0%
Freq_Pos	0.0911	1.524	12.7%

Table 3.8: Linear regression with frequencies of `Fin-Neg` and `Fin-Pos` words.

book-to-market ratio). Moreover, we add the logarithm of the market capitalization and dummy variables for listings on NASDAQ and NYSE. Our regressors of interest are `Freq_Neg` and `Freq_Pos`, which indicate the proportions of negative and respectively positive words in each 10-K. As dependent variable we use `y`. The regression summary is presented in Table 3.8[2]. We see that the estimated coefficient of `Freq_Neg` is negative and the one of `Freq_Pos` is positive, matching our expectation. However, the t-statistics are too small in terms of absolute values to speak of significance. The multiple R^2 of the presented regression is 2.7%, similar to the original paper. Moreover, we observe that Fama and French factors are highly significant and so is the logarithmic market capitalization.

Now we will inspect specifically individual impacts of the terms. Matching the dependent variable we create a `Data Frame` called `x`, in which every row corresponds to a filed 10-K and the columns are `Mkt.RF`, `SMB`, `HML` and all words from the union of `Fin-Neg` and `Fin-Pos`. In the entries of the latter we simply put the amount of occurrences of the specific word in the given 10-K form (raw count). The first 5 rows and 8 columns of `x` are presented in Table 3.9. The corresponding first five elements of the dependent variable are

$$\texttt{y} = (-0.09353, 0.00000, -0.00329, 0.01816, 0.01334, \dots).$$

Now we call

```
model = lm(y ~ .,data = x)
```

to create the linear regression model. We find a surprisingly high multiple R^2 of 7.2%. However, we also see immediately that the adjusted R^2 is just 3.04%, destroying the illusion of a reasonable fit. Moreover, inspecting the

[2]In this and future similar tables, a p-value of 0 means that the observed figure was less than 0.001, so highly significant

Mkt.RF	**SMB**	**HML**	**abl**	**absenc**	**accid**	**achiev**	**adequ**	$\cdots$
-2.38	-0.3	-0.34	8	1	5	1	10	
-0.06	0.66	0.01	7	1	6	1	8	
1.36	1.05	-0.4	6	1	5	1	9	
1.04	0.18	0.37	6	1	6	2	12	
-0.59	-0.24	-0.54	5	1	3	3	11	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Table 3.9: `Data Frame` of the regressor variables.

significance we find that extremely few terms have a p-value of less than 5%, namely just 52 of 1028, which is about 5%. Of course the factors of Fama and French are again highly significant (with p-values of less than 10^{-10} each). It is clear the regression is highly overfitted, however we can still do some interesting performance analysis of the dictionary. First, we are interested in the correlation between all negatively/positively labelled words by Loughran and Mcdonald and the negative/positive coefficient in our regression. It turns out that from the 881 `Fin-Neg` words only 460 have a negative sign in the regression (52.2%). On the other hand from the 143 `Fin-Pos` words only 76 have a positive sign (53.1%). In total roughly 52.3% of the words admit the "correct" sign. The total confusion matrix is given in Table 3.10.

	Negative	**Positive**
Fin-Neg	460	421
Fin-Pos	67	76

Table 3.10: Confusion matrix for `Fin-Neg` $\cup$ `Fin-Pos`.

Assuming that the probability of a negative coefficient sign in our linear regression is roughly 0.5, we can calculate that the standard deviation of the amount of negative signs is $\sqrt{1025}/2 \approx 16$, or 1.6%. It follows that it cannot be claimed that the 52.3% matching signs are significant, since they are within two standard deviations. We observe a similar story when looking at frequent words only, namely just at the top half of used words (see Table 3.6). Table 3.11 is the extension of that table and contains information on each word, whether it is an element of `Fin-Neg` or `Fin-Pos` and which sign the coefficient in the regression has. The words for which the sign fits the classification are marked bold, these are 9 out of 16 (56.3%).

Summarizing the analysis of the dictionaries of Loughran and Mcdonald, we conclude that it has the correct pattern when tested on quantiles (Figure

Word	% of total count	% Cumulative	Fin-Pos or Fin-Neg	Sign in Regression
cost	7.01%	7.01%	Neg	−
loss	6.41%	13.42%	Neg	+
will	4.65%	18.07%	Neg	−
effect	4.44%	22.51%	Pos	+
inform	3.14%	25.65%	Pos	+
benefit	3.10%	28.75%	Pos	+
content	3.02%	31.77%	Neg	+
subject	2.65%	34.42%	Neg	+
contract	2.38%	36.80%	Neg	−
limit	2.30%	39.10%	Neg	+
impair	2.29%	41.39%	Neg	−
advers	2.25%	43.64%	Neg	−
defer	2.20%	45.84%	Neg	+
gain	1.77%	47.61%	Pos	−
posit	1.53%	49.14%	Pos	−
claim	1.49%	50.63%	Neg	−

Table 3.11: `Fin-Neg`/`Fin-Pos` and the sign in the regression.

2.1). However, at the same time, regarding significance, it performs rather poorly. Also the linear model with all raw counts of terms is highly overfitted and insignificant. Moreover, nearly half of the previously classified words have the exact opposite coefficient sign in the regression. Reproducing the linear fit with the proportions of total negative/positive words yields the correct signs for `Freq-Neg` and `Freq-Pos`, though the observed t-statistics again are too small in absolute value to claim significance.

3.3.3 New generated word list

In this section we generate a completely new dictionary without relying on `Fin-Neg` or `Fin-Pos`. We do so by using the LASSO (least absolute shrinkage and selection operator) variable selection method on the linear regression with all words of `tdm`.

To apply LASSO in a reasonable computation time, we need to reduce the number of terms. We create a smaller `TermDocumentMatrix` by reducing sparsity even further and allowing only for terms that appear in at least 15% of the documents rather than 5%:

```
tdm_s = removeSparseTerms(tdm_total,0.85)
```

Heuristically we can argue that if a word appears in less than 15% of the documents then it will not be significant either way. Of course, we perform the linear selection model perform on a `TermDocumentMatrix` which has the frequency-inverse document frequency weighting

```
tdm_tfidf = weightTfIdf(tdm_s)
```

Finally, we need no exclude pathologies by forcing the weighting to be larger than 0.1

```
tdm_tfidf = tdm_tfidf[findFreqTerms(tdm_tfidf,0.1,Inf),]
```

Now the procedure is very similar to the analysis before. We create a `Data Frame` again called `x` with the regressor variables which has now dimension $24,089 \times 2,706$. Note that we do not include yet the data of Fama and French and the matrix elements are given by the frequency-inverse document frequency and not the usual term count. Again, `y` is used for the dependent variable. Then we run

```
cvfit <- cv.glmnet(as.matrix(x), y, nfolds = 10, trace.it=1)
```

using the glmnet package [FHT10]. This operation takes about 5 minutes on a regular computer. The output of `cvfit` is summarized in the top parts

	Lambda	Measure	SE	Nonzero
min	0.0004853	0.001357	1.141e-05	116
1se	0.0013504	0.001360	1.185e-05	0
min	0.0006665	0.0.00136	8.673e-06	9
1se	0.0008810	0.00136	8.667e-06	0

Table 3.12: LASSO `cvfit` for creating a completely new dictionary (top) and the selection of `Fin-Neg` ∪ `Fin-Pos` (bottom).

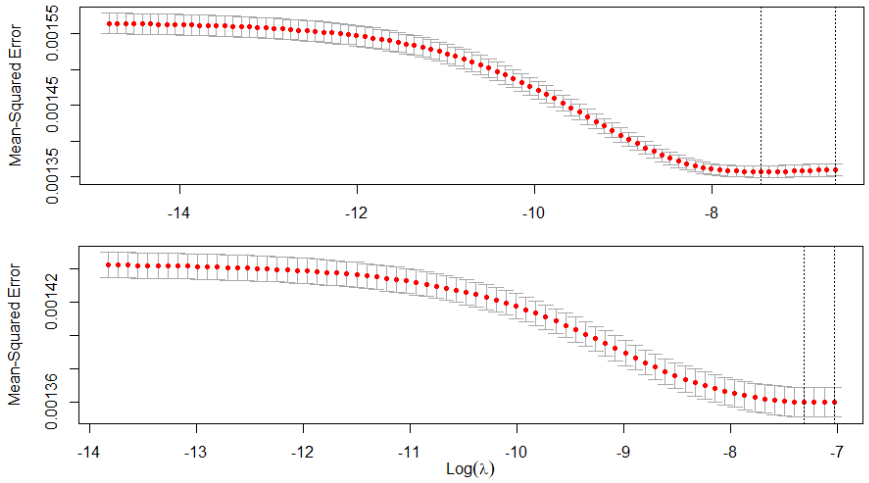

Figure 3.6: LASSO `cvfit` for creating a completely new dictionary (top) and the selection of `Fin-Neg` ∪ `Fin-Pos` (bottom).

of Table 3.12 and Figure 3.6. We use the optimal `lambda.min` as λ and obtain 116 most representative terms. The most common 17 of these, which are responsible for 2/3 of the cumulative frequency, are presented in the first column of Table 3.13. We shall call the list of these words `10K_Dict` and we will perform on it the same analysis as above.

To test the freshly generated word list we start with a new `Data Frame` `x` which now includes the three factors of Fama and French and all terms of `10K_Dict`. Now we use the raw count of terms, so `x` resembles Table 3.9, just with other words. For the dependent variable we again use `y` and run

```
model = lm(y ~ .,data = x)
```

This time we obtain a multiple R^2 value of 4.9%, which is more realistic compared to the analysis before. Also the adjusted R^2 attains a similar magnitude by being equal to 4.4%. We analyze the significance and find that

Word	Frequency %	Cumulative %	Sign in regression	Significance (0.05)
perform	10.94%	10.94%	+	✓
record	10.42%	21.36%	−	✓
accord	5.86%	27.21%	−	✗
adopt	4.52%	31.74%	−	✗
octob	4.04%	35.78%	+	✗
line	3.70%	39.47%	−	✗
juli	3.64%	43.11%	−	✗
sever	3.40%	46.51%	−	✗
august	3.37%	49.88%	−	✗
anticip	2.79%	52.67%	−	✗
store	2.71%	55.38%	+	✓
second	2.68%	58.06%	+	✓
local	2.57%	60.63%	+	✗
mainten	1.96%	62.60%	−	✓
noncontrol	1.84%	64.44%	+	✓
transport	1.73%	66.17%	+	✗
criteria	1.55%	67.72%	+	✓

Table 3.13: Most popular words of `10K_Dict`.

	Estimate	t-statistic	$\mathbb{P}(>\|t\|)$
(Intercept)	0.0024	0.819	41.3%
Mkt.RF	0.0046	17.442	0
SMB	0.0048	10.885	0
HML	0.0032	7.022	0
MarketCap	0.0003	2.816	0.5%
ExchangeNASDAQ	0.0020	1.285	19.9%
ExchangeNYSE	0.0018	1.133	25.7%
Freq_Neg	−1.2269	−14.731	0
Freq_Pos	0.6766	9.056	0

Table 3.14: Linear regression with frequencies of `10K_Dict`.

54 of the 116 terms (46.6%) have a p-value of less than 0.05. Again, most significant are the factors of Fama and French with similar significance levels as before.

While being a much more significant fit than the previous model, it is still clear that it suffers overfitting. Nevertheless, in Table 3.13 we show the sign of the regression coefficient of each of the 17 most popular words in `10K_Dict` and check whether the p-value of this assertion is smaller than 5% in our model. In total we have 70 negative and 46 positive terms when judging by the coefficient sign.

Using the coefficient sign, we can classify words from `10K_Dict` in positive and negative. For each report we calculate the relative frequency of these words and obtain new `Freq_Neg` and `Freq_Pos`, vectors of proportions of negative and positive words. Now we create Table 3.14, which corresponds to the first linear regression in the approach before, see Table 3.8. This time the variables of our interest are highly significant with p-values of less that 10^{-10} each. The coefficients of the other factors are similar. Naturally we observe a better R^2 of 3.8%.

We also create plots analogous to Figures 2.1 and 3.5 in which we match the quantiles of negative and positive `10K_Dict` words with the stock price performance. We obtain Figure 3.7. Compared to the same test of `Fin-Neg` and `Fin-Pos` on our new data this is a much better fit. The linear approximation in the plots explains 64.4% and 93.9% of the variance respectively and the t-statistics are −3.30 and +9.62.

Summarizing this approach, we can say that it produces a dictionary of words which explains our data much better than the lists of Loughran and Mcdonald. However, there is a huge drawback: in contrary to the justified selected of words in [LM11], we cannot ensure the significance and meaning-

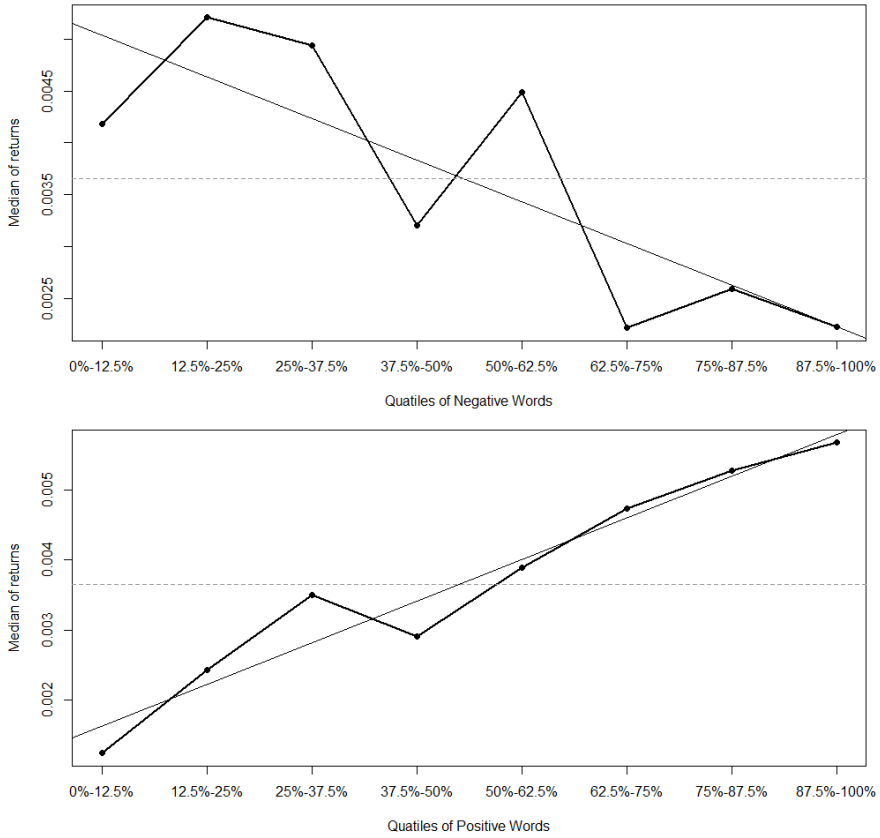

Figure 3.7: `Fin-Neg` (top) and `Fin-Pos` (bottom) quantiles vs. median of stock returns.

fullness of our list. Indeed, only roughly a half of the terms correspond to a p-value of less than 5% in our model. For example, statistically it seems that the occurrence of the words *July*, *August* and *October* in a 10-K has a negative impact on the stock performance, but there is no obvious reason why this should be the case from the practical point of view.

In order to test the overfitting apprehension, we split the data into a `train` and a `test` set (with ratio 80:20) and perform the explained procedure on the prior. The obtained dictionary is then evaluated on the `test` set. Indeed we see that the significance and relevance disappear: the analogue to Figure 3.7 is completely reversed and similarly in the linear regression performed with frequencies of the words like in Table 3.14 gives miserable results.

Based on this conclusion, we propose the second approach in which we try to combine the positive aspects of the previous one and the original word list by Loughran and Mcdonald.

Word	Frequency %	Cumulative %	Sign in regression	Fin-Neg or Fin-Pos	Significance (0.05)
sever	20.14%	20.14%	−	Neg	✗
negat	16.20%	36.34%	+	Neg	✓
restructur	14.40%	50.74%	+	Neg	✗
lead	10.46%	61.20%	+	Pos	✓
concern	7.10%	68.30%	−	Neg	✗
unfavor	4.62%	72.93%	+	Neg	✗
hazard	3.75%	76.67%	−	Neg	✗
problem	3.61%	80.28%	+	Neg	✓
strengthen	2.12%	82.40%	+	Pos	✗
satisfact	1.78%	84.19%	−	Pos	✗

Table 3.15: Most popular words of `LM_10K_Dict`.

3.3.4 Selection from Fin-Neg and Fin-Pos

In this section we present a combination of the two previous methods. Like in the first procedure, we will first reduce the `TermDocumentMatrix` to only those words which appear in `Fin-Neg` or `Fin-Pos` and then perform a further LASSO selection of these. In this way we try to ensure both, a rather good performance and that the terms we are using are indeed in a way significant.

We start with the sequence `words` of the union of `Fin-Neg` and `Fin-Pos` like before. Then we create a `TermDocumentMatrix` called `LM_tdm_tfidf`, containing only terms from `words` with the frequency-inverse document frequency weighting. Like in the previous approach this allows us to create the `Data Frame x`, however now its dimensions are $24,089 \times 1,025$. So compared to that procedure we just start with roughly 37% of the terms, but now we know that they are decisive. We run

```
cvfit <- cv.glmnet(as.matrix(x), y, nfolds = 10, trace.it=1)
```

and obtain the bottom parts of Table 3.12 and Figure 3.6. This time we cannot use the optimal value of $\lambda =$ `lambda.min`, since in this case we would obtain only 9 terms. Rather we set λ to be equal to 0.00045 ($\ln(\lambda) = -7.7$) so that we earn 85 total words. The new dictionary we call `LM_10K_Dict`.

Now that we have created a dictionary we can test its performance like we did before. The usual linear regression yields a model which can explain reasonable 4% of the variance in the data. The adjusted $R^2 = 3.6\%$ does not differ by much. In terms of significance we find that 30 out of

	Estimate	t-statistic	$\mathbb{P}(>\|t\|)$
(Intercept)	−0.0032	-1.146	25.2%
Mkt.RF	0.0046	17.474	0
SMB	0.0049	10.954	0
HML	0.0034	7.347	0
MarketCap	0.0003	2.421	1.6%
ExchangeNASDAQ	0.0027	1.689	9.1%
ExchangeNYSE	0.0027	1.671	9.5%
Freq_Neg	−3.1514	−8.817	0
Freq_Pos	0.9249	5.855	0

Table 3.17: Linear regression with frequencies of `10K_Dict`.

85 terms (35.3%) have a p-value of less than 0.05. As always, the Fama and French three factor model is highly significant. Table 3.15 contains the top 10 most common used words in `LM_10K_Dict`, their relative frequencies, the corresponding sign of the coefficient in the regression, their relation to `Fin-Neg`/`Fin-Pos` and the significance level. Terms in which the sign fits the previous dictionary are marked bold.

We notice immediately that from the ten top common words only five have the "correct" sign when compared with `Fin-Neg` or `Fin-Pos`. For the total list this ratio is not different, namely 54.1%, as can be seen from the complete confusion matrix (Table 3.16). Moreover, this table shows that `LM_10K_Dict` has 48 negative and 37 positive words according to their coefficient sign.

	Negative	Positive
Fin-Neg	42	33
Fin-Pos	6	4

Table 3.16: Confusion matrix for `LM_10K_Dict`.

Now that we have a new classification of words from the original dictionary, we can use it to create the sequences `Freq_Neg` and `Freq_Pos` as before containing relative frequencies of negative and positive words. Using these variables we employ again a linear model, which we summarize in Table 3.17. Similarly, this time the frequencies are highly significant and $R^2 = 3.1\%$.

Finally, as before, we explore the interaction between the amount of negative/positive words and the logarithmic stock performance in Figure 3.8. In this case using linear approximation it is possible to explain 62.8% of the variance in the top case of negative words and 80.7% in the bottom by identifying positive terms. The corresponding t-statistics are -3.19 and $+5.01$.

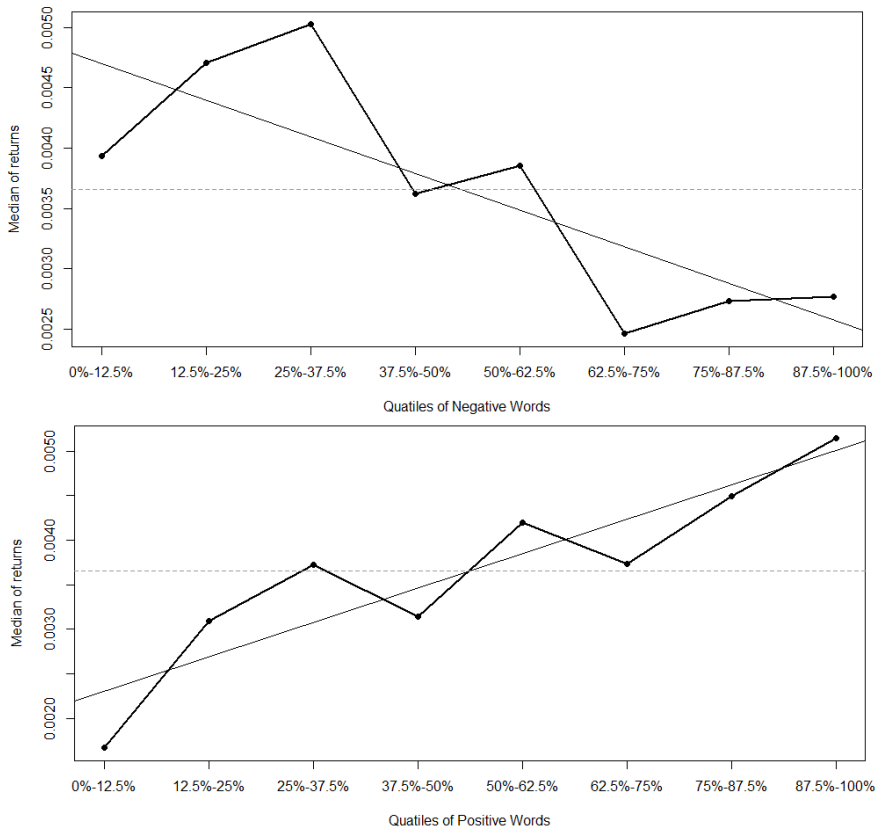

Figure 3.8: `Fin-Neg` (top) and `Fin-Pos` (bottom) quantiles vs. median of stock returns.

This is slightly worse than in the previous approach, however still reasonably good. In these plots it is clearly visible that the reports can be divided into three categories: few, average and high count of negative (respectively positive) words, say for example 0%–35%, 35%–70% and 70%–100% quantiles. Then the stock performance is evidently different in these categories and fits to the count of terms naturally.

While looking more promising, it turns out that this approach suffers overfitting as well. A closer cross-validated inspection using the `train/test` sets reveals the same problem as in the approach before. The correlation in Figure 3.8 loses its meaning and the significance in the linear regressions disappears.

We conclude that the third combined approach tried to incorporate the positive aspects of the previous two: the terms are by definition decisive

since they were selected by Loughran and Mcdonald and the performance on the `train` set is as good as in the case of the completely new dictionary. However, this performance is delusive, since a more detailed analysis unveils strong overfitting problems.

Chapter 4

Conclusion

"We've all acquired some education
A bit of this a bit of that."
A. Pushkin, *Eugene Onegin*

This short final chapter is devoted to the summary and conclusion of the presented work. Based on the results we will answer the three research questions stated in the introduction:

1. Are the word lists by Loughran and Mcdonald still up-to-date? Is it possible to reproduce the statistical findings using new data?

2. Can we create a new dictionary algorithmically using financial text data such as 10-K reports?

3. Is it possible to make better predictions on firm development by analyzing reports with the new word list?

In Section 3.3.2 we analyzed the dictionaries `Fin-Neg` and `Fin-Pos` for the time period 2009–2019. Performing a similar procedure like Loughran and Mcdonald in [LM11] we could find similar patterns as in the original paper. To be more precise, the linear regression in Table 3.8 shows a negative coefficient of `Freq_Neg` and a positive one for `Freq_Pos`. Similarly, Figure 3.5 suggests that the amount of positive words correlates with a better stock performance and (at least in theory) the quantile increase of negative terms indicates a worse logarithmic stock return. Yet, we notice that the observed t-statistics do not speak for the same level of significance as in the original work. Moreover, the linear regression summarized in Table 3.11 shows that many words classified as negative have actually a positive impact on the stock performance and vice versa.

To create a new dictionary using financial data we tried several approaches, two of the most promising and representative are summarized in the Sections 3.3.3 and 3.3.4. These strategies result in word lists containing 116 and 85 unique terms respectively. As it is evident from Figures 3.7 and 3.8 as well as Tables 3.14 and 3.17 the new dictionaries describe the given text data well and better than the previous word list. However, in all cases in which new dictionaries were selected by an algorithmic procedure based on given 10-K data, we found that major overfitting problems occurred. Hence, addressing the second research question, we provided approaches which lead to automatically generated word lists based on text data. However, judging by the performance on a `test` set we are forced to conclude that the answer to the third research question is negative, in the sense that the resulting dictionaries perform much worse, at least if methods like ours are used. This circumstance is another justification for the arduous procedure of Loughran and Mcdonald in [LM11] for the creation of a meaningful dictionary.

In conclusion it is important to note that the given answers to the second and third research questions depend on the setting we were working with. The possibility of an automatically generated and well performing word list is not ruled out, just the presented methods based on linear regressions did not prove to be fruitful. The continuation of this work with different approaches is an essential task for textual analysis not only for the financial sector, but in general.

Bibliography

[SB88] Gerard Salton and Christopher Buckley. "Term-weighting approaches in automatic text retrieval". In: *Information Processing & Management* 24.5 (1988), pp. 513–523. ISSN: 0306-4573. DOI: https://doi.org/10.1016/0306-4573(88)90021-0. URL: http://www.sciencedirect.com/science/article/pii/0306457388900210.

[BBS94] Kenneth A. Borokhovich, Robert J. Bricker, and Betty J. Simkins. "Journal Communication and Influence in Financial Research". In: *The Journal of Finance* 49.2 (1994), pp. 713–725. DOI: 10.1111/j.1540-6261.1994.tb05159.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1994.tb05159.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1994.tb05159.x.

[ZM98] Justin Zobel and Alistair Moffat. "Exploring the Similarity Space". In: *SIGIR FORUM* 32 (1998), pp. 18–34.

[Hea99] Marti A. Hearst. "Untangling Text Data Mining". In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*. ACL '99. College Park, Maryland: Association for Computational Linguistics, 1999, pp. 3–10. ISBN: 1-55860-609-3. DOI: 10.3115/1034678.1034679. URL: https://doi.org/10.3115/1034678.1034679.

[Gri03] Paul Griffin. "Got information? Investor response to Form 10-K and Form 10-Q EDGAR filings". In: *Review of Accounting Studies* 8 (Dec. 2003), pp. 433–460. DOI: 10.1023/A:1027351630866.

[Wei+04] Sholom Weiss et al. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Jan. 2004. DOI: 10.1007/978-0-387-34555-0.

[OTT05] Elisabeth Oltheten, Vasilis Theoharakis, and Nickolaos G. Travlos. "Faculty Perceptions and Readership Patterns of Finance Journals: A Global View". In: *Journal of Financial and Quantitative Analysis* 40.1 (2005), pp. 223–239. DOI: 10.1017/S0022109000001800.

[FHM08] Ingo Feinerer, Kurt Hornik, and David Meyer. "Text Mining Infrastructure in R". In: *Journal of Statistical Software, Articles* 25.5 (2008), pp. 1–54. ISSN: 1548-7660. DOI: 10.18637/jss.v025.i05. URL: https://www.jstatsoft.org/v025/i05.

[JM09] D. Jurafsky and J.H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, 2009. ISBN: 9780131873216. URL: https://books.google.at/books?id=fZmj5UNK8AQC.

[FHT10] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1 (2010), pp. 1–22. URL: http://www.jstatsoft.org/v33/i01/.

[Li10] Feng Li. "Textual Analysis of Corporate Disclosures: A Survey of the Literature". In: *Journal of Accounting Literature* 29 (Feb. 2010), pp. 143–165.

[LM11] Tim Loughran and Bill Mcdonald. "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks". In: *The Journal of Finance* 66.1 (2011), pp. 35–65. DOI: 10.1111/j.1540-6261.2010.01625.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.2010.01625.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2010.01625.x.

[KL13] Colm Kearney and Sha Liu. "Textual Sentiment in Finance: A Survey of Methods and Models". In: *International Review of Financial Analysis* 33 (Apr. 2013), pp. 171–185. DOI: 10.2139/ssrn.2213801.

[Das14] Sanjiv Das. "Text and Context: Language Analytics in Finance". In: *Foundations and Trends® in Finance* 8 (Nov. 2014), pp. 145–261. DOI: 10.1561/0500000045.

[LM16] Tim Loughran and Bill Mcdonald. "Textual Analysis in Accounting and Finance: A Survey". In: *Journal of Accounting Research* 54.4 (2016), pp. 1187–1230. DOI: 10.1111/1475-679X.12123. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1475-679X.12123. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1475-679X.12123.

[Ulr19] Joshua Ulrich. *TTR: Technical Trading Rules*. R package version 0.23-6. 2019. URL: https://CRAN.R-project.org/package=TTR.

[R C20] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2020. URL: https://www.R-project.org/.

[RU20] Jeffrey A. Ryan and Joshua M. Ulrich. *quantmod: Quantitative Financial Modelling Framework*. R package version 0.4-16. 2020. URL: https://CRAN.R-project.org/package=quantmod.