PhD thesis submitted for the joint degree of:

Doctor of Finance
Institute of Financial Analysis, Université de Neuchâtel
and
Doctor of Business Economics
Faculty of Social Sciences & Solvay Business School, Vrije Universiteit Brussel

---

Sentiment and Econometrics: Toward A Unified Framework of Textual Sentiment Analysis for Economic and Financial Applications

Doctoral Dissertation of:
Samuel Borms

Advisors:
Prof. David Ardia & **Prof. Kris Boudt**
Jury members:
Prof. Tim Kroencke (President)
Prof. Marie Lambert
Prof. Steven Vanduffel
Prof. Tim Verdonck

July 2020

UNIVERSITÉ DE NEUCHÂTEL

IMPRIMATUR POUR LA THÈSE

(cotutelle avec Vrije Universiteit Brussel)

Sentiment and Econometrics: Toward A Unified Framework of Textual Sentiment Analysis for Economic and Financial Applications

Samuel BORMS

UNIVERSITÉ DE NEUCHÂTEL

FACULTÉ DES SCIENCES ÉCONOMIQUES

La Faculté des sciences économiques,
sur le rapport des membres du jury

Prof. David Ardia (co-directeur de thèse, Université de Neuchâtel)
Prof. Kris Boudt (co-directeur de thèse, Vrije Universiteit Brussel)
Prof. Tim Kröncke (président du jury, Université de Neuchâtel)
Prof. Steven Vanduffel (Vrije Universiteit Brussel)
Prof. Marie Lambert (HEC Liège)
Prof. Tim Verdonk (Universiteit Antwerpen)

Autorise l'impression de la présente thèse.

Neuchâtel, le 22 septembre 2020

Annik Dubied

La doyenne
Annik Dubied

■ Téléphone : +41 32 718 15 00 ■ E-mail : secretariat.seco@unine.ch ■ www.unine.ch

Preface

Doing research is "building on the shoulders of Giants." The idiom presents itself as if Giants are stable and fixed and without holes, as if they are whole. They are not. I did not build on top of a Giant. I took some of its existing parts, linked them together, and made as such a (tiny) compartment of the Giant I delved into stronger. Still of similar size, but more compact and more versatile, and I am happy with it. I hope the Giant is too.

I did not do this alone, however.

Thank you Kris for taking me on board without really knowing me. Thank you Dave for taking me on board knowing me even less. I could not have wished for a better supervisory duo than you two. Many thanks for presenting me with an abundance of interesting opportunities and challenges, I will not forget. We had a productive run.

A sincere thank you to the members of the jury, Prof. Tim Kroencke, Prof. Marie Lambert, Prof. Steven Vanduffel and Prof. Tim Verdonck, for taking the time to evaluate my thesis. Sorry to Tim that we never actually had that coffee you proposed me a few times in Neuchâtel.

A special thanks to my closest research friends Nabil, Andres, Nico, Jeroen and Brecht (we liked to call ourselves the Academic Wanderers... or Wannabes). Together we shared many nice chats and moments, especially during (and after) conferences.

To my kind and smart colleagues in Neuchâtel (with an honorable mention earned by Keven and Hongliang), and the whole IAF department, I wish you all the very best in your professional and personal pursuits.

I also send a (most likely never arriving) word of appreciation to all modern musicians who have fed me over the past years: John Frusciante, Josh Klinghoffer, Paul Banks, Alex Turner, Nicolas Jaar, Adam Granduciel, Thom Yorke, Spinvis, and many more (yes, Marc Rebillet too). True creativity is yours.

Finally, thank you to my family, a union above all, and other friends. A very warm thank you to my girlfriend Roberta. Keep being the amazing person you are. You make my world and the entire world a better place. Você é a melhor. True love is (y)ours.

Every end is a new beginning. Sometimes the *via negativa* approach of taking something away is the best way to sail forward. Let this be the start of a new inspiring adventure.

(I suggest the reader to fill in their favorite quote here.)

SAMUEL BORMS
Somewhere between Brussels & Neuchâtel
July 2020

ABSTRACT

This doctoral thesis has three key contributions. First, it overviews the current methodologies and empirical applications at the intersection of sentiment analysis and econometrics, while also proposing new avenues for improvement. Doing so, it formalizes this emerging research field, terming it "sentometrics", which is a portmanteau of sentiment and econometrics. Second, this thesis describes an open-source implementation of a complete workflow to go from qualitative textual data to quantitative (time series) sentiment variables and extract (econometric) insights from those. These two contributions fill in the previously existing lack of a unified approach to using alternative sentiment data to obtain insights for economic and financial analysis. Third, the workflow is adopted and adjusted for two applications. The first one is in sustainable asset management. A large corpus of Belgian and Dutch news is transformed into daily signals that track news reporting relevant to Environmental, Social and corporate Governance (ESG) topics. The textual signals prove useful to restrict an investment portfolio universe to a smaller subset of more sustainable yet at least equally performing companies. The second application deals with the construction and analysis of monthly and daily news-based Economic Policy Uncertainty (EPU) indices for Belgium.

Keywords: Aggregation, Econometrics, EPU, ESG, Penalized Regression, Qualitative Data, R, Sentiment Analysis, Sentometrics, **sentometrics**, Sustainable Investment, Textual Analysis, Time Series

RÉSUMÉ

Cette thèse de doctorat apporte trois contributions clés. Premièrement, il donne un aperçu des méthodologies actuelles et des applications empiriques à l'intersection de l'analyse de sentiment et de l'économétrie, tout en proposant de nouvelles pistes d'amélioration. Ce faisant, il formalise ce domaine de recherche émergent, le qualifiant "sentométrie" (*sentometrics*), qui est un portemanteau de sentiment et d'économétrie (*econometrics*). Deuxièmement, il décrit un logiciel open source d'un flux de travail complet pour passer de données textuelles qualitatives à des variables (séries temporelles) quantitatives de sentiment et extraire des informations (économétriques) de celles-ci. La première et la deuxième contribution comblent le manque existant auparavant d'une approche unifiée de l'utilisation de données de sentiment alternatives pour obtenir des informations pour une analyse économique et financière. Troisièmement, le flux de travail est adopté et ajusté pour deux applications. La première se situe dans la gestion d'actifs durables. Un grand corpus de nouvelles presse belges et néerlandaises est transformé en signaux quotidiens qui suivent les articles aux sujets d'Environnement (*Environmental*), de Social et de Gouvernance d'entreprise (*corporate Governance*), en abrégé ESG. Les signaux textuels se montrent utiles pour sélectionner d'un univers de portefeuille un plus petit sous-ensemble d'entreprises plus durables mais (au moins) aussi performantes. La deuxième application s'occupe de la construction et de l'analyse des indicateurs mensuels et quotidiens sur la base des articles de presse belges à propos de l'incertitude de la politique économique (*economic policy uncertainty*; ou EPU).

Mots clés: Agrégation, Analyse de Sentiment, Analyse Textuelle, Données Qualitatives, Econométrie, EPU, ESG, Investissement Durable, Régression Pénalisée, R, Sentométrie, **sentometrics**, Séries Temporelles

SAMENVATTING

Deze doctoraatsthesis omvat drie hoofdcontributies. Ten eerste vat het de meest belangrijke methodologieën en empirische toepassingen op de kruising tussen sentiment analyse en econometrie samen, en stelt het ook mogelijkheden tot uitbreiding voor. Dit groeiende onderzoeksgebied wordt zo geformaliseerd, ernaar refererende als "sentometrie" (*sentometrics*), een porte-manteau van sentiment en econometrie (*econometrics*). Ten tweede beschrijft deze thesis een open-source implementatie van een volledige *worfklow* om van kwalitatieve tekstuele data te gaan naar kwantitatieve (tijdreeks) sentiment variabelen en hier (econometrische) inzichten uit af te leiden. Deze twee contributies vullen de leemte in rond een uniforme aanpak in het gebruiken van alternatieve sentiment data om inzichten voor economische en financiële analyse te onttrekken. Ten derde wordt de *workflow* specifiek aan het werk gezet in twee toepassingen. De eerste betreft duurzaam investeringsbeheer. Een grote corpus aan Belgisch en Nederlands nieuws is omgezet tot dagelijkse signalen die nieuws relevant voor de thema's milieu (*Environmental*), maatschappij (*Social*) en bestuur (*corporate Governance*), afgekort ESG, opvolgen. De tekstuele signalen bewijzen nuttig te zijn bij het selecteren van een meer duurzame doch (minstens) even goed presterende selectie aan bedrijven vanuit een groter investeringsuniversum. De tweede toepassing behelst de constructie en analyse van maandelijkse en dagelijkse indicatoren omtrent economische beleidsonzekerheid (*economic policy uncertainty*; oftewel EPU) op basis van Belgische persartikels.

Kernwoorden: Aggregatie, Duurzaam Investeren, Econometrie, EPU, ESG, Kwalitatieve Data, Penaliserende Regressie, R, Sentiment Analyse, Sentometrie, **sentometrics**, Tekstuele Analyse, Tijdreeksen

Acknowledgements

Chapter 2

This chapter circulates as a separate paper titled "Econometrics Meets Sentiment: An Overview of Methodology and Applications", co-authored with Andres Algaba, David Ardia, Keven Bluteau, and Kris Boudt. The paper is published in The Journal of Economic Surveys.

We thank the Associate Editors (Les Oxley and Stelios Bekiros) and two anonymous Referees, seminar participants at Ca' Foscari University of Venice, the European Commission JRC Ispra "Big Data and Forecasting Workshop" (Ispra, 2019), Ghent University, HEC Montréal, the International Conference on Computational and Financial Econometrics (London, 2019), Skema Business School, University of Delaware, and Vrije Universiteit Brussel for their useful comments. We are also grateful to Francesco Audrino, Leopoldo Catania, Maxime De Bruyn, William Doehler, Tim Kroencke, Nitish Sinha, and Leif Anders Thorsrud for stimulating discussions and feedback. This project benefited from financial support from Innoviris (`https://innoviris.brussels`), IVADO (`https://ivado.ca`), swissuniversities (`https://www.swissuniversities.ch`), and the Swiss National Science Foundation (`http://www.snf.ch`, grants #179281 and #191730).

Chapter 3

This chapter circulates as a separate paper titled "The R Package **sentometrics** to Compute, Aggregate and Predict with Textual Sentiment", co-authored with David Ardia, Keven Bluteau, and Kris Boudt. The paper is forthcoming in The Journal of Statistical Software.

We thank the Associate Editors (Toby Hocking and Torsten Hothorn) and three anonymous Referees, Andres Algaba (*package contributor*), Nabil Bouamara, Peter Carl, Leopoldo Catania, Thomas Chuffart, Dries Cornilly, Serge Darolles, William Doehler, Arnaud Dufays, Matteo Ghilotti, Kurt Hornik, Siem Jan Koopman, Tim Kroencke, Julie Marquis, Linda Mhalla, Brian Peterson, Laura Rossetti, Tobias Setz, Majeed Siman, Stefan Theussl, Wouter Torsin, Jeroen Van Pelt (*package contributor*), Marieke Vantomme, Tim Verdonck, and participants at the CFE (London, 2017), eRum (Budapest, 2018), R/Finance (Chicago, 2018), SwissText (Winterthur, 2018), SoFiE (Brussels, 2018), "Data Science in Finance with R" (Vienna, 2018),

"New Challenges for Central Bank Communication" (Brussels, 2018), (EC)^2 (Roma, 2018), and useR! (Toulouse, 2019) conferences for helpful comments. We acknowledge Google Summer of Code 2017 and 2019 (`https://summerofcode.withgoogle.com`), Innoviris (`https://innoviris.brussels`), IVADO (`https://ivado.ca`), swissuniversities (`https://www.swissuniversities.ch`), and the Swiss National Science Foundation (`http://www.snf.ch`, grants #179281 and #191730) for their financial support.

Chapter 4

This chapter circulates as a separate paper titled "Semi-Supervised Text Mining for Monitoring the News About the ESG Performance of Companies", co-authored with Kris Boudt, Frederiek Van Holle, and Joeri Willems. The paper will appear as a standalone chapter in the Springer book "Data Science for Economics and Finance: Methodologies and Applications."

We are grateful to the book editors (Sergio Consoli, Diego Reforgiato Recupero and Michaela Saisana) and three anonymous Referees, seminar participants at the CFE (London, 2019) conference, Andres Algaba, David Ardia, Keven Bluteau, Maxime De Bruyn, Tim Kroencke, Marie Lambert, Steven Vanduffel, Jeroen Van Pelt, Tim Verdonck, and the Degroof Petercam Asset Management division for stimulating discussions and helpful feedback. Many thanks to Sustainalytics (`https://www.sustainalytics.com`) for providing us with their historical dataset, and to Belga for giving us access to their news archive. This project received financial support from Innoviris, swissuniversities (`https://www.swissuniversities.ch`), and the Swiss National Science Foundation (`http://www.snf.ch`, grant #179281).

Chapter 5

This chapter circulates as a separate research note titled "The Economic Policy Uncertainty Index for Flanders, Wallonia and Belgium", co-authored with Andres Algaba, Kris Boudt, and Jeroen Van Pelt. The note is published in Bank- en Financiewezen digitaal edition 2020/6 (see `https://www.financialforum.be/nl/articles/economic-policy-uncertainty-index-flanders-wallonia-and-belgium`), and the monthly time series output is available on the `https://www.policyuncertainty.com` website.

We thank David Ardia, Scott Baker, Nick Bloom, Keven Bluteau, Steven Davis, Jolan Stevens, and Jan Wijffels for helpful feedback. We are grateful to the Belga News Agency for providing the news data. This project received financial support from the National Bank of Belgium, the Swiss National Science Foundation (`http://www.snf.ch`, grant #17928), and Innoviris.

Contents

List of Figures

List of Tables

Glossary

This glossary provides easy-to-understand definitions for a selection of concepts mentioned throughout the thesis, to help non-specialists get started. The definitions are mostly my own. Not all concepts dealt with in this work are covered.

Confusion matrix A confusion matrix summarizes the performance of a classification model. It categorizes the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Two important metrics that can be computed from the confusion matrix are precision (the proportion of classified positives that is correct, i.e. $TP/(TP+FP)$), and recall (the proportion of the actual positives detected, i.e. $TP/(TP+FN)$).

Corpus A corpus in linguistics jargon designates the collection of textual data units (e.g. documents) to be analyzed. It can be generalized to indicate the collection of data units from textual, audio, or visual data.

EPU EPU is an acronym popularized by Baker *et al.* (2016) that stands for Economic Policy Uncertainty.

ESG ESG is an acronym for Environmental (e.g. climate change), Social (e.g. employee relations) and Governance (e.g. audits). It designates the three main areas that socially responsible investors evaluate.

Features A feature is a broad term to represent any type of metadata attached to the original textual, audio, or visual data as stored in a corpus. Examples are source, expresser, entity, location, topic, and so on. This definition is slightly different but in line with how features are used in a machine learning context, where they refer to the set of explanatory variables. In video and audio data, (low-level) features are compact, mathematical representations of the physical properties of the data (Wang *et al.*, 2003).

Generative modeling The generative modeling paradigm views a statistical model as the explicit source behind generating the observations, based on a particular step-by-step process.

For example, all words in a news article can assumed to be generated sequentially as draws from an estimated statistical distribution of a large vocabulary of words.

Lexicon A lexicon is a list of tokens (e.g. words, a sequence of words, a facial expression, or a sound) with, for each token, an associated score that represents its average sentiment. Also interchangeably called a sentiment lexicon, a sentiment word list, or a sentiment dictionary.

Natural language processing Natural language processing (NLP) is the vast subfield within artificial intelligence occupied with the understanding, interpretation, and manipulation of human language. It integrates computer science, computational linguistics, and machine learning.

Polarity The polarity (or semantic orientation) of an expression (whether it is a text, a sound, or something else) represents its degree of positivity. Polarity categories go from very negative to very positive, either on a discrete or on a continuous scale.

Sentiment Sentiment equals the disposition of an entity toward an entity, expressed via a certain medium. This working definition consists of (1) the expression by an entity of its disposition, in the form of verbal or nonverbal communication, (2) the expression has a polarity or a semantic orientation measurable on a discrete or a continuous scale, and (3) the expression is oriented toward (an aspect of) an entity.

Sentiment analysis Sentiment analysis is about the extraction of sentiment from the medium it is expressed through. Multimodal sentiment analysis covers textual, audio, and visual media.

Sentometrics The term "sentometrics" is a portmanteau of sentiment and econometrics. It deals with the computation of sentiment from any type of qualitative data, the evolution of sentiment, and the application of sentiment in an economic analysis using econometric methods.

Supervised learning Supervised learning is a branch of machine learning that requires an annotated dataset (i.e. a set of input data with associated output values) to train a model.

Term frequency–inverse document frequency The term frequency–inverse document frequency (tf-idf) statistic is a widely used term weighting scheme. It weighs terms (e.g. words) based on their frequency of occurrence in a corpus ("tf"), but revalues upward the terms appearing across few documents ("idf"). The idea behind the latter adjustment is that less frequent terms on corpus level are of greater value to detect the specificity of a document.

Text mining Text mining is the process of extracting signals from and discovering structure

in large and heterogeneous textual data. It extensively uses tools from NLP.

Time series A time series is a sequence of (typically numerical) observations orderly indexed at a specified time interval (e.g. weekly).

Unsupervised learning Unsupervised learning is a branch of machine learning where the input data decide the output categories or representation by themselves. Any unsupervised method is typically hybrid or semi-supervised, as there is often need for certain minimal inputs from the modeler.

Introduction 1

Data can be regarded as the prime kingdom within analytics, with models and business integration as the two others. Its most intriguing tribe is made up of alternative data. Textual, audio, and visual data are battling to rule for power, individually or as a clan. As our world runs on feelings, sentiment might be the ultimate alter ego of these data sources. But it is very good at hiding itself...

As argued, less stylistically, by Shiller (2017) and others, the field of economics should accommodate for a proper quantitative study of the ever-evolving alternative and rich data sources it faces. Sentiment has already been used in various ways as either a parameter or a variable in econometric modeling. In recent years, it has become popular to analyze the sentiment embedded in textual, audio, and visual data, rather than using questionnaires or preexisting proxies to quantify sentiment. The increasing availability of these data thanks to the digitization of communication media has steered the interest and belief that they are carriers of potentially interesting information useful for economic analysis. To grasp the information, a thorough "text-to-data" approach is necessary. This has spurred the development of econometric methodology to transform qualitative sentiment data into quantitative sentiment variables, and to use those variables in an econometric analysis of the relationships between sentiment and other variables. In this thesis, this emerging research field is referred to as *sentometrics*, which is a portmanteau of sentiment and econometrics.

This doctoral thesis focuses on textual data and its representation as a sentiment variable useful for economic and financial research. As such, the goal of this thesis is to provide a point of reference for researchers in econom(etr)ics, finance, and machine learning interested in the analysis of qualitative sentiment data, and in particular textual data.

The general contribution of this thesis is threefold. It summarizes the current research field

and formally packages it into a new field termed sentometrics, it offers a fast and effective computational implementation immediately usable by researchers and practitioners, and it applies the proposed framework to two use cases. The thesis is therefore subdivided in below four key components as separate chapters.[1]

A Methodological Digress

I provide an overview of the main practices in studying alternative (in particular textual) data in economics and finance, and motivate their unification.

Augmenting a given econometric model with sentiment information is considered "econometrics meeting sentiment." The simplest example is including one or more sentiment variables in a regression model. The transformation of sentiment data into variables useful for your analysis is what "sentiment meeting econometrics" is about. Sentiment data needs to be converted into interpretable numbers and then aggregated. The sentiment can be modeled as such. The first main chapter (**Chapter 2**) explains sentiment and its origins, and discusses the various extraction methodologies together with some applications, empirical results, and useful Python and R software.

Sentiment is defined as "the disposition of an entity toward an entity, expressed via a certain medium." This definition is on purpose broad. The advantage is that it is simple and thus useful from an econometric modeling perspective. The disadvantage is that it seems to come at the cost of limiting the possibility to shed light on the economic channels that might drive results. However, the definition serves only as a frame of thought. The precise meaning and extraction of sentiment is filled in along the application of a sentometrics analysis, depending on the desired outcome.

To give an example, suppose one is interested in predicting stock returns using sentiment indices. To learn something about the "why" behind (changes in) the predictions, it is necessary to make specific indices for each channel of interest. For instance, the indices could represent sentiment about expected cash flows, about risk aversion, and about outright irrationality. Some channels are more easily captured by alternative data than others, so each index needs to be cautiously validated in order to draw correct conclusions.

A Computational Framework and Implementation

I implement and describe an R software package that allows executing the majority of the methodological analysis proposed in the previous chapter.

Textual data requires particular techniques for computational processing and analysis. I identify a gap in the current open-source software landscape in terms of an overall framework

[1] Each chapter can be read on its own, having, for instance, its own notation. The four main chapters are written in we-form, as they are joint work.

for fast sentiment computation, aggregation, and the use of the resulting (time series) variables into econometric models. The second main chapter (**Chapter 3**) explains the implementation of a software package that fills in the majority of this gap.

I created and maintain the fully open-source R package **sentometrics** (see `https://github.com/SentometricsResearch/sentometrics` for the source code and more), and provide a hands-on introduction to using the software for optimized textual sentiment indexation. The **sentometrics** package implements an intuitive framework to efficiently compute sentiment scores of numerous texts, to aggregate the scores into multiple time series, and to use these time series to predict other variables. I end by illustrating how to use the package, describing the entire workflow with a built-in corpus of news articles from two major U.S. journals to forecast the CBOE Volatility Index.

The package has been downloaded over 14000 times, and was already used successfully in several textual data science courses.[2] It is truly unique in the way it places textual sentiment extraction and aggregation oriented to time series into a flexible and modular framework. No other R software with an explicit focus on text-based (sentiment) time series exists.

An Application to News-Based Sustainability Scoring

I develop an innovative application dealing with extracting news signals around events relevant to Environmental, Social and Governance (ESG) issues on company-level, applying insights from the broad methodological analysis and making use of the computational implementation.

Investors base their sustainable investment decisions on in-house research and external data providers. There has been consensus that third party ESG ratings are diverse, not transparent, and lack standardisation (see Berg *et al.*, 2019, Amel-Zadeh and Serafeim, 2018, and Escrig-Olmedo *et al.*, 2010).[3] Most agencies only provide up to monthly updates, and the ratings they provide are often reporting-driven and not signal-driven. This implies that a company can be ESG-compliant "by the book" when it is transparent, but that the ratings might miss out on evolutions related to the true current underlying sustainability profile. Some of these evolutions can be more quickly picked up by automatically analyzing news data. The third main chapter (**Chapter 4**) presents a monitoring methodology that summarizes news relevant to ESG aspects about companies into tractable indices. This is a sentometrics analysis at heart. The approach embeds the use of text mining techniques to query and transform news data into frequency- and sentiment-based indicators. Word embeddings are used to achieve maximally

[2] Download instructions and code examples are assembled at the package website (see `https://SentometricsResearch.github.io/sentometrics`). An extensive documentation manual is kept at `https://cran.r-project.org/web/packages/sentometrics/sentometrics.pdf`.

[3] The U.S. Securities and Exchange Commission (SEC) even expressed its concerns about the danger of using and mixing different ESG ratings, see the Financial Times article "SEC chair warns of risks tied to ESG ratings" available at `https://www.ft.com/content/2c662135-4fd3-4c1b-9597-2c6f8f17faed`.

informative news selection and scoring.

I discuss several ways to use the outcome of the methodology from the viewpoint of an asset manager wanting to actively follow ESG-related news about the companies in his or her portfolio. For instance, (s)he can use the timely text-based indicators to (temporarily) reduce exposure to companies that pose greater ESG risks. More so even than regular asset managers, sustainable asset managers should "say what they do" and "do what they say", as investors who put their money in sustainable funds often have a strong motive for ESG compliance to be enforced diligently. Hartzmark and Sussman (2019) find that U.S. mutual funds categorized as high sustainability indeed have had considerable more net inflows than funds in the low sustainability category. It thus makes sense for asset managers to pay attention. The empirical analysis suggests the presence of abnormal news dynamics leading up to ESG downgrades as seen by an external data provider, and of efficient (more sustainable and equally performing) portfolio screening.

An Application to News-Based Economic Policy Uncertainty

I take the methodology of Baker *et al.* (2016) to create news-based indices capturing economic policy uncertainty (EPU), and apply it to the case of Belgium. Baker *et al.* (2016) have become the standard when it comes to the creation of EPU indices, but coherent and maintained time series output for Belgium was missing.

The level of uncertainty regarding (future) economic policy is taken into account by economic actors when they decide what to do next. In the last main chapter (**Chapter 5**), I compute monthly and daily time series for Flanders, Wallonia and Belgium that follow up on how often terms related to economic policy uncertainty are covered in the press. The attention the press gives to these terms is an effective indicator of the latent aggregate level of uncertainty. The indices turn out to reflect both national (such as federal elections) and international (such as the global financial crisis) evolutions. The indices are in line with other yet existing uncertainty measures. I also present an approach based on the biterm topic model (Yan *et al.*, 2013) to automatically detect the underlying drivers of prevailing uncertainty. As a case study, I look at the daily EPU trend during 2020, especially during the months in which the COVID-19 pandemic took hold. There was a clear rise in uncertainty up to Belgium's lockdown mid-March, after which uncertainty was at its highest point since the beginning of the time series in 2001.

The created time series for Belgium are updated every month on the website `https://www.policyuncertainty.com/belgium_monthly.html`. This platform also includes a whole range of other EPU indices, making a cross-sectional geographic study possible.

An Overview of Methodology and Applications 2

Abstract

The advent of massive amounts of textual, audio, and visual data has spurred the development of econometric methodology to transform qualitative sentiment data into quantitative sentiment variables, and to use those variables in an econometric analysis of the relationships between sentiment and other variables. We survey this emerging research field and refer to it as *sentometrics*, which is a portmanteau of sentiment and econometrics. We provide a synthesis of the relevant methodological approaches, illustrate with empirical results, and discuss useful software.

2.1 Introduction

There is a long-standing tradition of using sentiment as either a parameter or a variable in econometric modeling. Historically, the use of questionnaires and proxies to quantify sentiment variables has been predominant. In recent years, it has become popular to analyze the sentiment embedded in textual, audio, and visual data. Such data are becoming increasingly available in large amounts thanks to the digitization of communication media. These media are carriers of potentially interesting information useful for economic analysis. This has spurred a new strand of econometric research that investigates the transformation of large volumes of qualitative sentiment data into quantitative sentiment variables, and their subsequent application in an econometric analysis of the relationships between sentiment and other variables. We refer to this emerging field as *sentometrics*, which is a portmanteau of sentiment and econometrics.

In this survey, we overview the methodology and applications related to an econometric analysis of sentiment extracted from qualitative data. We first define sentiment as the disposition of an entity toward an entity, expressed via a certain medium. Examples of entities include individuals, news media, companies, government associations, industries, and markets. This disposition can be conveyed numerically but is primarily expressed qualitatively through text, audio, and visual media. Sentometrics studies the computation of sentiment from any type of qualitative data, the evolution of sentiment, and the application of sentiment in an economic analysis using econometric methods. Many approaches already exist for using econometrics with textual data, as recently overviewed by Gentzkow *et al.* (2019a), but our focus on qualitative sentiment data is unique. The overview by Lewis and Young (2019) is limited to the most important analytical approaches for analyzing textual content in accounting and finance.

The goal of our survey is to provide a synthesis of the relevant work that serves as a gateway for researchers in econom(etr)ics, finance, and machine learning interested in the analysis of qualitative sentiment data.[4] The survey takes a hands-on approach by synthesizing the research around the common challenges. The first critical step is the clarification of the problem that one is trying to solve. In function of the question, one collects, prepares, and selects the different data. The filtered qualitative data are then transformed into numbers using domain-specific sentiment quantification techniques. These numbers are next aggregated into meaningful sentiment variables. The different intermediate aggregation steps involve combinations of sentiment calculation methods, as well as the use of various within-text, across-text, and across-time aggregation methods. These variables are used as main input in an econometric model that is set up to solve the question at hand. An important part of the econometric analysis of qualitative sentiment data is a continuous validation activity. For each of the above topics, we discuss the relevant methodological approaches and illustrate

[4] Appendix 7.1 provides a shortened synthesis in a step-by-step workflow structure.

with empirical results. We also have a section that sums up some of the available software to perform each step.

2.2 Definition of Sentiment

The term "sentiment" is used in many different contexts and research areas, but there is no established definition. We propose a working definition that encapsulates the most important characteristics of sentiment from the perspective of a researcher wishing to transform textual, audio, and visual data into sentiment variables and to apply them in an economic analysis. We also summarize the literature that highlights other characteristics and alternative definitions of sentiment.

2.2.1 Working Definition

We propose the following generic definition of sentiment:

Sentiment is the disposition of an entity toward an entity,
expressed via a certain medium.

This working definition of sentiment embeds three components. First, the expression by an entity of a disposition in the form of verbal or nonverbal communication. To observe the private state of mind of any entity, one has to look at their subjective expressions through both qualitative sentiment data such as textual, audio, and visual data, as well as numeric data such as quantitative survey data and stock market data. The combined use of different sources of qualitative sentiment data is called multimodal sentiment, compared to the use of one data source, which is referred to as unimodal sentiment. Not many studies have assessed the added value of multimodal sentiment, but in general, findings confirm an increased level of accuracy over unimodal sentiment (see, for example, Soleymani *et al.*, 2017). Despite the fact that there are differences between, for example, the expression of emotion and sentiment (or other human subjective terms), it is not necessarily in the interest of a researcher interested in sentiment to make this distinction explicit. Second, the disposition has a measurable polarity or semantic orientation that shows through the medium of expression. It reveals the direction and intensity of the subjective expression, on a discrete or a continuous scale. Many definitions simply use positive or negative to indicate semantic orientation, but application-specific terminologies, such as bullish and bearish (Antweiler and Frank, 2004), dovish and hawkish (Picault and Renault, 2017), or Democrat and Republican (Gentzkow and Shapiro, 2010), can be helpful. Sentiment is usually asserted at different levels of granularity (e.g. a sentence, an entire article, a sequence of sounds, or an image). Third, the sentiment is oriented toward (an aspect of) another entity, or exceptionally the expressing entity itself.

This general synthesis encompasses a broad range of different sentiment definitions that are currently used in the field of economics. Casey and Owen (2013) describe consumer confidence as the consumer expectations about the future state of the economy. Ludvigson (2004) notes that surveys are most often used to measure consumer confidence. De Long *et al.* (1990) define investor sentiment as a belief about future cash flows and investment risks that is not justified by the fundamentals. Baker and Wurgler (2007) list several mediums through which this investor disposition is expressed. Notable examples include the use of investor surveys, proxies for investor mood changing variables such as the number of hours of daylight, and the analysis of market data such as trading volume, implied volatility, mutual fund flows, the premium on dividend-paying firms and the closed-fund discount. In a survey on sentiment in finance, Kearney and Liu (2014) argue there are two types of sentiment—namely, investor sentiment, which includes only the subjective judgments and behavioral characteristics of investors, and text-based or textual sentiment, which may also contain a more objective reflection of the conditions of a certain entity. Kräussl and Mirgorodskaya (2017) hypothesize that media sentiment translates into investor sentiment. Moreover, Chang *et al.* (2015) say sentiment affects the formation of investors' beliefs and thereby their reactions to information shocks.

In the remainder of the paper, we focus on the use of qualitative sentiment data as the medium through which the sentiment is expressed.

2.2.2 Other Definitions

From a psychological viewpoint, Munezero *et al.* (2014) state that sentiment is one of the so-called human subjectivity terms that reflects a person's desires, beliefs, and feelings. These human subjectivity terms are features of a person's private state of mind that can only be observed through textual, audio, or visual communication. For instance, the tonality of one's voice, or the frequency and length of pauses are informative about underlying sentiment. The same holds for pitch, facial and bodily gestures, the word use in a written article, and the colors present in a picture. Other human subjectivity terms include affect, feeling, emotion, and opinion. While Munezero *et al.* (2014) argue that there are some notable differences, these terms are mostly used interchangeably in different strands of the literature. For instance, the distinction that sentiment involves enduring emotional dispositions toward an object, whereas emotions are briefer, is not of direct interest for most economic applications. Based on social theory, Evans and Aceves (2016) classify sentiment as a human state that reflects its condition at a given time and place. Other states include preference, uncertainty, and ideology, among others.

Taboada (2016) details linguistic sentiment as the expression of subjectivity either as a positive or as a negative opinion through language. Soleymani *et al.* (2017) define sentiment as a long-term disposition with a certain polarity toward an entity. From a text mining perspective,

Taboada *et al.* (2011) treat sentiment as equivalent to semantic orientation, containing an evaluative factor (i.e. positive or negative) and a corresponding strength. In a survey on sentiment analysis, Liu (2015) makes no distinction between sentiment and opinion and defines an opinion as a quintuple of (1) the expressed sentiment, (2) the entity toward which it is expressed, (3) the particular (aspect of the) entity that is mentioned, (4) the opinion holder, and (5) a time stamp. This definition is closest to ours. Van de Kauter *et al.* (2015) distinguish between explicit sentiment (conveying subjective private states) and implicit sentiment (conveying factual information). Shapiro *et al.* (2018) deploy a characterization of emotions along the two dimensions valence (how positive) and arousal (how charged). The valence of word-of-mouth in marketing research is referred to by Gelper *et al.* (2018) as a discrete or continuous metric that captures the attitudes toward a brand. Additionally, sentiment can be looked at from the perspective of the sender (e.g. the sentiment attached by the author of a text), and from the perspective of the receiver (e.g. the sentiment perceived by the average reader).

In their analysis of political sentiment, Grimmer and Stewart (2013) use both the terms "sentiment" and "tone." They determine tone based on whether information is conveyed positively or negatively. Tone is more often used in the accounting and finance literature. Bajo and Raimondo (2017) characterize tone of news as a combination of the degree of positiveness, negativeness, and uncertainty. Feldman *et al.* (2010) define tone as the optimism or pessimism of the information embedded in qualitative verbal disclosures. Henry (2008) defines tone in earnings press releases as the affect of a communication. In the construction of a news-based coincident index of business cycles, Thorsrud (2020) uses tone as a synonym for sentiment and identifies it by determining whether news articles are positive or negative. In this survey, we also treat tone as a synonym for sentiment.

2.3 Problem Definition

Sentiment data have the potential to help solve or understand many problems involving the use of econometrics, across the fields of economics, finance, accounting, marketing, psychology, and computer science, among others. The adequate choice of methods to analyze sentiment data depends on the goal of the analysis. A common ground for econometrics applied to sentiment analysis is that one first needs to measure sentiment. Below, we discuss the use of qualitative sentiment data in applied economic theory and as an information source in nowcasting and forecasting economic variables.

2.3.1 The Role of Sentiment Data in Applied Economic Theory

At least since the work of Keynes (1936), economists have been wondering what role sentiment plays in influencing economic decision making. Understanding the relationship between

sentiment data and decision making at the micro and macro level is still important in economic theory. Sentiment can be considered either to contain fundamental information in the news sense (typically about cash flows, and risk aversion), or to capture irrationality up to "animal spirits" in the noise sense.[5] Both types of shocks move economic expectations and market outcomes at different horizons (Angeletos *et al.*, 2018). The challenge according to Angeletos and La'O (2013) is that economists first have to model and then to quantify the sentiment forces behind the formation of market expectations. They develop an economic theory that does not depart from rationality but rather connects market expectations with market outcomes through external shocks they call sentiments. Barsky and Sims (2012) create a dynamic stochastic general equilibrium (DSGE) model that accommodates both the information and animal spirits view of confidence. They find most empirical evidence for the perspective that innovations in confidence reflect information about future economic prospects. One explanation for the importance of economic sentiment is that it acts frequently as a self-fulfilling prophecy (see Petropoulos Petalas *et al.*, 2017 and the references therein). When there is consumer or business pessimism about economic growth, actual negative growth can be a direct consequence of it. A more specific example is bank runs. When too many depositors' sentiment about other depositors is negative, the unwanted outcome, a bank run, is more likely to materialize (Diamond and Dybvig, 1983).

In this regard, sentiment indices based on qualitative data can provide a more direct data-driven instrument to assess various types of economic shocks, or proxy for matters such as confidence or expectations. Larsen and Thorsrud (2019) use structural vector autoregression to identify news and noise shocks in a panel of text-based measures and other economic variables.

Sentiment proxies provide a way to test behavioral hypotheses on the aggregate level or on the individual level. In general, the key questions pondered in a behavioral analysis are "What drives sentiment?" and "What is the behavioral impact of the sentiment transmitted?" An example of a behavioral hypothesis is whether entities inflate the tone in their written communication to influence market reactions. Both Picault and Renault (2017) (for the European Central Bank) and Arslan-Ayaydin *et al.* (2016) (for firms) validate this hypothesis. Sentiment in texts can be argued to be driven by a self-interest to generate particular external outcomes. In Garz (2014), the evidence shows a strong bias in terms of the number of negative and positive reports related to unemployment that is not the consequence of an asymmetric interpretation of the official numbers but rather associated with noneconomic information and the process of news production itself. Along these lines, the degree of sentiment involved in images appended to advertisements can also have clear intentions to impact customer behavior. Kalogeropoulos (2018) studies the impact of various media outlets on individual economic expectations, not finding tone to be a good predictor. In the finance literature, behavioral theory predicts that short-horizon returns are reversed in the long run (Tetlock, 2007).

[5] Depending on the perspective, noise can also be seen as news.

There is a growing concern about the severity and impact of media bias. Closely related, Flaxman *et al.* (2016) explain there are two strands of thought about the impact of improved production, distribution, and discovery of news articles (or generally, any multimedia). Some defend it increases exposure to diverse perspectives; others argue that it increases ideological segregation. They find empirical support for both camps, thus a further investigation of the impact of a biased media production and consumption behavior would be worthwhile.

Boudt and Thewissen (2019) base their analysis of CEO letters on psychological phenomena such as framing (Tversky and Kahneman, 1981) and the serial position effect (Glanzer and Cunitz, 1966). These and similar phenomena can also be used to better understand the sentiment conclusions. Purely as an illustration, stronger weights of negative sentiment words in comparison to those of positive sentiment words could be supported by the negativity bias, claiming that negative things have a greater impact. It could also be related to the fact that news media tend to emphasize negative news (e.g. Lowry, 2008). The priming, agenda setting, and framing communication theories described in Scheufele and Tewksbury (2007) can also be subject to more precise testing using multimodal sentiment measures.

Many different entities can have sentiment attributed from different data sources. Sentiment across these different entities tends to interact in particular ways, possibly in a contagious manner. The assessment of these sentiment flows, the feedback effects, and the associated information dispersion over time concern a network analysis of sentiment. The Global Database of Events, Language, and Tone (GDELT) project[6] is the most comprehensive effort to date of a global network analysis of events and related sentiment. These data and social media data are useful to analyze what network structures would mitigate behavioral misperceptions, a question brought up by Teoh (2018).

Larsen and Thorsrud (2018) use a graphical Granger causality modeling framework to gain insights into the network of economically relevant news topics. Every node in the graph represents a sentiment/topic time series. The graph can be used to detect which narratives dominate and what the degree of news spillover is—that is, what news stories from which countries Granger cause the occurrence of any other news stories.

Eshbaugh-Soha (2010) emphasizes the role news coverage and tone can have on government trust and how it is central to explaining effective leadership. News provides country leaders a means to communicate their messages, but the local perception can differ significantly. Therefore, an interesting study would be to assess the spread between the sentiment of news reported in one region and the sentiment of similar news reported in another region.

[6] See `https://www.gdeltproject.org`.

2.3.2 Measuring, Nowcasting, and Forecasting of and with Sentiment

Sentiment time series indices aim to reflect the evolution of sentiment over time. A well-known text-based example is the Economic Policy Uncertainty (EPU) index of Baker *et al.* (2016).[7] This index measures uncertainty, a specific type of sentiment. Manela and Moreira (2017) create a news-based measure of option-implied uncertainty, arguing it incorporates disaster concerns expressed via the media. In many other applications, sentiment is also considered an explicit or implicit proxy for a certain desired output, such as for the visualization of company reputation (Saleiro *et al.*, 2017).

We refer to quantifying already observed sentiment as sentiment measurement, while the prediction of the unobserved current and future sentiment is called sentiment nowcasting and forecasting, respectively. Sentiment is a latent variable, meaning it is not readily observable. Measuring sentiment is a key task in any sentiment-based analysis.

In the now- and forecasting literature, sentiment measures are considered a timely driver of other variables. There are three approaches to the type of sentiment that is used. Sentiment is proxied using available (questionnaire-based) indices, sentiment is constructed itself from a qualitative data source commonly using relatively simple methods, or sentiment is bought from a data provider such as Reuters (e.g. Thomson Reuters MarketPsych Indices) who in general uses a more complex methodology for the computation. The obtained sentiment is then transformed for usage in prediction models, to obtain the best possible prediction at any time. Sentiment variables are rarely used alone as explanatory variables but are usually added to a set of standard explanatory variables to see whether its integration improves or deteriorates forecasting performance.

The integration of sentiment has indeed already shown its capacity to improve forecasting performance. A significant impact of sentiment expressed through diverse media on stock returns and trading volume is found by Heston and Sinha (2017), Jegadeesh and Wu (2013), Tetlock *et al.* (2008), Tetlock (2007), and Antweiler and Frank (2004). Ardia *et al.* (2019b) incorporate textual sentiment time series into the long-term forecasting of the U.S. industrial production growth rate using sparse regression techniques.

Beyond improved predictions, using sentiment data is very flexible and timely, especially compared to traditional sentiment extraction methods such as surveys. Changes in sentiment methodology can easily be backtested using the available data. Modifying the structure of a survey, however, necessitates the survey to be sent out again to obtain new results. Information derived from sentiment data hardly suffers from release lags, making timely sentiment an ideal variable to enhance nowcasting models and consequently to craft timelier policy responses.

As in Hamilton *et al.* (2016), sentiment analysis on word level can be used to measure the

[7] The EPU index for various countries can be retrieved from: `https://www.policyuncertainty.com`. The online publication of text-based indices is becoming prevalent, see also: `https://www.retriever-info.com/fni`.

time-varying perception surrounding certain words. For instance, "terrific" had a negative connotation up to 1960, but then became more positive. Lukeš and Søgaard (2018) find that words predictive of sentiment at one point in time remain not necessarily equally predictive at a later point, and that models trained on old data perform worse than models trained on recent data. They suggest a predictive feature selection approach to deal with temporal polarity shifts. The implication of changes in language over time is that the methods of sentiment quantification should evolve with it.

It is becoming well established in economics and finance that adding soft (qualitative) information on top of hard (quantitative) information results in predictive information gains. However, the soft information is usually explored through textual content. Audio and visual content have been explored less so but may deliver additional information value, according to Mayew and Venkatachalam (2012), who find that vocal cues of managers during conference calls predict a firm's future performance.

2.4 Qualitative Sentiment Data

The various ways in which economic agents express their sentiment leads to textual, audio, and visual sentiment data. Sentiment data are short for "sentiment-bearing" data. Most of the examples and methods in the remainder of this survey focus on textual data, because audio, and visual sentiment analysis is still in its infancy (Soleymani *et al.*, 2017). Teoh (2018) does so similarly but acknowledges the rising relevance of audio, and visual data. Currently, the main focus of current research in sentometrics effectively lies with textual data due to their wide availability in the digital form of news media articles, company filings, or social media posts (see e.g. Loughran and McDonald, 2016).

The choice for textual data thus comes from the fact that most research and applications have been developed for this type of data. An advantage of focusing on texts is that many other forms of unstructured data can be transformed into textual data and then analyzed as if they were textual. For instance, audio data are often transcribed into textual data and can thus be analyzed using tools from this domain.

Multimodal sentiment analysis techniques are expected to gain importance due to the internet, which has become more of a widespread multimedia platform. Where possible, we highlight close relations between the analysis of textual data and the analysis of audio, and visual data, covering potential similarities from one data type approach to the other. Doing so, we outline a uniform high-level framework that is applicable to all these data sources. The concepts of feature extraction, quantification, aggregation, modeling, and validation are very much transferable, though almost never presented as such.

2.4.1 Information Sources

The information sources for sentiment analysis in econometrics can be grouped in two ways. First, it can represent where the data were published. This includes news outlets (a journal, a social media channel, YouTube, a vlog, or a blog), companies and governments (regarding the publication of an official press release or an official report), or publication venues (an academic journal or a book publisher), among others. The source in this context should not be confused with the actual expresser of the sentiment; for instance, the source can be a journal, and the expresser a company or one of its top managers.

Second, it can represent from where the data were retrieved. The largest worldwide textual data providers are LexisNexis, Dow Jones' Factiva, and Reuters. Access to these databases is paid. A cheaper alternative, if allowed, is to scrape textual data from the web. A specific scraping procedure needs to be set up, which is a cumbersome activity, and in general goes with a considerable degree of hit-and-miss in terms of texts successfully collected. There also circulate some freely available datasets—for instance, the eight text datasets analyzed by Zhang *et al.* (2015) or the list of freely available text datasets provided by Ravi and Ravi (2015).[8]

The acquisition of the data requires a good data management system, able to structurally store many gigabytes, such as MySQL. The database should also have fast query functionalities, for example delivered by technologies such as Solr or Elasticsearch.

2.4.2 Alternative Sentiment Variables

Instead of the algorithmic extraction of sentiment from data, sentiment is also often proxied by asking people through surveys. The U.S. Consumer Confidence Index or the European Economic Sentiment Indicator are actively monitored examples of indices based on surveys (see for instance the analyses of Ludvigson, 2004 and Gelper and Croux, 2010). However, surveys have the downside of being costly, are hard to replicate, have a publication lag and cannot be backtested. Both survey-based measures and data-based measures have their value, and are in many cases complementary. Ardia *et al.* (2019b) show that the specification that includes both time series measures generates the best out-of-sample predictive power. Baker and Wurgler (2006) derive a sentiment index through a principal component procedure from six sentiment proxies proposed in the literature, without going through any sentiment quantification process themselves.

Similar to textual data providers, there exist a number of textual sentiment data providers. Two often-used solutions are the series from RavenPack, and the Refinitiv (formerly Thomson Reuters) MarketPsych Indices.

[8] A collection of open-source textual, audio, and visual data can be found at `https://pathmind.com/wiki/open-datasets`.

2.4.3 Data Limitations

A first limitation concerns data availability and the disagreement between textual databases. Ridout *et al.* (2012) find preliminary evidence that there are stories (in their study mostly international coverage) from printed newspapers that are systematically missing in electronic databases. Thus, not only do texts need to be collected, but the content from multiple sources also needs to be aggregated neatly. Chiou and Tucker (2017) cover some of the likely issues of content aggregation. The problem of data availability and data disagreement is small for open government databases, such as accounting textual data (e.g. EDGAR), court decisions (e.g. PACER), or patents (e.g. Espacenet). Much in the same way Riffe *et al.* (2019) note that the universe of online posts is "unlimited and unknowable and inherently unstable over time", the problem becomes more persistent for data coming from corporate resources, news media, or social media. Any sample drawn from that data might not be representative due to nonrandom sampling; true probability sampling is hard in the context of big data. Lacy *et al.* (2015) mention convenience sampling (a sample primarily defined by availability) and purposive sampling (a sample primarily defined by the nature of a research undertaking) as common practices.

An important aspect in data collection is the notion of data vintages. In a real-time setting, a researcher uses the data available at a given time, called a vintage or a snapshot. Yet, many data are subject to revisions. For instance, most data used in macroeconomics are updated one or more times until final numbers are reached (Croushore and Stark, 2003). The compilation of the FRED-MD historical vintage database of macroeconomic indicators was a response to this problematic (McCracken and Ng, 2016). This same difficulty persists in textual data, particularly with online publication and social media, with the data frequently updated, revised, or even removed from the information outlets. Saltzis (2012) reveals in a sample of breaking news stories on six major U.K. online news sites that the stories were updated on average 5.7 times. As such, the traditional process of scraping websites for historical news may lead to a forward-looking bias since the retrieved news will typically be the latest version of the news articles and not the one at the time of first publication. This phenomenon is crucial to deal with in intraday studies.

The problems described above lead to issues of reproducibility and limitations to generalizability of results.

2.5 Preprocessing, Enrichment, and Selection of Sentiment Data

Textual, audio, or visual data rarely arrive in a format that is ready for input into an algorithm. The data typically start off being very unstructured, and through a sequence of steps structure is imposed to make the data ready for further analysis. We define restructuring as doing two

things: preprocessing and enriching the data. Both the preprocessing and metadata enrichment ideally come before the actual data selection to have the most information available to do an optimal filtering.

2.5.1 Restructuring Textual Data

In this subsection, we describe the preprocessing and metadata generation concerning textual data. Bholat *et al.* (2015) provide a useful summary of many relevant text mining techniques for preprocessing and data enhancement.

2.5.1.1 Preprocessing

Raw textual data often come in a JSON or an XML file from which the actual text needs to be extracted first. This process is called parsing. Depending on the type of data available, this can be a relatively straightforward or tedious task. As part of this process, remaining garbage such as HTML tags, addresses, or other formatting is removed, or simply not selected through the parsing algorithm.

Furthermore, textual data are inherently (ultra)high-dimensional (Kelly *et al.*, 2019). Gentzkow *et al.* (2019a) highlight that to structure a text with a length of w words, each of which is drawn from a vocabulary of q possible words, the unique representations of this text has dimension q^w. Moreover, all characters in the text are probably not equally informative in assessing the sentiment of a particular document. For example, stop words such as 'the' are seldom indicative and are usually removed to reduce the noise and the dimensionality. Some type of further cleaning is often required to deal with issues such as spelling mistakes or (nonstandard) abbreviations (Nowak and Smith, 2017). Denny and Spirling (2018) outline several common preprocessing steps. The output is a corpus of cleaned texts.

Textual data comes in various granularities: words, sentences, paragraphs, and whole articles. Sentiment is the output of a function applied to specific components extracted from texts, also called terms. The most common kind of components are n-grams, a sequence of n words. Breaking up text into n-grams is called tokenization. If $n = 1$, tokens are referred to as unigrams. A bag-of-words approach presumes that the relative order of unigrams is irrelevant, but words are not necessarily independent of each other. More generally, a bag-of-words can be denoted by bag-of-tokens, where tokens can be any sequence of words. Further cleaning is needed to drop, for instance, punctuation marks, or transform all terms into lowercase, stemmed, or lemmatized form.

Terms are summarized into a document-term matrix, with the rows as the documents, the columns as the terms, and the cells as the values that measure the (weighted) frequency of occurrence of the terms. A document-term matrix is usually of high dimension and consequently very sparse, meaning, with a lot of zero entries. In a document-term matrix, the sparsest features are typically removed.

2.5.1.2 Metadata Enrichment

A corpus as is consisting of only documents can be enriched by adding all sorts of metadata. Metadata either already exist or are objective, such as a time stamp, the author, the news outlet, the language, or the geography. A good case for having metadata is that textual information is expressed across many different venues, including newspaper articles, newswires, and social media, all with a possibly differing degree of information value. Heston and Sinha (2017) emphasize the importance of studying news types to understand how financial markets process information and when underreaction and overreaction in returns occur. Aggregation across a metadata marker gives information about the sentiment concerning that particular metadata—for example, the sentiment about a given economic topic.

The available qualitative metadata need to be quantified for further use in the analysis. This can be done using binary or relevance variables. In the first case, one enumerates all qualitative metadata across the corpus for a given article and assigns a value of 1 if the metadata are of importance to that article, and 0 if not. A relevance variable follows the same principle but assigns a continuous score based on the connectedness of the metadata to the article. Some metadata lend better to be modeled as a dummy variable (e.g. language or geography), others as a relevance variable (e.g. predefined topics). If there are too many individual instances of the metadata, one can consider to group them. Other metadata can be generated using text mining models. The first type of metadata that can be generated are entities, using named entity recognition extraction techniques. The second type of useful metadata are topics and related keywords based on a supervised or an unsupervised topic model. The features can be valued as the probability score coming out of the topic model. Readability of a text (e.g. Loughran and McDonald, 2014) or the tense of a text are two other potentially useful metadata indicators.

2.5.2 Restructuring Audio and Visual Data

The underlying raw format of audio, and visual data is less comprehensible and vaster than textual data. One second of a video is size-wise equivalent to at least hundreds of pages of text; the maxim “An image is worth a thousand words” is no exaggeration. We emphasize some important aspects about the restructuring of audio, and visual data.

2.5.2.1 Preprocessing

For sentiment classification, visual and audio data are processed into emotional clues handy to discriminate between different sentiment categories. A major focus of sentiment extraction in visual data is on facial expressions. Secondary are other nonverbal expressions (e.g. hand gestures) and environmental factors such as what is happening in the background. There are seven basic emotion classes (danger, sadness, surprise, fear, disgust, joy, and contempt) that

can be inferred using a facial expression coding system originally proposed by Ekman and Friesen (1976). One can then construct variables that express the distance between several of these positional facial characteristics.

Visual data can be boiled down to image data. A video in that respect is a collection of segments, and every segment is a collection of images. Audio data can be boiled down to textual data using speech-to-text technology complemented with specific audio features (such as pause duration).

2.5.2.2 Metadata Enrichment

The principles of metadata enhancement for textual, audio, and visual data are similar, but the content of the metadata is different. Qualitative metadata such as author or time of publication are the same. Examples of useful audio features are pitch, pause, laughter, overlaps, and voice intensity; examples of visual features are color and motion. Videos have both visual and audio features. Wang *et al.* (2003) describe features and extraction techniques across four categories (spatial visual, motion, coding and audio).

A downside of features retrieved from nontextual data is that many of them require a large-dimensional representation. For a video, a manner to construct a feature called "smiling" could be to take the number of seconds a person smiles in the video. The decision of whether a person smiles is a function of various facial characteristic points that should be mapped deterministically to the binary outcome "smiling" or "not smiling."

2.5.3 Selection of the Relevant Data

Following a general-to-specific approach, the original and vast corpus of documents needs to be trimmed to a subselection of relevant texts. If the selection procedure is too restrictive, important data may be omitted; however, if irrelevant data are included, it may drastically lower the signal-to-noise ratio. This can be considered as "querying" the corpus database to extract the right selection of texts. Querying can be based on a search of keywords in the database using, for example, a *regular expression*. It should be made clear beforehand which texts are necessary to include in the analysis, or the selection can be approached as an optimization problem itself. The latter strategy would require defining different sets of keywords and finding out which keywords give the best outcome in terms of an objective (e.g. forecast accuracy).

To model an outcome variable, it is not always needed to focus exclusively on (sentiment) measures directly related to that variable. On the contrary, Larsen *et al.* (2020) argue that many news topics are, in their case, of interest to form inflation expectations, thus it would be limiting to only target media mentioning terms related to inflation. Kelly *et al.* (2019) tackle the problem of selection simultaneously with modeling a set of observed covariates.

They propose a method that only includes phrases of interest when useful, conditioning on the observed covariates, building on the model of Taddy (2015a).

2.6 Quantification of Sentiment

Any sentiment measure is a proxy for the actual prevailing sentiment and needs to be *estimated*. This can be done by human annotators or by a statistical function. A wide variety of techniques exist to infer the sentiment embedded in qualitative data, but measuring sentiment is inherently application- and data-specific. Therefore, it is neither possible nor recommended to consider sentiment computation in a single manner.

Sentiment is quantified for a given observational data unit—for instance, a text or a video. Quantification of sentiment is either on a discrete scale (classification into two or more classes, such as negative, positive, and neutral) or on a continuous scale. Based on decision rules, one can go from continuous to discrete output. Some methods produce a tuple of a positive and a negative sentiment (probability) score. Multiple sentiment scores from one computation method can be considered as separate methods and can turn out to be more informative.[9] Sentiment scores might benefit from a normalization for interpretation purposes and possible outlier elimination. Sentiment analysis can also take up a more fine-grained externalization, called aspect-based sentiment analysis (De Clercq *et al.*, 2017). This type of sentiment analysis separately measures the sentiment for different aspects and entities mentioned in the data unit. This is a combined problem that requires the extraction of entities and their aspect terms, classifying the aspect terms, before doing the sentiment calculation for each of the extracted combinations. One could draw an analogy with "feature-based opinion summarization" (Hu and Liu, 2004b), which is less specific.

2.6.1 Textual Data

Textual sentiment quantification uses tools from the broad field of natural language processing (NLP) to quantify the sentiment of a given text. It consists of many NLP-related subtasks, such as identifying entities and extracting relevant features. We briefly discuss the lexicon-based and machine learning approaches as the two main types of methods for sentiment computation. The data unit is usually a document, a paragraph, or a sentence. Some fields prefer one over the other. Sentiment can be detected more precisely at sentence level, but in political science, for instance, most often the analysis remains at document level since it requires less heavy NLP (Grimmer and Stewart, 2013). For a broader treatment of textual sentiment computation and associated subtasks, we refer to Liu (2015) and Ravi and Ravi (2015).

[9] For instance, a net textual sentiment value of two can be obtained from both two positive and zero negative words, or 20 positive and 18 negative words. The number of polarized words can be retained as separate sentiment scores, else its information can be used for within-unit aggregation.

The classification accuracy of the various sentiment approaches varies. Typically, machine learning algorithms outperform lexicon-based methods out-of-sample, at the expense of computational efficiency and model transparency. The difference in performance is a function of the type of texts and the domain specificity of the lexicon employed. Ribeiro *et al.* (2016) provide an extensive overview of the accuracy of both lexicon-based and machine learning–based sentence-level sentiment analysis. They compare 24 popular sentiment methods over 18 labeled datasets. Their experiments convey first of all a rather low average level of accuracy. More importantly, there are large differences in the accuracy across sentiment methods and across datasets. Their results also reveal no outstanding method at the sentence level. The conclusion is that a sentiment quantification method needs to be selected carefully depending on the purpose.

2.6.1.1 Lexicon-Based Approaches

A lexicon-based computation of sentiment is the most straightforward, efficient, and parsimonious method. Turney (2002) defines lexicon-based sentiment analysis as "calculating sentiment for a document from the sentiment of words or phrases in the document." Mechanically, this requires the use of a sentiment lexicon with sentiment information about important (combinations of) words, which is then matched with a text. A lexicon is thus a collection of pairs of words (or a sequence of words) and associated sentiment scores. In most cases, lexicons stick to unigrams, but for some applications, it is more effective to use n-grams. Picault and Renault (2017) construct a lexicon specific to European Central Bank communication and explicitly consider n-grams, such as the positive bigram "lower unemployment." The size of a lexicon ranges on average from in the hundreds to in the thousands. There is no preferred lexicon size; too large can mean inaccuracy due to noise, and too small might mean not enough coverage or a lack of important words. Comparing lexicons is not always easy, given the often varying sizes but also because there is no universal polarity grading system (Ravi and Ravi, 2015).

There is a distinction between general lexicons and domain-specific lexicons. Both the Henry lexicon (Henry, 2008) and the Loughran & McDonald lexicon (Loughran and McDonald, 2011) were developed as a response to the suboptimal applicability of generic lexicons to texts in the finance domain—for example, earnings press releases. The Lexicoder Sentiment Dictionary (Young and Soroka, 2012) is tailored to news content about politics. Lexicons are simple and the least black-box solution, and usable at any text level. However, lexicons can be brittle when facing domain shift and complex syntactic constructions (Täckström and McDonald, 2011). Very few lexicons are domain-portable, meaning applicable across several domains and text structures. It is difficult to achieve, if it is at all, and therefore hardly desirable.

Liu (2015) sees three broad ways of generating lexicons—namely manually, dictionary based, and corpus based. An additional approach to building lexicons involves a combination

of manual labor and a statistical methodology, which may arise from the machine learning literature. It is important to differentiate between machine learning algorithms for lexicon construction and those algorithms to measure sentiment but with no explicit intention to obtain a sentiment lexicon. We cover the latter in the next subsection. Apart from the manual approach, all methods entail automatic processes to varying degrees.

The manual approach to building lexicons has annotators assigning a sentiment score to selected words. Notable fully hand-curated lexicons are the Stone *et al.* (1963) General Inquirer and the Bradley and Lang (1999) ANEW word lists. Crowdsourcing platforms such as Amazon Mechanical Turk have made the task of developing high-quality manual lexicons more accessible nowadays. To our knowledge, the NRC lexicon of Mohammad and Turney (2013) was the first to be built using crowdsourcing services.

A dictionary based approach allows producing lexicons more cheaply while keeping a good level of accuracy. This method starts from a list of seed sentiment words with known polarity (often found using the manual approach) and then expands this list by using synonyms and antonyms coming from a large base dictionary. A suitable base dictionary is the WordNet database (Miller, 1995). This lexicon in conjunction with sentiment seed words was used to produce WordNet-Affect (Strapparava and Valitutti, 2004) and SentiWordNet Baccianella *et al.* (2010).

The corpus based method adapts an existing lexicon using information from a domain-specific corpus. The researcher first needs to adjust the sentiment orientation of the words to the new domain. Second, it may use linguistic rules to include new words in the lexicon. In this regard, Hatzivassiloglou and McKeown (1997) introduce the notion of sentiment consistency. For instance, adjectives with a similar sentiment orientation are often used in groups. Kanayama and Nasukawa (2006) propose the idea of sentiment coherency; the same sentiment orientation tends to be expressed in consecutive sentences, while sentiment change is expressed by an adversarial expression (e.g. “but” or “however”).

Statistical methodologies are the fastest and cheapest but the most prone to error. They typically start from a set of words from a previously built lexicon or a corpus, then a statistical methodology is used to find the sentiment orientation of those words. Jegadeesh and Wu (2013) use a regression framework to measure the sensitivity of words (“word power”) to stock returns; this could then be used to form a finance-specific sentiment lexicon. Lexicons can also be derived from (Bayesian) regularized methods, such as the Ridge, the LASSO, or the elastic net regression (see e.g. Nowak and Smith, 2017 and Pröllochs *et al.*, 2015). Pröllochs *et al.* (2015) argue that a shrinkage approach (Ridge regression) is superior over a variable selection approach (LASSO regression) because multicollinearity among the token predictors tends to be strong. In the corpus-based category, Engle *et al.* (2020) create a climate change vocabulary based on a collection of climate change white papers and glossaries. Their final lexicon is composed of the unique stemmed unigrams and bigrams, weighted by their respective term frequency–inverse document frequency (tf-idf) scores. To create a daily climate change index,

instead of term matching, they use the cosine similarity between the tf-idf scores of a given article and the scores in the lexicon.

Lexicons do not cope with the linguistic context around which the sentiment words appear. To this end, advanced lexicon-based methods integrate so-called valence shifters in the sentiment computation. Common types of valence shifters are amplifiers (e.g. very), downtoners (e.g. barely), negators (e.g. not), and adversative conjunctions (e.g. but). These valence shifters act on polarized words in the lexicon in particular ways depending on how close they appear to these polarized words. Taking the example of negation, one way to apply it to lexical entries is termed shift negation (Taboada *et al.*, 2011) as opposed to switch negation.[10] Having lexicons consisting of n-grams would also allow disambiguating of word use in different contexts. According to Young and Soroka (2012), even a modest integration of contextual (preprocessing) routines is fruitful. Taboada (2016) enumerates multiple linguistic insights to account for in sentiment analysis.

Not only domain specificity, but also language specificity is important. Most resources are still in English (Ravi and Ravi, 2015). In practice, one often sticks to translation. Either one translates the focused text from a resource-poor language into a resource-rich language (usually English) for which a robust sentiment method (e.g. lexicon) is available, or one translates an existing word list into the focused language. A third option is to translate annotated corpus resources from a resource-rich language to the focused language and use these to develop (or improve) another sentiment method. In many circumstances, however, the performance of translation results in a loss of accuracy. Mohammad *et al.* (2016) surprisingly find that, with Arabic social media as the focused texts, sentiment analysis of automatic English translations is competitive to existing Arabic sentiment analysis systems. On the other hand, translation made the human annotations become worse than sentiment analysis, and adding Arabic translations of sentiment-labeled English tweets data to Arabic training data resulted in a drop in accuracy, due to bad translations. Translation invariably comes with additional problems to solve. Bannier *et al.* (2019) start from the English Loughran & McDonald lexicon by doing word-by-word translation to German. On top of that, they deal with distinct grammatical features of the German language related to inflectional and lexical morphology, as well as compound wording. They claim to have described a comprehensive framework for future adaptations of dictionaries into other languages. To test the equivalence between their lexicon and the Loughran & McDonald one across positive, negative, and neutral categories, they rely on the two-sided equivalence test of Blair and Cole (2002). The test checks for accordance in terms of the mean number of detected polarity categories, given a confidence interval.

[10] The importance and application of valence shifters is also a function of the document type. Hutto and Gilbert (2014) created the VADER sentiment analysis system for social media texts, letting word shape (e.g. capitalization), slang (e.g. "kinda"), and emoticons, among others, act as valence shifters.

2.6.1.2 Machine Learning Approaches

The extraction of sentiment as a standalone problem is studied by machine learning and computational linguistics scientists. The purpose is to optimize the measurement of sentiment based on a learning algorithm typically benchmarked against an annotated dataset of text with corresponding sentiment values. The objective, in this case, is well-defined and dependent on the type of data source (e.g. product reviews or images) and the type of sentiment output (e.g. classification into positive or negative). The learning algorithm identifies the characteristics among the preprocessed smaller pieces of textual characteristics (i.e. words, n-grams, phrases, counts, and other information) that are most important in measuring sentiment. A survey of different machine learning algorithms applicable to text is given in Evans and Aceves (2016). Machine learning can be branched into supervised and unsupervised learning, both used on many occasions for sentiment analysis.

Supervised machine learning requires an annotated dataset—meaning, a set of documents with, for every document, a sentiment value, leading to what is often called the gold standard. Annotation can already exist from the data (e.g. product rating stars), but, in most cases, is constructed manually. Building such a dataset from scratch can be expensive and time-consuming while also prone to bias. Especially for domain corpora, annotation can be hard due to possibly complex specific sociolinguistic contexts (Hamilton *et al.*, 2016). The annotation cost also depends on the type of text. Van de Kauter *et al.* (2015) review some of the complexities of doing annotation. Taddy (2013a) outlines a procedure to select from a large corpus the texts that are most useful to annotate. Determining the best data examples to be labeled is referred to as (pool-based) active learning. Once the tagged dataset is obtained, a specific machine learning algorithm is trained with it. Pang *et al.* (2002) clarify the sentiment classification problem, and experiment with the Naive Bayes, maximum entropy classification, and Support Vector Machine (SVM) learning techniques. Naive Bayes and SVM are essentially text regressions of the sentiment target variable on a large-dimensional space of textual elements, such as words, which get assigned a weight. More recently, neural networks, primarily due to the emergence of deep learning, have become more prominent. One can also combine several learning algorithms. For instance, Das and Chen (2007) employ a majority voting scheme across five classifiers, claiming it minimizes false positives.

An unsupervised learning approach lets the data decide the categories or representation by themselves. Any unsupervised method is typically hybrid or semi-supervised, as there is need for specific minimal inputs from the modeler. A classic example is the suggested approach by Turney (2002), which ranks phrases based on their pointwise mutual information (PMI) with respect to two seed words, one negative ("poor") and one positive ("excellent"). It infers the semantic orientation from the semantic association with respect to a manual set of seed words. Remus *et al.* (2010) develop the German SentiWortschatz dictionary using the PMI approach. A vector space model (VSM) is a more complex undertaking. These models generate word embeddings, which are latent quantitative vector representations of textual

information, such as documents, paragraphs, words, phrases, and even letters. A VSM learns distributed vector representations that capture many precise syntactic and semantic word relationships. Words closer to each other in terms of linguistic context receive a more similar quantitative representation because they are assumed to share the same semantic meaning.[11] Global matrix factorization methods (co-occurrence counts based) and local context window methods (prediction based) are the two main families for learning word vectors.

Latent semantic analysis (LSA) is a notable example of a global matrix factorization method. It reduces high-dimensional count vectors to a lower-dimensional latent semantic vector space. Hofmann (2001) introduces a probabilistic version of LSA, defining the semantic space over a set of latent variables referred to as "aspects" based on a generative model for word-to-document co-occurrences. His model allows figuring out, for instance, which latent aspects are most likely to generate a word, or what the latent class posterior probabilities are given a certain document and word. Liu *et al.* (2009) refactor the model to capture a multidimensional measure of blog sentiment, considering sentiment as a joint contribution of a few hidden factors. They call their work S-PLSA (sentiment probabilistic latent semantic analysis). In a subsequent time series regression, they form sentiment variables as the average sentiment mass attributed to each of the hidden sentiment factors.

Most of the recent research on word embeddings has gravitated toward the prediction-based method using neural network architectures. The Word2Vec approach of Mikolov *et al.* (2013) is one of the earliest and best-known techniques in this category. Word2Vec uses the continuous bag-of-words (CBOW) or the continuous skip-gram model architecture. In CBOW, one tries to predict the current word in a text from a window of surrounding context words. In contrast, in the skip-gram model, one tries to predict the surrounding context words using the current word. Mikolov *et al.* (2013) also formalized the idea of using vector operation, such as $vec(\text{“}Madrid\text{”}) - vec(\text{“}Spain\text{”}) + vec(\text{“}France\text{”}) \approx vec(\text{“}Paris\text{”})$. GloVe (Pennington *et al.*, 2014) aims at taking the best of the count-based and prediction-based methods, with a first attempt to integrate both global and local statistics. Pennington *et al.* (2014) find that the quality of GloVe's learned representations is slightly better than Word2Vec's vectors, but it depends on the task at hand. A more recent method is fastText (Bojanowski *et al.*, 2017). It incorporates subword information into the learning process such that words not observed in the training corpus (out-of-vocabulary) can still be assigned a word vector. The current state of the art in word embeddings are the deep neural network Bidirectional Encoder Representations from Transformers (BERT) models and its variants (Devlin *et al.*, 2018). These models most explicitly integrate global and local context. For example, the word vector for "right" in "I am right" and "I take a right turn" will be different.

Estimated word embeddings are used as an input to more traditional sentiment classification methods (e.g. logistic regression), or to probabilistic methods such as the one proposed by

[11] Word embeddings are an advanced way of doing text vectorization, compared to, for instance, the simpler construction of a document-term matrix.

Taddy (2015b). Alternatively, by selecting several known positive and negative seed words, the vector space can be used to pinpoint words adjacent to those seed words and consider them as carrying the same polarity. The SENTPROP method from Hamilton *et al.* (2016) first constructs a lexical graph from a VSM with the words connected according to their embedding using cosine similarity, and then performs label propagation to define the polarity. The sentiment score of a word is proportional to the probability of a random walk hitting that word, as propagated starting from a seed set. To obtain confidence bands around the scores, they bootstrap over random subsets of seed words.

In the same vein as for lexicons, learning algorithms are ideally adapted for specific domains and languages to optimize the sentiment quantification. Thus, for optimal accuracy, the analysis for a specific domain needs a separate annotated dataset, as opposed to using an annotated broad corpus and the resulting generic trained algorithm. Transfer learning is the strand that investigates the optimal conversion of methods in one domain or one language to another. Good transfer learning minimizes the burden on the researcher to acquire equally informative domain-specific annotated corpora for all domains of interest. An application of transfer learning is to deduce sentence-level sentiment from document-level sentiment labels. Täckström and McDonald (2011) use hidden conditional random fields as a latent variable structure model to deduce the latent sentence-level sentiment.

2.6.2 Audio and Visual Data

Some of the tools discussed for textual sentiment computation are also of value for the extraction of sentiment from audio, and visual data. A lexicon can be constructed with entries such as “light smile”, “big smile”, “eye contact”, “crying”, “shouting”, “high pitch”, or “low pitch”, all with a certain calibrated polarity, and the number of seconds the action is held as a measure of polarity strength.

Domain specificity can be thought of as speaker specificity in the context of audio data. Speaker-dependent approaches give (much) better results than speaker-independent approaches (Poria *et al.*, 2016). The number of possible speakers is almost always larger than the number of possible languages or domains, making it infeasible to develop a specific algorithm for every individual speaker. However, making algorithms for types of speakers (e.g. political speakers) makes sense and is achievable.

Rousseeuw *et al.* (2018) define a measure of directional outlyingness that is applied on image data to detect (sudden) changes in how a video frame appears relative to another frame. A transformed aggregation of the various outlyingness measures would make a good candidate as a proxy for sentiment.

2.7 Aggregation of Sentiment Variables

Most researchers are not interested in an entity's or a data unit's sentiment at one specific point in time but in the average value on several moments, or across many entities, methods, and data sources. Therefore, appropriate aggregation is required.

2.7.1 Within-Unit

An essential aspect of the sentiment quantification as discussed in Section 2.6 is within-unit aggregation. For textual data, this becomes within-document or intratextual aggregation. Within-document aggregation is the weighting of the document-level sentiment information (e.g. the sentiment of a word or of a sentence) into a score that represents sentiment for that document. For visual data, this becomes, for instance, within-video aggregation, which consists of the aggregation of sentiment of the different segments of the video into a whole video sentiment score.

A widely used weighting scheme, in preprocessing and for text aggregation, is the tf-idf statistic. This scheme weighs terms based on their frequency of occurrence ("tf"), but revalues upward the words appearing across few documents ("idf"), under the idea that less frequent terms can be of greater value to detect the specificity of a document. This weighting approach makes document specificity a function of term use rather than term meaning. Another option is to weight based on reader's attention, which could be assumed higher in the beginning and end of a text. Allee and DeAngelis (2015) find an important degree of dispersion of sentiment in financial disclosures. Documents have typically one dominant sentiment class but no uniform sentiment across paragraphs or sentences. Boudt and Thewissen (2019), for example, show a clearly U-shaped pattern of sentiment within CEO letters.

Poria *et al.* (2016) outline two approaches to aggregating, or *fusing*, textual, audio, and visual signals, which happens when dealing with video material. A first strategy is to combine characteristics from every type of data into a joint vector and use this vector as input in a classification algorithm. The second strategy is to model sentiment individually per data stream, and then combine the unimodal results based on suitable metrics and weighting. The dynamic weighting of the unimodal results is an interesting research issue to explore. Pham *et al.* (2018) propose a third strategy, closest related to the first strategy, aiming at a joint multimodal representation. They use an unsupervised encoder-decoder framework but admit that a unimodal textual approach led to the best overall results in their empirical video analysis.

2.7.2 Cross-Sectional

Cross-sectional aggregation can occur at multiple levels. A first level is across documents at a given frequency, which results in a time series. This across-document aggregation is the natural next step after within-document aggregation. For example, to obtain a weekly

time series, all sentiment scores need to be aggregated at a weekly frequency. An interesting possibility for the aggregation is to let the weights depend on the articles' reach (e.g. the number of reads). One can then decide to further adjust the weights based on some empirical knowledge—for example, to cope with the underrepresentation of far-right voters on social media, as suggested in Ceron *et al.* (2014).

A second level is across documents for a given metadata marker. For instance, one could aggregate sentiment scores for all documents coming from a given source, or discussing a certain entity. Only considering a limited number of sources to measure sentiment for a given period risks to give a biased estimate due to an unrepresentative sample. Typically, the first and the second level are combined to obtain a time series for a given metadata occurrence. Many of such combinations capture different dynamics of the corpus and its metadata. Borovkova *et al.* (2017) obtain weekly sentiment values by a weighting that takes into account the relevance and novelty scores supplied by the Thomson Reuters News Analytics database.

A third possible level of cross-sectional aggregation is across sentiment methods. The order of when to do this aggregation depends on the goal. In the simplest scenario, only one method is used, or multiple methods are kept side by side—meaning, no across-method aggregation at all. Another simple scenario is to average the sentiment scores from any given number of methods to obtain an averaged sentiment score. Boudt *et al.* (2018) take the centered average of the scores coming from the lexicons they apply. A more statistical approach is commonality extraction, using principal component analysis or latent factor modeling. Rogers *et al.* (2011) define sentiment as the first principal component over a range of sentiment measures. Last, an objective-based approach optimally weighs different methods based on their relationship with a target variable or based on another quantifiable objective. We further develop the techniques, problems, and open questions regarding the last two approaches in Section 2.8.

2.7.3 Across-Time

Across-time aggregation aims to smooth obtained sentiment time series or, more generally, to infuse a certain time dependency pattern. There are various valid reasons for smoothing. One of those is to remove outliers. This especially holds for short-term sentiment series, for example at a daily frequency. Thorsrud (2020) applies a 60-day moving average to his daily tone-adjusted textual topic time series to filter out the noise. Another motivation for smoothing is related to the belief that sentiment at a certain time usually also partly reflects earlier sentiment. Sentiment needs to be updated when new information arrives but remains affected by previous information. Ardia *et al.* (2019b), for example, use beta weighting schemes covering a large number of possible time dynamics. They use a data-driven calibration to deal with the problem of not knowing in advance which time pattern has the most value for forecasting. The Kalman filter is also an appropriate technique to smooth out sentiment time series. It can be used to retrieve the unobserved sentiment state from the observed (already

aggregated) sentiment variable. Borovkova *et al.* (2017) employ a simple local level state space model, leading to significantly less noisy sentiment variables. Shapiro *et al.* (2018) use a monthly fixed effect as time series sentiment indicator, controlling for newspaper and article type fixed effects.

2.7.4 Across Variables or Proxies

The combination of the likely heterogeneity in the input data, the number of variables that can be associated to the data, and the number of sentiment implementations and aggregations may give rise to many constructed sentiment time series. For instance, Gelper and Croux (2010) use a one-factor model, estimated either as the first principal component or using partial least squares, to form an aggregate sentiment indicator from 160 sentiment proxies. In Ardia *et al.* (2019b), the different sentiment variables are weighted and assembled into a sentiment-based index using the elastic net regression. The obtained sentiment index is specific to the dependent variable used in the regression. Aggregation here is thus across metadata as well, which is usually not done at the across-document level. For example, to measure the sentiment around the economy, one may want to obtain this sentiment as a weighted average of several components such as employment, production, and the business cycle. Borovkova *et al.* (2017) obtain a final aggregated weekly sentiment index as an average of sentiment indices about important financial institutions, weighted by a bank-related measure such as net debt.

In a multivariate setting, one can repeat this process of creating separate sentiment indices for a series of proxies and then aggregate across these sentiment time series to obtain a final sentiment measure. That measure ought to be the optimized representation of the latent variable that is assumed to be represented by the collection of proxies. An example of a latent variable is the reputation of a company, which depends on observable variables such as profitability, market share, stock returns, and sustainability. Simplicity in weighting might be desired (e.g. equal weighting), but more complex (aggregation) schemes deserve to be studied. Larsen and Thorsrud (2019) use the marginal likelihoods across predictive regression models to form weights aggregating text-based time series into an index that best captures the variable to predict. Going forward, the idea of forecast combination could be useful for across-proxy aggregation.

Nimark and Pitschner (2019) define two interesting aggregated measures based on topic probabilities coming from a probabilistic topic model. The first is topic-specific deviation of a certain news topic ("specialization"); the second measures the news homogeneity in terms of agreement which topic is deemed most important. Empirically, they use the measures to show that different news sources emphasize different topics, but major events make news coverage more homogeneous. Similar measures could be constructed to test for the sentiment agreement across various sources.

Creating interactions of sentiment time series with other variables allows testing their interplay in explaining a dependent variable. The joint assessment of sentiment and topics is most prevalent in the literature (see e.g. the sentiment-adjusted topic measures of Thorsrud, 2020, or the context-specific sentiment time series in Calomiris and Mamaysky, 2019). Calomiris and Mamaysky (2019) and Glasserman and Mamaysky (2019) use an entropy-based measure to characterize a collection of news during a given time frame in terms of "unusualness" and create simple interaction terms with sentiment variables aggregated at the same frequency. These interaction terms add information, allowing one, for example, to uncover that negative unusual news leads to an increase in U.S. stock market volatility (Glasserman and Mamaysky, 2019). Boudt *et al.* (2018) assess the interaction of sentiment with various company variables (finding that the informativeness of sentiment depends on the level of information asymmetry), while Arslan-Ayaydin *et al.* (2016) interact sentiment with managerial compensation (finding that the informativeness of sentiment depends on the incentives to manipulate the sentiment). García (2013) interacts a measure based on the *New York Times* news with a dummy variable to indicate a recession and concludes that daily stock returns are better predicted during recessions.

2.8 Modeling

This section is mainly approached as the problem of modeling an outcome variable Y as a function of the sentiment variables stored in a matrix $\mathbf{S}$, and possibly a number of control variables in another matrix $\mathbf{X}$. It can very generally be seen as modeling the joint density function $f(Y, \mathbf{S}, \mathbf{X})$.

2.8.1 Time Series Models

A very simple setup exists in modeling the output variable with a small number of sentiment variables and possibly other explanatory variables through a linear regression. Simple means it can be solved with ordinary least squares (OLS) regression. Penalized, or regularized, regression is required when OLS regression cannot be applied—that is, when the number of explanatory variables is too high relative to the sample size, or when there is a severe problem of multicollinearity. Regularization of a high-dimensional variables set shrinks the coefficients of the least informative variables toward zero. The Ridge (Hoerl and Kennard, 1970) and the LASSO (Tibshirani, 1996) approaches are the most common ways to specify the penalized regression. The elastic net regularization of Zou and Hastie (2005) embeds both the Ridge and the LASSO.

Factor models extract one or more latent common patterns among a set of time series. Thorsrud (2020) develops a mixed-frequency time-varying dynamic factor model from which he extracts a daily news-based coincident index of business cycles. Both the mixed-frequency

and (dynamic) factor aspects are useful approaches. For the first, for example, sentiment aggregated at both weekly and quarterly frequency could be fed through a mixed-frequency factor model to obtain a short-term, a long-term and an overall trend. Similarly, grouped datasettings can be used to extract common sentiment in groups of time series—for example, a common factor for every industry group consisting of all firms' sentiment measures. Andreou *et al.* (2019) derive asymptotics to identify common and group-specific factors in such a setting. Specifically, they introduce a test to assess which factors are common across a set of group-specific vectors.

The news-based measure from Manela and Moreira (2017) is an estimate from an SVM regression using the VIX index as dependent variable and normalized n-gram counts from texts as independent variables. This is a valid way to create a final optimized index—that is, to let an index be constructed from how well it captures a target variable. However, using such sentiment proxies in a second-stage regression usually has an impact on the uncertainty surrounding the then estimated coefficients. Manela and Moreira (2017) adjust the standard errors around the eventual point estimates to account for the uncertainty that is introduced by the first-stage regression.

Many target variables of interest could be discrete—for instance, an indicator variable whether a month lies in a recession period or not. Regularization is also perfectly applicable in a nonlinear context. Pure machine learning algorithms, such as SVM, neural networks, or Random Forest, are more relevant in a nonlinear setup, also applicable in case of time series variables.

Multiple sentiment variables and target variables can be jointly modeled in a multivariate regression framework, such as vector autoregression (VAR) models (see Qin, 2011 for a historical development of VAR models, and Lütkepohl, 2017 for a survey on structural VAR models). These frameworks are in general less prone to identification issues, since the variables are treated as endogenous, unless when explicitly considered exogenous or not modeled.

2.8.2 Generative Models

One can distinguish between two key econometric approaches to measuring sentiment (Gentzkow *et al.*, 2019a). Sentiment is either seen as a function of the written text (sentiment $=$ $f(\text{text})$), or the written text is seen as a function of the underlying sentiment ($\text{text} = f(\text{sentiment})$). In the latter case, sentiment can be considered as a parameter of a stochastic process that generates texts as realizations. A seminal research paper in this is field is by Blei *et al.* (2003) proposing the latent Dirichlet allocation (LDA) model. Under this model, documents are assumed to be random mixtures over a predefined number of latent topics, where each topic is characterized by a distribution over words. Fitting such a model on a corpus of texts allows studying topic prevalence (the proportion of a document devoted to a topic) and topic content (the words used to discuss a topic).

Blei and Lafferty (2006) come up with a dynamic topic model that allows the content of the topics to change over time. Blei and Lafferty (2007) extend the LDA model by making correlation across topic proportions possible. Roberts *et al.* (2016) develop a structural topic model that lets the discovery of topics be a function of both word counts and observable covariates. These covariates can consist of sentiment variables, or metadata such as author and time of publication. The generative paradigm in a sentiment context thus starts from a statistical model that should be viewed as the source of all statements generated. For example, a model can be set up in which tokens are hypothesized to follow a generative model conditioned on a sentiment variable.

Taddy (2013b) introduces a framework to obtain low-dimensional document representations rich in sentiment information, called multinomial inverse regression (MNIR). He defines sentiment as the observable variables (e.g. product rating or whether a text is positive or not) impacting the composition of text data. Hence, his approach clearly follows the "text $= f($sentiment$)$" assumption. The most probable sentiment output can be associated with any unseen text using forward regression. Taddy (2015a) extends the MNIR framework to also account for potentially larger dimensions of the sentiment variables, referred to as distributed multinomial regression (DMR).

2.8.3 Combining Time Series Models and Joint Generative Models

Given the natural role that topics play as metadata features, the joint generative modeling of topics and sentiment is very useful, especially when a time series perspective is included. The dynamic topic model framework of Blei and Lafferty (2006) can be deemed a time series generalization of the topic models proposed earlier. Eguchi and Lavrenko (2006) address both the topic and sentiment of a text unit using probabilistic generative modeling. Every statement is considered to have a set of topic-bearing and a set of sentiment-bearing words, each coming from respectively an underlying topic and sentiment language model. The dependence between both models is explicitly taken into account, under the assumption that sentiment depends on the topic. This assumption is, for example, supported by the importance of domain-specific sentiment lexicons.

Lin and He (2009) jointly extract document-level sentiment and the mixture of topics using an unsupervised procedure. They go from the three-layered LDA (topics associated with documents, and words associated with topics) to their joint sentiment/topic (JST) model, having four layers (sentiment labels associated to documents, topics associated with sentiment labels, and words associated with sentiment labels and topics). The joint sentiment and topic modeling answers to the need for domain specificity of sentiment analysis. It generally is approached as a two-stage process: first the detection of topics, then the assignment of sentiment labels.

He *et al.* (2013) and Fu *et al.* (2015) further develop two related joint sentiment-topic models that allow selected dynamic parameters to account for the time variation in topics and sentiment. The inclusion of external variables makes it easier to interpret the driving processes behind discourse and content of qualitative material. In the approach of Gentzkow *et al.* (2019b) to measure trends in the degree of polarization in political speech, one can, for instance, include observed and unobserved speaker-specific characteristics.

There does not seem to be any longitudinal approach that uses the current state of a set of external variables (e.g. representing the economic and financial markets) as drivers for the time variation of the used sentiment and topics in written media articles. Such a holistic parametric model has, however, clear advantages in terms of econometric inference about the relationship between the observed news coverage, the features of the news sources, and the time variation in the variables system.

2.8.4 Normal and Abnormal Sentiment

There are several modeling approaches to decomposing sentiment into a normal and an abnormal component. Huang *et al.* (2014) distinguish between normal tone and abnormal tone, defining abnormal tone as the residual of a regression of tone on firm-specific characteristics. Ardia *et al.* (2019a) make the same distinction. They similarly consider a regression approach but use a static observable factor model, more precisely a market-cap weighted sentiment index, with abnormal tone also the residual. Other alternatives could be to use the residuals of a simple mean model or of a latent factor model.

Hubert and Labondance (2018) identify sentiment as the unpredictable component of lexicon-based textual tone, orthogonal to a series of variables representing economic fundamentals. In other words, they define sentiment as the soft information conveyed through the tone of a communication beyond traditional quantitative and qualitative information conveyed through the content. Sentiment is obtained as the residual, with its first-order autoregressive component removed, from a regression on the variables representing the fundamental content.

2.8.5 Attribution Analysis for Model Interpretation

Interpretation is strongly tied to the problem definition and generally qualitative. On the statistical side, we point out attribution analysis to interpret measured, nowcasted, and forecasted sentiment.

Sentiment aggregation and modeling condenses a lot of information into a few quantitative sentiment representations of interest. A natural question is then how much of the final value is explained by the input data. Obtaining such a decomposition of the final value into the contributions of the component input data is the purpose of a top-down attribution analysis. These constituents are weighted based on their relationship with a target variable and thus

allows studying the relative importance of every constituent or of groups of constituents. This in fact is a more fine-grained approach to doing sentiment decomposition, though typically not model based. Aggregation based on the metadata features allows obtaining a predefined decomposition of the relevant sentiment and may help with identifying the underlying sentiment drivers in relation to a target variable. Because of the linearity of the aggregation performed in Ardia *et al.* (2019b), the attribution to any of the aggregation dimensions could be easily obtained. For example, they attribute the full sentiment-based forecast of the U.S. industrial production growth to six clusters of separate economic topics. The aggregate news index from Thorsrud (2020) can also be decomposed in terms of topic contribution. An interesting avenue to explore is to do the same attribution to various news sources and bring this into relation to how readers are exposed to these sources and their potential media biases. Larsen *et al.* (2020) analyze the variation in attribution by looking at the proportion of attribution that is unchanged for model updates up to 60 months in the future. During the global financial crisis, the predictive attribution relationship turned out to be much less stable, with only a small proportion of the explanatory news variables remaining important. This speaks in favor of doing regular model reestimations when times are troubling to incorporate the relevant news. Calomiris and Mamaysky (2019) also detect strongly time-varying coefficient estimates for news measures when forecasting the stock market. This is due to both the changing mix of the news sources as well as the actual impact of the news. Interestingly, Larsen and Thorsrud (2018) find that narratives mostly go viral during downs in the business cycle, albeit for a duration of only a few months.

In case of multivariate economic systems, impulse response functions in the vector autoregression (VAR) framework are usually used for interpretation. An impulse response function describes a variable's evolution along a specified time horizon after a shock in the regression system. When a meaningful sentiment shock is infused, its impact on all other variables can be quantified and understood across time. Borovkova *et al.* (2017) analyze the impact of a one standard deviation change in sentiment on various macroeconomic variables and find it to last significantly up to two months later.

2.9 Validation

The entire workflow is about extracting sentiment variables from qualitative data and using those variables in an economic analysis. Validation takes place at the end of every step but can be broken down into four categories: (1) evaluation of the quality and selection of the data, (2) evaluation of the sentiment quantification and aggregation, (3) model estimation and hypothesis testing, and (4) evaluation of the out-of-sample statistical and economic performance of the model-based predictions.

Many choices in the econometric analysis of textual, audio, and visual sentiment remain *ad hoc*. To adequately gauge the presence and impact of sentiment, the entire analysis

should be frequently validated in a problem-specific way, both quantitatively and qualitatively. Comprehensive validation combines tools from econometrics with tools from machine learning. Machine learning is mostly about accuracy of prediction; econometrics is about uncovering (causal) relationships between economic variables.[12] Validation essentially jointly tests the current step and all previous steps as to whether they satisfy the assumptions for correct further (econometric) analysis.

When a sentiment variable does not seem to have a significant effect on the variable of interest, it may be due to two things. Either there is no significant effect of sentiment, or there is a significant effect, but the sentiment variables used are a weak proxy for real sentiment and do not capture the significant relationship. This can be conceived as a "joint hypothesis" problem. In order to mitigate this problem, the validation in the field of sentometrics is largely twofold. First, one should validate the sentiment variables created and then the model. When a model is deemed adequate in a statistical sense, further validation includes the interpretation of the results. A sentiment-based model that cannot be interpreted is not useful to convincingly answer the question outlined.

2.9.1 Data Quality and Data Selection

Since textual, audio, and visual data arrive in raw formats, the quality can vary substantially. Chances are not all data units are fully cleaned even after preprocessing. Data quality checking is an iterative process. It is natural to go back to the cleaning and selection when some errors are found a few steps further in the workflow.

A basic quality check asks whether everything necessary for analysis is present. For instance, to be able to do a time series analysis, time stamps are inevitable. Any preprocessing of data involve a clear trade-off between simplifying the data and information loss. Denny and Spirling (2018) document the sensitivity of textual preprocessing choices on the outcome of an unsupervised analysis. They devise a scoring and regression approach to quantify this sensitivity.

Validation of the data quality and its selection exists in minimizing the exposure to the limitations described in Section 2.4.3 or acknowledging them going forward. Ideally, the selected data are maximally spread out across relevant data sources. If there are several major broadcasters but data for only one is available, there is a severe risk of bias when generalizing any obtained results from this restricted dataset, as opposed to being only interested in and sticking with the conclusions of the particular data source studied.

Directing the analysis of audio data via speech-to-text to a textual analysis brings up the question of how trustworthy the conversion was. It is important to treat every transformation

[12] Advancements in machine learning and econometrics have been going more hand in hand. An interesting example is "double" or "orthogonal" machine learning, a development that aims to deal with the invalidity of inference infused by many machine learning methods (see mainly Chernozhukov *et al.*, 2017 and related work).

step and its possible errors as such, not confusing the textual data for the actual source audio data.

The data should be controlled for duplicates or near duplicates. If the duplicated data entries come from a different source, the content has likely been consumed more widely. A way to omit duplication but still maintain the implications it has is to add a metadata component that counts the number of duplicated occurrences. Wang *et al.* (2014) provide a (technical) overview with different techniques useful for duplicate detection.

2.9.2 Sentiment Quantification and Aggregation

The quantification of sentiment is highly important because it provides the numbers that any further step and interpretation is based on.

Relying on machine learning to train sentiment classifiers works under the assumption that the annotated dataset is a faithful representation of the actual sentiment. Not every annotation procedure leads to a reliable annotation set. The quality of the gold standard can be measured by the level of inter-annotator agreement using, for instance, Cohen's kappa. To measure the effectiveness of a sentiment classifier or a lexicon, one has to compare the model-generated scores with the gold standard. More precisely, the trade-off between precision (the proportion of predicted positives that is correct) and recall (the proportion of actual positives that is found) is at stake.[13] Recall and precision extend easily from a two-class problem (e.g. positive sentiment versus negative sentiment) to a multiclass setting doing micro or macro averaging (see e.g. Zhang and Zhou, 2014).

Every lexicon tends to undergo one or more rounds of expert-based checks, to explicitly classify words into positive or negative, delete irrelevant words, and correct obvious mistakes. The validity of individual entries of lexicons are thus still mainly evaluated by humans. Overall, lexicons should undergo the same level of scrutiny as any other sentiment computation method in terms of validation. It should be tested if the accuracy of domain-specific lexicons is higher than generic lexicons. Loughran and McDonald (2011) use careful inspection of frequently occurring words as the only basis to create their alternative word lists. To validate this procedure, they relate tone computed from their negative lexicon to filing period excess stock returns, finding this sentiment measure to be in general more significant than tone based on the generic Harvard dictionary negative lexicon. The approach of Labille *et al.* (2017) compares a set domain lexicons on other domain texts. If the domain-specific lexicon is well constructed, it should rank first in terms of accuracy for the domain it is designed for. Apart from accuracy levels, another simple comparison procedure is an ANOVA analysis to see which lexicon's score variability is best captured by human coders. When the lexicon is generated with a

[13] The precision and recall metrics can be combined in the F_β-score, with $F_\beta \equiv (1+\beta^2)\frac{\text{precision}\times\text{recall}}{\beta^2\times\text{precision}+\text{recall}}$. The β factor defines the relative level of importance put on recall. If $\beta = 1$, both metrics are weighted equally in a harmonic mean sense.

regression, one looks at fit or information criteria statistics to validate the overall power of a lexicon (e.g. Pröllochs *et al.*, 2015). An imbalance between positive and negative entries might make sense from a domain-specific perspective but should be defendable, since bias in the sentiment quantification algorithm can also be due to a biased training set. The Loughran & McDonald dictionary, for instance, is left with the large proportion of 78% negative words as a result of the domain adaptation to financial disclosures.

When creating sentiment measures, a first and simple analysis is to determine the correlation with existing related sentiment time series. All EPU indices of Baker *et al.* (2016) were validated using a very diligent human audit process, showing that the computer-generated indices are highly correlated with the human-generated ones. Soo's (2018) media sentiment housing index correlates strongly with the University of Michigan Survey of Consumers, albeit lagging. She further validates her index by confirming a reasonably strong lagged correlation with a multifactor index that combines multiple proxies, constructed based on the methodology of Baker and Wurgler (2006). The most difficult task can end up to find related proxies, as sometimes they are rare or do not exist at all. Another simple time series validation procedure is what is referred to as event validation. This entails visualizing a sentiment measure and confirming whether sharp increases or drops coincide with the incidence of important events that intuitively would result in a strong increase or decrease in sentiment, respectively.

2.9.3 Econometric Modeling and Interpretation

Many models are evaluated by measuring the accuracy in an out-of-sample prediction exercise. However, prediction is not always of interest; measuring which words and how they convey sentiment can be a more important objective that is not always related to prediction accuracy. As mentioned in Justin Grimmer's comment on Taddy (2013b), how to do trustworthy task-specific sentiment evaluation still needs to be formalized. How is one to know with a high degree of confidence whether a token can be attributed to a particular sentiment feature? This problem can be particularly apparent when doing a large-dimensional regression of a sentiment variable on unigrams, for instance, with the resulting coefficients of the unigrams not always easy to interpret and sensitive to change across different specifications.

The problem of extensive and problem-specific validation is brought up in detail in Grimmer and Stewart (2013). For supervised methods, validation is fairly straightforward; it boils down to minimizing the prediction error in replicating a set of annotated outputs or maximizing the classification accuracy (typically making use of confusion matrices). A dataset is best divided into a training, a validation, and a testing set to avoid a biased view on accuracy due to overfitting (Varian, 2014). Alternatively, one can do k-fold cross-validation or rolling forecasting origin cross-validation when dealing with time series. Unsupervised methods require combining experimental, substantive, and statistical evidence to show the conceptual validity of a model output. Proper validation of unsupervised models is especially important

when used for inference or measurement rather than prediction or exploration (Roberts *et al.*, 2016).

2.9.3.1 Model Estimation and Hypothesis Testing

It is common to evaluate the in-sample goodness of fit of a sentiment-based regression model with the (adjusted) R^2 statistic. When adding their word flow measures, Calomiris and Mamaysky (2019) find a substantial increase in the R^2 for predicting returns, volatility, and drawdown risk. The main concerned parameters are those associated with the sentiment variables. Their significance should be assessed statistically and economically. Statistical significance shows whether an effect exists, but its applicability is mainly limited to low-dimensional models. Gandomi and Haider (2015) review various issues of doing econometrics in a big data environment, pointing out the "irrelevance of statistical significance." Economic significance inspects the sign and strength of the association. Economic meaning can be given through, for instance, an attribution analysis.

In general, textual, audio, and visual data bring the known endogeneity challenges to econometricians. The creation and publication of texts, videos, and speeches is correlated with many factors, so positing a cause and effect remains dangerous when no further insights into the (many) underlying factors of the data are available. Is it the sentiment of the alternative dataset that is at the heart of a certain correlation or causality, or is the sentiment a reflection of associated underlying factors? Does the sentiment impact the outcome variable directly or indirectly, and through what mechanism? Larsen and Thorsrud (2018) partition their network of sentiment/topic variables into more and less exogenous variables. Variables are considered exogenous if they have predictive power for other topics but are not (often) predicted themselves. The most exogenous variables seem to be associated with economic fundamentals. Hubert and Labondance (2018) correct for endogeneity in their central bank tone measure by stripping away fundamentals, expectations of future fundamentals, standard monetary shocks, investor sentiment, and past sentiment shocks. Benhabib and Spiegel (2019) deal with endogeneity and, more specifically, reverse causality using instrumental variables. They use political data to instrument for differences in (survey-based) sentiment levels by state. When testing the effect of sentiment on the target variable and finding significant results, it is recommended to also test the effect of the target variable on sentiment via a reverse (lagged) regression specification. Few research papers on sentiment have carried forward this robustness step.

Model uncertainty is assessed through analyzing the impact of sentiment parameter estimates across various model specifications. This has to do with both a good and exhaustive definition of the control variables $\mathbf{X}$ and with testing for enough different model structures. Soo (2018) creates robustness variables from the qualitative data themselves. She computes indices from those news articles that convey fundamental market information rather than sentiment, adds those to her regression specifications, and finds that her major sentiment

index remains significant. Varian (2014) states it is important "to be explicit about examining how parameter estimates vary with respect to choices of control variables and instruments." Validation is rarely a black-and-white matter. The researcher should identify when and how sentiment is informative and when it is not.

2.9.3.2 Out-of-Sample Evaluation

An out-of-sample version of the R^2 statistic can be used to measure the relative reduction or increase in the mean square out-of-sample prediction error of a sentiment-based forecasting strategy with respect to a baseline strategy. Using the out-of-sample R^2, Adämmer and Schüssler (2020) document statistically significant increased predictive power of the monthly U.S. equity premium when using news-aggregated variables combined with a model-switching strategy. Caporin and Poli (2017) use five metrics (mean absolute error, mean square error, heteroskedasticity adjusted mean square error, QLIKE loss function, and R^2 of Mincer-Zarnowitz forecasting regressions) to compare the forecasting performance of a news-based realized volatility model versus a baseline.

The magnitude of the impact of sentiment variables on economic and financial variables is highly subject to time variation. Stability needs to be tested by performing the analysis, and measuring the performance, on various subsamples, or by doing rolling forward regressions.

A simple way to sidestep the issue of endogeneity is by comparing existing linear models to models enhanced with the quantified alternative data sources and to simply focus on whether predictive power improves or existing (significant) relationships hold. This could be formulated as testing models "controlling for sentiment." The model confidence set procedure from Hansen *et al.* (2011) allows testing whether different model specifications are truly different according to some significance level.

2.10 Software

This section points to a selection of useful software tools to carry out a detailed sentometrics analysis from textual data.[14] The selection is by no means exhaustive—meaning, there exist plenty of other software tools equally of use to perform (parts of) a sentometrics analysis. We

[14] A well-known open-source software tool for audio data processing is openSMILE (Eyben *et al.*, 2013). The LVA software (`https://lva650.com`) can be used for preprocessing, deconstruction, and immediate emotion analysis of audio data (see Mayew and Venkatachalam, 2012 for an application in finance). For visual data processing, alternatives are the commercial softwares OKAO Vision System from OMRON (`https://plus-sensing.omron.com/technology`) or Luxand FaceSDK (`https://www.luxand.com/facesdk`), both mainly for facial features extraction. A good commercial speech-to-text technology is Vocapia (`https://www.vocapia.com`). In the open-source sphere, the DeepSpeech project (Hannun *et al.*, 2014) and associated software packages are very useful. Generally, the outputs returned by the above tools can be easily loaded into any programming environment to perform the remaining steps in the analysis.

limit ourselves to the open-source R and Python programming environments, due to their large popularity, strong communities of developers, and relatively gradual learning curves. For instance, MATLAB rarely comes to mind for doing textual analysis, but its Text Analytics Toolbox has many capabilities for doing powerful preprocessing, vectorization, sentiment analysis, and topic modeling. One does not necessarily have to choose one programming environment. Like Glasserman and Mamaysky (2019), a common workflow includes doing the handling of textual, audio, and visual data in Python, and the statistical analysis in R. The available software is linked to specific tasks involved in an econometric analysis of qualitative sentiment data and is summarized in Table 2.1.

The **quanteda** package (Benoit *et al.*, 2018) is a general text mining toolkit in R. Its development has been actively supported by the European Commission. The package **tidytext** (Silge and Robinson, 2016) can also be used to do many text processing tasks, following "tidy" data principles. The **tm** package (Feinerer *et al.*, 2008) is an older textual analysis framework, but is still used as a backend in many text-related R packages.[15]

The **NLTK** library (Bird *et al.*, 2009), short for Natural Language Toolkit, is the text mining toolkit counterpart in Python, albeit even more exhaustive.[16] In Python, the **spaCy** library (Honnibal and Montani, 2017) is the most complete alternative. It is faster, but more black box. The **TextBlob** library (Loria, 2019) is built on the **NLTK** library and is therefore more specialized as to what concerns several textual extractions, such as sentiment analysis through machine learning classification.

The R package **sentometrics** (Ardia *et al.*, 2020) provides a collection of functions to do sentiment computation, sentiment aggregation, and (high-dimensional) sentiment-based regression. Wischnewsky *et al.* (2019) use the package to create a "Sentoindex" that represents financial stability sentiment as expressed during testimonies at U.S. Congressional hearings. The sentiment computation in **sentometrics** is lexicon based, but other sentiment scores can be used as input for further aggregation. The **sentometrics** package also provides a simple keywords-based approach to generating metadata features. The **SentimentAnalysis** package (Feuerriegel and Pröllochs, 2019) can be used to create lexicons and compute sentiment according to the method of Pröllochs *et al.* (2015).

The regression framework in **sentometrics** relies on both the **caret** (Kuhn, 2018) and **glmnet** (Friedman *et al.*, 2010) packages but is specific to the sentiment time series generated within the package. The **glmnet** package implements various penalized regressions; the **caret** package provides more generic classification and regression modeling. The inverse text regression methods developed by Taddy (2013b) and Taddy (2015a) are available in the

[15] A helpful starting point to explore the plethora of textual analysis tools in R is CRAN's Task View "NaturalLanguageProcessing" (`https://CRAN.R-project.org/view=NaturalLanguageProcessing`).

[16] The **quanteda** package website gives an overview of the actual functions across the packages referred to perform several specific tasks (see `https://quanteda.io/articles/pkgdown/comparison.html`).

R package **textir** (Taddy, 2018).[17] The **rJST** package (Boiten, 2019) implements the joint sentiment/topic model of Lin and He (2009).[18]

Python's **scikit-learn** (Pedregosa *et al.*, 2011) is one if its most established machine learning libraries. It supports the majority of the common learning algorithms used in sentiment analysis and is easy to use with respect to feature engineering. To do the same, but also particularly deep learning, Google's **TensorFlow** library (Abadi *et al.*, 2016) is the standard, albeit imposing more on the user in terms of setting up the individual components of a chosen model. Since recently, there also exists a comprehensive R interface to the **TensorFlow** framework.

These days, research papers also go increasingly accompanied with standalone open-source replication code (see e.g. the MATLAB code used in Thorsrud, 2020).[19] Another example, to do sentiment analysis benchmarking, is the online tool iFeel 2.0 (Araújo *et al.*, 2016) based on Ribeiro *et al.* (2016).

A shortcoming of the current software landscape is that there are no libraries that propose a full and easy integration of the required data handling, machine learning, and econometric tools. The preprocessing and sentiment quantification packages have very little in common with the packages used for modeling. Having to combine too many packages or even multiple programming languages is prone to error, for instance, due to the usage of different types of object classes that need to be converted.

[17] The DMR from Taddy (2015a) is also implemented with the programming language Julia, available at `https://github.com/AsafManela/HurdleDMR.jl`, which mainly includes the Hurdle Distributed Multiple Regression algorithm from Kelly *et al.* (2019).

[18] The package seems to have been removed from CRAN, but the source code is still available at `https://github.com/maxboiten/rJST`.

[19] The code is available at `https://github.com/leifandersthorsrud/NCI`.

Table 2.1: Nonexhaustive overview of textual data analysis tools in R and Python. The abbreviation ML stands for machine learning. A tick indicates that the software can be directly or indirectly (i.e. by minimally chaining with other available tools) used to perform a particular workflow step. The packages included for R are **caret** (Kuhn, 2018), **glmnet** (Friedman *et al.*, 2010), **quanteda** (Benoit *et al.*, 2018), **rJST** (Boiten, 2019), **SentimentAnalysis** (Feuerriegel and Pröllochs, 2019), **sentometrics** (Ardia *et al.*, 2020), **textir** (Taddy, 2018), **tidytext** (Silge and Robinson, 2016), and **tm** (Feinerer *et al.*, 2008). For Python, the libraries tabulated are **NLTK** (Bird *et al.*, 2009), **scikit-learn** (Pedregosa *et al.*, 2011), **spaCy** (Honnibal and Montani, 2017), **TensorFlow** (Abadi *et al.*, 2016), and **TextBlob** (Loria, 2019).

Tasks	**Restructuring**			**Sentiment quantification**		**Time series**		**Econometric analysis**	
Software	Cleaning	Metadata	Tokens	Lexicon-based	ML	Aggregation	Visualization	Regression	Validation
R									
caret					✓			✓	✓
glmnet					✓			✓	✓
quanteda	✓	✓	✓	✓	✓				
rJST		✓			✓				
SentimentAnalysis				✓	✓				
sentometrics		✓		✓		✓	✓	✓	✓
textir		✓						✓	✓
tidytext	✓	✓	✓				✓		
tm	✓	✓	✓	✓					
Python									
NLTK	✓	✓	✓	✓	✓			✓	
scikit-learn	✓	✓	✓		✓			✓	✓
spaCy	✓	✓	✓						
TensorFlow					✓			✓	✓
TextBlob	✓	✓	✓		✓				

2.11 Concluding Remarks

Sentiment analysis allows us to accurately and automatically map alternative data into quantitative statistics as a support for decision making across many business applications. Economists and investors, and also politicians and journalists, have started to embrace the utilization of econometric methods in the analysis and application of textual, audio, and visual data, to understand historical evolutions and better forecast future evolutions.

We overview the emerging field of sentometrics that investigates the transformation of qualitative data into quantitative sentiment variables, and their subsequent application in an econometric analysis of the relationships between sentiment and other variables. This survey is organized around the different steps of a typical analysis. The most important terminology is collected in the glossary.

Textual, audio, and visual data will continue to become more cheaply and widely available, together with becoming more easily accessible. The interest of public and private institutions to monetize these data and their proprietary data will grow as well. We recommend further research on multimodal sentiment analysis in econometrics. The future will be exceedingly multimedia in terms of content generated, hence the analysis indispensably multimodal. A major challenge is the development of appropriate technology for unified multimodal sentiment analysis systems.

Progress toward better integrated and more reproducible sentiment data research will require collaborative cross-disciplinary efforts. We end this paper with a call for more efforts toward reproducibility in the econometric study of sentiment from qualitative data. It would benefit greatly from reference data and associated state-of-the-art performance, for different sentiment quantification techniques, data, and econometric approaches. In the field of computer science, such practices are more widespread. Other researchers can evaluate any new approach on the reference data and as such provide a consistent picture of reproducibility or improved performance. Even though the sharing of code and data has gained adoption, there are yet no standard practices on how to do so. The reference data and results should be made available through an open database with easy access and well-documented formats. This matches with the proposition of Lacy *et al.* (2015) to set up a standard scholarly repository to share research-related materials. As a companion to this survey paper, we have therefore set up a collaborative econometrics and sentiment GitHub project to gather such resources.[20]

[20] See `https://sborms.github.io/econometrics-meets-sentiment`.

A Computational Framework to Compute, Aggregate and Predict with Textual Sentiment 3

Abstract

We provide a hands-on introduction to optimized textual sentiment indexation using the R package **sentometrics**. Textual sentiment analysis is increasingly used to unlock the potential information value of textual data. The **sentometrics** package implements an intuitive framework to efficiently compute sentiment scores of numerous texts, to aggregate the scores into multiple time series, and to use these time series to predict other variables. The workflow of the package is illustrated with a built-in corpus of news articles from two major U.S. journals to forecast the CBOE Volatility Index.

3.1 Introduction

Individuals, companies, and governments continuously consume written material from various sources to improve their decisions. The corpus of texts is typically of a high-dimensional longitudinal nature requiring statistical tools to extract the relevant information. A key source of information is the sentiment transmitted through texts, called *textual sentiment*. Algaba *et al.* (2020a) review the notion of sentiment and its applications, mainly in economics and finance. They define sentiment as "the disposition of an entity toward an entity, expressed via a certain medium." The medium in this case is texts. The sentiment expressed through texts may provide valuable insights on the future dynamics of variables related to firms, the economy, political agendas, product satisfaction, and marketing campaigns, for instance. Still, textual sentiment does not live by the premise to be equally useful across all applications. Deciphering when, to what degree, and which layers of the sentiment add value is needed to consistently study the full information potential present within qualitative communications. The econometric approach of constructing time series of sentiment by means of optimized selection and weighting of textual sentiment is referred to as *sentometrics* by Algaba *et al.* (2020a) and Ardia *et al.* (2019b). The term sentometrics is a composition of (textual) sentiment analysis and (time series) econometrics.

The release of the R (R Core Team, 2019) text mining infrastructure **tm** (Feinerer *et al.*, 2008) over a decade ago can be considered the starting point of the development and popularization of textual analysis tools in R. A number of successful follow-up attempts at improving the speed and interface of the comprehensive natural language processing capabilities provided by **tm** have been delivered by the packages **openNLP** (Hornik, 2016), **cleanNLP** (Arnold, 2017), **quanteda** (Benoit *et al.*, 2018), **tidytext** (Silge and Robinson, 2016), and **qdap** (Rinker, 2017).

The notable tailor-made packages for sentiment analysis in R are **meanr** (Schmidt, 2019), **SentimentAnalysis** (Feuerriegel and Pröllochs, 2019), **sentimentr** (Rinker, 2018), and **syuzhet** (Jockers, 2017). Many of these packages rely on one of the larger above-mentioned textual analysis infrastructures. The **meanr** package computes net sentiment scores fastest, but offers no flexibility.[21] The **SentimentAnalysis** package relies on a similar calculation as used in **tm**'s sentiment scoring function. The package can additionally be used to generate and evaluate sentiment dictionaries. The **sentimentr** package extends the polarity scoring function from the **qdap** package to handle more difficult linguistic edge cases, but is therefore slower than packages which do not attempt this. The **SentimentAnalysis** and **syuzhet** packages also become comparatively slower for large input corpora. The **quanteda** and **tidytext** packages have no explicit sentiment function but their toolsets can be used to construct one.

[21] In Appendix 7.2, we provide an illustrative comparison of the computation time for various lexicon-based sentiment calculators in R, including the one from the **sentometrics** package. The appendix and the replication script `run_timings.R` are also available on our package's GitHub repository in the `appendix` folder.

Our R package **sentometrics** proposes a well-defined modeling workflow, specifically targeted at studying the evolution of textual sentiment and its impact on other quantities. It can be used (i) to *compute* textual sentiment, (ii) to *aggregate* fine-grained textual sentiment into various sentiment time series, and (iii) to *predict* other variables with these sentiment measures. The combination of these three facilities leads to a flexible and computationally efficient framework to exploit the information value of sentiment in texts. The package presented in this paper therefore addresses the present lack of analytical capability to extract time series intelligence about the sentiment transmitted through a large panel of texts.

Furthermore, the **sentometrics** package positions itself as both integrative and supplementary to the powerful text mining and data science toolboxes in the R universe. It is integrative, as it combines the strengths of **quanteda** and **stringi** (Gagolewski, 2020) for corpus construction and manipulation. It uses **data.table** (Dowle and Srinivasan, 2019) for fast aggregation of textual sentiment into time series, and **glmnet** (Friedman *et al.*, 2010) and **caret** (Kuhn, 2018) for (sparse) model estimation. It is supplementary, given that it easily extends any text mining workflow to compute, aggregate and predict with textual sentiment.

The remainder of the paper is structured as follows. Section 3.2 introduces the methodology behind the R package **sentometrics**. Section 3.3 describes the main control functions and illustrates the package's typical workflow. Section 3.4 applies the entire framework to forecast the Chicago Board Options Exchange (CBOE) Volatility Index. Section 3.5 concludes.

3.2 Use Cases and Workflow

The typical use cases of the R package **sentometrics** are the fast computation and aggregation of textual sentiment, the subsequent time series visualization and manipulation, and the estimation of a sentiment-based prediction model. The use case of building a prediction model out of textual data encompasses the previous ones.

We propose a modular workflow that consists of five main steps, *Steps 1–5*, from corpus construction to model estimation. All use cases can be addressed by following (a subset of) this workflow. The R package **sentometrics** takes care of all steps, apart from corpus collection and cleaning. However, various conversion functions and method extensions are made available that allow the user to enter and exit the workflow at different steps. Table 3.1 pieces together the key functionalities of **sentometrics** together with the associated functions and `S3` class objects. All steps are explained below. We minimize the mathematical details to clarify the exposition, and stay close to the actual implementation. Section 3.3 demonstrates how to use the functions.

Table 3.1: Taxonomy of the R package **sentometrics**. This table displays the functionalities of the **sentometrics** package along with the associated functions and `S3` output objects. We explain the (generic) R(-style) methods in Appendix 7.3.

Functionality	Functions	Output
1. Corpus management		
(a) Creation	`sento_corpus()`	*sento_corpus*
(b) Manipulation	**quanteda** corpus functions (e.g. `docvars()`, `corpus_sample()`, or `corpus_subset()`), `as.data.frame()`, `as.data.table()`, `as.sento_corpus()`	
(c) Features generation	`add_features()`	
(d) Summarization	`corpus_summarize()`, `print()`	
2. Sentiment computation		
(a) Lexicon management	`sento_lexicons()`	*sento_lexicons*
(b) Computation	`compute_sentiment()`	*sentiment*
(c) Manipulation	`merge()`, `as.sentiment()`	
(d) Summarization	`peakdocs()`	
3. Sentiment aggregation		
(a) Specification	`ctr_agg()`	
(b) Aggregation	`sento_measures()`, `aggregate()`	*sento_measures*
(c) Manipulation	`subset()`, `merge()`, `diff()`, `scale()`, `as.data.frame()`, `as.data.table()`, `measures_fill()`, `measures_update()`	
(d) Visualization	`plot()`	
(e) Summarization	`summary()`, `peakdates()`, `print()`, `nobs()`, `nmeasures()`, `get_dimensions()`, `get_dates()`	
4. Modeling		
(a) Specification	`ctr_model()`	
(b) Estimation	`sento_model()`	*sento_model*, *sento_modelIter*
(c) Prediction	`predict()`	
(d) Diagnostics	`summary()`, `print()`, `get_loss_data()`, `attributions()`	*attributions*
(e) Visualization	`plot()`	

3.2.1 Pre-Process Texts and Generate Relevant Features (Step 1)

We assume the user has a corpus of texts of any size at its disposal. The data can be scraped from the web, retrieved from news databases, or obtained from any other source. The texts should be cleaned such that graphical and web-related elements (e.g. HTML tags) are removed. To benefit from the full functionality of the **sentometrics** package, a minimal requirement is that every text has a timestamp and an identifier. This results in a set of documents $d_{n,t}$ for $n = 1, \ldots, N_t$ and time points $t = 1, \ldots, T$, where N_t is the total number of documents at time t. If the user has no interest in an aggregation into time series, desiring to do only sentiment calculation, the identifiers and especially the timestamps can be dropped. The corpus can also be given a language identifier, for a sentiment analysis across multiple languages at once. The identifier is used to direct the lexicons in the different languages to the right texts.

Additionally, *features* have to be defined and mapped to the documents. Features can come in many forms: news sources, entities (individuals, companies or countries discussed in the texts), or text topics. The mapping implicitly permits subdividing the corpus into many smaller groups with a common interest. Many data providers enrich their textual data with information that can be used as features. If this is not the case, topic modeling or entity recognition techniques are valid alternatives. Human classification or manual keyword(s) occurrence searches are simpler options. The extraction and inclusion of features is an important part of the analysis and should be related to the variable that is meant to be predicted.

The texts and features have to be structured in a rectangular fashion. Every row represents a document that is mapped to the features through numerical values $w_{n,t}^k \in [0, 1]$ where the features are indexed by $k = 1, \ldots, K$. The values are indicative of the relevance of a feature to a document. Binary values indicate which documents belong to which feature(s).

This rectangular data structure is turned into a *`sento_corpus`* object when passed to the `sento_corpus()` function. The reason for this separate corpus structure is twofold. It controls whether all corpus requirements for further analysis are met (specifically, dealing with timestamps and numeric features), and it allows performing operations on the corpus in a more structured way. If no features are of interest to the analysis, a dummy feature valued $w_{n,t}^k = 1$ throughout is automatically created. The `add_features()` function is used to add or generate new features, as will be shown in the illustration. When the corpus is constructed, it is up to the user to decide which texts have to be kept for the actual sentiment analysis.

3.2.2 Sentiment Computation and Aggregation (Steps 2 and 3)

Overall, in the sentiment computation and aggregation framework, we define three weighting parameters: ω, θ and b. They control respectively the within-document, across-document, and across-time aggregation. Section 3.3.3 explains how to set the values for these parameters. Appendix 7.4 gives a overview of the implemented formulae for weighting.

3.2.2.1 Compute Document- or Sentence-Level Textual Sentiment (Step 2)

Every document requires at least one sentiment score for further analysis. The **sentometrics** package can be used to assign sentiment using the *lexicon*-based approach, possibly augmented with information from *valence shifters*. The sentiment computation always starts from a corpus of documents. However, the package can also automatically decompose the documents into sentences and return sentence-level sentiment scores. The actual computation of the sentiment follows one of the three approaches explained below. Alternatively, one can align own sentiment scores with the **sentometrics** package making use of the `as.sentiment()` and `merge()` functions.

The lexicon-based approach to sentiment calculation is flexible, transparent, and computationally convenient. It looks for words (or unigrams) that are included in a predefined word list of polarized (positive and negative) words. The package benefits from built-in word lists in English, French, and Dutch, with the latter two mostly as a checked web-based translation. The **sentometrics** package allows for three different ways of doing the lexicon-based sentiment calculation. These procedures, though simple at their cores, have proven efficient and powerful in many applications. In increasing complexity, the supported approaches are:

(i) A **unigrams** approach. The most straightforward method, where computed sentiment is simply a (weighted) sum of all detected word scores as they appear in the lexicon.

(ii) A **valence-shifting bigrams** approach. The impact of the word appearing before the detected word is evaluated as well. A common example is 'not good', which under the default approach would get a score of 1 ('good'), but now ends up, for example, having a score of -1 due to the presence of the negator 'not.'

(iii) A **valence-shifting clusters** approach. Valence shifters can also appear in positions other than right before a certain word. We implement this layer of complexity by searching for valence shifters (and other sentiment-bearing words) in a cluster of at maximum four words before and two words after a detected polarized word.

In the first two approaches, the sentiment score of a document $d_{n,t}$ (d in short) is the sum of the adjusted sentiment scores of all its unigrams. The adjustment comes from applying weights to each unigram based on its position in the document and adjusting for the presence of a valence shifting word. This leads to:

$$s_{n,t}^{\{l\}} \equiv \sum_{i=1}^{Q_d} \omega_i v_i s_{i,n,t}^{\{l\}}, \tag{3.1}$$

for every lexicon $l = 1, \ldots, L$. The total number of unigrams in the document is equal to Q_d. The score $s_{i,n,t}^{\{l\}}$ is the sentiment value attached to unigram i from document $d_{n,t}$, based on

lexicon l. It equals zero when the word is not in the lexicon. The impact of a valence shifter is represented by v_i, being the shifting value of the *preceding* unigram $i-1$. No valence shifter or the simple unigrams approach boils down to $v_i = 1$. If the valence shifter is a negator, typically $v_i = -1$. The weights ω_i define the within-document aggregation. The values ω_i and v_i are specific to a document $d_{n,t}$, but we omit the indices n and t for brevity.

The third approach differs in the way it calculates the impact of valence shifters. A document is decomposed into C_d clusters around polarized words, and the total sentiment equals the sum of the sentiment of each cluster. The expression (3.1) becomes in this case $s_{n,t}^{\{l\}} \equiv \sum_{J=1}^{C_d} s_{J,n,t}^{\{l\}}$. Given a detected polarized word, say unigram j, valence shifters are identified in a surrounding cluster of adjacent unigrams $J \equiv \{J^L, J^U\}$ around this word (irrespective of whether they appear in the same sentence or not), where $J^L \equiv \{j-4, j-3, j-2, j-1\}$ and $J^U \equiv \{j+1, j+2\}$. The resulting sentiment value of cluster J around associated unigram j is $s_{J,n,t}^{\{l\}} \equiv n_N(1+\max\{0.80(n_A - 2n_A n - n_D), -1\})\omega_j s_{j,n,t}^{\{l\}} + \sum_{m \in J^U} \omega_m s_{m,n,t}^{\{l\}}$. The number of amplifying valence shifters is n_A, those that deamplify are counted by n_D, $n = 1$ and $n_N = -1$ if there are is odd number of negators, else $n = 0$ and $n_N = 1$.[22] All n_A, n_D, n and n_N are specific to a cluster J. The unigrams in J are first searched for in the lexicon, and only when there is no match, they are searched for in the valence shifters word list. Clusters are non-overlapping from one polarized word to the other; if another polarized word is detected at position $j+4$, then the cluster consists of the unigrams $\{j+3, j+5, j+6\}$. This clusters-based approach borrows from how the R package **sentimentr** does its sentiment calculation. Linguistic intricacies (e.g. sentence boundaries) are better handled in their package, at the expense of being slower.

In case of a clusters-based sentence-level sentiment calculation, we follow the default settings used in **sentimentr**. This includes, within the scope of a sentence, a cluster of 5 words (not 4 as above) before and 2 words after the polarized word, limited to occurring commas. A fourth type of valence shifters, adversative conjunctions (e.g. however), is used to reweight the first expression of $\max\{\cdot, -1\}$ by $1 + 0.25 n_{AC}$, where n_{AC} is the difference between the number of adversative conjunctions within the cluster before and after the polarized word.

The scores obtained above are subsequently multiplied by the feature weights to spread out the sentiment into lexicon- and feature-specific sentiment scores, as $s_{n,t}^{\{l,k\}} \equiv s_{n,t}^{\{l\}} w_{n,t}^{k}$, with k the index denoting the feature. If the document does not correspond to the feature, the value of $s_{n,t}^{\{l,k\}}$ is zero.

[22] Amplifying valence shifters, such as 'very', strengthen a polarized word. Deamplifying valence shifters, such as 'hardly', downtone a polarized word. The strengthening value of 0.80 is fixed and acts, if applicable, as a modifier of 80% on the polarized word. Negation inverses the polarity. An even number of negators is assumed to cancel out the negation. Amplifiers are considered (but not double-counted) as deamplifiers if there is an odd number of negators. Under this approach, for example, the occurrence of 'not very good' receives a more realistic score of -0.20 ($n = 1$, $n_N = -1$, $n_A = 1$, $n_D = 0$, $s = +1$), instead of -1.80, in many cases too negative.

In **sentometrics**, the `sento_lexicons()` function is used to define the lexicons and the valence shifters. The output is a *sento_lexicons* object. Any provided lexicon is applied to the corpus. The sentiment calculation is performed with `compute_sentiment()`. Depending on the input type, this function outputs a `data.table` with all values for $s_{n,t}^{\{l,k\}}$. When the output can be used as the basis for aggregation into time series in the next step (that is, when it has a `"date"` column), it becomes a *sentiment* object. To do the computation at sentence-level, the argument `do.sentence = TRUE` should be used. The `as.sentiment()` function transforms a properly structured table with sentiment scores into a *sentiment* object.

3.2.2.2 Aggregate the Sentiment into Textual Sentiment Time Series (Step 3)

In this step, the purpose is to aggregate the individual sentiment scores and obtain various representative time series. Two main aggregations are performed. The first, across-document, collapses all sentiment scores across documents within the same frequency (e.g. day or month, as defined by t) into one score. The weighted sum that does so is:

$$s_t^{\{l,k\}} \equiv \sum_{n=1}^{N_t} \theta_n s_{n,t}^{\{l,k\}}. \tag{3.2}$$

The weights θ_n define the importance of each document n at time t (for instance, based on the length of the text). The second, across-time, smooths the newly aggregated sentiment scores over time, as:

$$s_u^{\{l,k,b\}} \equiv \sum_{t=t_\tau}^{u} b_t s_t^{\{l,k\}}, \tag{3.3}$$

where $t_\tau \equiv u - \tau + 1$. The time weighting schemes $b = 1, \ldots, B$ go with different values for b_t to smooth the time series in various ways (e.g. according to an upward sloping exponential curve), with a time lag of τ. The first $\tau - 1$ observations are dropped from the original time indices, such that $u = \tau, \ldots, T$ becomes the time index for the ultimate time series. This leaves $N \equiv T - \tau + 1$ time series observations.

The number of obtained time series, P, is equal to L (number of lexicons) $\times$ K (number of features) $\times$ B (number of time weighting schemes). Every time series covers one aspect of each dimension, best described as "the textual sentiment as computed by lexicon l for feature k aggregated across time using scheme b." The time series are designated by $s_u^p \equiv s_u^{\{l,k,b\}}$ across all values of u, for $p = 1, \ldots, P$, and the triplet $p \equiv \{l,k,b\}$. The ensemble of time series captures both different information (different features) and the same information in different ways (same features, different lexicons, and aggregation schemes).

The entire aggregation setup is specified by means of the `ctr_agg()` function, including the within-document aggregation needed for the sentiment analysis. The `sento_measures()` function performs both the sentiment calculation (via `compute_sentiment()`) and time

series aggregation (via `aggregate()`), outputting a *`sento_measures`* object. The obtained sentiment measures in the *`sento_measures`* object can be further aggregated across measures, also with the `aggregate()` function.

3.2.3 Specify Regression Model and Obtain Predictions (Step 4)

The sentiment measures are now regular time series variables that can be applied in regressions. In case of a linear regression, the reference equation is:

$$y_{u+h} = \delta + \boldsymbol{\gamma}^\top \mathbf{x}_u + \beta_1 s_u^1 + \ldots + \beta_p s_u^p + \ldots + \beta_P s_u^P + \varepsilon_{u+h}. \tag{3.4}$$

The target variable y_{u+h} is often a variable to forecast, that is, $h > 0$. Let $s_u \equiv (s_u^1, \ldots, s_u^P)^\top$ encapsulate all textual sentiment variables as constructed before, and $\boldsymbol{\beta} \equiv (\beta_1, \ldots, \beta_P)^\top$. Other variables are denoted by the vector $\mathbf{x}_u$ at time u and $\boldsymbol{\gamma}$ is the associated parameter vector. Logistic regression (binomial and multinomial) is available as a generalization of the same underlying linear structure.

The typical large dimensionality of the number of predictors in (3.4) relative to the number of observations, and the potential multicollinearity, both pose a problem to ordinary least squares (OLS) regression. Instead, estimation and variable selection through a penalized regression relying on the elastic net regularization of Zou and Hastie (2005) is more appropriate. As an example, Joshi *et al.* (2010) and Yogatama *et al.* (2011) use regularization to predict movie revenues, and scientific article downloads and citations, respectively, using many text elements such as words, bigrams, and sentiment scores. Ardia *et al.* (2019b) similarly forecast U.S. industrial production growth based on a large number of sentiment time series extracted from newspaper articles.

Regularization, in short, shrinks the coefficients of the least informative variables toward zero. It consists of optimizing the least squares or likelihood function including a penalty component. The elastic net optimization problem for the specified linear regression is expressed as:

$$\min_{\tilde{\delta},\tilde{\boldsymbol{\gamma}},\tilde{\boldsymbol{\beta}}} \left\{ \frac{1}{N} \sum_{u=\tau}^{T} (y_{u+h} - \tilde{\delta} - \tilde{\boldsymbol{\gamma}}^\top \tilde{\mathbf{x}}_u - \tilde{\boldsymbol{\beta}}^\top \tilde{s}_u)^2 + \lambda \left[\alpha \left\| \tilde{\boldsymbol{\beta}} \right\|_1 + (1-\alpha) \left\| \tilde{\boldsymbol{\beta}} \right\|_2^2 \right] \right\}. \tag{3.5}$$

The tilde denotes standardized variables, and $\|.\|_p$ is the ℓ_p-norm. The standardization is required for the regularization, but the coefficients are rescaled back once estimated. The rescaled estimates of the model coefficients for the textual sentiment indices are in $\widehat{\boldsymbol{\beta}}$, usually a sparse vector, depending on the severity of the shrinkage. The parameter $0 \leq \alpha \leq 1$ defines the trade-off between the Ridge (Hoerl and Kennard, 1970), ℓ_2, and the LASSO (Tibshirani, 1996), ℓ_1, regularizaton, respectively for $\alpha = 0$ and $\alpha = 1$. The $\lambda \geq 0$ parameter defines the level of regularization. When $\lambda = 0$, the problem reduces to OLS estimation. The two

parameters are calibrated in a data-driven way, such that they are optimal to the regression equation at hand. The **sentometrics** package allows calibration through cross-validation, or based on an information criteria with the degrees of freedom properly adjusted to the elastic net context according to Tibshirani and Taylor (2012).

A potential analysis of interest is the sequential estimation of a regression model and obtaining all out-of-sample predictions. For a given sample size $M < N$, a regression is estimated with M observations and used to predict some next observation of the target variable. This procedure is repeated rolling forward from the first to the last M-sized sample, leading to a series of estimates. These are compared with the realized values to assess the (average) out-of-sample prediction performance.

The type of model, the calibration approach, and other modeling decisions are defined via the `ctr_model()` function. The (iterative) model estimation and calibration is done with the `sento_model()` function that relies on the R packages **glmnet** and **caret**. The user can define here additional (sentiment) values for prediction through the `x` argument. The output is a *`sento_model`* object (one model) or a *`sento_modelIter`* object (a collection of iteratively estimated *`sento_model`* objects and associated out-of-sample predictions).

A forecaster, however, is not limited to using the models provided through the **sentometrics** package—(s)he is free to guide this step to whichever modeling toolbox available, continuing with the sentiment variables computed in the previous steps.

3.2.4 Evaluate Performance and Sentiment Attributions (Step 5)

A *`sento_modelIter`* object carries an overview of out-of-sample performance measures relevant to the type of model estimated. Plotting the object returns a time series plot comparing the predicted values with the corresponding observed ones. A more formal way to compare the forecasting performance of different models, sentiment-based or not, is to construct a model confidence set (Hansen *et al.*, 2011). This set isolates the models that are statistically the best regarding predictive ability, within a confidence level. To do this analysis, one needs to first call the function `get_loss_data()` which returns a loss data `matrix` from a collection of *`sento_modelIter`* objects, for a chosen loss metric (like squared errors); see `?get_loss_data` for more details. This loss data `matrix` is ready for use by the R package **MCS** (Catania and Bernardi, 2017) to create a model confidence set.

The aggregation into textual sentiment time series is entirely linear. Based on the estimated coefficients $\widehat{\boldsymbol{\beta}}$, every underlying dimension's sentiment *attribution* to a given prediction can thus be computed easily. For example, the attribution of a certain feature k in the forecast of the target variable at a particular date is the weighted sum of the model coefficients and the values of the sentiment measures constructed from k. Attribution can be computed for all features, lexicons, time-weighting schemes, time lags, and individual documents. Through attribution, a prediction is broken down in its respective components. The attribution to documents is

useful to pick the texts with the most impact to a prediction at a certain date. The function `attributions()` computes all types of possible attributions.

3.3 The R Package sentometrics

In what follows, several examples show how to put the steps into practice using the **sentometrics** package.[23] The subsequent sections illustrate the main workflow, using built-in data, focusing on individual aspects of it. Section 3.3.1 studies corpus management and features generation. Section 3.3.2 investigates the sentiment computation. Section 3.3.3 looks at the aggregation into time series (including the control function `ctr_agg()`). Section 3.3.4 briefly explains further manipulation of a *sento_measures* object. Section 3.3.5 regards the modeling setup (including the control function `ctr_model()`) and attribution.

3.3.1 Corpus Management and Features Generation

The very first step is to load the R package **sentometrics**. We also load the **data.table** package as we use it throughout, but loading it is in general not required.

```
R> library("sentometrics")
R> library("data.table")
```

We demonstrate the workflow using the `usnews` built-in dataset, a collection of news articles from The Wall Street Journal and The Washington Post between 1995 and 2014.[24] It has a `data.frame` structure and thus satisfies the requirement that the input texts have to be structured rectangularly, with every row representing a document. The data are loaded below.

```
R> data("usnews", package = "sentometrics")
R> class(usnews)

[1] "data.frame"
```

For conversion to a *sento_corpus* object, the `"id"`, `"date"`, and `"texts"` columns have to come in that order. One could also add an optional `"language"` column for a multi-language sentiment analysis (see the multi-language sentiment computation example in the next section). All other columns are reserved for features, of type numeric. For this particular corpus, there are four original features. The first two indicate the news source, the latter two

[23] All computational details to replicate the examples are listed in Appendix 7.5.

[24] The data originate from https://www.figure-eight.com/data-for-everyone (under "Economic News Article Tone and Relevance").

the relevance of every document to the U.S. economy. The feature values $w_{n,t}^{k}$ are binary and complementary (when `"wsj"` is 1, `"wapo"` is 0; similarly for `"economy"` and `"noneconomy"`) to subdivide the corpus to create separate time series.

```
R> head(usnews[, -3])
```

```
         id       date wsj wapo economy noneconomy
1 830981846 1995-01-02   0    1       1          0
2 842617067 1995-01-05   1    0       0          1
3 830982165 1995-01-05   0    1       0          1
4 830982389 1995-01-08   0    1       0          1
5 842615996 1995-01-09   1    0       0          1
6 830982368 1995-01-09   0    1       1          0
```

To access the texts, one can simply do `usnews[["texts"]]` (i.e. the third column omitted above). An example of one text is:

```
R> usnews[["texts"]][2029]
```

```
[1] "Dow Jones Newswires NEW YORK -- Mortgage rates rose in the
past week after Fridays employment report reinforced the perception
that the economy is on solid ground, said Freddie Mac in its weekly
survey. The average for 30-year fixed mortgage rates for the week
ended yesterday, rose to 5.85 from 5.79 a week earlier and 5.41 a
year ago. The average for 15-year fixed-rate mortgages this week was
5.38, up from 5.33 a week ago and the year-ago 4.69. The rate for
five-year Treasury-indexed hybrid adjustable-rate mortgages, was
5.22, up from the previous weeks average of 5.17. There is no
historical information for last year since Freddie Mac began tracking
this mortgage rate at the start of 2005."
```

The built-corpus is cleaned for non-alphanumeric characters. To put the texts and features into a corpus structure, call the `sento_corpus()` function. If you have no features available, the corpus can still be created without any feature columns in the input `data.frame`, but a dummy feature called `"dummyFeature"` with a score of 1 for all texts is added to the *`sento_corpus`* output object.

```
R> uscorpus <- sento_corpus(usnews)
R> class(uscorpus)
```

```
[1] "sento_corpus" "corpus" "list"
```

The `sento_corpus()` function creates a *sento_corpus* object on top of the **quanteda**'s package *corpus* object. Hence, many functions from **quanteda** to manipulate corpora can be applied to *sento_corpus* objects as well. Take `quanteda::corpus_subset(uscorpus, date < "2014-01-01")`, which limits the corpus to all articles before 2014. The presence of the date document variable (the `"date"` column) and all other metadata as numeric features valued between 0 and 1 are the two distinguishing aspects between a *sento_corpus* object and any other corpus-like object in R. Having the date column is a requirement for the later aggregation into time series. The function `as.sento_corpus()` transforms a **quanteda** *corpus* object, a **tm** *SimpleCorpus* object or a **tm** *VCorpus* object into a *sento_corpus* object; see `?as.sento_corpus` for more details.

To round off *Step 1*, we add two metadata features using the `add_features()` function. The features `uncertainty` and `election` give a score of 1 to documents in which respectively the word `"uncertainty"` or `"distrust"` and the specified regular expression `regex` appear. Regular expressions provide flexibility to define more complex features, though it can be slow for a large corpus if too complex. Overall, this gives $K = 6$ features. The `add_features()` function is most useful when the corpus starts off with no additional metadata, i.e. the sole feature present is the automatically created `"dummyFeature"` feature.[25]

```
R> regex <- paste0("\\bRepublic[s]?\\b|\\bDemocrat[s]?
+    \\b|\\belection\\b|")
R> uscorpus <- add_features(uscorpus,
+    keywords = list(uncertainty = c("uncertainty", "distrust"),
+        election = regex),
+    do.binary = TRUE, do.regex = c(FALSE, TRUE))
R> tail(quanteda::docvars(uscorpus))

               date wsj wapo economy noneconomy uncertainty election
842616931 2014-12-22   1    0       1          0           0        0
842613758 2014-12-23   1    0       0          1           0        0
842615135 2014-12-23   1    0       0          1           0        0
842617266 2014-12-24   1    0       1          0           0        0
842614354 2014-12-26   1    0       0          1           0        0
842616130 2014-12-31   1    0       0          1           0        0
```

The `corpus_summarize()` function is useful to numerically and visually display the evolution of various parameters within the corpus.

[25] To delete a feature, use `quanteda::docvars(corpusVariable, field = "featureName") <- NULL`. The `docvars()` method is extended for a *sento_corpus* object—for example, if all current features are deleted, the dummy feature `"dummyFeature"` is automatically added.

```
R> summ <- corpus_summarize(uscorpus, by = "year")
R> summ$plots$feature_plot +
+    guides(color = guide_legend(nrow = 1))
```

FIGURE 3.1: Yearly evolution of the features presence across the corpus.

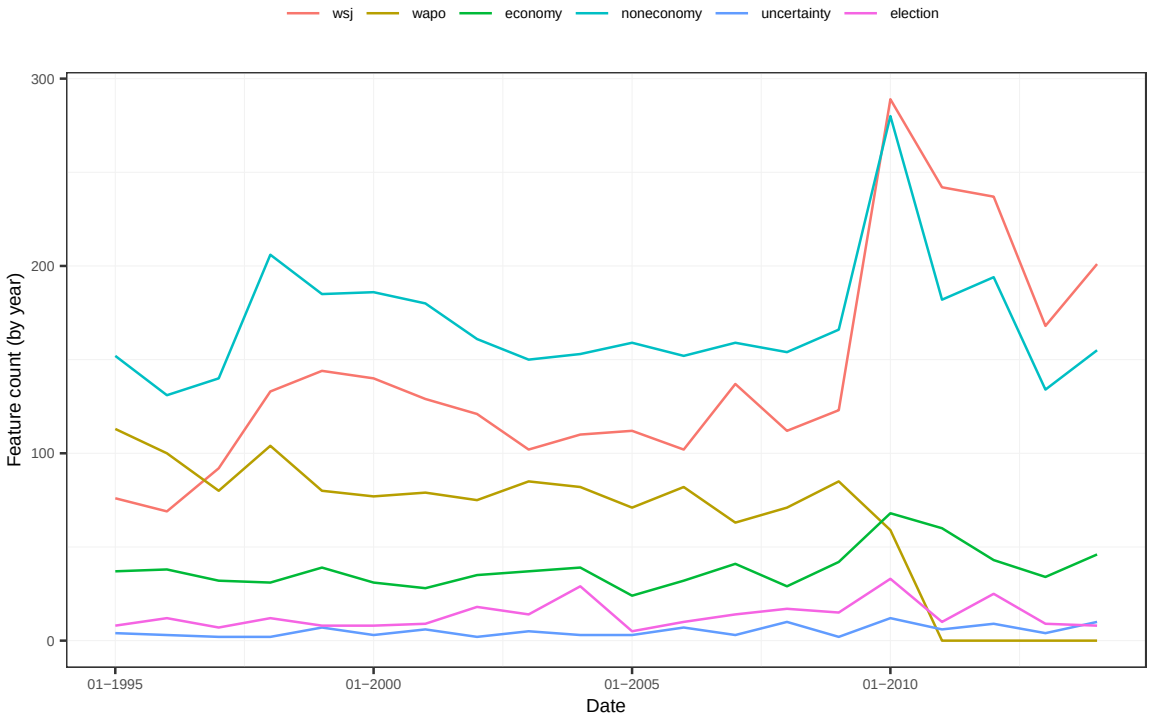


Figure 3.1 shows the counts of the corpus features per year.[26] The values are obtained by counting a feature for a document if it is not zero. Around 2010, the Wall Street Journal and non-economic news dominated the corpus.

3.3.2 Lexicon Preparation and Sentiment Computation

As *Step 2*, we calculate sentiment using lexicons. We supply the built-in Loughran & McDonald (Loughran and McDonald, 2011) and Henry (Henry, 2008) word lists, and the more generic Harvard General Inquirer word list. We also add six other lexicons from the R package **lexicon** (Rinker, 2019): the NRC lexicon (Mohammad and Turney, 2010), the Hu & Liu

[26] For more control over the plots, our replication script also loads the **ggplot2** (Wickham, 2016) and **gridExtra** (Auguie, 2017) packages.

lexicon (Hu and Liu, 2004a), the SentiWord lexicon (Baccianella *et al.*, 2010), the Jockers lexicon (Jockers, 2017), the SenticNet lexicon (Cambria *et al.*, 2016), and the SO-CAL lexicon (Taboada *et al.*, 2011).[27] This gives $L = 9$.

We pack these lexicons together in a named list, and provide it to the `sento_lexicons()` function, together with an English valence word list. The `valenceIn` argument dictates the complexity of the sentiment analysis. If `valenceIn = NULL` (the default), sentiment is computed based on the simplest unigrams approach. If `valenceIn` is a table with an `"x"` and a `"y"` column, the valence-shifting bigrams approach is considered for the sentiment calculation. The values of the `"y"` column are those used as v_i. If the second column is named `"t"`, it is assumed that this column indicates the type of valence shifter for every word, and thus it forces employing the valence-shifting clusters approach for the sentiment calculation. Three types of valence shifters are supported for the latter method: negators (value of `1`, defines n and n_N), amplifiers (value of `2`, counted in n_A), and deamplifiers (value of `3`, counted in n_D). Adversative conjunctions (value of `4`, counted in n_{AC}) are an additional type only picked up during a sentence-level calculation.

```
R> data("list_lexicons", package = "sentometrics")
R> data("list_valence_shifters", package = "sentometrics")
R> lexiconsIn <- c(list_lexicons[c("LM_en", "HENRY_en", "GI_en")],
+    list(NRC = lexicon::hash_sentiment_nrc,
+         HULIU = lexicon::hash_sentiment_huliu,
+         SENTIWORD = lexicon::hash_sentiment_sentiword,
+         JOCKERS = lexicon::hash_sentiment_jockers,
+         SENTICNET = lexicon::hash_sentiment_senticnet,
+         SOCAL = lexicon::hash_sentiment_socal_google))
R> lex <- sento_lexicons(lexiconsIn = lexiconsIn,
+    valenceIn = list_valence_shifters[["en"]])
```

The `lex` output is a *`sento_lexicons`* object, which is a list. Duplicates are removed, all words are put to lowercase and only unigrams are kept.

```
R> lex[["HENRY_en"]]
```

```
                x     y
  1:        above     1
  2:    accomplish     1
  3:  accomplished     1
```

[27] You can add any two-column table format as a lexicon, as long as the first column is of type character and the second column is numeric.

```
  4:  accomplishes     1
  5: accomplishing     1
 ---
185:         worse    -1
186:        worsen    -1
187:     worsening    -1
188:       worsens    -1
189:         worst    -1
```

The individual word lists themselves are `data.tables`, as displayed above.

Document-Level Sentiment Computation

The simplest way forward is to compute sentiment scores for every text in the corpus. This is handled by the `compute_sentiment()` function, which works with either a character vector, a *sento_corpus* object, a **quanteda** *corpus* object, a **tm** *SimpleCorpus* object, or a **tm** *VCorpus* object. The core of the sentiment computation is implemented in C++ through **Rcpp** (Eddelbuettel and Francois, 2011). The `compute_sentiment()` function has, besides the input corpus and the lexicons, other arguments. The main one is the `how` argument, to specify the within-document aggregation. In the example below, `how = "proportional"` divides the net sentiment score by the total number of tokenized words. More details on the contents of these arguments are provided in Section 3.3.3, when the `ctr_agg()` function is discussed. See below for a brief usage and output example.

```
R> sentScores <- compute_sentiment(usnews[["texts"]],
+    lexicons = lex, how = "proportional")
R> head(sentScores[, c("id", "word_count", "GI_en",
+    "SENTIWORD", "SOCAL")])
```

```
      id word_count     GI_en SENTIWORD   SOCAL
1: text1        213 -0.013146 -0.003436 0.20679
2: text2        110  0.054545  0.004773 0.15805
3: text3        202  0.004950 -0.006559 0.11298
4: text4        153  0.006536  0.021561 0.15393
5: text5        245  0.008163 -0.017786 0.08997
6: text6        212  0.014151 -0.009316 0.03506
```

For a character vector input, the `compute_sentiment()` function returns a `data.table` with an identifier column, a word count column and the computed sentiment scores for all lexicons, as no features are involved. When the input is a *sento_corpus* object, the

output is a *sentiment* object. The **tm** *SimpleCorpus* and *VCorpus* objects are treated as a character vector input. Both could be transformed into a *sento_corpus* object with the `as.sento_corpus()` function. A **tm** *VCorpus* object as input, for instance, thus leads to a plain `data.table` object, with a similar structure as above.

```
R> reuters <- system.file("texts", "crude", package = "tm")
R> tmVCorp <- tm::VCorpus(tm::DirSource(reuters),
+    list(reader = tm::readReut21578XMLasPlain))
R> class(compute_sentiment(tmVCorp, lex))

[1] "data.table" "data.frame"
```

In the example below, we exhibit the use of the `as.sento_corpus()` and `as.sentiment()` functions, showing how to integrate computations from another package into the workflow.

```
R> sentoSent <- compute_sentiment(as.sento_corpus(tmVCorp,
+    dates = as.Date("1993-03-06") + 1:20), lex, "UShaped")
R> tmSentPos <- sapply(tmVCorp, tm::tm_term_score,
+    lex$NRC[y > 0, x])
R> tmSentNeg <- sapply(tmVCorp, tm::tm_term_score,
+    lex$NRC[y < 0, x])
R> tmSent <- cbind(sentoSent[, 1:3],
+    "tm_NRC" = tmSentPos - tmSentNeg)
R> sent <- merge(sentoSent, as.sentiment(tmSent))
R> sent[6:9, c(1, 11:13)]

    id SENTICNET--dummyFeature SOCAL--dummyFeature tm_NRC--dummyFeature
1: 236                 0.1611              0.3927                    -2
2: 237                 0.0542              0.3441                    10
3: 242                 0.0025              0.1547                    -3
4: 246                 0.2203              0.2581                    -4
```

The `tmSent` table is appropriately formatted for conversion into a *sentiment* object. The `sent` object embeds the different scoring approaches and can be used for aggregation into time series following the remainder of the **sentometrics** workflow.

Sentence-Level Sentiment Computation

A sentiment calculation at sentence-level instead of the given corpus unit level requires to set `do.sentence = TRUE` in the `compute_sentiment()` function.

```
R> sSentences <- compute_sentiment(uscorpus, lex,
+    do.sentence = TRUE)
R> sSentences[1:11, 1:6]
```

```
           id sentence_id       date word_count LM_en--wsj LM_en--wapo
 1: 830981846           1 1995-01-02         12          0     0.08333
 2: 830981846           2 1995-01-02         17          0     0.00000
 3: 830981846           3 1995-01-02         13          0     0.00000
 4: 830981846           4 1995-01-02         19          0    -0.05263
 5: 830981846           5 1995-01-02         19          0    -0.15789
 6: 830981846           6 1995-01-02         40          0     0.00500
 7: 830981846           7 1995-01-02         49          0    -0.01633
 8: 830981846           8 1995-01-02         30          0     0.00000
 9: 830981846           9 1995-01-02         21          0     0.00000
10: 830981846          10 1995-01-02          8          0     0.00000
11: 842617067           1 1995-01-05         19          0     0.00000
```

The obtained sentence-level *sentiment* object can be aggregated into document-level scores using the `aggregate()` function with `do.full = FALSE`. The value `do.full = TRUE` (by default) aggregates the scores into sentiment time series. The aggregation across sentences from the same document is set through the `howDocs` aggregation parameter.

```
R> aggDocuments <- aggregate(sSentences,
+    ctr_agg(howDocs = "equal_weight"),
+    do.full = FALSE)
R> aggDocuments[1:2, 1:6]
```

```
          id       date word_count LM_en--wsj LM_en--wapo LM_en--economy
1: 830981846 1995-01-02        228    0.00000     -0.0277        -0.0277
2: 842617067 1995-01-05        122    0.06471      0.0000         0.0000
```

The output shows the aggregated sentiment scores for document with identifier `"830981846"` as a simple average of the sentiment scores of its ten sentences.

Multi-Language Sentiment Computation

For a sentiment analysis with multiple languages, the *sento_corpus* object needs to have a character `"language"` identifier column. The names should map to a named list of *sento_lexicons* objects to be applied to the different texts. The language information should be expressed in the different unique lexicon names. The values for the columns pertaining to a lexicon in another language than the document are set to zero.

```
R> usnewsLang <- usnews[1:5, 1:3]
R> usnewsLang[["language"]] <- c("fr", "en", "en", "fr", "en")
R> corpusLang <- sento_corpus(corpusdf = usnewsLang)
R> sLang <- compute_sentiment(corpusLang, list(
+    en = sento_lexicons(list("GI_en" = list_lexicons$GI_en)),
+    fr = sento_lexicons(list("GI_fr" = list_lexicons$GI_fr_tr))))
R> head(sLang)
```

```
          id       date word_count GI_fr--dummyFeature GI_en--dummyFeature
1: 830981846 1995-01-02        213            0.004695            0.000000
2: 842617067 1995-01-05        110            0.000000            0.054545
3: 830982165 1995-01-05        202            0.000000            0.004950
4: 830982389 1995-01-08        153            0.000000            0.000000
5: 842615996 1995-01-09        245            0.000000            0.008163
```

3.3.3 Creation of Sentiment Measures

To create sentiment time series, one needs a well-specified aggregation setup defined via the control function `ctr_agg()`. To compute the measures in one go, the `sento_measures()` function is to be used. Sentiment time series allow to use the entire scope of the package. We focus the explanation on the control function's central arguments and options, and integrate the other arguments in their discussion:

- `howWithin`: This argument defines how sentiment is aggregated within the same document (or sentence), setting the weights ω_i in (3.1). It is passed on to the `how` argument of the `compute_sentiment()` function. For binary lexicons and the simple unigrams matching case, the `"counts"` option gives sentiment scores as the difference between the number of positive and negative words. Two common normalization schemes are dividing the sentiment score by the total number of words (`"proportional"`) or by the number of polarized words (`"proportionalPol"`) in the document (or sentence). A wide number of other weighting schemes are available. They are, together with those for the next two arguments, summarized in Appendix 7.4.
- `howDocs`: This argument defines how sentiment is aggregated across all documents at the same date (or frequency), that is, it sets the weights θ_n in (3.2). The time frequency at which the time series have to be aggregated is chosen via the `by` argument, and can be set to daily (`"day"`), weekly (`"week"`), monthly (`"month"`) or yearly (`"year"`). The option `"equal_weight"` gives the same weight to every document, while the option `"proportional"` gives higher weights to documents with more words, relative to the document population at a given date. The `do.ignoreZeros` argument forces ignoring documents with zero sentiment in the computation of the across-document weights. By

default these documents are overlooked. This avoids the incorporation of documents not relevant to a particular feature (as in those cases $s_{n,t}^{\{l,k\}}$ is exactly zero, because $w_{n,t}^{k} = 0$), which could lead to a bias of sentiment toward zero.[28] When applicable, this argument also defines the aggregation across sentences within the same document.

- `howTime`: This argument defines how sentiment is aggregated across dates, to smoothen the time series and to acknowledge that sentiment at a given point is at least partly based on sentiment and information from the past. The `lag` argument has the role of τ dictating how far to go back. In the implementation, `lag = 1` means no time aggregation and thus $b_t = 1$. The `"equal_weight"` option is similar to a simple weighted moving average, `"linear"` and `"exponential"` are two options which give weights to the observations according to a linear or an exponential curve, `"almon"` does so based on Almon polynomials, and `"beta"` based on the Beta weighting curve from Ghysels *et al.* (2007). The last three curves have respective arguments to define their shape(s), being `alphasExp` and `do.inverseExp`, `ordersAlm` and `do.inverseAlm`, and `aBeta` and `bBeta`. These weighting schemes are always normalized to unity. If desired, user-constructed weights can be supplied via `weights` as a named `data.frame`. All the weighting schemes define the different b_t values in (3.3). The `fill` argument is of sizeable importance here. It is used to add in dates for which not a single document was available. These added, originally missing, dates are given a value of 0 (`"zero"`) or the most recent value (`"latest"`). The option `"none"` accords to not filling up the date sequence at all. Adding in dates (or not) impacts the time aggregation by respectively combining the latest *consecutive* dates, or the latest *available* dates.
- `nCore`: The `nCore` argument can help to speed up the sentiment calculation when dealing with a large corpus. It expects a positive integer passed on to the `setThreadOptions()` function from the **RcppParallel** package (Allaire *et al.*, 2019), and parallelizes the sentiment computation across texts. By default, `nCore = 1`, which indicates no parallelization. Parallelization is expected to improve the speed of the sentiment computation only for sufficiently large corpora, or when using many lexicons.
- `tokens`: Our unigram tokenization is done with the R package **stringi**; it transforms all tokens to lowercase, strips punctuation marks and strips numeric characters (see the internal function `sentometrics:::tokenize_texts()`). If wanted, the texts could be tokenized separately from the **sentometrics** package, using any desired tokenization setup, and then passed to the `tokens` argument. This way, the tokenization can be tailor-made (e.g. stemmed[29]) and reused for different sentiment computation function calls, for example to compare the impact of several normalization or aggregation choices for the same tokenized corpus. Doing the tokenization once for multiple subsequent

[28] It also ignores the documents which are relevant to a feature, but exhibit zero sentiment. This can occur if none of the words have a polarity, or the weighted number of positive and negative words offset each other.

[29] In this case, also stem the lexical entries before you provide these to the `sento_lexicons()` function.

computation calls is more efficient. In case of a document-level calculation, the input should be a list of unigrams per document. If at sentence-level (`do.sentence = TRUE`), it should be a list of tokenized sentences as a list of the respective tokenized unigrams.

In the example code below, we aggregate sentiment at a weekly frequency, choose a counts-based within-document aggregation, and weight the documents for across-document aggregation proportionally to the number of words in the document. The resulting time series are smoothed according to an equally-weighted and an exponential time aggregation scheme ($B = 2$), using a lag of 30 weeks. We ignore documents with zero sentiment for across-document aggregation, and fill missing dates with zero before the across-time aggregation, as per default.

```
R> ctrAgg <- ctr_agg(howWithin = "counts",
+    howDocs = "proportional",
+    howTime = c("exponential", "equal_weight"),
+    do.ignoreZeros = TRUE,
+    by = "week", fill = "zero", lag = 30, alphasExp = 0.2)
```

The `sento_measures()` function performs both the sentiment calculation in *Step 2* and the aggregation in *Step 3*, and results in a *sento_measures* output object. The generic `summary()` displays a brief overview of the composition of the sentiment time series. A *sento_measures* object is a list with as most important elements `"measures"` (the textual sentiment time series), `"sentiment"` (the original sentiment scores per document) and `"stats"` (a selection of summary statistics). Alternatively, the same output can be obtained by applying the `aggregate()` function on the output of the `compute_sentiment()` function, if the latter is computed from a *sento_corpus* object.

```
R> sentMeas <- sento_measures(uscorpus, lexicons = lex,
+    ctr = ctrAgg)
```

There are 108 initial sentiment measures (9 lexicons $\times$ 6 features $\times$ 2 time weighting schemes). An example of one created sentiment measure and its naming (each dimension's component is separated by `"--"`), is shown below.

```
R> as.data.table(sentMeas)[, c(1, 23)]

          date NRC--noneconomy--equal_weight
1: 1995-07-24                         3.916
2: 1995-07-31                         4.109
3: 1995-08-07                         3.945
```

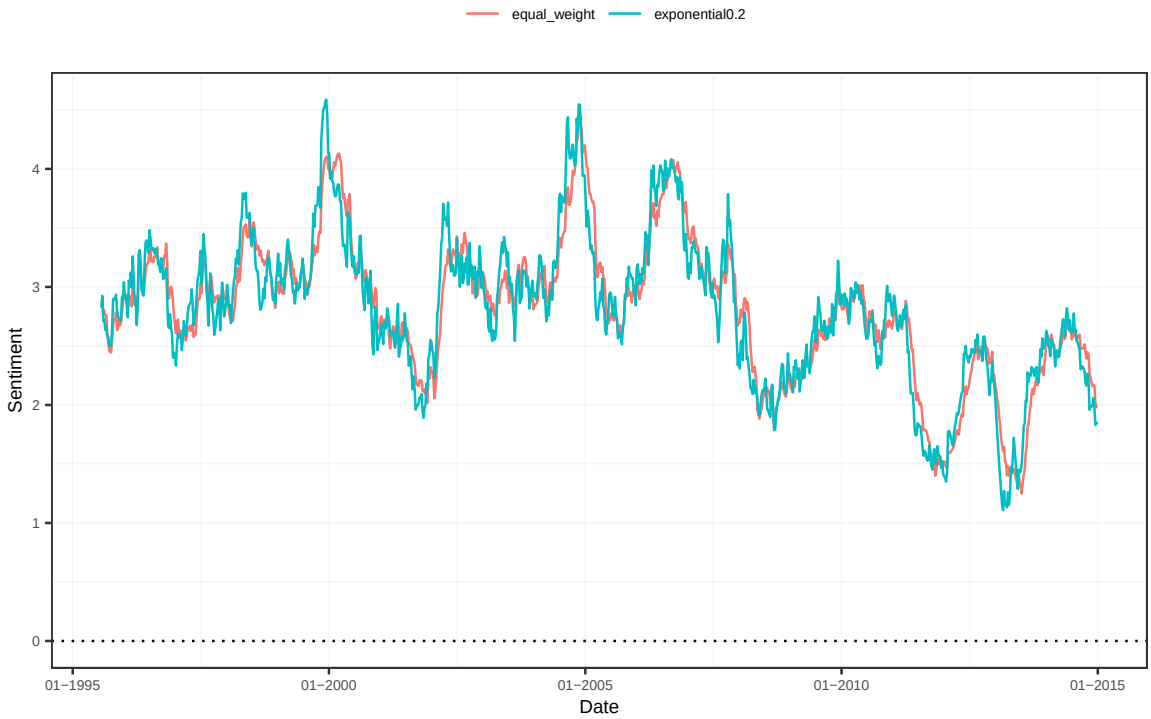
```
   4: 1995-08-14                          4.044
   5: 1995-08-21                          3.831
  ---
1011: 2014-12-01                          4.694
1012: 2014-12-08                          4.694
1013: 2014-12-15                          4.639
1014: 2014-12-22                          4.698
1015: 2014-12-29                          4.876
```

A *sento_measures* object is easily plotted across each of its dimensions. For example, Figure 3.2 shows a time series of average sentiment for both time weighting schemes involved.[30] A display of averages across lexicons and features is achieved by altering the group argument from the plot() method to "lexicons" and "features", respectively.

```
R> plot(sentMeas, group = "time")
```

[30] The averaged sentiment measures can be accessed from the plot object. Given p <- plot(...), this is via p[["data"]]. The data are in long format.

FIGURE 3.2: Textual sentiment time series averaged across time weighting schemes.



3.3.4 Manipulation of the Sentiment Measures

There are a number of methods and functions implemented to facilitate the manipulation of a *sento_measures* object. Useful methods are `subset()`, `diff()`, and `scale()`. The `subset()` function can be used to subset rows by index or by a condition, as well as to select or delete certain sentiment measures. The `scale()` function can take `matrix` inputs, handy to add, deduct, multiply or divide the sentiment measures up to the level of single observations. The functions `get_dates()` and `get_dimensions()` are convenient functions that return the dates and the aggregation dimensions of a *sento_measures* object. The functions `nobs()` and `nmeasures()` give the number of time series observations and the number of sentiment measures. The `as.data.table()` function extracts the sentiment measures, either in long or wide format.

The `select` and `delete` arguments in the `subset()` function indicate which combinations of sentiment measures to extract or delete. Here, the `subset()` function call returns a new object without all sentiment measures created from the `"LM_en"` lexicon and without the single time series consisting of the specified combination `"SENTICNET"` (lexicon), `"economy"`

(feature), and `"equal_weight"` (time weighting).[31]

```
R> subset(sentMeas, 1:600, delete = list(c("LM_en"),
+    c("SENTICNET", "economy", "equal_weight")))

A sento_measures object (95 textual sentiment time series,
                                                ... 600 observations).
```

Subsetting across rows is done with the `subset()` function without specifying an exact argument. A typical example is to subset only a time series range by specifying the dates part of that range, as below, where 50 dates are kept. One can also condition on specific sentiment measures being above, below, or between certain values, or directly indicate the row indices (as shown above).

```
R> subset(sentMeas, date %in% get_dates(sentMeas)[50:99])

A sento_measures object (108 textual sentiment time series,
                                                ... 50 observations).
```

To *ex-post* fill the time series date-wise, the `measures_fill()` can be used. The function at minimum fills in missing dates between the existing date range at the prevailing frequency. Dates before the earliest and after the most recent date can be added too. The argument `fill = "zero"` sets all added dates to zero, whereas `fill = "latest"` takes the most recent known value. This function is applied internally depending on the `fill` parameter from the `ctr_agg()` function. The example below pads the time series with trailing dates, taking the first value that occurs.

```
R> sentMeasFill <- measures_fill(sentMeas, fill = "latest",
+    dateBefore = "1995-07-01")
R> head(as.data.table(sentMeasFill)[, 1:3])

         date LM_en--wsj--equal_weight LM_en--wapo--equal_weight
1: 1995-06-26                  -2.021                    -3.064
2: 1995-07-03                  -2.021                    -3.064
3: 1995-07-10                  -2.021                    -3.064
4: 1995-07-17                  -2.021                    -3.064
5: 1995-07-24                  -2.021                    -3.064
6: 1995-07-31                  -2.061                    -3.059
```

[31] The new number of sentiment measures is not necessarily equal to $L \times K \times B$ anymore once a *sento_measures* object is modified.

The sentiment visualized using the `plot()` function when there are many different lexicons, features, and time weighting schemes may give a distorted image due to the averaging. To obtain a more nuanced picture of the differences in one particular dimension, one can ignore the other two dimensions. For example, `corpusPlain` below has only the dummy feature, and there is no time aggregation involved (`lag = 1`). This leaves the lexicons as the sole distinguishing dimension between the sentiment time series.

```
R> corpusPlain <- sento_corpus(usnews[, 1:3])
R> ctrAggLex <- ctr_agg(howWithin = "proportionalPol",
+    howTime = "own", howDocs = "equal_weight",
+    by = "month", fill = "none", lag = 1)
R> sentMeasLex <- sento_measures(corpusPlain,
+    lexicons = lex[-length(lex)], ctr = ctrAggLex)
R> mean(as.numeric(sentMeasLex$stats["meanCorr", ]))

[1] 0.3598
```

Figure 3.3 plots the nine time series, each belonging to a lexicon. There are differences in levels. For example, the Loughran & McDonald lexicon's time series lies almost exclusively in the negative sentiment area. The overall average correlation is close to 36%.

One can do additional aggregation of the sentiment time series with, again, the `aggregate()` function. This functionality equips the user with a way to either further diminish or further expand the dimensionality of the sentiment measures. For instance, all sentiment measures composed of three particular time aggregation schemes can be shrunk together, by averaging across each fixed combination of the other two dimensions, resulting in a set of new measures.

In the example, both built-in lexicons, both news sources, both added features, and similar time weighting schemes are collapsed into their respective new counterparts. The `do.keep = FALSE` option indicates that the original measures are not kept after merging, such that the number of sentiment time series effectively goes down to $7 \times 4 \times 1 = 28$.

```
R> sentMeasAgg <- aggregate(sentMeas,
+    time = list(W = c("equal_weight", "exponential0.2")),
+    lexicons = list(LEX = c("LM_en", "HENRY_en", "GI_en")),
+    features = list(JOUR = c("wsj", "wapo"),
+        NEW = c("uncertainty", "election")),
+    do.keep = FALSE)
R> get_dimensions(sentMeasAgg)

$features
[1] "economy" "noneconomy" "JOUR" "NEW"
```

FIGURE 3.3: Textual sentiment time series averaged across lexicons.


```
$lexicons
[1] "NRC" "HULIU" "SENTIWORD" "JOCKERS" "SENTICNET" "SOCAL" "LEX"

$time
[1] "W"
```

The `aggregate()` function with argument `do.global = TRUE` merges the sentiment measures into single dimension-specific time series we refer to as global sentiment. Each of the different components of the dimensions has to receive a weight that stipulates the importance and sign in the global sentiment index. The weights need not sum to one, that is, no condition is imposed. For example, the global lexicon-oriented sentiment measure is computed as $s_u^{G,L} \equiv \frac{1}{P}\sum_{p=1}^{P} s_u^p w_{l,p}$. The weight to lexicon l that appears in p is denoted by $w_{l,p}$. An additional, "fully global", measure is composed as $s_u^G \equiv (s_u^{G,L} + s_u^{G,K} + s_u^{G,B})/3$.

In the code excerpt, we define *ad-hoc* weights that emphasize the initial features and the Henry lexicon. Both time weighting schemes are weighted equally.[32] The output is a four-

[32] This can also be achieved by setting `time = 1` (similarly for the other arguments), and is in fact the default.

column `data.table`. The global sentiment measures provide a low-dimensional summary of the sentiment in the corpus, depending on some preset parameters related to the importance of the dimensions.

```
R> glob <- aggregate(sentMeas,
+    lexicons = c(0.10, 0.40, 0.05, 0.05, 0.08, 0.08,
+        0.08, 0.08, 0.08),
+    features = c(0.20, 0.20, 0.20, 0.20, 0.10, 0.10),
+    time = c(1/2, 1/2), do.global = TRUE)
```

The `peakdates()` function pinpoints the dates with most extreme sentiment (negatively, positively, or both). The example below extracts the date with the lowest sentiment time series value across all measures. The `peakdocs()` function can be applied equivalently to a *sentiment* object to get the document identifiers with most extreme document-level sentiment.

```
R> peakdates(sentMeas, n = 1, type = "neg")

[1] "2008-01-28"
```

3.3.5 Sparse Regression Using the Sentiment Measures

Step 4 consists of the regression modeling. The **sentometrics** package offers an adapted interface to sparse regression. Other model frameworks can be explored with as input the sentiment measures extracted through the `as.data.table()` (or `as.data.frame()`) function. For example, one could transform the computed sentiment time series into a *zoo* object from the **zoo** package (Zeileis and Grothendieck, 2005), use any of **zoo**'s functionalities thereafter (e.g. dealing with an irregular time series structure), or run a simple linear regression (here on the first six sentiment variables) as follows:

```
R> y <- rnorm(nobs(sentMeas))
R> dt <- as.data.table(sentMeas)
R> z <- zoo::zoo(dt[, !"date"], order.by = dt[["date"]])
R> reg <- lm(y ~ z[, 1:6])
```

We proceed by explaining the available modeling setup in the **sentometrics** package. The `ctr_model()` function defines the modeling setup. The main arguments are itemized, all others are reviewed within the discussion:

- `model`: The `model` argument can take `"gaussian"` (for linear regression), and `"binomial"` or `"multinomial"` (both for logistic regression). The argument `do.intercept = TRUE` fits an intercept.

- `type`: The `type` specifies the calibration procedure to find the most appropriate α and λ in (3.5). The options are `"cv"` (cross-validation) or one of three information criteria (`"BIC"`, `"AIC"` or `"Cp"`). The information criterium approach is available in case of a linear regression only. The argument `alphas` can be altered to change the possible values for `alpha`, and similarly so for the `lambdas` argument. If `lambdas = NULL`, the possible values for λ are generated internally by the `glmnet()` function from the R package **glmnet**. If `lambdas = 0`, the regression procedure is OLS. The arguments `trainWindow`, `testWindow` and `oos` are needed when model calibration is performed through cross-validation, that is, when `type = "cv"`. The cross-validation implemented is based on the "rolling forecasting origin" principle, considering we are working with time series.[33] The argument `do.progress = TRUE` prints calibration progress statements. The `do.shrinkage.x` argument is a logical vector to indicate on which external explanatory variables to impose regularization. These variables, x_u, are added through the `x` argument of the `sento_model()` function.
- `h`: The integer argument `h` shifts the response variable up to y_{u+h} and aligns the explanatory variables in accordance with (3.4).[34] If `h = 0` (by default), no adjustments are made. The logical `do.difference` argument, if `TRUE`, can be used to difference the target variable `y` supplied to the `sento_model()` function, if it is a continuous variable (i.e. `model = "gaussian"`). The lag taken is the absolute value of the `h` argument (given $|h| > 0$). For example, if `h = 2`, and assuming the `y` variable is aligned time-wise with all explanatory variables, denoted by X here for sake of the illustration, the regression performed is of $y_{t+2} - y_t$ on X_t. If $h = -2$, the regression fitted is $y_{t+2} - y_t$ on X_{t+2}.
- `do.iter`: To enact an iterative model estimation and a one-step ahead out-of-sample analysis, set `do.iter = TRUE`. To perform a one-off in-sample estimation, set `do.iter = FALSE`. The arguments `nSample`, `start` and `nCore` are used for iterative modeling, thus, when `do.iter = TRUE`. The first argument is M, that is, the size of the sample to re-estimate the model with each time. The second argument can be used to only run a later subset of the iterations (`start = 1` by default runs all iterations). The total number of iterations is equal to `length(y) - nSample - abs(h) - oos`, with `y` the response variable as a vector. The `oos` argument specifies partly, as explained above, the

[33] As an example, take 120 observations in total, `trainWindow = 80`, `testWindow = 10` and `oos = 5`. In the first round of cross-validation, a model is estimated for a certain α and λ combination with the first 80 observations, then 5 observations are skipped, and predictions are generated for observations 86 to 95. The next round does the same but with all observations moved one step forward. This is done until the end of the total sample is reached, and repeated for all possible parameter combinations, relying on the `train()` function from the R package **caret**. The optimal (α, λ) couple is the one that induces the lowest average prediction error (measured by the root mean squared error for linear models, and overall accuracy for logistic models).

[34] If the input response variable is not aligned time-wise with the sentiment measures and the other explanatory variables, `h` cannot be interpreted as the exact prediction horizon. In other words, `h` only shifts the input variables as they are provided.

cross-validation, but also provides flexibility in defining the out-of-sample exercise. For instance, given t, the one-step ahead out-of-sample prediction is computed at $t+$`oos`$+1$. As per default, `oos = 0`. If `nCore > 1`, the `%dopar%` construct from the R package **foreach** (Weston, 2019) is utilized to speed up the out-of-sample analysis.

To enhance the intuition about attribution, we estimate a contemporaneous in-sample model and compute the attribution decompositions.

```
R> ctrInSample <- ctr_model(model = "gaussian",
+    h = 0, type = "BIC", alphas = 0, do.iter = FALSE)
R> fit <- sento_model(sentMeas, y, ctr = ctrInSample)
```

The `attributions()` function takes the *`sento_model`* object and the related sentiment measures object as inputs, and generates by default attributions for all in-sample dates at the level of individual documents, lags, lexicons, features, and time weighting schemes. The function can be applied to a *`sento_modelIter`* object as well, for any specific dates using the `refDates` argument. If `do.normalize = TRUE`, the values are normalized between -1 and 1 through division by the ℓ_2-norm of the attributions at a given date. The output is an *`attributions`* object.

```
R> attrFit <- attributions(fit, sentMeas)
R> head(attrFit[["features"]])
```

```
         date      wsj     wapo  economy noneconomy uncertainty  election
1: 1995-07-24 0.001422 0.004144 -0.01382    0.01675   -0.002531 -0.004931
2: 1995-07-31 0.001900 0.004747 -0.01336    0.01739   -0.002468 -0.004824
3: 1995-08-07 0.001626 0.004348 -0.01258    0.01636   -0.002409 -0.004723
4: 1995-08-14 0.001243 0.004861 -0.01297    0.01590   -0.002353 -0.004231
5: 1995-08-21 0.001524 0.004248 -0.01179    0.01553   -0.002509 -0.004148
6: 1995-08-28 0.001433 0.004373 -0.01237    0.01584   -0.002462 -0.004071
```

Attribution decomposes a prediction into the different sentiment components along a given dimension, for example, lexicons. The sum of the individual sentiment attributions per date, the constant, and other non-sentiment measures are thus equal to the prediction. Indeed, the piece of code below shows that the difference between the prediction and the summed attributions plus the constant is equal to zero throughout.

```
R> X <- as.matrix(as.data.table(sentMeas)[, -1])
R> yFit <- predict(fit, newx = X)
R> attrSum <- rowSums(attrFit[["lexicons"]][, -1]) +
+    fit[["reg"]][["a0"]]
R> all.equal(as.numeric(yFit), attrSum)
```

```
[1] TRUE
```

3.4 Application to Predicting the CBOE Volatility Index

A noteworthy amount of finance research has pointed out the impact of sentiment expressed through various corpora on stock returns and trading volume, including Heston and Sinha (2017), Jegadeesh and Wu (2013), Tetlock *et al.* (2008), and Antweiler and Frank (2004). Caporin and Poli (2017) create lexicon-based news measures to improve daily realized volatility forecasts. Manela and Moreira (2017) explicitly construct a news-based measure closely related to the CBOE Volatility Index (VIX) and a good proxy for uncertainty. A more widely used proxy for uncertainty is the Economic Policy Uncertainty (EPU) index (Baker *et al.*, 2016). This indicator is a normalized text-based index of the number of news articles discussing economic policy uncertainty, from ten large U.S. newspapers. A relationship between political uncertainty and market volatility is found by Pástor and Veronesi (2013).

The VIX measures the annualized option-implied volatility on the S&P 500 stock market index over the next 30 days. It is natural to expect that media sentiment and political uncertainty partly explain the expected volatility measured by the VIX. In this section, we test this using the EPU index and sentiment variables constructed from the `usnews` dataset. We analyze if our textual sentiment approach is more helpful than the EPU index in an out-of-sample exercise of predicting the end-of-month VIX in six months. The prediction specifications we are interested in are summarized as follows:

$$\begin{aligned}
\mathscr{M}_s: &\qquad VIX_{u+h} = \delta + \gamma VIX_u + \beta^\top s_u + \varepsilon_{u+h},\\
\mathscr{M}_{epu}: &\qquad VIX_{u+h} = \delta + \gamma VIX_u + \beta EPU_{u-1} + \varepsilon_{u+h},\\
\mathscr{M}_{ar}: &\qquad VIX_{u+h} = \delta + \gamma VIX_u + \varepsilon_{u+h}.
\end{aligned}$$

The target variable VIX_u is the most recent available end-of-month daily VIX value. We run the predictive analysis for $h = 6$ months. The sentiment time series are in s_u and define the sentiment-based model ($\mathscr{M}_s$). As primary benchmark, we exchange the sentiment variables for EPU_{u-1}, the level of the U.S. economic policy uncertainty index in month $u-1$ we know is fully available by month u ($\mathscr{M}_{epu}$). We also consider a simple autoregressive specification ($\mathscr{M}_{ar}$).

We use the built-in U.S. news corpus of around 4145 documents, in the `uscorpus` object. Likewise, we proceed with the nine lexicons and the valence shifters list from the `lex` object used in previous examples. To infer textual features from scratch, we use a structural topic modeling approach as implemented by the R package **stm** (Roberts *et al.*, 2019). This is a prime example of integrating a distinct text mining workflow with our textual sentiment analysis workflow. The **stm** package works on a **quanteda** document-term matrix as an input. We perform a fairly standard cleaning of the document-term matrix, and use the default parameters of the `stm()` function. We group into eight features.

```
R> dfm <- quanteda::dfm(uscorpus, tolower = TRUE,
+    remove_punct = TRUE,
+    remove_numbers = TRUE,
+    remove = quanteda::stopwords("en")) %>%
+    quanteda::dfm_remove(min_nchar = 3) %>%
+    quanteda::dfm_trim(min_termfreq = 0.95,
+    termfreq_type = "quantile") %>%
+    quanteda::dfm_trim(max_docfreq = 0.15,
+    docfreq_type = "prop")
R> dfm <- quanteda::dfm_subset(dfm, quanteda::ntoken(dfm) > 0)
R> topicModel <- stm::stm(dfm, K = 8, verbose = FALSE)
```

We then define the keywords as the five most statistically representative terms for each topic. They are assembled in `keywords` as a list.

```
R> topTerms <- t(stm::labelTopics(topicModel, n = 5)[["prob"]])
R> keywords <- lapply(1:ncol(topTerms), function(i) topTerms[, i])
R> names(keywords) <- paste0("TOPIC_", 1:length(keywords))
```

We use the `add_features()` function to generate the features based on the occurrences of these keywords in a document, scaling the feature values between 0 and 1 by setting `do.binary = FALSE`. We also delete all current features. Alternatively, one could use the predicted topics per text as a feature and set `do.binary = TRUE`, to avoid documents sharing mutual features, instead of relying on the generated keywords. We see a relatively even distribution of the corpus across the generated features.

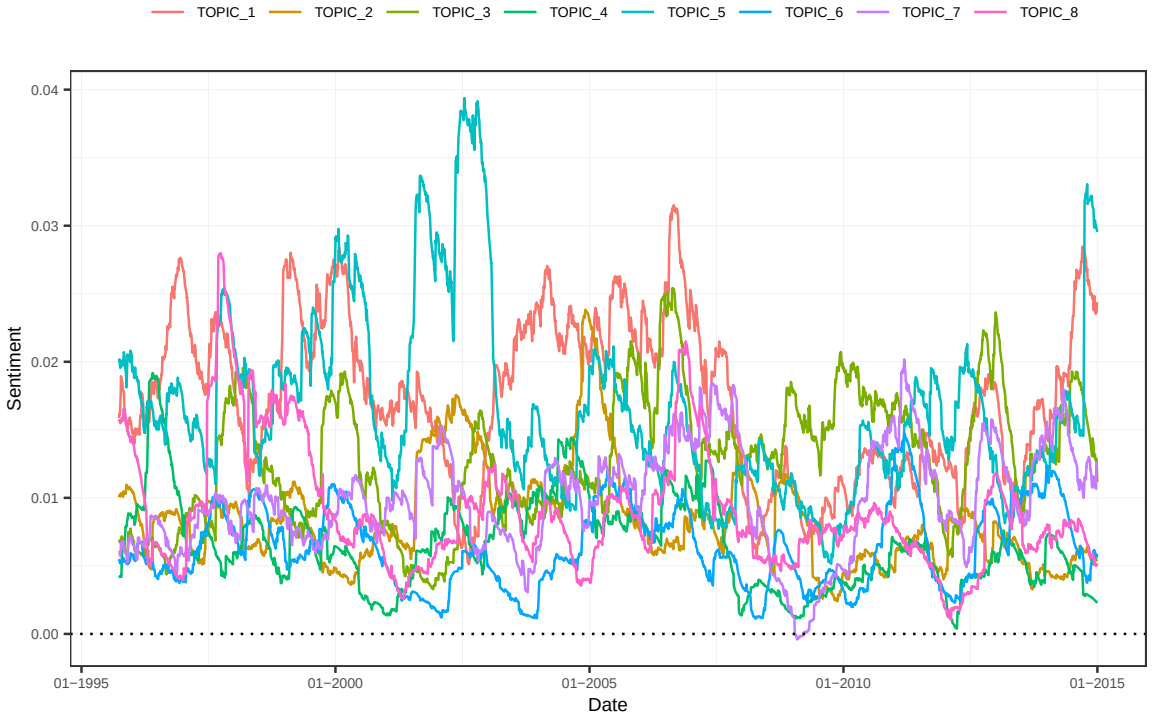
```
R> uscorpus <- add_features(uscorpus, keywords = keywords,
+    do.binary = FALSE, do.regex = FALSE)
R> quanteda::docvars(uscorpus, c("uncertainty", "election",
+    "economy", "noneconomy", "wsj", "wapo")) <- NULL
R> colSums(quanteda::docvars(uscorpus)[, -1] != 0)

TOPIC_1 TOPIC_2 TOPIC_3 TOPIC_4 TOPIC_5 TOPIC_6 TOPIC_7 TOPIC_8
   1389    1079    1261     648     994    1005     856    1041
```

The frequency of our target variable is monthly, yet we aggregate the sentiment time series on a daily level and then across time using a lag of about nine months (`lag = 270`). While the lag is substantial, we also include six different Beta time weighting schemes to capture time dynamics other than a slowly moving trend.

```
R> ctrAggPred <- ctr_agg(howWithin = "proportionalPol",
+    howDocs = "equal_weight", howTime = "beta",
+    by = "day", fill = "latest", lag = 270, aBeta = 1:3, bBeta = 1:2)
R> sentMeasPred <- sento_measures(uscorpus, lexicons = lex,
+    ctr = ctrAggPred)
```

In Figure 3.4, we see sufficiently differing average time series patterns for all topics. The drop in sentiment during the recent financial crisis is apparent, but the recovery varies along features.

FIGURE 3.4: Textual sentiment time series across latent topic features.



The package has the EPU index as a dataset `epu` included. We consider the EPU index values one month before all other variables and use the **lubridate** package (Grolemund and Wickham, 2011) to do this operation. We assure that the length of the dependent variable is equal to the number of observations in the sentiment measures by selecting based on the proper monthly dates. The pre-processed VIX data are represented by the `vix` variable.[35]

[35] The VIX data are retrieved from `https://fred.stlouisfed.org/series/VIXCLS`.

```
R> data("epu", package = "sentometrics")
R> sentMeasIn <- subset(sentMeasPred, date %in% vix$date)
R> datesIn <- get_dates(sentMeasIn)
R> datesEPU <- lubridate::floor_date(datesIn, "month")
+    %m-% months(1)
R> xEPU <- epu[epu$date %in% datesEPU, "index"]
R> y <- vix[vix$date %in% datesIn, "value"]
R> x <- data.frame(lag = y, epu = xEPU)
```

We apply the iterative rolling forward analysis (`do.iter = TRUE`) for our six-month prediction horizon. The target variable is aligned with the sentiment measures in `sentMeasIn`, such that `h = 6` in the modeling control means forecasting the monthly averaged VIX value in six months. The calibration of the sparse linear regression is based on a Bayesian-like information criterion (`type = "BIC"`) proposed by Tibshirani and Taylor (2012). We configure a sample size of $M = 60$ months for a total sample of $N = 232$ observations. Our out-of-sample setup is nonoverlapping; `oos = h - 1` means that for an in-sample estimation at time u, the last available explanatory variable dates from time $u - h$, and the out-of-sample prediction is performed at time u as well, not at time $u - h + 1$. We consider a range of `alpha` values that allows any of the Ridge, LASSO, and pure elastic net regularization objective functions.[36]

```
R> h <- 6
R> oos <- h - 1
R> M <- 60
R> ctrIter <- ctr_model(model = "gaussian",
+    type = "BIC", h = h, alphas = c(0, 0.1, 0.3, 0.5,
+        0.7, 0.9, 1),
+    do.iter = TRUE, oos = oos, nSample = M, nCore = 1)
R> out <- sento_model(sentMeasIn, x = x[, "lag", drop = FALSE],
+    y = y, ctr = ctrIter)
R> summary(out)

Model specification
- - - - - - - - - - - - - - - - - - - -

Model type: gaussian
Calibration: via BIC information criterion
Sample size: 60
```

[36] When a sentiment measure is a duplicate of another or when at least 50% of the series observations are equal to zero, it is automatically discarded from the analysis. Discarded measures are put under the `"discarded"` element of a *sento_model* object.

```
Total number of iterations/predictions: 161
Optimal average elastic net alpha parameter: 0.94
Optimal average elastic net lambda parameter: 2.52

Out-of-sample performance
- - - - - - - - - - - - - - - - - - - -

Mean directional accuracy: 52.5 %
Root mean squared prediction error: 10.16
Mean absolute deviation: 7.73
```

The output of the `sento_model()` call is a *sento_modelIter* object. Below we replicate the analysis for the benchmark regressions, without the sentiment variables. The vector `preds` assembles the out-of-sample predictions for model $\mathscr{M}_{epu}$, the vector `predsAR` for model $\mathscr{M}_{ar}$.

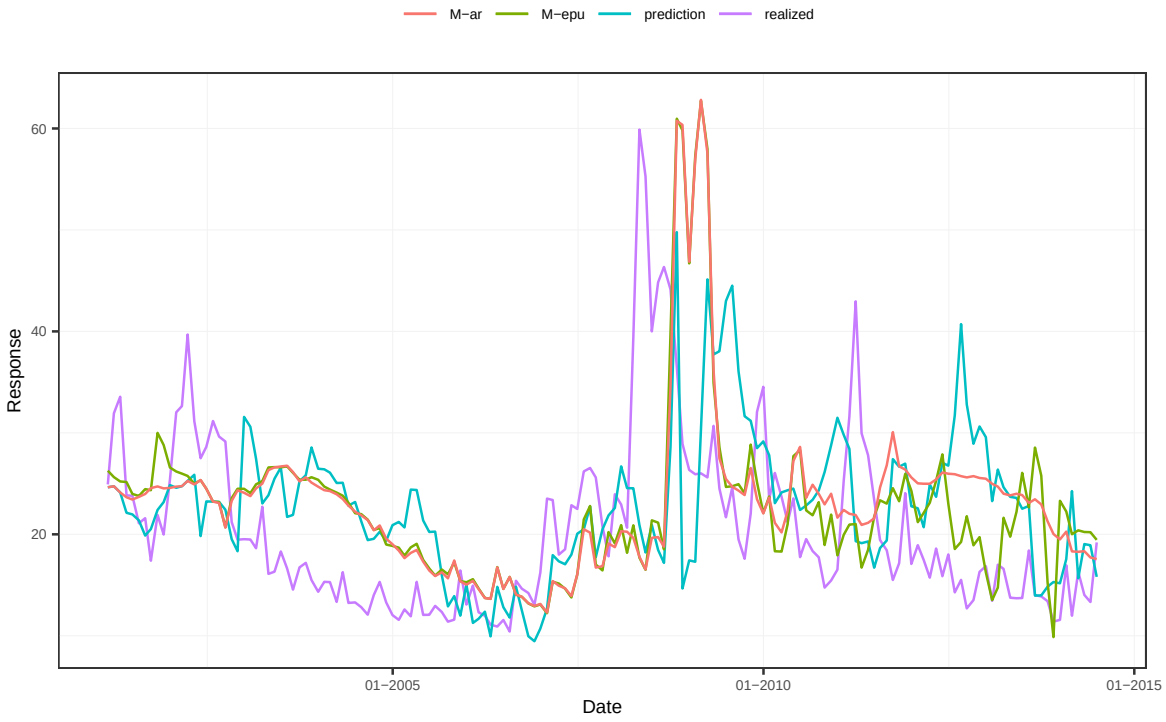
```
R> preds <- predsAR <- rep(NA, nrow(out[["performance"]]$raw))
R> yTarget <- y[-(1:h)]
R> xx <- x[-tail(1:nrow(x), h), ]
R> for (i in 1:(length(preds))) {
+    j <- i + M
+    data <- data.frame(y = yTarget[i:(j - 1)], xx[i:(j - 1), ])
+    reg <- lm(y ~ ., data = data)
+    preds[i] <- predict(reg, xx[j + oos, ])
+    regAR <- lm(y ~ ., data = data[, c("y", "lag")])
+    predsAR[i] <- predict(regAR, xx[j + oos, "lag", drop = FALSE])
R> }
```

A more detailed view of the different performance measures, in this case directional accuracy, root mean squared, and absolute errors, is obtained via `out[["performance"]]`. A list of the individual *sento_model* objects is found under `out[["models"]]`. A simple plot to visualize the out-of-sample fit of any *sento_modelIter* object can be produced using `plot()`. We display in Figure 3.5 the realized values and the different predictions.

```
R> plot(out) +
+    geom_line(data = melt(data.table(date = names(out$models),
+    "M-epu" = preds, "M-ar" = predsAR, check.names = FALSE),
+    id.vars = "date"))
```

Table 3.2 reports two common out-of-sample prediction performance measures, decomposed in a pre-crisis period (spanning up to June 2007, from the point of view of the prediction

FIGURE 3.5: Realized six-month ahead VIX_{u+6} values (purple, `realized`) and out-of-sample predictions from models $\mathscr{M}_s$ (blue, `prediction`), $\mathscr{M}_{ar}$ (red), and $\mathscr{M}_{epu}$ (green).



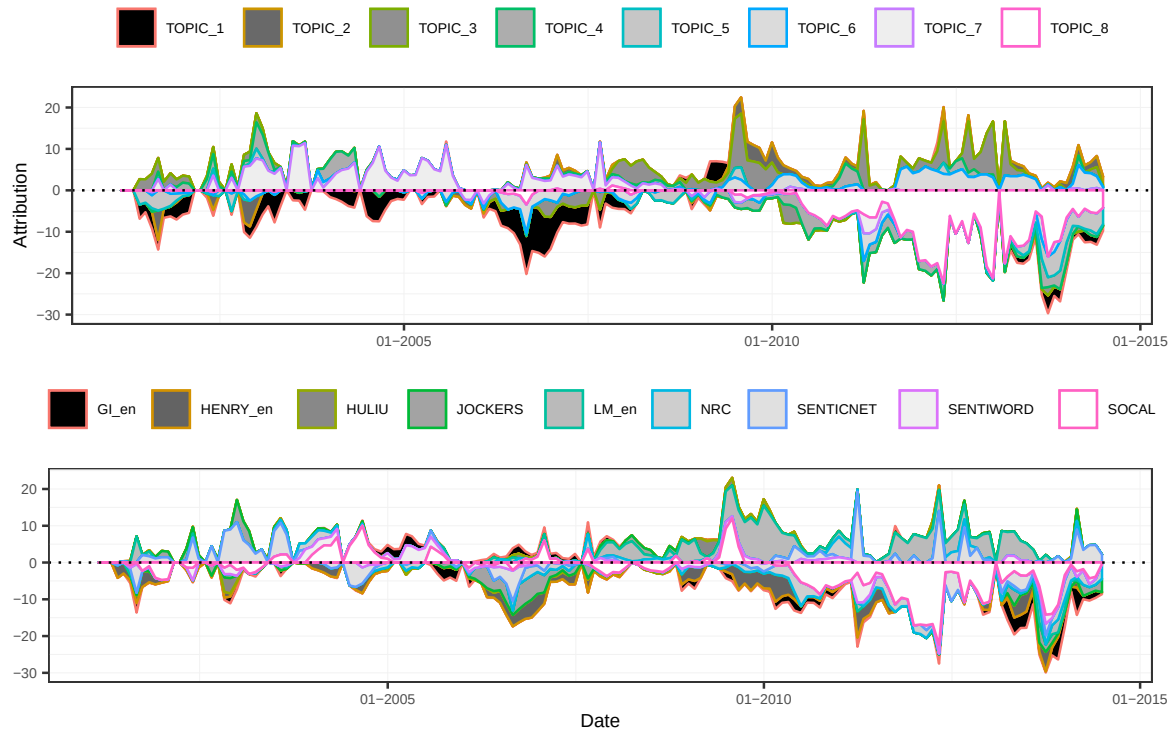
date), a crisis period (spanning up to December 2009) and a post-crisis period. It appears that sentiment adds predictive power during the crisis period. The flexibility of the elastic net avoids that predictive power is too seriously compromised when adding sentiment to the regression, even when it has no added value.

Table 3.2: Performance measures. The root mean squared error (RMSE) is computed as $\sum_{i=1}^{N_{oos}} e_i^2/N_{oos}$, and the mean absolute deviation (MAD) as $\sum_{i=1}^{N_{oos}} |e_i|/N_{oos}$, with N_{oos} the number of out-of-sample predictions, and e_i the prediction error. The sentiment-based model is $\mathscr{M}_s$, the EPU-based model is $\mathscr{M}_{epu}$, the autoregressive model is $\mathscr{M}_{ar}$.

	Full			Pre-crisis			Crisis			Post-crisis		
	$\mathscr{M}_s$	$\mathscr{M}_{epu}$	$\mathscr{M}_{ar}$	$\mathscr{M}_s$	$\mathscr{M}_{epu}$	$\mathscr{M}_{ar}$	$\mathscr{M}_s$	$\mathscr{M}_{epu}$	$\mathscr{M}_{ar}$	$\mathscr{M}_s$	$\mathscr{M}_{epu}$	$\mathscr{M}_{ar}$
RMSE	10.16	10.42	10.72	7.01	6.64	6.52	16.34	19.23	19.46	9.42	7.43	8.42
MAD	7.73	7.46	7.93	5.88	5.78	5.60	12.94	14.23	14.56	7.43	6.09	7.54

The last step is to perform a post-modeling attribution analysis. For a *sento_modelIter* object, the `attributions()` function generates sentiment attributions for all out-of-sample dates. To study the evolution of the prediction attribution, the attributions can be visualized with the `plot()` function applied to the *attributions* output object. This can be done according to any of the dimensions, except for individual documents. Figure 3.6 shows two types of attributions in separate panels. The attributions are displayed stacked on top of each other, per date. The y-axis represents the attribution to the prediction of the target variable. The third topic was most impactful during the crisis, and the eighth topic received the most positive post-crisis weight. Likewise, the lexicons attribution conveys an increasing influence of the SO-CAL lexicon on the predictions. Finally, it can be concluded that the predictive role of sentiment is least present before the crisis.

FIGURE 3.6: Attribution to features (top) and lexicons (below).



This illustration shows that the **sentometrics** package provides useful insights in predicting variables like the VIX starting from a corpus of texts. Results could be improved by expanding the corpus, or by optimizing the features generation. From a theoretical point of view, we know that the VIX can be explained by fundamental uncertainty and by risk aversion (e.g. Bollerslev *et al.*, 2009). One can try to build sentiment indices that proxy these different VIX components, through well-specified features. If it happens that changes in risk aversion are

easier to predict than fundamental uncertainty, this would help to understand why text helps to predict the VIX.

For a larger application of the entire workflow, we refer to Ardia *et al.* (2019b). They find that the incorporation of textual sentiment indices results in better prediction of the U.S. industrial production growth rate compared to using a panel of typical macroeconomic indices only.

3.5 Conclusion and Future Development

The R package **sentometrics** provides a framework to calculate sentiment for texts, to aggregate textual sentiment scores into many time series at a desired frequency, and to use these in a flexible prediction modeling setup. It can be deployed to quantify a qualitative corpus of texts, relate it to a target variable, and retrieve which type of sentiment is most informative through visualization and attribution analysis.

The main priorities for further development are integrating better prediction tools, enhancing the complexity of the sentiment engine, allowing user-defined weighting schemes, and adding intraday aggregation. Interfacing with tools that provide principal component analysis, partial least squares regression, and (dynamic) factor analysis would provide interesting ways to extract from many sentiment time series the most prominent trend(s), possibly related to an outcome variable. The high-dimensional regression framework can also be extended to include other regression penalties, for example, the group LASSO penalty (Yuan and Lin, 2006). At the minimum, in the future, we intend to release more extensive examples on how to use the output from our package and use it appropriately in other existing dimensionality reduction and modeling packages.

If you use R or **sentometrics**, please cite the software in publications. In case of the latter, use `citation("sentometrics")`. Additional code examples are found in the "Examples" and "Contributions" sections at `https://SentometricsResearch.github.io/sentometrics`.

Semi-Supervised Text Mining for Monitoring the News About the ESG Performance of Companies 4

Abstract

We present a general monitoring methodology to summarize news about predefined entities and topics into tractable time-varying indices. The approach embeds text mining techniques to transform news data into numerical data, which entails the querying and selection of relevant news articles, and the construction of frequency- and sentiment-based indicators. Word embeddings are used to achieve maximally informative news selection and scoring. We apply the methodology from the viewpoint of a sustainable asset manager wanting to actively follow news covering Environmental, Social and Governance (ESG) aspects. In an empirical analysis, using a Dutch-written news corpus, we create news-based ESG signals for a large list of companies and compare these to scores from an external data provider. We find preliminary evidence of abnormal news dynamics leading up to downward score adjustments, and of efficient portfolio screening.

4.1 Introduction

Automated analysis of textual data such as press articles can help investors to better screen the investable universe. News coverage, how often news discusses a certain topic, and textual sentiment analysis, if news is perceived as positive or negative, serve as good proxies to detect important events and their surrounding perception. Text-based signals have at least the advantage of timeliness, and often also that of complementary information value. The challenge is to transform the textual data into useful numerical signals through application of proper text mining techniques.

Key research in finance employing text mining includes Heston and Sinha (2017), Jegadeesh and Wu (2013), Tetlock *et al.* (2008), and Antweiler and Frank (2004). These studies point out the impact of textual sentiment on stock returns and trading volume. Lately, the focus has shifted to using text corpora for more specific goals. For instance, Engle *et al.* (2020) form portfolios hedged against climate change news based on news indicators.

This chapter takes the use of textual data science in sustainable investment as running example. Investors with a goal of Socially Responsible Investing (SRI) consider alternative measures to assess investment risk and return opportunities. They evaluate portfolios by how well the underlying assets align with a Corporate Social Responsibility (CSR) policy—for instance, if they commit to environmental-friendly production methods. A corporation's level of CSR is often measured along the Environmental, Social and corporate Governance (ESG) dimensions.

Investors typically obtain an investable universe of ESG-compliant assets by comparing companies to their peers, using a best-in-class approach (e.g. including the top 40% companies), or a worst-in-class approach (e.g. excluding the bottom 40% companies). To do so, investors rely on in-house research, and third party agency reports and ratings. Berg *et al.* (2019), Amel-Zadeh and Serafeim (2018), and Escrig-Olmedo *et al.* (2010), among others, find that these ESG ratings are diverse, not transparent, and lack standardisation. Moreover, most agencies only provide at best monthly updates. Furthermore, ratings are often reporting-driven and not signal-driven. This implies that a company can be ESG-compliant "by the book" when it is transparent (akin to greenwashing), but that the ratings are not an accurate reflection of the true current underlying sustainability profile.

In the remainder of the chapter, we introduce a methodology to create and validate news-based indicators allowing to follow entities and topics of interest. We then empirically demonstrate the methodology in a sustainable portfolio monitoring context, extracting automatically from news an objective measurement of the ESG dimensions. Moniz (2016) is an exception in trying to infer CSR-related signals from media news using text mining in this otherwise largely unexplored territory.

4.2 Methodology to Create Text-Based Indicators

We propose a methodology to extract meaningful time series indicators from a large collection of texts. The indicators should represent the dimensions and entities one is interested in, and their time variation should connect to real-life events and news stories. The goal is to turn the indicators into a useful decision-making signal. This is a hard problem, as there is no underlying objective function to optimize, text data are not easy to explore, and it is computationally cumbersome to iterate frequently. Our methodology is therefore semi-supervised, altering between rounds of algorithmic estimation and human expert validation.

4.2.1 From Text to Numerical Data

A key challenge is to transform the stream of qualitative textual data into quantitative indicators. This involves first the selection of the relevant news and the generation of useful metadata, such as the degree to which news discusses an entity or an ESG dimension, or the sentiment of the news message. We tackle this by using domain-specific keywords to query a database of news articles and create the metadata. The queried articles need to undergo a second round of selection, to filter out the irrelevant news. Lastly, the kept corpus is aggregated into one or more time series.

To classify news as relevant to sustainability, we rely on keywords generated from a word embedding space. Moniz (2016) uses a latent topic model, which is a probabilistic algorithm that clusters a corpus into a variety of themes. Some of these themes can then be manually annotated as belonging to ESG. We decide to go with word embeddings as it gives more control over the inclusion of keywords and the resulting text selection. Another approach is to train a named entity recognition (NER) model, to extract specific categories of concepts. A NER model tailored to ESG concepts is hard to build from scratch, as it needs fine-grained labeled data.

The methodology laid out below assumes that the corpus is in a single language. However, it can be extended to a multi-language corpus in various ways. The go-to approach, in terms of accuracy, is to consider each language separately by doing the indicators construction independently for every language involved. After that, an additional step is to merge the various language-specific indicators into an indicator that captures the evolution across all languages. One could, for simplicity, generate keywords in one language and then employ translation. Another common way to deal with multiple languages is to translate all incoming texts into a target language, and then proceed with the pipeline for that language.

4.2.1.1 Keywords Generation

Three types of keywords are required. The **query lexicon** is a list of keywords per dimension of interest (*in casu*, the three ESG dimensions). Its use is twofold. First, to identify the articles

from a large database with at least one of these keywords, second, to measure the relevance of the queried articles (i.e. more keywords present in an article means it is more relevant). The **sentiment lexicon** is a list of words with an associated sentiment polarity, used to calculate document-level textual sentiment. The polarity defines the average connotation a word has, for example -1 for 'violence', or 1 for 'happy.' **Valence shifters** are words that change the meaning of other words in their neighborhood. There are several categories of valence shifters, but we focus on amplifiers and deamplifiers. An amplifier strengthens a neighboring word, for instance the word 'very' amplifies the word 'strong' in case of 'very strong.' Deamplifiers do the opposite, for example, 'hardly' weakens the impact of 'good' when 'hardly good.' The reason to integrate valence shifters in the sentiment calculation is to better account for context in a text. The unweighted sentiment score of a document i with Q_i words under this approach is $s_i = \sum_{j=1}^{Q_i} v_{j,i} s_{j,i}$. The score $s_{j,i}$ is the polarity value attached in the sentiment lexicon to word j, and is zero when the word is not in the lexicon. If word $j-1$ is a valence shifter, its impact is measured by $v_{j,i} = 1.8$ for amplifiers or $v_{j,i} = 0.2$ for deamplifiers. By default, $v_{j,i} = 1$.

To generate the keywords, we rely on expansion through a word embedding space. Word embeddings are vector representations optimized so that words closer to each other in terms of linguistic context have a more similar quantitative representation. Word embeddings are usually a means to an end. In our case, based on an initial set of seed keywords, analogous words can be obtained by analyzing the words closest to them in the embedding space. Many word embeddings computed on large-scale datasets (e.g. on Wikipedia) are freely available in numerous languages.[37] The availability of pretrained word embeddings makes it possible to skip the step of estimating a new word embedding space, however, in this chapter, we describe a straightforward approach to do the estimation oneself.

Word2Vec (Mikolov *et al.*, 2013) and GloVe (Pennington *et al.*, 2014) are two of the most well-known techniques to construct a word embedding space. More recent and advanced methods include fastText (Bojanowski *et al.*, 2017) and the BERT family (Devlin *et al.*, 2018). Word2Vec is structured as a continuous bag-of-words (CBOW) or as a skip-gram architecture, both relying only on local word information. A CBOW model tries to predict a given word based on its neighboring words. A skip-gram model tries to use a given word to predict the neighboring words. GloVe (Pennington *et al.*, 2014) is a factorization method applied to the corpus word-word co-occurrence matrix. A co-occurrence matrix stores the number of times a column word appears in the context of a row word. As such, GloVe integrates both global (patterns across the entire corpus) and local (patterns specific to a small context window) statistics. The intuition is that words which co-occur frequently are assumed to share a related semantic meaning. This is apparent in the co-occurrence matrix, where these words as a row-column combination will have higher values.

[37] For example, pretrained word embeddings by Facebook are available for download at `https://fasttext.cc/docs/en/crawl-vectors.html`.

GloVe's optimization outputs two v-dimensional vectors per word (the word vector, and a separate context word vector), that is, $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^v$. The final word vector to use is defined as $\boldsymbol{w} \equiv \boldsymbol{w}_1 + \boldsymbol{w}_2$. To measure the similarity between word vectors, say $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$, the cosine similarity metric is commonly used. We define $cs_{ij} \equiv \boldsymbol{w}_i \boldsymbol{w}_j / \|\boldsymbol{w}_i\| \left\|\boldsymbol{w}_j\right\|$, where $\|.\|$ is the ℓ_2-norm. The measure $cs_{ij} \in [-1, 1]$, and the higher the more similar words i and j are in the embedding space.

FIGURE 4.1: Representation of the flow from seed words to the keywords of interest.

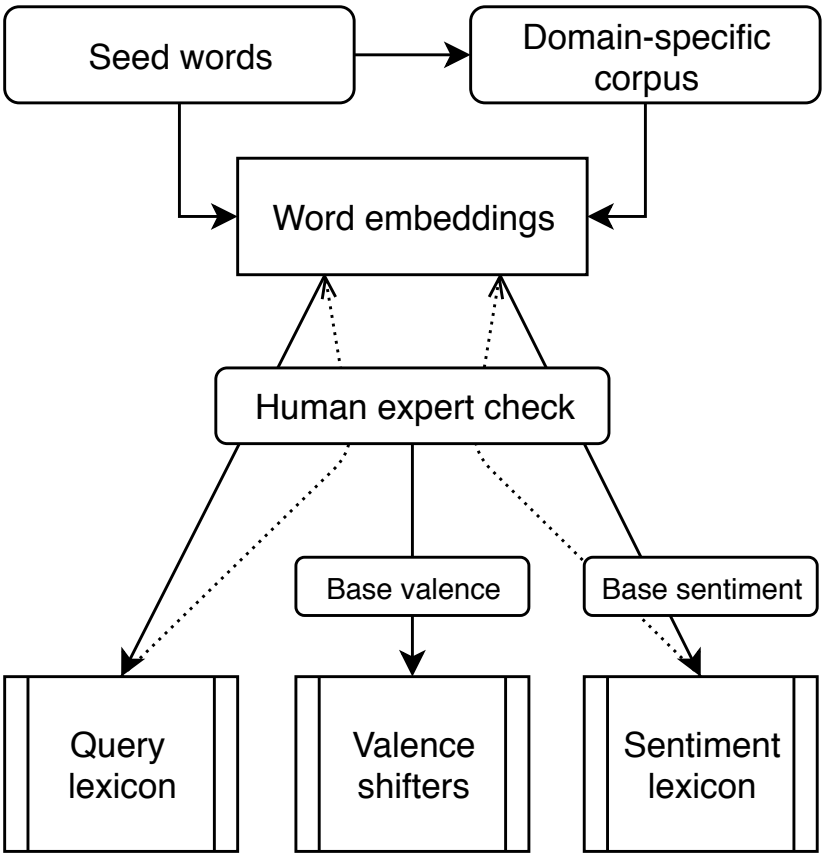


Figure 4.1 displays the high-level process of expanding an initial set of seed words into the final three types of keywords needed. The seed words are the backbone of the analysis. They are defined manually and should relate strongly to the study domain. Alternatively, they can be taken from an existing lexicon, as done in Theil *et al.* (2018) who start from the uncertainty terms in the Loughran & McDonald lexicon (Loughran and McDonald, 2011). The seed words include both query seed words, and sentiment seed words (often a subset of the former). The base valence and base sentiment word lists are existing dictionaries in need for a domain-specific twist to the application of interest.

All seed words are first used to query a more confined corpus from which the word embeddings will be estimated. The seed words are then expanded into the final query keywords by adding words that are similar, based on a ranking using the cs_{ij} metric, and a human check. The human expert chooses between keeping the word, discarding the word, or assigning the word as a valence shifter. The same step is done for the sentiment seed words. As sentiment lexicons are typically larger, the words from a base sentiment lexicon not too far from the obtained query lexicon are added as well. The words coming from the word embeddings might be considered more important, and thus weighted differently. The valence shifters are a combination of a base valence shifters list with the words assigned as valence shifter. Section 4.3.2.1 further explains the implementation for the ESG use case.

This keywords generation framework has as limitation that it only considers unigrams, i.e. single words. Maintaining a valence shifters list adds a contextual layer in the textual sentiment calculation, and the number of keywords present in an article is a good overall indicator of the ESG-relevance of news.

4.2.1.2 Database Querying

The database of texts is the large corpus that contains the subset of news relevant for the analysis. The task is to extract that subset as accurately as possible. The trade-off at play is that a large subset may guarantee full relevance, but it also adds more noise so it requires to think more carefully about the filtering step. In the process described in Figure 4.1, a first query is needed to obtain a decent domain-specific corpus to estimate the embeddings.

Once the final query lexicon is composed, the batch of articles including the words in this lexicon as well as the entities to analyze needs to be retrieved and stored. To avoid a very time-consuming query, the querying is best approached as a loop over pairs of a given entity and the query lexicon keywords. A **list of entities** with the exact names to extract needs to be curated, possibly dynamic over time to account for name changes. Only the articles in which at least one entity name and at least one of the keywords is present are returned.

4.2.1.3 News Filtering

Keywords-based extraction does not guarantee that all articles retrieved are pertinent. It must be expected that a considerable degree of noise still remains. For example, press articles about a thief driving a BMW is not ESG-worthy news about the company BMW. Therefore, we recommend the following negative filters:

- Removal of texts that have no connection with the topic to study, for example articles dealing with sports or lifestyle.
- Removal of articles that are too long (e.g. lengthy interviews) or too short (being more prone to a biased measurement of relevance and sentiment). Instead of removing

the longer than usual articles, one could proceed with the leading paragraph(s), or a summary.
- Removal of exact duplicated entries or highly related (near-duplicated) entries.
- Removal of texts that are subject to database-specific issues, such as articles with a wrong language tag.

The level of filtering is a choice of the researcher. For instance, one can argue to leave (near-)duplicates in the corpus if one wants to represent the total news coverage, irrespective of whether news rehashes an already published story or not. In this sense, it is also an option to reweight an article based on its popularity, proxied by the number of duplicates within a chosen interval of publication, or by the number of distinct sources expressing related news.

4.2.1.4 Indicators Construction

A corpus with N documents between daily time points $t = 1, \ldots, T$ has a $N \times p$ matrix $\boldsymbol{Z}$ associated to it. This matrix maps the filtered corpus for a given entity to p numerical metadata variables. It stores the values used for optional additional filtering, and ultimately for the aggregation into the time series indicators. Every row corresponds to a news article with its time stamp. The number of articles at time t is equal to N_t, such that $N \equiv N_1 + \ldots + N_T$.

The ultimate indices are obtained applying a function $f : \boldsymbol{Z} \mapsto \boldsymbol{I}$, where $\boldsymbol{I}$ is a $U \times P$ time series matrix that represents the "suite" of P final text-based indices, with $U \leq T$. The (linear or nonlinear) aggregation function depends on the use case.

Specific computation of the metadata and the aggregation into indices are elaborated upon in the application described in Section 4.3.

4.2.2 Validation and Decision Making

Not all ESG information is so-called material. The created indicators only become useful when explicitly mapped into practical and validated decision-making signals.

Qualitative validation involves surveying the news to assess remaining irrelevance of the articles. It also includes a graphical check in terms of peaks around the appearance of important events. Quantitative validation statistically measures the leading properties in regards to a certain target variable (e.g. existing sustainability scores), and the effectiveness of an investment strategy augmented with text-based information (in terms of out-of-sample risk and return, and the stability and interpretation of formed portfolios).

In a real-life setting, when wanting to know which companies face a changing sustainability profile ("positives") and which not ("negatives"), false positives are acceptable but false negatives are typically not, in the same vein doctors do not want to tell sick patients they are healthy. It is more important to bring up all cases subject to a potentially changed underlying ESG profile (capturing all the actual positives at the cost of more false positives), rather than

missing out on some (the false negatives) but bringing only the certain cases to the surface (merely a subset of the true positives). In machine learning classification lingo, this would mean aiming for excellent recall performance. An analyst will always proceed to investigation based on the signals received before recommending a portfolio action. Still, only an amount of signals that can reasonably be coped with should get through.

4.3 Monitoring the News About Company ESG Performance

In this section, we further motivate the integration of news-based ESG indices in sustainable investment practices. Secondly, we implement the described methodology, and validate its applicability.

4.3.1 Motivation and Applications

We believe there is a high added value of news-implied time-varying ESG indicators for asset managers and financial analysts active in both risk management and investment. These two main types of applications in the context of sustainable investment are motivated below.

4.3.1.1 Text-Based ESG Scoring as a Risk Management Tool

According to Riedl and Smeets (2017), social preferences are the driving factor behind why investors are willing to forgo financial performance when investing in SRI-compliant funds. This class of investors might be particularly interested in enhanced ESG risk management. An active sustainable portfolio manager should react appropriately when adverse news comes out, to avoid investors becoming worried, as the danger of reputational damage lurks.

The degree to which a company is sustainable does not change much at a high frequency, but unexpected events such as scandals may immediately cause a corporation to lose its ESG-compliant stamp. An investor relying on low-frequency rating updates may be invested wrongly for an extended time period. Thus, it seems there is the need for a timelier filter, mainly to exclude corporations that suddenly cease to be ESG-compliant. News-based indicators can improve this type of negative screening. In fact, both negative and positive ESG screening are considered among the most important future investment practices (Amel-Zadeh and Serafeim, 2018). A universe of stocks can be split into a sustainable and a non-sustainable subuniverse. The question is whether news-based indicators can anticipate a change in the composition of the subuniverses.

Portfolio managers need to be proactive by choosing the right response among the various ESG signals they receive, arriving from different sources and at different times. In essence, this makes them an "ESG signals aggregator." The more signals, the more flexibility in the

ESG risk management approach. An important choice in the aggregation of the signals is which value to put on the most timely signal, usually derived from news analysis. Final decisions regarding flipping the portfolio or not should always be aligned with the stipulated sustainable investment objectives.

Overall, the integration of textual data can lead to a more timely and a more conservative investment screening process, forcing asset managers as well as companies to continuously do well at the level of ESG transparency and ESG news presence.

4.3.1.2 Text-Based ESG Scoring as an Investment Tool

Increased investment performance may occur while employing suitable sustainable portfolio strategies, or strategies relying on textual information. These phenomena are not new, but doing both at the same time has been less frequently investigated. A global survey by Amel-Zadeh and Serafeim (2018) shows that the main reason for senior investment professionals to follow ESG information is investment performance. Their survey does not discuss the use of news-based ESG data. Investors can achieve improved best-in-class stock selection, or do smarter sector rotation. Targeted news-based indices can also be exploited as a means to tilt portfolios toward certain sustainability dimensions, in the spirit of Engle *et al.* (2020). All of this can generate extra risk-adjusted returns.

4.3.2 Pipeline Tailored to the Creation of News-Based ESG Indices

To display the methodology, we create text-based indices from press articles written in Dutch, for an assortment of European companies. We obtain the news data from the combined archive of the Belga News Agency and Gopress, covering all press sources in Belgium, as well as the major press outlets from the Netherlands. The data are not freely available.

The pipeline is incremental with respect to the companies and dimensions monitored. One can add an additional company or an extra sustainability (sub)dimension by coming up with new keywords and applying it to the corpus, which will result in a new specified time series output. This is important for investors that keep an eye on a large and changing portfolio, who therefore might benefit from the possibility of building the necessary corpus and indicators incrementally. The keywords and indicators can be built first with a small corpus, and then improved based on a growing corpus. Given historical availability of the news data, it is always easy to generate updated indicators for backtesting purposes. If one is not interested in defining keywords, one can use the keywords used in this work, available upon request.

4.3.2.1 Word Embeddings and Keywords Definition

We manually define the seed words drawing inspiration from factors deemed of importance by Vigeo Eiris and Sustainalytics, leading global providers of ESG research, ratings, and

Table 4.1: Dutch E, S, G and negative sentiment seed words.

E	S	G	sentiment[a]
milieu (*environment*), energie (*energy*), mobiliteit (*mobility*), nucleair (*nuclear*), klimaat (*climate*), biodiversiteit (*biodiversity*), koolstof (*carbon*), vervuiling (*pollution*), water, verspilling (*waste*), ecologie (*ecology*), duurzaamheid (*sustainability*), uitstoot (*emissions*), hernieuwbaar (*renewable*), olie (*oil*), olielek (*oil leak*)	samenleving (*society*), gezondheid (*health*), mensenrechten (*human rights*), sociaal (*social*), discriminatie (*discrimination*), inclusie (*inclusion*), donatie (*donation*), staking (*strike*), slavernij (*slavery*), stakeholder, werknemer (*employee*), werkgever (*employer*), massaontslag (*mass fire*), arbeid (*labor*), community, vakbond (*trade union*), depressie (*depression*), diversiteit (*diversity*)	gerecht (*court*), budget, justitie (*justice*), bestuur (*governance*), directie (*management*), omkoping (*bribery*), corruptie (*corruption*), ethiek (*ethics*), audit, patentbreuk (*patent infringement*), genderneutraal (*gender neutral*), witwaspraktijken (*money laundering*), dierproeven (*animal testing*), lobbyen (*lobbyism*), toploon (*top wage*)	vervuiling, verspilling, olielek, discriminatie, staking, slavernij, massaontslag, depressie, omkoping, corruptie, patentbreuk, witwaspraktijken

[a] These are a subset of the words in E, S and G.

data. Environmental factors are for instance climate change and biodiversity, Social factors are elements such as employee relations and human rights, and Governance factors are for example anti-bribery and gender diversity. We define a total of 16, 18 and 15 seed words for the Environmental, Social and Governance dimensions, respectively. Out of those, we take 12 negative sentiment seed words. There are no duplicates across categories. Table 4.1 shows the seed words.

The time horizon for querying (and thus training the word embeddings) spans from January 1996 to November 2019 included. The corpus is queried separately for each dimension using each set of seed words. We then combine into a large corpus, consisting of 4290370 unique news articles. This initial selection assures a degree of domain-specificity in the obtained

word vectors, as taking the entire archive would result in a too general embedding.

We tokenize the corpus into unigrams and take as vocabulary the 100000 most frequent tokens. A preceding cleaning step drops Dutch stop words, all words with less than 4 characters, and words that do not appear in at least 10 articles or in more than 10% of the corpus. We top the vocabulary with the 49 ESG seed words.

To estimate the GloVe word embeddings, we rely on the R package **text2vec** (Selivanov and Wang, 2018). We choose a symmetric context window of 7 words, and set the vector size to 200. Word analogy experiments in Pennington *et al.* (2014) show that a larger window or a larger vector size does not result in significantly better accuracy. Hence, this hyperparameters choice offers a good balance between expected accuracy and estimation time. In general, small context windows pick up substitutable words (e.g. due to enumerations), while large windows tend to better pick up topical connections. Creating the word embeddings is the most time-consuming part of the analysis, which might take from start to finish around half a day on a regular laptop. Figure 4.2 shows the fitted embedding space, shrunk down to two dimensions, focused on the seed words 'duurzaamheid' and 'corruptie.'

To expand the seed words, for every seed word in each dimension, we start off with the 25 closest words based on cs_{ij}, i.e. those with the highest cosine similarity. By hand, we discard irrelevant words, or tag words as an amplifying or as a deamplifying valence shifter. An example in the first valence shifter category is 'chronische' (*chronic*), an example in the second category is 'afgewend' (*averted*). We reposition duplicates to the most representative category. This leads to 197, 226 and 166 words respectively for the Environmental, Social and Governance dimensions.

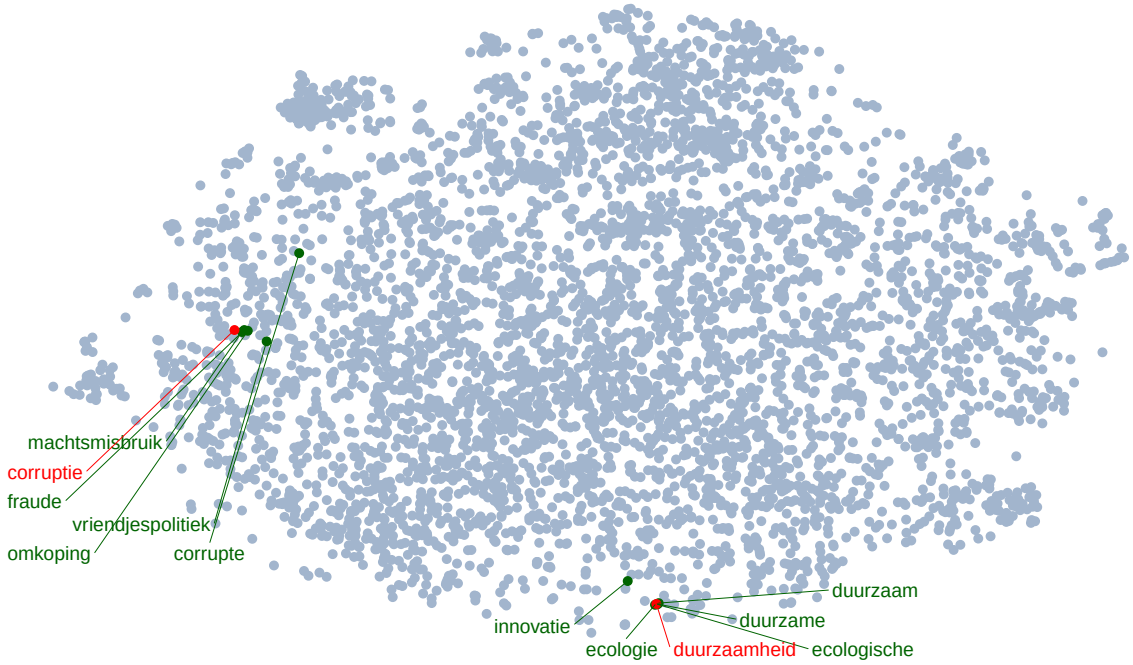
To expand the sentiment words, we take the same approach. The obtained words (151 in total) receive a polarity score of -2 in the lexicon. From the base lexicon entries that also appear in the vocabulary, we discard the words for which none of its closest 200 words is an ESG query keyword. If at least one of these top 200 words is a sentiment seed word, the polarity is set to -1 if not already. In total, the sentiment lexicon amounts to 6163 words, and we consider 84 valence shifters.

4.3.2.2 Company Selection and Corpus Creation

To query the news related to companies we use a reasonable trade-off between their commonplace name and their legal name.[38] Counting the total entity occurrences (measured by $n_{i,t}$, see Section 4.3.2.3) happens less strict by also accounting for company subnames. Our assumption is that often the full company name is mentioned once, and further references

[38] Suffixes (e.g. N.V. or Ltd) and too generic name parts (e.g. International) are excluded. We also omit companies with names that could be a noun or a place (for instance Man, METRO, Partners, Restaurant, or Vesuvius). Our querying system is case-insensitive, but case-sensitivity would solve the majority of this problem. We only consider fully merged companies, such as Unibail-Rodamco-Westfield and not Unibail-Rodamco.

FIGURE 4.2: Visualization of the embedding for a 5% fraction of the 100049 vocabulary words. The t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm implemented in the R package **Rtsne** (Krijthe and van der Maaten, 2018) is used with the default settings to reduce the 200-dimensional space to a two-dimensional space. In red, focal seed words 'duurzaamheid' and 'corruptie', and in green the respective five closest words according to the cosine similarity metric given the original high-dimensional word embeddings.



are made in an abbreviated form. As an example, to query news about the company Intercontinental Hotels we require the presence of 'Intercontinental' and 'Hotels', as querying 'Intercontinental' alone would result in a lot of unrelated news. To count the total matches, we consider both 'Intercontinental' and 'Intercontinental Hotels.'

We look at the 403 European companies that are included in both the Sustainalytics ESG dataset (ranging from August 2009 to July 2019) and (historically) in the S&P Europe 350 stock index between January 1999 and September 2018. The matching is done based on the tickers.

We run through all filters enumerated in Section 4.2.1.3. Articles without minimum 450 or with more than 12000 characters are deleted. To detect near-duplicated news, we use the locality-sensitive hashing approximate nearest neighbor algorithm (Leskovec *et al.*, 2014) as implemented in the R package **textreuse** (Mullen, 2016).

In total, 1453349 company-specific and sustainability-linked news articles are queried, of which 1022898 are kept after the aforementioned filtering. On average 33.4% of the articles

are removed. Most comes from the removal of irrelevant articles (20.5 p.p.), only a minor part is the result of filtering out too short and too long articles (6.4 p.p.). Pre-filtering, 42.2%, 71%, and 64.3% are marked belonging to the E, S or G dimension, respectively. Post-filtering, the distribution is similar (38.1%, 70.2%, and 65.9%). Additionally, we drop the articles which have only one entity mention. The total corpus size falls to 365319. The strictness of this choice is to avoid the inclusion of news in which companies are only mentioned in passing (Moniz, 2016). Furthermore, companies without at least 10 articles are dropped. We end up with 291 of the companies after the main filtering procedure, and move forward to the index construction with for each company a corpus.

4.3.2.3 Aggregation into Indices

As discussed in Section 4.2.1.4, we define a matrix $\boldsymbol{Z}_e$ for every entity e (i.e. a company) as follows:

$$\boldsymbol{Z}_e = \begin{bmatrix} n_{1,1} & n^E_{1,1} & n^S_{1,1} & n^G_{1,1} & a^E_{1,1} & a^S_{1,1} & a^G_{1,1} & s_{1,1} \\ n_{2,1} & n^E_{2,1} & n^S_{2,1} & n^G_{2,1} & a^E_{2,1} & a^S_{2,1} & a^G_{2,1} & s_{2,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n_{i,t} & n^E_{i,t} & n^S_{i,t} & n^G_{i,t} & a^E_{i,t} & a^S_{i,t} & a^G_{i,t} & s_{i,t} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n_{N^e-1,T} & n^E_{N^e-1,T} & n^S_{N^e-1,T} & n^G_{N^e-1,T} & a^E_{N^e-1,T} & a^S_{N^e-1,T} & a^G_{N^e-1,T} & s_{N^e-1,T} \\ n_{N^e,T} & n^E_{N^e,T} & n^S_{N^e,T} & n^G_{N^e,T} & a^E_{N^e,T} & a^S_{N^e,T} & a^G_{N^e,T} & s_{N^e,T} \end{bmatrix}. \tag{4.1}$$

Computed metadata for each news article are the number of times the company is mentioned (column 1), the total number of detected keywords for the E, S and G dimensions (columns 2 to 4), the proportions of the E, S and G keywords w.r.t. one another (columns 5 to 7), and the textual sentiment score (column 8). More specifically, n counts the number of entity mentions, n^E, n^S and n^G count the number of dimension-specific keywords, and s is the textual sentiment score. The proportion $a^d_{i,t}$ is equal to $n^d_{i,t}/(n^E_{i,t} + n^S_{i,t} + n^G_{i,t})$, for d one of the sustainability dimensions. It measures something distinct from keywords occurrence—for example, two documents can have the same number of keywords of a certain dimension yet one can be about one dimension only and the other about all three.

The sentiment score is calculated as $s_{i,t} = \sum_{j=1}^{Q_{i,t}} \omega_{j,i,t} v_{j,i,t} s_{j,i,t}$, where $Q_{i,t}$ is the number of words in article i at time t, $s_{j,i,t}$ is the polarity score for word j, $v_{j,i,t}$ is the valence shifting value applied to word j, and $\omega_{j,i,t}$ is a weight evolving as a U-shape across the document.[39] To do the sentiment computation we use the R package **sentometrics** (Ardia *et al.*, 2020).[40]

[39] Notably, $\omega_{j,i,t} = c\left(j - (Q_{i,t}+1)/2\right)^2$ with c a normalization constant. Words earlier and later in the document receive a higher weight than words in the middle of the document.

[40] See the accompanying package website at `https://sentometricsresearch.github.io/sentometr`

The metadata variables can also be used for further filtering, requiring, for instance, a majority proportion of one dimension in an article to include it. We divide $\boldsymbol{Z}_e$ into $\boldsymbol{Z}_{e,E}$, $\boldsymbol{Z}_{e,S}$ and $\boldsymbol{Z}_{e,G}$. In those subsets, we decide to keep only the news entries for which $n^d_{i,t} \geq 3$ and $a^d_{i,t} > 0.5$, such that each sustainability dimension d is represented by articles maximally related to it. This trims down the total corpus size to 166020 articles.[41]

For a given dimension d, the time series matrix that represents the suite of final text-based indices is a combination of 11 frequency-based and 8 sentiment-adjusted indicators. We do the full time series aggregation in two steps. This allows separating out the first simple from the subsequent (possibly time) weighted daily aggregation. We are also not interested in relative weighting within a single day, rather we will utilize absolute weights that are equally informative across the entire time series period.

We first create daily $T \times 1$ frequency vectors $\boldsymbol{f}$, $\boldsymbol{p}$, $\boldsymbol{d}$ and $\boldsymbol{n}$, and a $T \times 1$ vector $\boldsymbol{s}$ of a daily sentiment indicator. For instance, $\boldsymbol{f} = (f_1, \ldots, f_t, \ldots, f_T)'$ and $\boldsymbol{f}_{[k,u]} = (f_k, \ldots, f_t, \ldots, f_u)'$. The elements of these vectors are computed starting from the submatrix $\boldsymbol{Z}_{e,d}$, with at any time $N^{e,d}_t$ articles, as follows:

$$f_t = N^{e,d}_t, \quad p_t = 1/N^{e,d}_t \sum_{i=1}^{N^{e,d}_t} a^d_{i,t}, \quad d_t = \sum_{i=1}^{N^{e,d}_t} n^d_{i,t}, \quad n_t = \sum_{i=1}^{N^{e,d}_t} n_{i,t}. \tag{4.2}$$

For sentiment, $s_t = 1/N^{e,d}_t \sum_{i=1}^{N^{e,d}_t} s_{i,t}$. Missing days in $t = 1, \ldots, T$ are added with a zero value. Hence, we have that $\boldsymbol{f}$ is the time series of the number of selected articles, $\boldsymbol{p}$ is the time series of the average proportion of dimension-specific keyword mentions, $\boldsymbol{d}$ is the time series of the number of dimension-specific keyword mentions, and $\boldsymbol{n}$ is the time series of the number of entity mentions. Again, these are all specific to the dimension d.

The second step aggregates the daily time series over multiple days. The weighted frequency indicators are computed as $\boldsymbol{f}'_{[k,u]} \boldsymbol{B}_{[k,u]} \boldsymbol{W}_{[k,u]}$, with $\boldsymbol{B}_{[k,u]}$ a $(u-k+1) \times (u-k+1)$ diagonal matrix with the time weights $\boldsymbol{b}_{[k,u]} = (b_k, \ldots, b_t, \ldots, b_u)'$ on the diagonal, and $\boldsymbol{W}_{[k,u]}$ a $(u-k+1) \times 7$ metadata weights matrix defined as:

$$\boldsymbol{W}_{[k,u]} = \begin{bmatrix} p_k & g(d_k) & h(n_k) & p_k g(d_k) & p_k h(n_k) & g(d_k)h(n_k) & p_k g(d_k)h(n_k) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_t & g(d_t) & h(n_t) & p_t g(d_t) & p_t h(n_t) & g(d_t)h(n_t) & p_t g(d_t)h(n_t) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_u & g(d_u) & h(n_u) & p_u g(d_u) & p_u h(n_u) & g(d_u)h(n_u) & p_u g(d_u)h(n_u) \end{bmatrix}, \tag{4.3}$$

`ics` for code examples, and the survey paper by Algaba *et al.* (2020a) about the broader sentometrics research field concerned with the construction of sentiment indicators from alternative data such as texts.

[41] For some companies the previous lower bound of 10 news articles is breached, but we keep them aboard. The average number of documents per company over the embedding time horizon is 571.

where $g(x) = \ln(1+x)$ and $h(x) = x$. In our application, we choose to multiplicatively emphasize the number of keywords and entity mentions but alleviate the effect of the first, as in rare cases disproportionately many keywords pop up. The value p_t is a proportion between 0 and 1 and requires no transformation. The aggregate for the last column is $\sum_{t=k}^{u} f_t b_t p_t \ln(1+d_t) n_t$, for instance.

The aggregations repeated for $u = \tau, \ldots, T$, where τ pinpoints the size of the first aggregation window, gives the time series. They are assembled in a $U \times 7$ matrix of column vectors. Every vector represents a different weighting of the obtained information in the text mining step.

We opt for a daily moving fixed aggregation window $[k,u]$ with $k \equiv u - \tau + 1$. As time weighting parameter, we take $b_t = \alpha_t / \sum_{t=k}^{u} \alpha_t$, with $\alpha_t = \exp\left(0.3\left(\frac{t}{\tau} - 1\right)\right)$. We set τ to 30 days. The chosen exponential time weighting scheme distributes half of the weight to the last seven days in the 30-day period, therefore ensuring that peaks are not averaged away. To omit any time dynamic, it is sufficient to set $b_t = 1$.

The non-weighted frequency measures for time u are computed as $\boldsymbol{b}'_{[k,u]} \boldsymbol{A}_{[k,u]}$, where $\boldsymbol{A}_{[k,u]}$ is a $(u-k+1) \times 4$ weights matrix defined as:

$$\boldsymbol{A}_{[k,u]} = \begin{bmatrix} \boldsymbol{f}_{[k,u]} & \boldsymbol{p}_{[k,u]} & \boldsymbol{d}_{[k,u]} & \boldsymbol{n}_{[k,u]} \end{bmatrix}. \tag{4.4}$$

The frequency-based time series indicators are all stored into a $U \times 11$ matrix.

The computation of the (weighted) sentiment values follows the same logic as described and results in a $U \times 8$ matrix. The final indices combined are in a $U \times 19$ matrix $\boldsymbol{I}_{e,d}$. We do this for the three ESG dimensions, for a total of 57 unique text-based sustainability indicators, for each of the 291 companies.

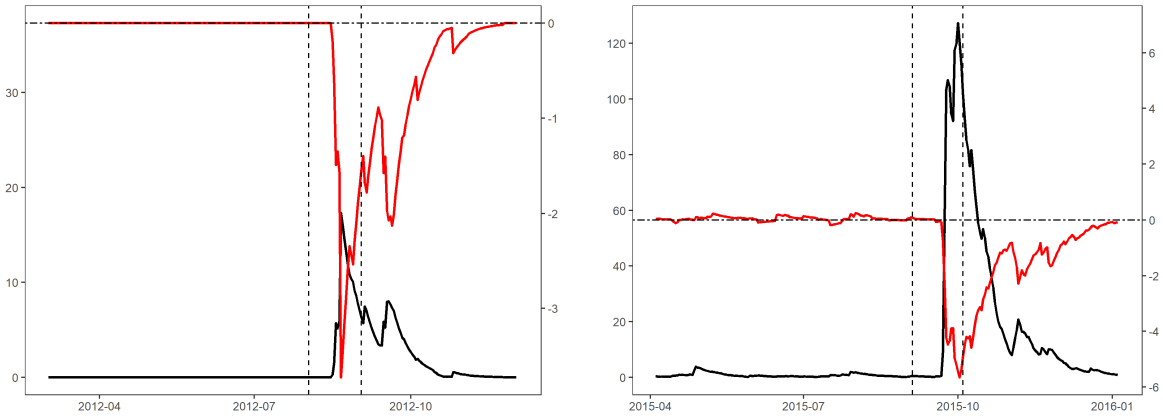
4.3.2.4 Validation

We first present a couple of sustainability crisis cases and how they are reflected in our indicators relative to the scores from Sustainalytics. Figure 4.3 shows the evolution of the indicators for the selected cases. For comparison, Figure 4.4 shows for these same cases the corresponding monthly Sustainalytics ratings.

Figure 4.3a displays Lonmin, a British producer of metals active in South Africa, whose mine workers and security were at the centre of strikes mid-August 2012 leading to unfortunate killings. This is a clear example of a news-driven sustainability downgrade. It was picked up by our constructed news indicators, in that news coverage went up and news sentiment went down, and later reflected in a severe downgrade by Sustainalytics in their Social score. Similar patterns are visible for the Volkswagen Dieselgate case (Figure 4.3b), for the Libor manipulation scandal (Figure 4.3c, with besides Barclays also other financial institutions impacted), and for a corruption lawsuit at Finmeccanica (Figure 4.3d).[42]

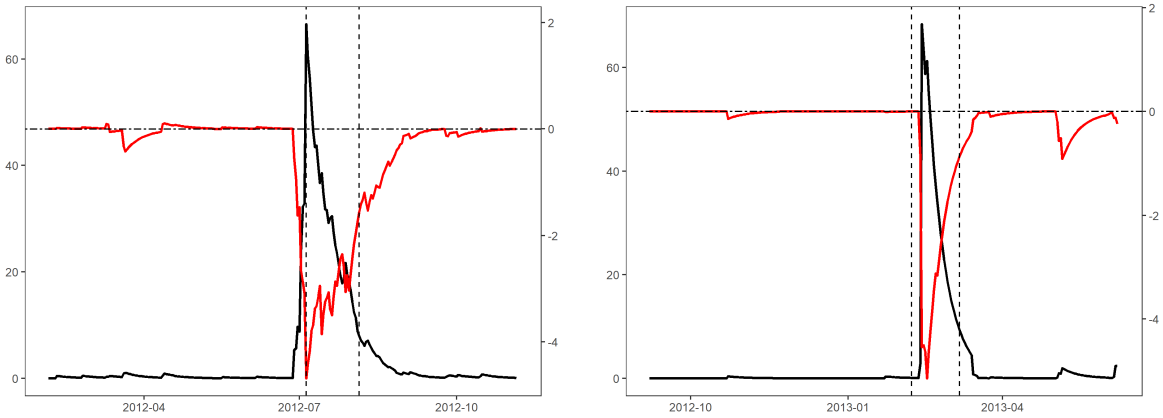
[42] The examples can also be framed as exposing "controversies." Some ESG data providers compute explicit

FIGURE 4.3: News-based indicators around a selection of severe Sustainalytics downgrades (a drop larger than 5 on their 0–100 scale). The vertical bars indicate the release date of the downgraded score, and one month before. The time frame shown is six months prior and three months after the release date. In black the average of the 11 frequency-based indicators (left axis), in red of the 8 sentiment-based measures (right axis, with a horizontal line through zero).



(a) Lonmin (Social)

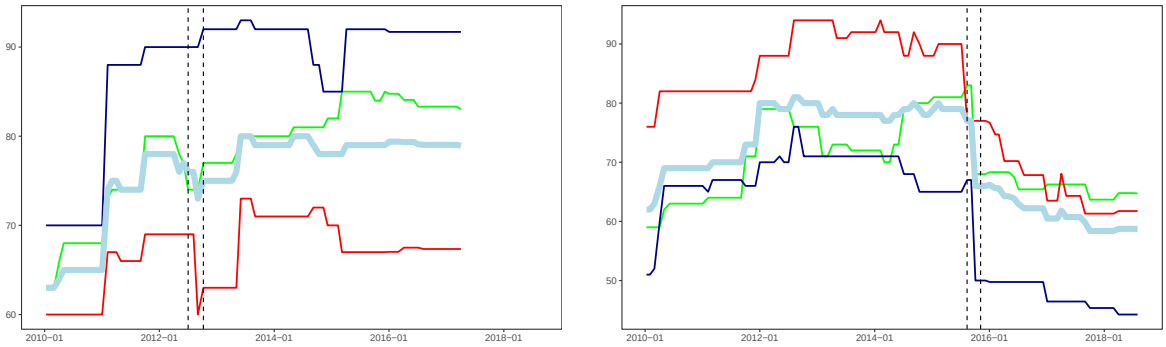
(b) Volkswagen (ESG)



(c) Barclays (Governance)

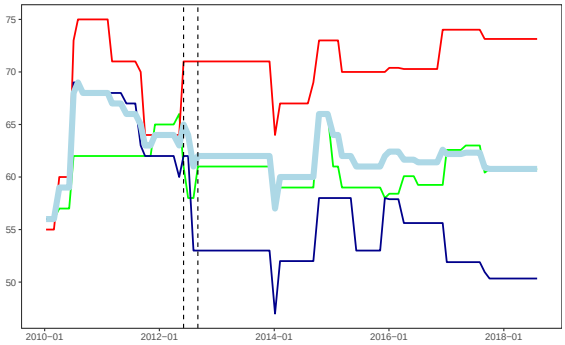
(d) Finmeccanica (Governance)

FIGURE 4.4: Time series of the available monthly Sustainalytics scores for the companies presented in Figure 4.3. In green the E scores, in red the S scores, in dark blue the G scores, and in thick light blue the ESG scores. The vertical bars show the period around the severe downgrade as in Figure 4.3, but with one month added at both ends for clarity. The average scores across all 291 companies between January 2010 and August 2018 are 64.53 (E), 64.91 (S), 68.24 (G), and 65.45 (ESG).

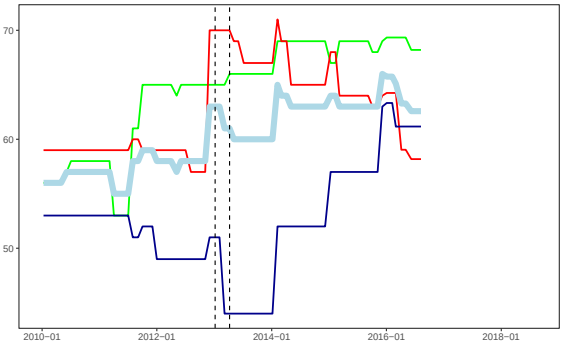


(a) Lonmin

(b) Volkswagen



(c) Barclays



(d) Finmeccanica

The main conclusions are the following. First, not all Sustainalytics downgrades (or sustainability changes in general) are covered in the press. Second, our indicators pick up severe cases faster, avoiding the lag of a few weeks or longer before adjustments in Sustainalytics scores are observed. The fact that media analysis does not pick up all events, but when it does, it does so fast(er), is a clear argument in favor of combining news-based ESG data with traditional ESG data.

In these illustrations, the general pattern is that the peak starts to wear out before the change in Sustainalytics score is published. Smoother time scaling would result in peaks occurring later, sometimes after the Sustainalytics reporting date, as well as phasing out slower (i.e. more persistence). This is because the news reporting is often clustered and spread out

(discrete) metrics to inform about current and likely future controversies, often mainly based on news data. For future research, it could be interesting to see to what extent the news measures in this work relate to these metrics.

over several days. Likewise, an analysis run without the strict relevance filtering revealed less obvious peaks. Therefore, for (abnormal) peak detection, we recommend short-term focused time weighting and strict filtering.

We restate that the indicators do not replace traditional ESG scores, but should be seen as alerts. Some of the news shocks coincide with structural changes, others may indicate only temporary struggles. In both cases, the news-based time series will revert. This is, as argued above, in part constructed behavior, but also inherent to news data. News coverage will in most situations eventually fade away. Mixed-frequency modeling is one way to actually nowcast existing ESG scores based on news data. This approach also allows disentangling the temporary from the structural changes, or can be used to decide how much focus to put on each type of shock. An example is Algaba *et al.* (2020b) who combine a monthly consumer confidence indicator with news data into daily nowcasts of consumer confidence.

In addition to the qualitative validation of the indicators, we present one possible way to quantitatively measure their ability to send early warnings for further investigation. We perform an ex-post analysis. Early warnings coming from the news-based indicators are defined as follows. We first split the period prior to a downward re-evaluation by Sustainalytics (a drop larger than 5) into two blocks of three months. The first three-month block is the reference period. The indicator values in the second three-month block are continuously benchmarked against an extreme outcome of the previous block. For the frequency-based indicators, a hypothetical early warning signal is sent when the indicator surpasses the 99% quantile of the daily values in the reference block. For the sentiment-based indicators, a signal is sent if the indicator dips below the 1% reference quantile. Less signals will be passed on if the cut-offs are more extreme, but they will more likely be relevant.

Table 4.2 displays the results of the analysis for the averaged frequency-based and sentiment-based indicators. Between 11% to 34% of downgrades correspond with more abnormal news dynamics as defined. When so, on average about 50 days ahead of a realized downgrade, an initial news-based early warning is sent. Note that these early warnings should be interpreted as reasonable *first* signals, not necessarily the optimal ones, nor the only ones. There is ample room to finetune these metrics, and especially the amplitude of the signals generated in line with investment needs, as hinted to in Section 4.2.2.

4.3.3 Stock and Sector Screening

Another test of the usefulness of the created indices is to input them in a sustainable portfolio construction strategy. This allows studying the information content of the indices in general, of the different types of indices (mainly frequency-based against sentiment-based), and of the three ESG dimensions. The analysis should be conceived as a way to gauge the value of using textual data science to complement standard ESG data, not as a case in favor of ESG investing in itself.

Table 4.2: Ex-post early warning ability of news-based indicators.

		detected		time gain (days)	
	events	f	s	f	s
E	53%	19%	11%	48	48
S	53%	34%	24%	52	52
G	63%	25%	19%	51	46
ESG	24%	28%	18%	52	47

This table shows ex-post early warning performance statistics. The "events" column is the proportion of the 291 companies analyzed that faced at least one substantial Sustainalytics downgrade released at a day t_D. The "detected" column is the proportion of downgrades for which minimum one early warning was generated within three months before t_D. The "time gain (days)" column is the average number of days the first early warning precedes t_D. The analysis is done for the average of the 11 frequency-based indicators (f), and of the 8 sentiment-based measures (s).

We run a small horse race between three straightforward monthly screening strategies. The investable universe consists of the 291 analyzed companies. The strategies employed are the following:

- Invest in the 100 top-performing companies. [S1]
- Invest in the companies excluding the 100 worst-performing ones. [S2]
- Invest in the companies in the 10 top-performing sectors. [S3]

All strategies equally weight the monthly rebalanced selection of companies. We include 24 sectors formed by combining the over forty peer groups defined in the Sustainalytics dataset. The notion of top-performing (resp. worst-performing) means having, at rebalancing date, the lowest (resp. the highest) news coverage, or the most positive (resp. the most negative) news sentiment. The strategies are run with the indicators individually for each ESG dimension. To benchmark, we run the strategies using the scores from Sustainalytics, and also compare with a portfolio equally invested in the total universe.

We take the screening one step further by imposing for all three strategies that companies should perform among the best both according to the news-based indicators and according to the ratings from Sustainalytics. We slightly modify the strategies per approach to avoid retaining a too limited group of companies; strategy S1 looks at the 150 top-performing companies, strategy S2 excludes the 50 worst-performing companies, and strategy S3 picks the 15 top-performing sectors. The total investment portfolio consists of the intersection of the selected companies by the two approaches.

We split the screening exercise in two out-of-sample time periods. The first period covers February 1999 to December 2009 (131 months), and the second period covers January 2010 to August 2018 (104 months). The rebalancing dates are at every end of the month, and range from January 1999 to July 2018.[43] To screen based on our news-based indicators, we take the daily value at rebalancing date. For the Sustainalytics strategy, we take the most recently available monthly score, typically dating from two to three weeks earlier.

An important remark is that to estimate the word embeddings we use a dataset whose range (i.e. January 1996–November 2019) is greater than that of the portfolio analysis. This poses a threat of lookahead bias—meaning, at a given point in time, we will have effectively already considered news data beyond that time point. This would be no problem if news reporting style is fixed over time, yet word use in news, and thus its relationships in a high-dimensional vector space, are subject to change.[44] It would be more correct (but also more compute intensive) to update the word embeddings rolling forward through time, for example, once a year. The advantage of a large dataset is an improved overall grasp of the word-to-word semantic relationships. Assuming the style changes are minor, and given the wide scope of our dataset, the impact on the outcome of the analysis is expected to be small.

4.3.3.1 Aggregate Portfolio Performance Analysis

We analyze the strategies through aggregate comparisons.[45] The results are summarized in Table 4.3. We draw several conclusions.

First, in both subsamples, we notice comparable or better performance for the S2 and S3 investment strategies versus the equally-weighted portfolio. The sector screening procedure seems especially effective. Similarly, we find that our news indicators, both the news coverage and the sentiment ones, are a more valuable screening tool, in terms of annualized Sharpe ratio, than using Sustainalytics scores. The approach of combining the news-based signals with the Sustainalytics ratings leads for strategies S1 and S2 to better outcomes compared to relying on the Sustainalytics ratings only. Most of the Sharpe ratios across ESG dimensions for the combination approach are close to the unscreened portfolio Sharpe ratio. The worst-in-class exclusion screening (strategy S2) performs better than the best-in-class inclusion screening (strategy S1), of which only part is explained by diversification benefits.

[43] Within this first period, the effective corpus size is 87611 articles. Within the second period, it is 60977 articles. The two periods have a similar monthly average number of articles.

[44] An interesting example is the Guardian who declared in May 2019 to start using more often 'climate emergency' or 'climate crisis' instead of 'climate change.'

[45] As a general remark, due to the uncertainty in expected return estimation, the impact of any sustainability filter on the portfolio performance (e.g. the slope of the linear function Boudt *et al.* (2013) derive to characterize the relationship between a sustainability constraint and the return of mean-tracking error efficient portfolios) is hard to evaluate accurately.

Table 4.3: Sustainable portfolio screening (across strategies).

(a) News engine

		E		S		G		ESG	
		f	*s*	*f*	*s*	*f*	*s*	*f*	*s*
P1	S1	0.50	0.46	**0.59**	0.40	**0.57**	0.50	**0.55**	0.46
	S2	**0.53**	0.49	**0.61**	0.47	**0.60**	**0.54**	**0.58**	0.50
	S3	**0.65**	0.46	**0.76**	0.44	**0.62**	**0.59**	**0.69**	0.50
P2	S1	0.88	0.81	0.93	0.85	0.91	0.86	0.91	0.84
	S2	**1.03**	0.99	0.99	**1.02**	**1.04**	**1.03**	**1.02**	**1.01**
	S3	**1.11**	**1.02**	**1.02**	0.98	0.99	**1.17**	**1.06**	**1.08**
All	S1	0.64	0.59	**0.72**	0.56	**0.70**	0.63	0.69	0.60
	S2	**0.71**	0.68	**0.76**	0.67	**0.77**	**0.73**	**0.75**	0.69
	S3	**0.82**	0.66	**0.86**	0.64	**0.76**	**0.81**	**0.82**	**0.71**

(b) Sustainalytics

		E	S	G	ESG
P2	S1	0.81	0.82	0.98	0.88
	S2	0.92	0.91	0.98	0.94
	S3	**1.07**	0.89	0.98	**1.00**

(c) News engine + Sustainalytics

		E		S		G		ESG	
		f	*s*	*f*	*s*	*f*	*s*	*f*	*s*
P2	S1	0.93	0.86	0.84	0.79	**1.09**	**1.08**	0.96	0.92
	S2	0.97	0.94	0.99	0.98	0.99	0.98	0.99	0.97
	S3	0.41	0.93	0.33	**1.03**	0.46	0.85	0.41	0.98

Table 4.3a shows the annualized Sharpe ratios for all strategies (S1–S3), averaged across the strategies on the 11 frequency-based indicators (f), and on the 8 sentiment-based indicators (s). The ESG column invests equally in the related E, S and G portfolios. Table 4.3b shows the Sharpe ratios for all strategies using Sustainalytics scores. Table 4.3c refers to the strategies based on the combination of both signals. P1 designates the first out-of-sample period (February 1999 to December 2009), P2 the second out-of-sample period (January 2010 to August 2018), and All the entire out-of-sample period. An equally-weighted benchmark portfolio consisting of all 291 assets obtains a Sharpe ratio of 0.52 (annualized return of 8.4%), of 1.00 (annualized return of 12.4%), and of 0.70 (annualized return of 10.1%) over P1, P2, and All, respectively. The screening approaches performing at least as good as the unscreened portfolio are indicated in bold.

There seems to be no performance loss when applying news-based sustainability screening. It is encouraging to find that portfolios based on simple universe screening procedures contingent on news analysis are competitive with an unscreened portfolio, and with screenings based on ratings from a reputed data provider.

Second, the indicators adjusted for sentiment are not particularly more informative than the frequency-based indicators. On the contrary, in the first subsample, the news coverage indicators result in higher Sharpe ratios. Not being covered (extensively) in the news is thus a valid screening criterion. In general, however, there is little variability in the composed portfolios across the news-based indicators, as many included companies simply do not appear in the news, and thus the differently weighted indices are the same.

Third, news has in both time periods satisfactory relative value. The Sharpe ratios are low in the first subsample due to the presence of the global financial crisis. The good performance in the second subperiod confirms the universally growing importance and value of sustainability screening. It is also consistent with the study of Drei *et al.* (2019), who find that, between 2014 and 2019, ESG investing in Europe led to outperformance.

Fourth, the utility of each dimension is not uniform across time or screening approach. In the first subperiod, the Social dimension is best. In the second period, the Governance dimension seems most investment worthy, but closely followed by the other dimensions. Drei *et al.* (2019) observe an increased relevance of the Environmental and Social dimensions since 2016, whereas the Governance dimension has been the most rewarding driver overall (Bennani *et al.*, 2018). An average across the three dimension-specific portfolios also performs well, but not better.

The conclusions stay intact when looking at the entire out-of-sample period, which covers almost twenty years.

4.3.3.2 Additional Analysis

Framing the main analysis in a Sharpe peer performance framework (Ardia and Boudt, 2018), the results are qualitatively similar. In the first subsample, there are 228 strategies in total (3 strategies for 19 indicators with 4 portfolios). Of these, 46% have a statistically superior Sharpe ratio over the equally-weighted benchmark, and 28% have an indistinguishable performance. In the second subsample, we add the combination strategies. The proportion of sustainable strategies that perform as good or better is 65%. In general, we corroborate that there is little loss of performance value when applying sustainable screening that involves news indicators.

We also assess the value of the different weighting schemes. Table 4.4 shows the results for strategy S3 across the 8 sentiment indices, in the second period. It illustrates that the performance discrepancy between various weighting schemes for the sentiment indicators is not clear-cut. More complex weighting schemes, in this application, do not clearly beat the simpler weighting schemes.

Table 4.4: Sustainable portfolio screening (across sentiment indicators).

	s1	*s2*	*s3*	*s4*	*s5*	*s6*	*s7*	*s8*
E	1.01	1.04	0.99	0.99	1.04	1.02	1.04	1.02
S	0.98	0.99	0.98	1.00	0.98	0.95	0.95	1.01
G	1.14	1.17	1.20	1.20	1.16	1.19	1.09	1.17
ESG	1.06	1.09	1.08	1.08	1.08	1.08	1.05	1.08

This table shows the Sharpe ratios in P2 for screening strategy S3, built on the sentiment-based indicators, being *s1*, and *s2*–*s8* as defined through the weighting matrix in (4.3).

An alternative approach for the strategies on the frequency-based indicators is to invert the ranking logic, so that companies with a high news coverage benefit, and low or no news coverage is penalized. We run this analysis but find that the results worsen markedly, indicating that attention in the news around sustainability topics is not a good screening metric.

To test the sensitivity to the strict filtering choice of leaving out articles not having at least three keywords and more than half of all keywords related to one dimension, we rerun the analysis keeping those articles in. Surprisingly, some strategies improve slightly, but not all. We did not examine other filtering choices.

We also tested a long/short strategy but the results were poor. The long leg performed better than the short leg, as expected, but there was no reversal effect for the worst-performing stocks.

Other time lag structures (different values for τ, or different functions in $\boldsymbol{B}$) are not tested, given this would make the analysis more a concern of market timing than of assessing the lag structure. A short-term indicator catches changes earlier, but they may have already worn out by the rebalancing date, whereas long-term indicators might still be around peak level, or not yet. We believe finetuning the time lag structure is more crucial for peak detection and visualization.

4.3.3.3 Possible Extensions

The analysis can be further improved. An interesting extension would be to compute news-based investment beta's that measure the sensitivity of a stock against factor portfolios created from news-based signals, or against the news indices themselves. The beta's can be estimated through a single-factor or a multi-factor return regression. Another extension would be to not consider levels of the indicators, but (cumulative) idiosyncratic shocks, as obtained from an autoregressive model. We also point to the potential rewards of applying portfolio weights optimization on a screened stock universe, for instance, at sector level.

4.4 Conclusion

This chapter presents a methodology to create frequency-based and sentiment-based indicators to monitor news about given topics and entities. We apply the methodology to extract company-specific news indicators relevant to Environmental, Social and Governance matters. These indicators can be used to timely detect abnormal dynamics in the ESG performance of companies, as an input in risk management and investment screening processes. They are not calibrated to automatically make investment decisions. Rather, the indicators should be seen as an additional source of information to the asset manager, or other decision makers.

We find that the indicators often anticipate substantial negative changes in the scores of the external ESG research provider Sustainalytics. Moreover, we also find that the news indices can be used as a sole input to screen a universe of stocks and construct simple but well-performing investment portfolios. In light of the active sustainable investment manager being an "ESG ratings aggregator", we show that combining the news signals with the scores from Sustainalytics leads to a portfolio selection that performs equally well as the entire universe.

Given the limited reach of our data (we use Flemish and Dutch news to cover a wide number of European stocks), better results are expected with geographically more representative news data as well as a larger universe of stocks. Hence, the information potential is promising. It would be useful to investigate the benefits local news data bring for monitoring companies with strong local ties.

Additional value to explore lies in more meaningful text selection and index weighting. Furthermore, it would be of interest to study the impact of more fine-grained sentiment calculation methods. Summarization techniques and topic modeling are interesting text mining tools to obtain a drill-down of sustainability subjects, or for automatic peak labeling. An analysis on news source contribution could be worthwile to find out which publication venues are driving the relevant news.

News analysis can also further illuminate the story of Hartzmark and Sussman (2019), who show that investors put more money into funds with a higher level of sustainability. The saliency of news metrics can help to more clearly identify why and when investors do so.

The Economic Policy Uncertainty Index for Flanders, Wallonia and Belgium 5

Abstract

This research note describes the construction of news-based Economic Policy Uncertainty (EPU) indices for Flanders, Wallonia and Belgium. The indices are computed from January 2001 until May 2020. Important domestic and more global events coincide with spikes in the indices. The COVID-19 pandemic represents the highest point, reflecting very strong consecutive Belgian newspaper attention to economic policy uncertainty. The monthly values of the EPU indices for Flanders, Wallonia and Belgium are published on `www.policyuncertainty.com`.

5.1 Introduction

In their seminal work, Baker *et al.* (2016) propose a novel approach for measuring economic policy uncertainty (EPU) based on newspaper coverage. The coverage is measured by words related to the economy, policy, or uncertainty. The normalized volume of news articles containing these words are considered a good measure of the prevailing underlying uncertainty regarding economic policy. Their focus is on the construction of an index for the U.S., from the country's ten largest newspapers.

Since, many have applied this methodology to construct text-based EPU indices for various geographies. For instance, Kroese *et al.* (2015) create an EPU index for the Netherlands, and Ghirelli *et al.* (2019) do so for Spain. Some have tried to improve on the methodology. Tobback *et al.* (2018) attempt to develop an EPU index for Belgium using, among others, support vector machine classification. However, their initiative to regularly publish index updates fell short due to data scraping issues. Azqueta-Gavaldon *et al.* (2020) extend the original methodology by employing two powerful machine learning methods (word embeddings and topic models).

This note presents three EPU indices constructed for Belgium based on a large archive of news articles. Belgium is situated in the very centre of Europe, split into the Flanders, Wallonia, and Brussels-Capital regions. The respective official languages are Flemish (a variant of Dutch), French, and both in Brussels. The press landscape is divided according to these two languages as well, allowing for a separate construction of two regional indices, before merging the two indices into a Belgium countrywide index.

The monthly EPU indices for Belgium correlate well with other uncertainty indicators. To label the sources of uncertainty, we present a technique to automatically extract non-EPU key terms in news. During 2020 and the COVID-19 pandemic, we reveal a clear daily build-up of news-based economic policy uncertainty.

5.2 Implementation

The generic methodological steps for the construction of monthly news-based EPU indices as proposed in Baker *et al.* (2016) are:

1. Select the newspapers of interest.
2. Count the number of newspaper articles containing at least one economic (E) keyword, one policy (P) keyword, and one uncertainty (U) keyword, in the native language of the newspaper in question. These are the raw number of EPU articles.
3. Scale the raw EPU count by a measure of the total number of articles in the same newspaper and month.

4. Standardize each newspaper-level monthly series to unit standard deviation prior to a certain date, and average across newspapers by month.
5. Normalize to a mean of 100 prior to a certain date. Perform other index scaling if deemed useful.
6. If applicable, average across a set of final indices to obtain an aggregated index (e.g. GDP-weighted per country, or, as in our case, per language).

Baker *et al.* (2016) emphasize that with each monthly update, data from the preceding (two) months may be revised marginally, driven by the fact that some newspapers do not immediately update their online archives with all articles.

The specific implementation requires choices in terms of news data provider, newspaper selection, keywords, and reference period for the standardization. Ghirelli *et al.* (2019) show the sensitivity of the index construction to the amount of newspapers considered, and the number of keywords. We accord with their guidelines by considering more than two newspapers, and enlarging the set of keywords used.

5.2.1 Data

We obtain the news articles for Belgium from the national Belgian News Agency (Belga). Their archive contains over 40 million media news articles in Flemish and French starting from 2001 until now.

We include following eight Flemish newspapers: “De Tijd", “De Standaard", “De Morgen", “Het Laatste Nieuws", “Het Nieuwsblad", “Gazet van Antwerpen", “Het Belang van Limburg", and “Het Volk." The newspaper “Het Volk" seized activities in 2008 but is an historically important news source. We include following five Walloon newspapers: “L’Avenir", “La Dernière Heure", “La Libre Belgique", “Le Soir", and “L’Echo."

We clean the news data by filtering out exact duplicate articles, and remaining articles of no relevance to economic policy uncertainty (e.g. sports or arts news). Near-duplicate entries are kept, as these are often the same publications but by different newspapers. We also trim too short (up to 450 characters) and too long (from 7500 characters) articles, as these are more sensitive to a biased measurement of EPU relevance. Between January 2001 and May 2020, we select around 109000 articles from the Flemish-speaking press, and around 81800 articles from the French-speaking press.

The database archive we use has no news available during 2006 for none of the Walloon newspapers. We decide to encode the resulting monthly index values as missing, and carry this forward to the Belgium index. Alternatively, we could have performed interpolation. Simple linear interpolation across a full year would be uninformative. Proportional interpolation (filling in the missing values in the Wallonia series by mimicking the trend of the Flanders

series) would suffer from lookahead bias as one series is based on future known values of another series.

It is also important to highlight that we exclude online newspaper content, for two main reasons. First, to have comparability of our data universe across time, as online publications have become growingly prevalent compared to the earlier years of the time period covered. Second, because web publications typically serve other (monetization) purposes, and might therefore be different when it comes to the level of uncertainty expressed. It would also involve a more complex management of possible news duplicates. The differences between offline and online news content in light of uncertainty coverage could be an interesting study in itself.

5.2.2 Keywords

The entire set of EPU keywords decide on the selection of articles to take into account. The E and P categories are required to trim down the news to the correct topic, in this case reporting about economic policy, and the U category can be seen as the “sentiment" driver.

We start from the Dutch word list provided to us by Kroese *et al.* (2015), who based themselves on the original paper of Baker *et al.* (2016). Terms specific for the Netherlands are deleted. We also limit the keywords to single words (called unigrams). We use a pretrained word embedding space on a Flemish corpus to generate candidate words to make the Flemish keywords more comprehensive.[46] The expanded Flemish word list is translated to a French word list. Both lists of keywords are checked manually to omit remaining dubious entries. The keywords are made available in Appendix 7.6.

5.2.3 Computation

We follow the computation and normalization approach as explained. We normalize the newspaper-level series to unit standard deviation for a reference period up to 2011 (i.e. divide each index by its standard deviation). As a final normalization, we bring the mean of the averaged series before 2011 to a level of 100 (i.e. divide the index by its mean and multiply by 100). The Belgium EPU index is a simple average of the resulting Flanders and Wallonia EPU indices.

[46] A word embedding space maps words into high-dimensional vectors. Similar words have a shorter distance between their vectors. The embedding we use is recycled from another project (hence, pretrained), and was fit with the GloVe technique (Pennington *et al.*, 2014).

5.3 Analysis

We validate the Belgian EPU indices in four ways. First, we plot the series and label important peaks. Second, we extract frequent terms of EPU articles around some of such peak periods. Third, we gauge the correlation between our indices and other related uncertainty time series. Fourth, we discuss alternative index creation methods. Additionally, in the last subsection, we dive deeper into the months covering the COVID-19 pandemic.

5.3.1 Belgian EPU Time Series and Events

Figure 5.1 displays the three final EPU indices for Belgium. Figure 5.1a compares the Flemish to the Walloon EPU index, including an annotation of noteworthy events, whereas Figure 5.1b plots the resulting index for Belgium. We perceive both peaks related to local events, and peaks related to European and worldwide events.

The global financial crisis starting in 2007 up to 2009 comes with a clear peak in the EPU indices. The period around the European debt crisis as well, with two peak moments (one with the Greek bailout, and one at a later stage of the crisis). Around the first Brexit referendum (mid-2016) and after, the indices bear volatile but not particularly high values. The crisis related to the COVID-19 pandemic reaches a level higher than the financial crisis. The uncertainty also increased from March to April. The surge in uncertainty as a result of the pandemic is also documented by Baker *et al.* (2020). The situation is unprecedented and not limited to the financial sector but impacts the entire economy. From April to May, there is a small decrease.

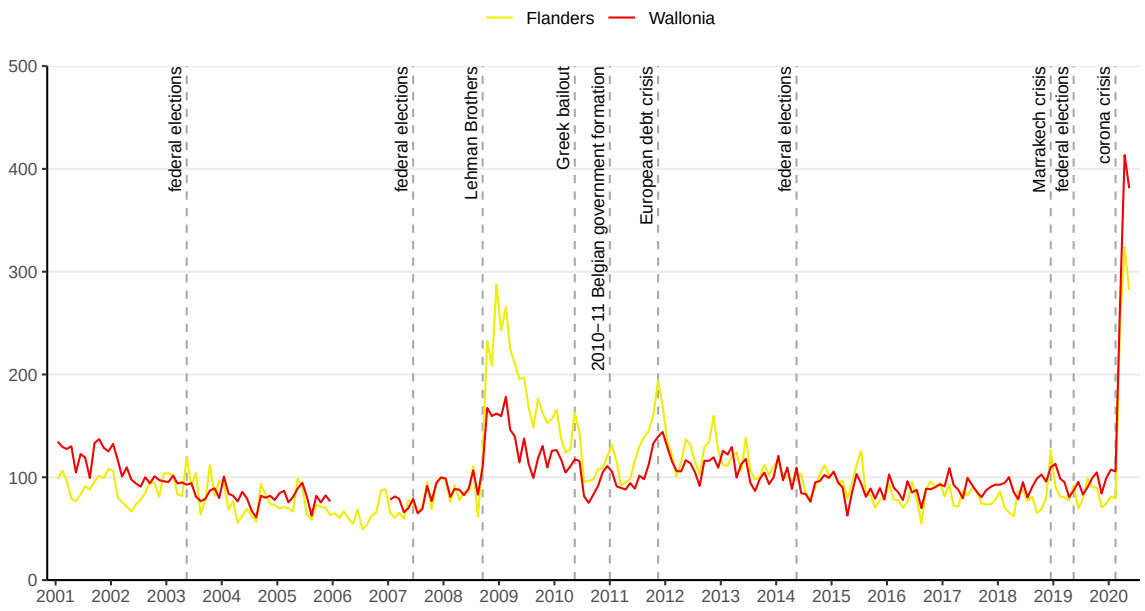
There are peaks around federal elections, albeit minimal ones. More apparent is the increase in economic policy uncertainty when Belgium was on its way to obtain the world record of longest government formation during 2010 and 2011. More recently, in December 2018, the Belgian government fell after disagreements about endorsing the United Nations Marrakech migration pact. Economic policy uncertainty raised accordingly.

The proportional evolution of the number of detected keywords per category in observed EPU articles is fairly constant. In the two severest peak times (the global financial crisis, and the corona crisis), the uncertainty category becomes slightly more important, which implies that the indices then capture a relative increase in uncertainty as opposed to increased relative reporting about economic policy matters. However, during the European debt crisis over 2011, this is the opposite.

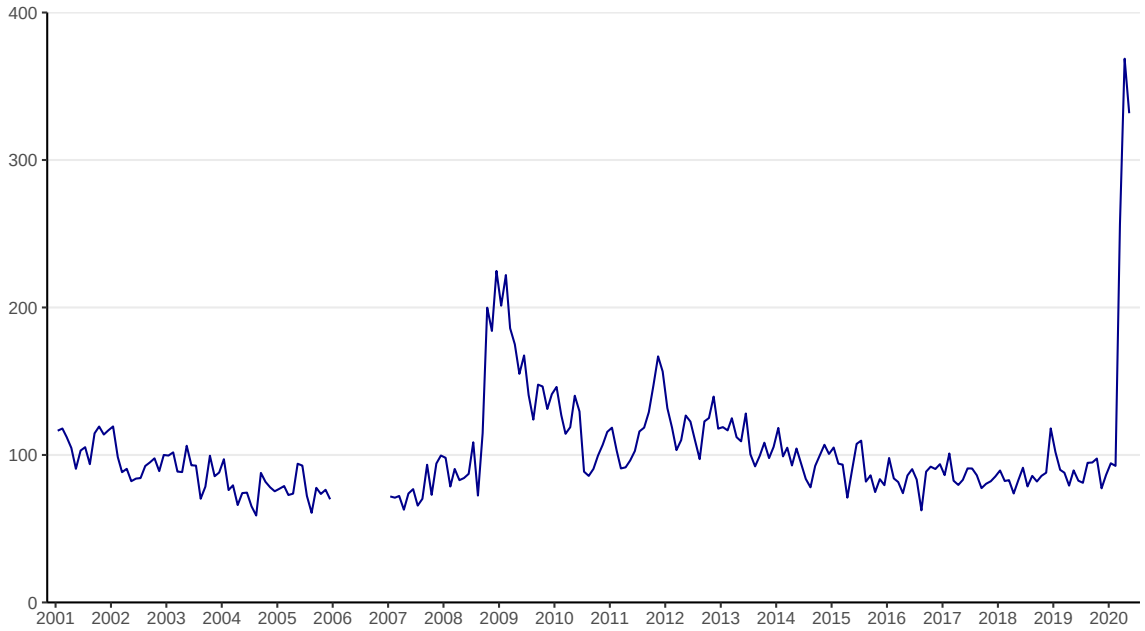
5.3.2 Key Terms Extraction for EPU Articles

To give a face to the sources of uncertainty, we apply a three-step technique to automatically summarize the main non-EPU (i.e. excluding the EPU keywords, except for the capitalized abbreviations such as ECB/BCE) terms found in news during a selection of peaks. First, we

FIGURE 5.1: EPU indices for Flanders, Wallonia and Belgium.



(a) Annotated chart of EPU index for Flanders and Wallonia.



(b) Aggregated EPU index for Belgium.

extract from the relevant corpus subset (here defined as comprising news during the month in which a peak occurs) the nouns and full proper nouns. Second, among these terms, we compute the co-occurrence frequency and continue with those sets of terms that co-occur at least five times. The first two steps are performed to maximize the informational content as the basis to form topics. Next, we apply the biterm topic model developed by Yan *et al.* (2013) with the above as input.

The generative process underlying biterm topic modeling does not consider individual news articles, but a reduction of the whole corpus as an aggregated biterm set. The entire corpus is seen as a mixture of topics. It are the word co-occurrence patterns across the corpus that convey the topics, not single words at the level of documents. Our version models in essence the noun/proper noun biterm (in any direction) co-occurrence relationships.

We infer six topics but drop the first cluster, which is set equal to the empirical word distribution to filter out the most common words. This gives for a given month five topics, which we each define by the ten words most related to it. A peak month is thus explained by at maximum 50 nouns or proper nouns.

We use the R packages **udpipe** (Wijffels, 2019) and **BTM** (Wijffels, 2020) to do the majority of the calculations. We present the full output of the news summary analysis in Table 5.1 and Table 5.3. Terms expected to pop up are indeed cited repeatedly in the news. Sometimes other subjects come up too, for instance, during the Marrakech crisis, news also discussed the ongoing developments about the Brexit and about the trade deal between the U.S. and China. One month after Lehman Brothers' bankruptcy, the U.S. presidential elections were coming up and also a heavy topic of discussion in the Flemish press.

Table 5.1: Automated display through biterm topic modeling of the most recurring non-EPU terms in Flemish press around some peak EPU events (cf. Figure 5.1a).

Event	Cluster	Top non-EPU terms
Lehman Brothers (10/2008)	1	banken, dollar, miljard, bank, geld, landen, bedrijven, week, VS, markt
	2	bedrijf, werknemers, directie, bedrijven, productie, miljoen, maanden, vraag, stuk, week
	3	Fortis, bank, Dexia, BNP Paribas, banken, miljard, geld, België, aandeel, week
	4	geld, Leterme, land, miljoen, België, partij, tijd, bedrijven, banken, CD&V
	5	Obama, McCain, president, debat, campagne, Bush, VS, Republikeinen, Palin, Barack Obama
Marrakech crisis (12/2018)	1	Macron, land, Europa, Frankrijk, werk, migratie, president, Antwerpen, België, bedrijven
	2	Trump, China, president, VS, Huawei, land, Europa, wereld, vraag, Congo
	3	rente, bank, VS, Trump, groei, Fed, dollar, miljard, bedrijven, president
	4	N-VA, Michel, partij, CD&V, land, meerderheid, MR, motie, steun, vertrouwen
	5	May, deal, brexit, miljoen, Bpost, land, bedrijf, Europa, stemming, Brussel

A more complex and bottom-up technique to analyze more precisely the uncertainty sources would be to use a regular topic model, as explained and carried out in Azqueta-Gavaldon (2017) and Azqueta-Gavaldon *et al.* (2020), for instance.

5.3.3 Belgian EPU Time Series and Related Indices

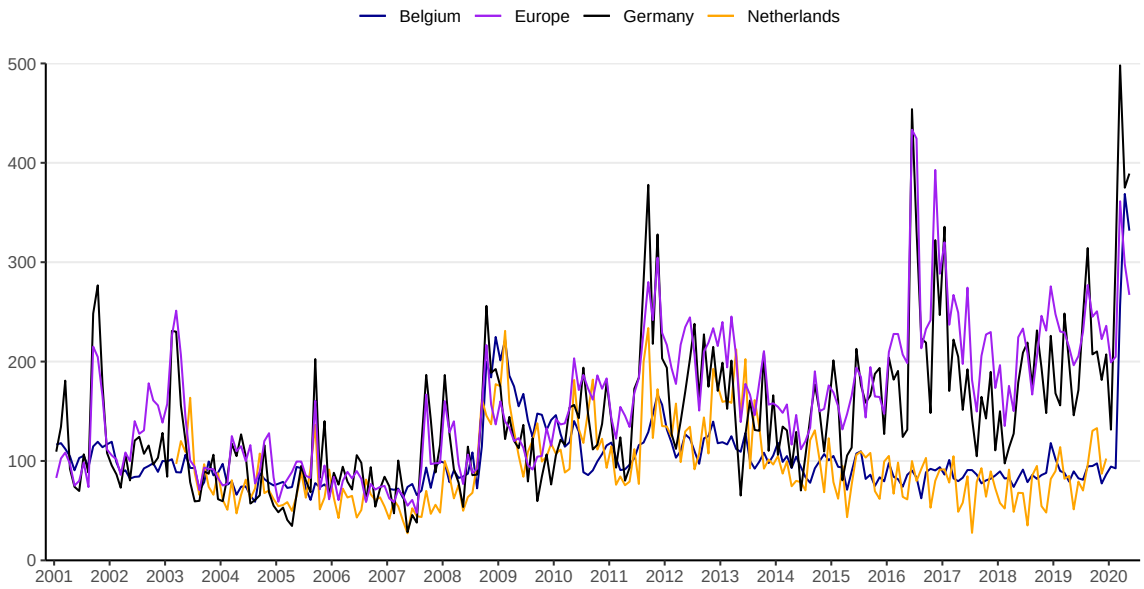
Table 5.2 exhibits the contemporaneous correlations between the three constructed EPU indices for Belgium and some other uncertainty indicators. As can also be seen in Figure 5.1a, the correlation between the Flanders (EPU_t^{FL}) and Walloon (EPU_t^{WL}) indices is strong at 79%, but not perfect. Some events are covered more in the Flemish press (e.g. the global financial crisis), others more in the Walloon press (e.g. the COVID-19 pandemic), but the patterns are similar. The keywords are very comparable (due to the translation), yet this does not guarantee ex-ante that the news coverage will be as well. The conclusion is thus meaningful.

Table 5.2: Contemporaneous correlations between the Belgian EPU indices and various other EPU indices and uncertainty indicators.

	EPU_t^{FL}	EPU_t^{WL}	EPU_t^{BE}	EPU_t^{NL}	EPU_t^{DE}	EPU_t^{EU}	VIX_t^{NL}	VIX_t^{EU}	CCI_t^{BEL}
EPU_t^{FL}	1	0.79	0.96	0.65	0.37	0.26	0.51	0.54	−0.66
EPU_t^{WL}	0.79	1	0.94	0.62	0.45	0.28	0.37	0.39	−0.49
EPU_t^{BE}	0.96	0.94	1	0.64	0.42	0.26	0.47	0.49	−0.62

The correlations with the EPU index of the Netherlands (EPU_t^{NL}), Belgium's most resembling country, are above 60%. The correlation of the Belgium EPU index (EPU_t^{BE}) with the German EPU index (EPU_t^{DE}) is a bit lower at 42%, and the correlation with the index for major European economies (EPU_t^{EU}) sinks to 26%. Figure 5.2 plots the benchmark EPU series and the one for Belgium. The most striking divergence took place during the initial Brexit struggles, when the EPU index for Europe (including a U.K. EPU index) and the EPU index for Germany soared severely more than the series for Belgium and the Netherlands did. The differences in the beginning of the time series are because of other normalization reference periods used.

FIGURE 5.2: Monthly Belgium EPU index and benchmark EPU indices. *Note*: the EPU from the Netherlands is not available in the beginning and end of the sample.



We also include the VIX index on the Dutch AEX stock market index (VIX_t^{NL}), and the

VIX index on the Euro Stoxx 50 stock market index (VIX_t^{EU}) in the comparison. The obtained correlations are all in the proximity of 50%. Lastly, we analyze the interaction with the consumer confidence indicator in Belgium (CCI_t^{BEL}). The correlations are strongly negative up to -66%, as anticipated. If economic policy uncertainty goes up, consumer confidence goes down, and vice versa.

Overall, the sign and strength of the correlations are in line with what other works have reported (such as Kroese *et al.*, 2015), thereby corroborating a correct construction of the Belgian EPU indices. The indices reveal sufficient domestic and foreign affairs.

5.3.4 Alternative Index Construction Methods

As a robustness check, we tested several alternative index construction approaches. First, in terms of corpus selection. We tried a filtering that keeps a news article only if keywords from all three categories show up within a span of six sentences at least once. This halves the corpus for both languages. Contrary to Kroese *et al.* (2015)'s additional index confined to the Netherlands only (i.e. their "EBO-NL" index), we stick with the broader EPU index as-is. Adding a complementary filter that ensures news articles discuss Belgium impacted the corpus size too heavily. Second, in terms of computation, for instance, counting the raw number of EPU keywords instead of the normalized number of EPU articles.

The different index versions have a positive correlation with the indices coming from the original indexation approach, but do not result in more qualitatively interpretable indices. They tend to be more volatile, and relate less well to the benchmark indices.

This supplementary analysis validates the effectiveness of our keywords and our corpus cleaning procedure. For the Belgian case, more stringent (and more time-consuming to obtain) news corpora, or other index measures, do not result in better EPU indices.

5.3.5 Economic Policy Uncertainty in Times of COVID-19

In this subsection, we briefly zoom in on the first five months of 2020 during which the COVID-19 crisis unfolded across the world. Table 5.3 (for Flemish press) and Table 5.4 (for Walloon press) show the key terms from the topic clusters obtained as above. The focus is on how the news coverage content changed from March to May, along with the high measurement of economic policy uncertainty.

In almost all the topic clusters, many terms refer explicitly to the pandemic. Over the months, there is an increasing focus on the U.S. and the possible overall economic and political consequences of the corona virus. The intention of parent company Lufthansa to restructure Brussels Airlines comes up as an important topic in May, in both Flemish and Walloon press.

Figure 5.3 shows the EPU indices for Flanders, Wallonia and Belgium on a daily scale, from January 2020 to May 2020. It presents the true series and a locally smoothed version

Table 5.3: Automated display through biterm topic modeling of the most recurring non-EPU terms in *Flemish* press during the COVID-19 pandemic.

Month	Cluster	Top non-EPU terms
03/2020	1	bedrijven, coronacrisis, aantal, werknemers, werk, banken, miljard, week, weken, miljoen
	2	virus, land, coronavirus, Trump, China, VS, president, tijd, Italië, wereld
	3	bedrijven, banken, miljard, coronavirus, geld, rente, landen, ECB, bank, impact
	4	N-VA, CD&V, PS, MR, noodregering, meerderheid, partij, Dewael, land, Laruelle
	5	Europa, Turkije, EU, Griekenland, grens, Erdogan, landen, Duitsland, president, steun
04/2020	1	bedrijven, miljoen, coronacrisis, bedrijf, miljard, maand, banken, werknemers, week, België
	2	week, N-VA, land, mei, weken, tijd, Veiligheidsraad, coronacrisis, CD&V, leven
	3	virus, land, landen, wereld, aantal, lockdown, China, Trump, leven, coronavirus
	4	landen, miljard, Italië, geld, Nederland, Europa, bedrijven, land, coronacrisis, EU
	5	miljoen, Brussels Airlines, coronacrisis, bedrijven, stad, vraag, Lufthansa, weken, geld, mei
05/2020	1	bedrijven, miljoen, Makhlouf, wereld, situatie, weken, vraag, Facebook, contract, coronacrisis
	2	bedrijven, Brussels Airlines, miljard, coronacrisis, miljoen, geld, Lufthansa, bedrijf, banken, landen
	3	China, land, landen, president, Trump, virus, wereld, coronacrisis, week, Europa
	4	N-VA, partij, PS, Magnette, CD&V, Vlaanderen, VLD, MR, coronacrisis, Vlaams Belang
	5	bedrijven, virus, aantal, weken, werk, lockdown, coronacrisis, maanden, tijd, land

Table 5.4: Automated display through biterm topic modeling of the most recurring non-EPU terms in *Walloon* press during the COVID-19 pandemic.

Month	Cluster	Top non-EPU terms
03/2020	1	taux, BCE, coronavirus, baisse, marché, cours, l'économie, mois, pays, cas
	2	pays, coronavirus, président, États, l'UE, zone, cas, l'épidémie, virus, l'Union
	3	N-VA, PS, président, coronavirus, confiance, mois, parti, temps, Parlement, pays
	4	coronavirus, pays, virus, Chine, cas, monde, temps, santé, pandémie, l'épidémie
	5	cas, coronavirus, secteur, mois, jours, situation, temps, travailleurs, travail, Belgique
04/2020	1	mai, président, mois, fin, pays, temps, monde, saison, N-VA, coronavirus
	2	confinement, secteur, temps, situation, travail, gens, cas, mois, personnel, mai
	3	prix, marché, coronavirus, mois, secteur, baisse, d'euros, groupe, pays, confinement
	4	confinement, monde, virus, pays, temps, santé, population, coronavirus, cas, tests
	5	pays, États, l'UE, d'euros, plan, pandémie, zone, l'Union, l'économie, relance
05/2020	1	pays, États, l'Union, relance, plan, pandémie, l'UE, PIB, zone, BCE
	2	secteur, mois, plan, temps, PS, mai, juin, fin, situation, travail
	3	pays, monde, pandémie, Chine, président, coronavirus, question, fin, Grèce, Donald Trump
	4	confinement, cas, virus, santé, situation, temps, déconfinement, mois, place, jours
	5	d'euros, groupe, mois, Lufthansa, compagnie, secteur, Belgique, plan, pays, Brussels Airlines

(using LOESS regression), with Sundays dropped. The dynamics are interesting. January and February are calm months with a level around 100, which indicates the same degree of uncertainty as on average up to 2011. Thereafter, the uncertainty unequivocally drives up, becomes more volatile transitioning from March to April, and has been decreasing in the last three weeks of April. Most of the uncertainty was accrued before the lockdown in Belgium was officially imposed. The uncertainty remains high in May, but at a fairly constant, and lower, daily level. The daily coverage in Flemish newspapers versus the one in Walloon newspapers is similar, as shown in Figure 5.3a, though the Wallonia series is consistently higher.

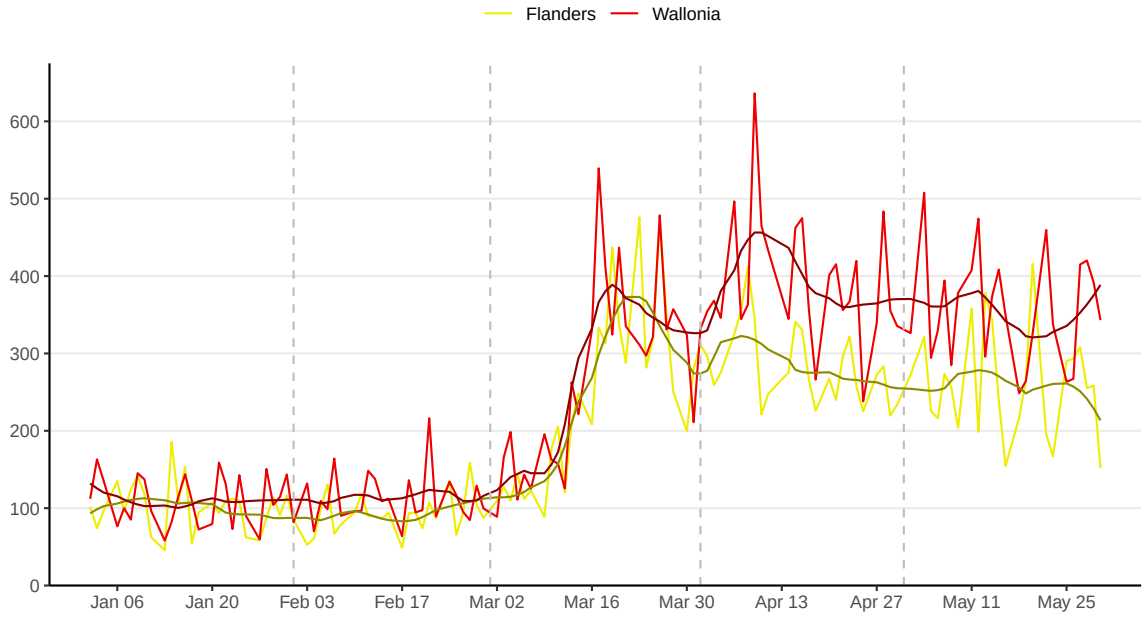
The increase in the monthly EPU indices for Belgium from March to April thus not stems from a day-to-day increasing trend, but from a sustained high level throughout April. Still, the overall observed daily trend in Belgian newspaper coverage surrounding economic policy uncertainty seems to point toward a soft decline going forward.

5.4 Conclusion

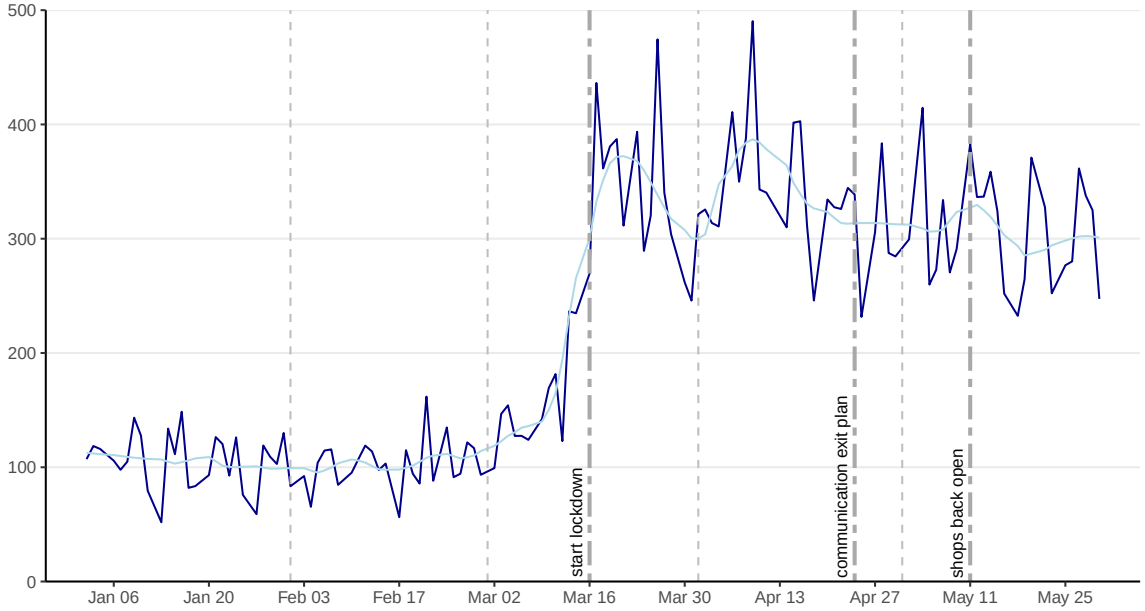
This paper describes the construction of EPU indices for Flanders, Wallonia and Belgium from press data in the style of Baker *et al.* (2016). The EPU index is an interesting descriptive measure of the degree to which newspapers are discussing economic policy concerns using terms related to uncertainty. The constructed indices correlate with existing European uncertainty time series but also capture national evolutions. The last part of our analysis focuses on 2020 and the COVID-19 pandemic. News-based economic policy uncertainty reaches unseen levels in March and April, but witnesses a decreasing trend in May. More timely (up to daily) and alternative calculations of the presented indices are available upon request.

The main question for further research is how to use text-based indices, including any version of a news-based EPU index, to improve nowcasts and forecasts about the (Belgian) economy. In an increasingly faster evolving world, nowcasting might have become the hottest practice within departments responsible for economic analysis. Text-based indices have the advantage of being flexible, timely, and are able to uncover latent variables. For an overview of the different steps in creating and researching the added value of textual indices, we refer to the survey of Algaba *et al.* (2020a).

FIGURE 5.3: Daily Belgian EPU indices in 2020. The lines on top are LOESS curves.



(a) Daily EPU index for Flanders and Wallonia in 2020.



(b) Daily EPU index for Belgium in 2020.

Conclusion 6

Analyzing qualitative data such as texts poses distinct challenges. I laid out a generic methodological and computational framework to handle these challenges. While only a minor aim of this thesis was to test the empirical power of sentiment and the multiple ways to compute it, many interesting hypotheses can be (re)assessed when current models in economics and finance are enhanced with sentiment information. The proposed sentometrics methodological framework is no precise recipe that will get you from A to Z in all situations, because in my opinion, there exists no such thing. The specificity of the research question as well as the creativity of the researcher stay central, but both can fall back on the guiding building blocks the framework provides. I summarize below some ideas for further research.

Grimmer and Stewart (2013) propose four principles for using automated text analysis in political applications, one of which is: "Validate, validate, validate." This resonates well with the three golden rules of econometrics as mentioned in Hendry (1980): "Test, test, test." One could say that "testing" is the hallmark of econometrics, typically the evaluation of a hypothesis about an underlying variable process, that then provokes new insights. More work can be done in the **validation** area of sentometrics, especially in light of the "joint hypothesis problem" I introduced. An important difficulty remains the simultaneous validation of the various steps, and the decoupling of which step contributes most to a weak model performance. For instance, it is currently complicated to directly link the filtering of the data to a classification performance statistic, given the many intermediate steps in between.

Additional value to explore lies in more meaningful **text selection** and **index weighting**. The use of keywords has many advantages, its inherent simplicity as the principal one, but is too generic at times (I am sure that some of the keywords used in this thesis stay debatable even after multiple rounds of validation). Model-oriented approaches for selection and scoring continue to have the best prospects for optimizing the balance between comprehensibility and

accuracy. Furthermore, it would be of interest to study how econometrics can be deployed to develop more **fine-grained sentiment calculation** methods. The **attribution** analysis can be extended further, possibly automated, to achieve a more informative analysis on the contribution of sentiment calculation methods, textual content, news sources, and other components.

This thesis generally defined and discussed qualitative sentiment data, but focused on textual data, and most so on news data. Print media data, as compared to social media data, is generally of higher quality and more reliable. However, for **intraday analyses**, social media data is more pertinent as it is even timelier and more diverse, yet also substantially more noisy and prone to fake news. Intraday analysis fits within the overall framework of data "acquisition–enrichment–filtering–quantification–aggregation–modeling", but an interesting area of research would be the refinement and empirical application of the methodology for social media data available at a very high frequency.

The future will be exceedingly multimedia in terms of content generated, hence the analysis indispensably **multimodal**. A major challenge is the development of appropriate technology for unified multimodal sentiment analysis systems. I recommend first the development of a more tailored individual framework for visual and audio data, and then one for combining all three alternative data sources.

I line with the above, the priorities for further development of the R package **sentometrics** are integrating better prediction and model assessment tools, enhancing the complexity of the sentiment engine, allowing user-defined weighting schemes, and adding intraday aggregation.

I end by reinitializing the call for more efforts toward **reproducibility** in the econometric study of sentiment from qualitative data, across all disciplines involved in the sentiment analysis value chain. To make progress, a view on reference data and associated state-of-the-art performance is needed for different sentiment quantification techniques, data, and (econometric) approaches. Other researchers can evaluate any new approach on the reference data, thereby providing a consistent picture of reproducibility or improved performance. There are no standard practices yet on how to share code, data, and results. These should be made available through an open database with easy access and well-documented formats. The field of computer science can be used as a guiding example, as such practices are more widespread there. The collaborative GitHub project about sentiment and econometrics research I set up (see `https://sborms.github.io/econometrics-meets-sentiment`) is meant to be a starting point to gather such resources.

Appendix 7

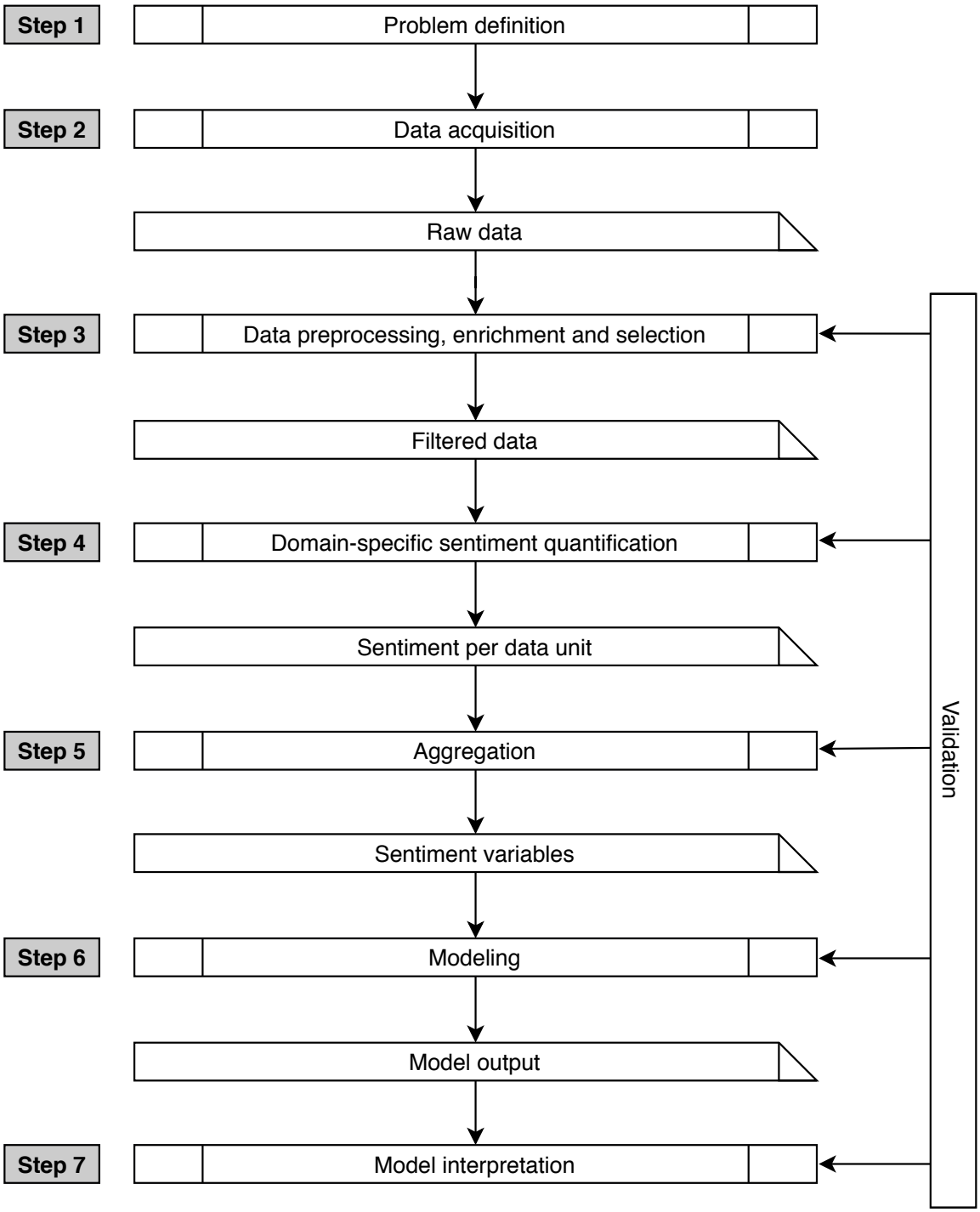
7.1 A Typical Sentometrics Analysis Workflow

A complete sentometrics workflow is depicted in Figure 7.1. It extends a typical applied econometrics process with the additional steps inherent to the use of qualitative data that needs to be transformed into numbers (see e.g. Gentzkow *et al.*, 2019a). The workflow serves to discover or confirm a certain theory or hypothesis, and therefore starts with a question. A clear question is important, because the tools to solve respective questions might be very different. For instance, in macroeconomics and politics one is particularly interested in aggregate categorizations (Ceron *et al.*, 2014), whereas in computer science, the interest lies more in the correct classification of individual data units. A generic example of a question is what, and how large, the effect of a set of particular sentiment variables is on a continuous variable of interest.

The initial step toward answering the question is the acquisition of the data needed to construct the sentiment variables from. Following the definition of sentiment given above, the sentiment data are of a qualitative nature and consist of expressions in the form of verbal or nonverbal communication, naturally leading to three datatypes, namely textual, audio, and visual data. Examples are news media articles, conference calls, and images. Each data point may already contain a rich metadata component including elements such as the date and the source.

Next is the preprocessing, enrichment and ultimately selection of the relevant data, to determine the exact data to construct the sentiment variable(s) from. All three datatypes are in unstructured form, requiring to undertake some preprocessing to impose a structure and clean the data. For the data retrieved, ideally much relevant metadata are generated, which we

FIGURE 7.1: Steps of a complete sentometrics workflow.



refer to as features. A feature is a broad term that can be anything attached to the original data, from source, expressor, entity, location, topic, and so on. The necessity of including features in the analysis depends on the problem. Selection of the qualitative data is application-specific. One should always be able to verify why a specific sample is representative for the sentiment variables that have to be modeled.

Once a structured and confined dataset is obtained it can be used to assess the individual sentiment, at the desired level of data unit granularity (e.g. a sentence, a full article, a sequence of sounds, a video, or an image). This leads to a quantitative sentiment dataset. The process of creating valuable and structured data from unstructured data appeals to the popular process of "datafication" (Cukier and Mayer-Schoenberger, 2013).

For the problems that require a cross-sectional approach, properly aggregated individual sentiment scores per data unit are sufficient. However, for many other analyses, especially in economics and finance, a time series perspective is necessary (see Ardia *et al.*, 2019b, Thorsrud, 2020, and Borovkova *et al.*, 2017, among others). For this, some type of aggregation into time series is required to construct meaningful sentiment variables. Firstly, the low-level sentiment scores per data unit are spread out across the metadata features. Secondly, these different feature-sentiment scores are cross-sectionally aggregated for a given time unit (e.g. daily) to obtain a first version of sentiment measures. Thirdly, these measures can be further aggregated across time using different time series weighting functions, to result in various sentiment variables. We consider a sentiment variable to be a time series of sentiment extracted from given data, with particular sentiment calculation, feature and aggregation characteristics.

Once the aggregated sentiment variables are obtained they can be used in an econometric analysis. A final aggregation of interest is across sentiment variables, which can be done as an additional pre-modeling aggregation step, or seen as the output of the modeling step. Given the problem defined in the first step, a model which describes the joint distribution of the dependent and the independent (sentiment) variables of interest is proposed. In the model interpretation step, the outcome of the model is qualitatively interpreted to explain the results and find an answer to the original question.

Every step needs to be subject to a validation analysis. One has to reiterate back to improve separate steps in the workflow depending on where a problem occurs. Problems across the workflow can arise as late as during the final validation step. For example, the model representation related to the question may have to be reformulated, or the variable construction improved to obtain more substantial sentiment proxies. Validation is particularly important in an econometric analysis with sentiment extracted from textual, audio or video sources, because it suffers from a "joint hypothesis" problem. A model not capturing a relationship between sentiment and other variables can be due to either a bad proxy of sentiment, or due to truly no existing connection. Validation in each step of the process and using several methods is crucial to counterbalance this problem. An econometric model is validated in terms of statistical and economic significance, out-of-sample forecast precision, visual fit, attribution,

parameter stability, validity of the assumptions, and a comparison to baseline models.

7.2 Efficiency of Lexicon-Based Sentiment Analysis in R

This appendix provides an illustrative comparison of the computation time of various lexicon-based sentiment analysis tools in R. The core of the sentiment computation in the R package **sentometrics** (Ardia *et al.*, 2020) is implemented in C++ through **Rcpp** (Eddelbuettel and Francois, 2011). We compare the speed of our computation with the R packages **meanr** (Schmidt, 2019), **SentimentAnalysis** (Feuerriegel and Pröllochs, 2019), **syuzhet** (Jockers, 2017), **quanteda** (Benoit *et al.*, 2018), and **tidytext** (Silge and Robinson, 2016). The first three of these five packages have proper sentiment functions. The **quanteda** and **tidytext** packages have no explicit sentiment computation function; it needs to be constructed first, based on their respective toolsets. This is an entry barrier for less-experienced programmers. The **SentimentAnalysis** package has the **tm** package as backend and uses internally a similar calculation as **tm**'s `tm_term_score()` function. The **sentimentr** package is not part of the exercise because it proved to be vastly slower than all others, which was anticipated as it aims to handle more difficult linguistic edge cases.

We perform two analyses. Sentiment is computed for 1000, 5000, 10000, 25000, 50000, 75000 and 100000 texts, and the average execution time in seconds across five repetitions, using the **microbenchmark** (Mersmann and Ulrich, 2019) package, is shown in Table 7.1.

The first analysis (see Panel 7.1a) benchmarks these implementations with three approaches using the `compute_sentiment()` function from **sentometrics**: one without valence shifters, one with valence shifters integrated from a bigrams perspective, and one with valence shifters integrated from a clusters perspective. The number of threads for parallel computation is set to one where appropriate. All other algorithms are run with a version of the the Hu & Liu lexicon (about 6600 single words). The computations are counts-based and constructed so as to give the same output across all packages for a binary lexicon, if the tokenization is the same. For example, the **sentometrics** and **tidytext** implementations give identical results.

The **meanr** implementation comes out fastest because everything is written in the C programming language. Yet, it offers no flexibility to define the input lexicon nor the scale on which the scores are returned. On the other spectrum, among these approaches, the **SentimentAnalysis** and **suyzhet** packages are slowest. The latter package further does not offer the flexibility of adding different sentiment lexicons than those available in their package. **SentimentAnalysis** becomes exponentially slower as it suffers to manage the memory required for larger corpora. The **quanteda** package is fast, but slower than the **sentometrics** and **tidytext** implementations. The **tidytext** package is faster, particularly for the two largest corpus sizes.

The second analysis (see Panel 7.1b) compares the computation time with nine lexicons as input. The comparison is against the **tidytext** package, for a unigrams and a bigrams

Table 7.1: Average computation time (in seconds) of various lexicon-based sentiment tools in R. All implementations consider the Hu & Liu lexicon (Panel 7.1a), or the nine lexicons specified in the `lex` object defined in the main text of the vignette (Panel 7.1b). Some implementations do not integrate valence shifters (*unigrams*), others do from a bigrams perspective (*bigrams*) or from a clusters perspective (*clusters*). The number of texts in the first column is denoted in thousands (i.e. 100 means 100000 texts).

(a) Average execution time of the sentiment computation for one lexicon

	sentometrics							
Texts	*unigrams*	*bigrams*	*clusters*	**meanr**	**SentimentAnalysis**	**syuzhet**	**quanteda**	**tidytext**
1	0.24	0.20	0.22	0.08	1.18	0.55	0.60	0.16
5	0.87	0.87	0.91	0.34	5.26	1.99	1.74	0.60
10	1.73	1.68	1.72	0.67	11.23	3.83	3.07	1.11
25	4.41	4.21	4.40	1.71	26.88	9.07	7.19	2.83
50	9.18	8.55	9.42	3.75	53.08	18.37	14.12	5.88
75	13.62	13.49	13.44	5.06	78.44	27.13	20.37	8.48
100	18.69	18.22	18.61	6.57	109.58	35.25	26.98	11.06

(b) Average execution time of the sentiment computation for nine lexicons

	sentometrics					**tidytext**	
Texts	*unigrams*	*unigrams*, feats.	*bigrams*	*clusters*	*clusters*, parallel	*unigrams*	*bigrams*
1	0.26	0.24	0.27	0.26	0.22	0.21	0.66
5	1.00	0.87	1.01	1.01	0.79	0.67	2.80
10	1.96	1.68	1.98	1.97	1.54	1.27	5.68
25	4.82	4.24	4.90	4.97	3.81	3.07	13.95
50	9.96	8.71	10.13	10.02	7.85	6.03	28.00
75	16.70	19.14	16.67	23.04	15.43	14.00	58.02
100	32.40	23.66	23.80	36.41	30.86	14.02	64.73

implementation. The lexicons are those in the `lex` object defined in the main text of the vignette. For the clusters approach, we also look at its parallelized version, using eight cores (see the '*clusters*, parallel' column). For the unigrams approach in **sentometrics**, we also assess the additional time it takes to spread out sentiment across features (see the '*unigrams*, feats.' column).

The **tidytext** package is, in general, faster for many lexicons as well. Differences are not large nonetheless, and running any **sentometrics** computation in parallel would make the speed differentials disappear. However, the bigrams calculation using **tidytext** is markedly slower. With **sentometrics**, the speed of the computation is comparable across all types of sentiment calculation. The **tidytext** framework thus copes more slowly with complexity.

Overall, the **sentometrics** package brings an off-the-shelf yet flexible sentiment calculator that is computationally efficient, being fast in itself, and independent as to the decision (how) to integrate valence shifters as well as (though to a smaller extent) the number of input lexicons.

7.3 Package Methods Overview

This appendix provides an overview of the R methods made available in the **sentometrics** package, as also highlighted in Table 3.1. The S3 class objects from the **sentometrics** package are created using their function counterpart with the same name, except for the *`sentiment`* object (created with the `compute_sentiment()` function) and the *`sento_modelIter`* object (created with the `sento_model(..., ctr = ctr_model(..., do.iter = TRUE))` function). Most of the methods are individually documented, accessible as `?method.object` (e.g. `?aggregate.sento_measures`).

Standard methods

`plot()` Classes: *attributions*, *sento_measures*, *sento_modelIter*.
Plots, all in a similar **ggplot2** style, respectively, the computed sentiment attributions of a run regression model, the constructed sentiment measures, and the target variable versus the predicted outcomes of an iteratively ran regression model. The first two can be grouped according a specific dimension (e.g. by `"features"`).

`summary()` Classes: *sento_measures*, *sento_model*, *sento_modelIter*.
Provides a short description of the contents of the respective object. The `print()` method simply displays the object class; it is also supported for a *sento_corpus* object and prints like in **quanteda**.

Statistical methods

`aggregate()` Classes: *sentiment*, *sento_measures*.
Aggregates a document-level or a sentence-level *sentiment* object into a *sento_measures* object, or a sentence-level *sentiment* object into a document-level *sentiment* object. A *sento_measures* object can also be further aggregated across sentiment measures.

`diff()` Classes: *sento_measures*.
Returns a *sento_measures* object with differenced sentiment measures.

`merge()` Classes: *sentiment*.
Combines multiple *sentiment* objects row-wise and/or column-wise.

`nobs()` Classes: *sento_measures*.
Gives the number of data points (i.e. rows) in the sentiment measures. The number of sentiment measures can be obtained with the `nmeasures()` function.

`predict()` Classes: *sento_model*.
Generates predictions from the model object for a data *matrix* of values for the explanatory sentiment measures and other variables.

`scale()` Classes: *sento_measures*.
Returns a *sento_measures* object with scaled sentiment measures. One can also use the `center` and `scale` arguments to define values to subtract from the sentiment measures or divide them by.

Coercion

`as.data.frame()`	Classes: *sento_corpus*, *sento_measures*. Converts the corpus or sentiment measures in a *data.frame* object.
`as.data.table()`	Classes: *sento_corpus*, *sento_measures*. Converts the corpus or sentiment measures in a *data.table* object.
`as.sentiment()`	Classes: *data.frame*, *data.table*. Converts a properly structured sentiment table in *data.frame* or *data.table* format into a document-level or a sentence-level *sentiment* object.
`as.sento_corpus()`	Classes: **quanteda** *corpus*, **tm** *SimpleCorpus*, **tm** *VCorpus*. Transforms the given corpus input object into a *sento_corpus* object, integrating available metadata, where possible, into corpus features.

Extraction

`subset()`	Classes: *sento_measures*. Can be used to do three things: subset the rows (either by index or by a condition), select certain sentiment measures, or delete certain sentiment measures. The selection and deletion is based on the names of the sentiment measures along the features, lexicons, and time-weighting schemes dimensions.

7.4 Package Aggregation Weighting Schemes

This appendix presents the formulas that define the weights used in the different sentiment aggregation schemes available in the package. The constant c indicates a normalization factor that makes sure the considered weights sum up to 1. When not specified, arguments referred to are from the `ctr_agg()` function.

Within-Document and Within-Sentence Weighting

We outline here the different options available for the `howWithin` argument of the `ctr_agg()` function and the `how` argument of the `compute_sentiment()` function, for the sentiment calculation in (3.1). The weight ω_i is associated to the unigram at the ith position in a document (resp. sentence) $d_{n,t}$, where d serves as a notational shorthand. The number of unigrams in a document (resp. sentence) is Q_d, the number of unigrams in a document (resp. sentence) that appear in the lexicon is n_{pol}, N is the total number of documents (resp. sentences) in the corpus, and q_i is the number of documents (resp. sentences) across the entire corpus containing unigram i.

The package lets the user choose between the following constant and unigram-specific

weights:

- `"counts"`:

$$\omega_i = 1$$

- `"proportional"`:

$$\omega_i = \frac{1}{Q_d}$$

- `"proportionalPol"`:

$$\omega_i = \frac{1}{\max\{n_{pol}, 1\}}$$

- `"proportionalSquareRoot"`:

$$\omega_i = \frac{1}{\sqrt{Q_d}}$$

- `"UShaped"`:

$$\omega_i = \left(i - \frac{Q_d + 1}{2}\right)^2 \times c$$

- `"inverseUShaped"`:

$$\omega_i = \left(0.25 - \frac{\left(i - \frac{Q_d+1}{2}\right)^2}{Q_d^2}\right) \times c$$

- `"exponential"`:

$$\omega_i = \exp\left(5 \times \left(\frac{i}{Q_d} - 1\right)\right) \times c$$

- `"inverseExponential"`:

$$\omega_i = \exp\left(5 \times \left(1 - \frac{i}{Q_d}\right)\right) \times c$$

- `"TFIDF"`:

$$\omega_i = \log_{10}\left(\frac{N}{1 + q_i}\right)$$

Note: The `"TFIDF"` option represents term frequency–inverse document frequency weighting (Spärck, 1972). The weight covers only the inverse document frequency (IDF) part, and we follow the implementation of the **quanteda** package's `docfreq(..., scheme = "inverse", k = 1, base = 10)` function. The term frequency (TF) component is inherent in equation (3.1)—for instance, it will pertain to the raw count convention when using no valence shifters.

Across-Document and Across-Sentence Weighting

We outline here the different options available for the `howDocs` argument of the `ctr_agg()` function, for the aggregation in (3.2). The weight θ_n values a document (resp. sentence) $d_{n,t}$ (again, we use d as a shorthand) in the aggregation window (per date for across-document, and per document for across-sentence). Recall that N_t is the total number of documents at time t, or, abusing the notation, it can similarly represent the number of sentences within a document. The total number of unigrams of all documents (resp. sentences) included in the aggregation window is z.

For a given document (resp. sentence) $d_{n,t}$, the weight can be one of:

- `"equal_weight"`:
$$\theta_n = \frac{1}{N_t}$$
- `"proportional"`:
$$\theta_n = Q_d \times c$$
- `"inverseProportional"`:
$$\theta_n = \frac{1}{Q_d} \times c$$
- `"exponential"`:
$$\theta_n = \exp\left(\alpha \times \left(\frac{Q_d}{z} - 1\right)\right) \times c$$
- `"inverseExponential"`:
$$\theta_n = \exp\left(\alpha \times \left(1 - \frac{Q_d}{z}\right)\right) \times c$$

The value α is set equal to $10 \times$ `alphaExpDocs`.

Across-Time Weighting

We outline here the options available for the `howTime` argument of the `ctr_agg()` function, for the aggregation in (3.3). The weight b_t represents the time weight for a sentiment value at the time point t relative to a starting date and a lag τ.

For positional time points $t = 1, \ldots, \tau$, the weighting schemes available are:

- `"equal_weight"`:
$$b_t = \frac{1}{\tau}$$

- `"almon"`:

$$b_t = \left[\left(1 - \frac{t}{\tau}\right)^{R-r} \times \left(1 - \left(1 - \frac{t}{\tau}\right)^{r}\right)\right] \times c$$

- `"beta"`:

$$b_t = f\left(\frac{t}{\tau}; a, b\right) / \sum_{t=1}^{\tau} f\left(\frac{t}{\tau}, a, b\right) \times c, \text{ where}$$

$$f(x; a, b) \equiv \frac{x^{a-1}(1-x)^{b-1}\Gamma(a+b)}{\Gamma(a)\Gamma(b)}, \text{ and } \Gamma(\cdot) \text{ is the gamma function}$$

- `"linear"`:

$$b_t = \frac{t}{\tau} \times c$$

- `"exponential"`:

$$b_t = \exp\left(\alpha \times \left(\frac{t}{\tau} - 1\right)\right) \times c$$

The value r is a specific element from the `ordersAlm` vector, and R is the maximum value in that vector. If `do.inverseAlm = TRUE`, the inverse Almon polynomials are computed too, modifying $\frac{t}{\tau}$ to $1 - \frac{t}{\tau}$. If `do.inverseExp = TRUE`, the inverse exponential curves are added. In the Beta density $f(\cdot;\cdot,\cdot)$, a is `aBeta`, and b is `bBeta`. The α here is defined as $10 \times$ `alphasExp`. The functions `weights_almon()`, `weights_beta()` and `weights_exponential()` allow generating these time weight separately.

7.5 Computational Details

The results for everything related to Chapter 3 were obtained using R 3.6.2 (R Core Team, 2019), **sentometrics** version 0.8.0, and underlying or used packages **caret** version 6.0.85 (Kuhn, 2018), **data.table** version 1.12.8 (Dowle and Srinivasan, 2019), **foreach** version 1.4.7 (Weston, 2019), **ggplot2** version 3.2.1 (Wickham, 2016), **glmnet** version 3.0.2 (Friedman *et al.*, 2010), **gridExtra** version 2.3.0 (Auguie, 2017), **ISOweek** version 0.6.2 (Block, 2011), **lexicon** version 1.2.1 (Rinker, 2019), **lubridate** version 1.7.4 (Grolemund and Wickham, 2011), **quanteda** version 1.5.2 (Benoit *et al.*, 2018), **Rcpp** version 1.0.3 (Eddelbuettel and Francois, 2011), **RcppArmadillo** version 0.9.800.3.0 (Eddelbuettel and Sanderson, 2014), **RcppRoll** version 0.3.0 (Ushey, 2018), **RcppParallel** version 4.4.4 (Allaire *et al.*, 2019), **stm** version 1.3.5 (Roberts *et al.*, 2019), **stringi** version 1.4.5 (Gagolewski, 2020), **tm** version 0.7.7 (Feinerer *et al.*, 2008), and **zoo** version 1.8.7 (Zeileis and Grothendieck, 2005). Computations were performed on a Windows 10 Pro machine, x86 64–w64–mingw32/x64 (64-bit) with Intel(R) Core(TM) i7–7700HQ CPU 2x 2.80 GHz. The code displayed is available in the R script `run_vignette.R` located in the `examples` folder on the dedicated **sentometrics** GitHub repository at `https://github.com/SentometricsResearch/sentometrics`.

R, **sentometrics**, and all other packages are available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org`. Any version under development will be available on our GitHub repository.

The timings comparison can be replicated using the R script `run_timings.R`, available on the **sentometrics** GitHub repository in the `appendix` folder. To generate the results, we have also used the packages **dplyr** version 0.8.3 (Wickham *et al.*, 2019), **meanr** version 0.1.2 (Schmidt, 2019), **microbenchmark** version 1.4.7 (Mersmann and Ulrich, 2019), **SentimentAnalysis** version 1.3.3 (Feuerriegel and Pröllochs, 2019), **syuzhet** version 1.0.4 (Jockers, 2017), **tidytext** version 0.2.2 (Silge and Robinson, 2016), and **tidyr** version 1.0.0 (Wickham and Henry, 2019).

7.6 EPU Keywords

This appendix lists the used keywords for the three EPU categories, in Flemish (Table 7.2) and in French (Table 7.3), sorted alphabetically.

7.6.1 Flemish EPU Keywords

Table 7.2: Flemish EPU keywords.

Category	Keywords
Economic	arbeidsmarkt, arbeidsplaatsen, banen, BBP, bedrijfsinvesteringen, bestedingen, BNP, concurrentiekracht, conjuncturele, conjunctuur, consumentenbestedingen, consumeren, consumptie, economie, economisch, economische, investeringen, jeugdwerkloosheid, jobs, koopkracht, levensstandaard, lonen, recessie, tewerkstelling, welvaart, werkgelegenheid, werkloosheid, werklozen
Policy	aftrekbaar, balans, bedrag, begroting, begrotingen, begrotingsoverschot, begrotingstekort, belast, belasting, belastingdienst, belastingen, belastingplichtigen, belastingverhoging, beleid, bespaard, bespaart, besparen, besparing, betalingsbalans, bewindslieden, bewindsman, bezuinigd, bezuinigen, bezuiniging, bezuinigingen, bonden, BTW, budget, budgetair, budgetten, coalitie, coalitiepartner, coalitiepartners, ECB, federale, financiën, fiscaal, fiscale, fiscus, fractie, fractieleider, fracties, heffing, hervormen, hervorming, hervormingen, kabinet, kamerlid, klimaatbeleid, lastenverhoging, lastenverhogingen, lastenverlaging, lastenverlagingen, lastenverzwaring, loonkostenverlaging, loonlastenverlaging, maatregel, maatregelen, maatregels, milieubeleid, minister, ministers, NBB, oppositie, overheden, overheid, overheidsbegroting, overheidsbeleid, overheidsbudget, overheidsschuld, overheidstekort, overheidsuitgaven, parlement, parlementaire, parlementsleden, planbureau, politici, politicus, politiek, politieke, premier, regeerakkoord, regelgeving, regering, regeringen, regeringsbeleid, regeringsleider, reglementen, reglementering, senaat, staatsschuld, staatssecretaris, taks, tax, tekort, uitgave, uitgaven, uitgeven, vakbond, vakbonden, verkiezingsprogramma, wet, wetgeving, wetten
Uncertainty	absurd, achterdocht, afwijzend, alarmerend, argwaan, argwanend, besluiteloos, besluiteloosheid, chaos, crisis, dubbelzinnig, dubbelzinnigheid, ergernis, getreuzel, incompetentie, leugenachtig, misleidend, onbegrip, onbekwaamheid, onbetrouwbaar, onbetrouwbaarheid, onbetrouwbare, oncontroleerbaar, onduidelijkheid, ongeloofwaardig, ongeloofwaardigheid, ongerustheid, onkunde, onrust, onvoorspelbare, onvoorzien, onvoorziene, onzeker, onzekere, onzekerheden, onzekerheid, risico, scepsis, scepticisme, sceptisch, schok, schokken, schokkend, schommelingen, shock, spanningen, tegenstrijdig, twijfel, twijfelachtig, twijfelen, twijfels, vaag, vaagheid, vage, verontrustend, verwarrend, verwarring, wantrouwen, zorgwekkend

7.6.2 French EPU Keywords

Table 7.3: French EPU keywords.

Category	Keywords
Economic	chômage, compétitivité, consommation, consommatrice, consommer, cycle, cyclique, achat, économie, économique, économiques, emploi, emplois, investissement, investissements, PIB, PNB, prospérité, récession, salaires, vie
Policy	augmentation, augmentations, baisses, balance, BCE, BNB, budget, budgetaire, budgets, cabinet, coalition, contribuables, d'impôt, déductible, déficit, déficits, dépense, dépenser, dépenses, députés, dette, économies, électoral, excédent, fédéral, finances, fiscale, fiscalement, fiscales, gouvernement, gouvernemental, gouvernementale, gouvernements, gouvernment, impôt, impôts, législation, loi, lois, mesure, mesures, ministre, ministres, opposition, parlement, parlementaire, politicien, politiciens, politique, politiques, prélèvement, réforme, réformer, réformes, réglementation, réglementations, régulation, régulations, secrétaire, sénat, syndicat, syndicats, taxe, taxé, taxes, TVA
Uncertainty	absurde, alarmant, ambigu, ambiguïté, anxiété, chaos, choc, choquant, confusion, contradictoire, crise, doute, douter, doutes, douteux, ennuyeux, fiabilité, fluctuations, ignorance, imprévisible, imprévu, incertain, incertains, incertitude, incertitudes, incompétence, incompréhension, incontrôlable, incroyable, incroyance, indécis, indécision, indétermination, indéterminé, infidélité, inquiétant, irrésolution, malhonnête, méconfiance, méfiant, mensongère, risques, scepticisme, sceptique, suspect, suspicion, tensions, trompeur, troubles, vague

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. **TensorFlow**: A system for large-scale machine learning, in: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, USENIX Association. pp. 265–283. URL: https://ai.google/research/pubs/pub45381.

Adämmer, P., Schüssler, R.A., 2020. Forecasting the equity premium: Mind the news! doi:10.1093/rof/rfaa007. Forthcoming in Review of Finance.

Algaba, A., Ardia, D., Bluteau, K., Borms, S., Boudt, K., 2020a. Econometrics meets sentiment: An overview of methodology and applications. Journal of Economic Surveys 34, 512–547. doi:10.1111/joes.12370.

Algaba, A., Borms, S., Boudt, K., Verbeken, B., 2020b. Monitoring consumer confidence: A real-time approach using media news articles. doi:10.2139/ssrn.3609297. Working paper.

Allaire, J., Francois, R., Ushey, K., Vandenbrouck, G., Geelnard, M., Intel, 2019. **RcppParallel**: Parallel programming tools for **Rcpp**. URL: https://CRAN.R-project.org/package=RcppParallel. R package version 4.4.4.

Allee, K.D., DeAngelis, M.D., 2015. The structure of voluntary disclosure narratives: Evidence from tone dispersion. Journal of Accounting Research 53, 241–274. doi:10.1111/1475-679X.12072.

Amel-Zadeh, A., Serafeim, G., 2018. Why and how investors use ESG information: Evidence from a global survey. Financial Analysts Journal 74, 87–103. doi:https://doi.org/10.2469/faj.v74.n3.2.

Andreou, E., Gagliardini, P., Ghysels, E., Rubin, M., 2019. Inference in group factor models with an application to mixed frequency data. Econometrica 87, 1267–1305. doi:10.3982/ECTA14690.

Angeletos, G.M., Collard, F., Dellas, H., 2018. Quantifying confidence. Econometrica 86, 1689–1726. doi:10.3982/ECTA13079.

Angeletos, G.M., La'O, J., 2013. Sentiments. Econometrica 81, 739–779. doi:10.3982/ECTA10008.

Antweiler, W., Frank, M., 2004. Is all that talk just noise? The information content of internet stock message boards. Journal of Finance 59, 1259–1294. doi:10.1111/j.1540-6261.2004.00662.x.

Araújo, M., Diniz, J.P., Bastos, L., Soares, E., Júnior, M., Ferreira, M., Riberio, F., Benevenuto, F., 2016. iFeel 2.0: A multilingual benchmarking system for sentence-level sentiment analysis, in: Proceedings of the 10th International AAAI Conference on Web and Social Media, pp. 758–759. URL: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM16/paper/view/13039.

Ardia, D., Bluteau, K., Borms, S., Boudt, K., 2020. The R package **sentometrics** to compute, aggregate and predict with textual sentiment. doi:10.2139/ssrn.3067734. Forthcoming in Journal of Statistical Software.

Ardia, D., Bluteau, K., Boudt, K., 2019a. Media and the stock market: Their relationship and abnormal dynamics around earnings announcements. doi:10.2139/ssrn.3192064. Working paper.

Ardia, D., Bluteau, K., Boudt, K., 2019b. Questioning the news about economic growth: Sparse forecasting using thousands of news-based sentiment values. International Journal of Forecasting 35, 1370–1386. doi:10.1016/j.ijforecast.2018.10.010.

Ardia, D., Boudt, K., 2018. The peer performance ratios of hedge funds. Journal of Banking and Finance 87, 351–368. doi:10.1016/j.jbankfin.2017.10.014.

Arnold, T., 2017. A tidy data model for natural language processing using **cleanNLP**. The R Journal 9, 1–20. URL: https://journal.r-project.org/archive/2017/RJ-2017-035/RJ-2017-035.pdf.

Arslan-Ayaydin, Ö., Boudt, K., Thewissen, J., 2016. Managers set the tone: Equity incentives and the tone of earnings press releases. Journal of Banking and Finance 72, 132–147. doi:10.1016/j.jbankfin.2015.10.007.

Auguie, B., 2017. **gridExtra**: Miscellaneous functions for "grid" graphics. URL: https://CRAN.R-project.org/package=gridExtra. R package version 2.3.0.

Azqueta-Gavaldon, A., 2017. Developing news-based economic policy uncertainty index with unsupervised machine learning. Economics Letters 158, 47–50. doi:10.1016/j.econlet.2017.06.032.

Azqueta-Gavaldon, A., Hirschbühl, D., Onorante, L., Saiz, L., 2020. Economic policy uncertainty in the Euro area: An unsupervised machine learning approach. URL: https://ssrn.com/abstract=3516756. ECB working paper no. 2359.

Baccianella, S., Esuli, A., Sebastiani, F., 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, in: Proceedings of the Seventh conference on International Language Resources and Evaluation. URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf.

Bajo, E., Raimondo, C., 2017. Media sentiment and IPO underpricing. Journal of Corporate Finance 46, 139–153. doi:10.1016/j.jcorpfin.2017.06.003.

Baker, M., Wurgler, J., 2006. Investor sentiment and the cross-section of stock returns. Journal of Finance 61, 1645–1680. doi:10.1111/j.1540-6261.2006.00885.x.

Baker, M., Wurgler, J., 2007. Investor sentiment in the stock market. Journal of Economic Perspectives 21, 129–152. doi:10.1257/jep.21.2.129.

Baker, S., Bloom, N., Davis, S., Terry, S., 2020. COVID-induced economic uncertainty. URL: https://www.nber.org/papers/w26983. NBER working paper no. 26983.

Baker, S.R., Bloom, N., Davis, S.J., 2016. Measuring economic policy uncertainty. The Quarterly Journal of Economics 131, 1593–1636. doi:10.1093/qje/qjw024.

Bannier, C., Pauls, T., Walter, A., 2019. Content analysis of business communication: Introducing a German dictionary. Journal of Business Economics 89, 79–123. doi:10.1007/s11573-018-0914-8.

Barsky, R.B., Sims, E.R., 2012. Information, animal spirits, and the meaning of innovations in consumer confidence. American Economic Review 102, 1343–1377. doi:10.1257/aer.102.4.1343.

Benhabib, J., Spiegel, M.M., 2019. Sentiments and economic activity: Evidence from US states. Economic Journal 129, 715–733. doi:10.1111/ecoj.12605.

Bennani, L., Le Guenedal, T., Lepetit, F., Ly, L., Mortier, V., 2018. The alpha and beta of ESG investing. URL: http://research-center.amundi.com. Amundi working paper 76.

Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., Matsuo, A., 2018. **quanteda**: An R package for the quantitative analysis of textual data. Journal of Open Source Software 3, 774. doi:10.21105/joss.00774.

Berg, F., Koelbel, J., Rigobon, R., 2019. Aggregate confusion: The divergence of ESG ratings. URL: 10.2139/ssrn.3438533. MIT Sloan School working paper 5822-19.

Bholat, D., Hans, S., Santos, P., Schonhardt-Bailey, C., 2015. Text mining for central banks. Technical Report. Centre for Central Banking Studies, Bank of England. URL: https://www.bankofengland.co.uk/ccbs/text-mining-for-central-banks.

Bird, S., Klein, E., Loper, E., 2009. Natural language processing with Python. O'Reilly Media. URL: https://www.nltk.org/book.

Blair, R.C., Cole, S.R., 2002. Two-sided equivalence testing of the difference between two means. Journal of Modern Applied Statistical Methods 1, 139–142. doi:10.22237/jmasm/1020255540.

Blei, D.M., Lafferty, J.D., 2006. Dynamic topic models, in: Proceedings of the 23rd International Conference on Machine Learning, ACM. pp. 113–120. doi:10.1145/1143844.1143859.

Blei, D.M., Lafferty, J.D., 2007. A correlated topic model of Science. Annals of Applied Statistics 1, 17–35. doi:10.1214/07-AOAS114.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent Dirichlet allocation. Journal of Machine Learning Research 3, 993–1022. URL: http://dl.acm.org/citation.cfm?id=944919.944937.

Block, U., 2011. **ISOweek**: Week of the year and weekday according to ISO 8601. URL: https://CRAN.R-project.org/package=ISOweek. R package version 0.6.2.

Boiten, M., 2019. **rJST**: Joint sentiment topic modelling. URL: https://CRAN.R-project.org/package=rJST.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146. doi:10.1162/tacl_a_00051.

Bollerslev, T., Tauchen, G., Zhou, H., 2009. Expected stock returns and variance risk premia. Review of Financial Studies 22, 4463–4492. doi:10.1093/rfs/hhp008.

Borovkova, S., Garmaev, E., Lammers, P., Rustige, J., 2017. SenSR: A sentiment-based systemic risk indicator. Technical Report 553. De Nederlandsche Bank. URL: https://www.dnb.nl/binaries/Working%20Paper%20No.%20553_tcm46-356707.pdf.

Boudt, K., Cornelissen, J., Croux, C., 2013. The impact of a sustainability constraint on the mean-tracking error efficient frontier. Economics Letters 119, 255–260. doi:10.1016/j.econlet.2013.03.020.

Boudt, K., Thewissen, J., 2019. Jockeying for position in CEO letters: Impression management and sentiment analytics. Financial Management 48, 77–115. doi:10.1111/fima.12219.

Boudt, K., Thewissen, J., Torsin, W., 2018. When does the tone of earnings press releases matter? International Review of Financial Analysis 57, 231–245. doi:10.1016/j.irfa.2018.02.002.

Bradley, M.M., Lang, P.J., 1999. Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical Report. URL: https://www.uvm.edu/pdodds/teaching/courses/2009-08UVM-300/docs/others/everything/bradley1999a.pdf.

Calomiris, C.W., Mamaysky, H., 2019. How news and its context drive risk and returns around the world. Journal of Financial Economics 133, 299–336. doi:10.1016/j.jfineco.2018.11.009.

Cambria, E., Poria, S., Bajpai, R., Schuller, B., 2016. SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives, in: Proceedings of the 26th International Conference on Computational Linguistics, pp. 2666–2677. URL: https://www.aclweb.org/anthology/C16-1251.

Caporin, M., Poli, F., 2017. Building news measures from textual data and an application to volatility forecasting. Econometrics 5, 1–46. doi:10.3390/econometrics5030035.

Casey, G.P., Owen, A.L., 2013. Good news, bad news, and consumer confidence. Social Science Quarterly 94, 292–315. doi:10.1111/j.1540-6237.2012.00900.x.

Catania, L., Bernardi, M., 2017. **MCS**: Model confidence set procedure. URL: https://CRAN.R-project.org/package=MCS. R package version 0.1.3.

Ceron, E., Curini, L., Iacus, S., Porro, G., 2014. Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to Italy and France. New Media & Society 16, 340–358. doi:10.1177/1461444813480466.

Chang, C.C., Hsieh, P.F., Wang, Y.H., 2015. Sophistication, sentiment, and misreaction. Journal of Financial and Quantitative Analysis 50, 903–928. doi:10.1017/S0022109015000290.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., 2017. Double/debiased/Neyman machine learning of treatment effects. American Economic Review 107, 261–265. doi:10.1257/aer.p20171038.

Chiou, L., Tucker, C., 2017. Content aggregation by platforms: The case of the news media. Journal of Economics & Management Strategy 26, 782–805. doi:10.1111/jems.12207.

R Core Team, 2019. R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org. R version 3.6.2.

Croushore, D., Stark, T., 2003. A real-time data set for macroeconomists: Does the data vintage matter? Review of Economics and Statistics 85, 605–617. doi:10.1162/003465303322369759.

Cukier, K., Mayer-Schoenberger, V., 2013. The rise of big data: How it's changing the way we think about the world. Foreign Affairs 92, 28–40. URL: https://www.jstor.org/stable/23526834.

Das, S.R., Chen, M.Y., 2007. Yahoo! for Amazon: Sentiment extraction from small talk on the web. Management Science 53, 1375–1388. doi:10.1287/mnsc.1070.0704.

De Clercq, O., Lefever, E., Jacobs, G., Carpels, T., Hoste, V., 2017. Towards an integrated pipeline for aspect-based sentiment analysis in various domains, in: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, ACM. pp. 136–142. doi:10.18653/v1/W17-5218.

De Long, J.B., Shleifer, A., Summers, L.H., Waldmann, R.J., 1990. Noise trader risk in financial markets. Journal of Political Economy 98, 703–738. URL: https://www.jstor.org/stable/2937765.

Denny, M.J., Spirling, A., 2018. Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. Political Analysis 26, 168–189. doi:10.1017/pan.2017.44.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. URL: https://arxiv.org/abs/1810.04805v2. Working paper.

Diamond, D.W., Dybvig, P.H., 1983. Bank runs, deposit insurance, and liquidity. Journal of Political Economy 91, 401–419. doi:10.1086/261155.

Dowle, M., Srinivasan, A., 2019. **data.table**: Extension of '**data.frame**'. URL: https://CRAN.R-project.org/package=data.table. R package version 1.12.8.

Drei, A., Le Guenedal, T., Lepetit, F., Mortier, V., Roncalli, T., Sekine, T., 2019. ESG investing in recent years: New insights from old challenges. URL: http://research-center.amundi.com. Amundi discussion paper 42.

Eddelbuettel, D., Francois, R., 2011. **Rcpp**: Seamless R and C++ integration. Journal of Statistical Software 40, 1–18. doi:10.18637/jss.v040.i08.

Eddelbuettel, D., Sanderson, C., 2014. **RcppArmadillo**: Accelerating R with high-performance C++ linear algebra. Computational Statistics and Data Analysis 71, 1054–1063. doi:10.1016/j.csda.2013.02.005.

Eguchi, K., Lavrenko, V., 2006. Sentiment retrieval using generative models, in: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, ACM. pp. 345–354. doi:10.3115/1610075.1610124.

Ekman, P., Friesen, W.V., 1976. Measuring facial movement. Environmental Psychology and Nonverbal Behavior 1, 56–75. doi:10.1007/BF01115465.

Engle, R., Giglio, S., Kelly, B., Lee, H., Stroebel, J., 2020. Hedging climate change news. Review of Financial Studies 33, 1184–1216. doi:10.1093/rfs/hhz072.

Escrig-Olmedo, E., Muñoz-Torres, M.J., Fernandez-Izquierdo, M.A., 2010. Socially responsible investing: Sustainability indices, ESG rating and information provider agencies. International Journal of Sustainable Economy 2, 442–461.

Eshbaugh-Soha, M., 2010. The tone of local presidential news coverage. Political Communication 27, 121–140. doi:10.1080/10584600903502623.

Evans, J.A., Aceves, P., 2016. Machine translation: Mining text for social theory. Annual Review of Sociology 42, 21–50. doi:10.1146/annurev-soc-081715-074206.

Eyben, F., Weninger, F., Gross, F., Schuller, B., 2013. Recent developments in openSMILE, the Munich open-source multimedia feature extractor, in: Proceedings of the 21st ACM International Conference on Multimedia, ACM. pp. 835–838. doi:10.1145/2502081.2502224.

Feinerer, I., Hornik, K., Meyer, D., 2008. Text mining infrastructure in R. Journal of Statistical Software 22, 1–54. doi:10.18637/jss.v025.i05.

Feldman, R., Govindaraj, S., Livnat, J., Segal, B., 2010. Management's tone change, post earnings announcement drift and accruals. Review of Accounting Studies 15, 915–953. doi:10.1007/s11142-009-9111-x.

Feuerriegel, S., Pröllochs, N., 2019. **SentimentAnalysis**: Dictionary-based sentiment analysis. URL: https://CRAN.R-project.org/package=SentimentAnalysis.

Flaxman, S., Goel, S., Rao, J., 2016. Filter bubbles, echo chambers, and online news consumption. Public Opinion Quarterly 80, 298–320. doi:10.1093/poq/nfw006.

Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33, 1–22. doi:10.18637/jss.v033.i01.

Fu, X., Yang, K., Huang, J.Z., Cui, L., 2015. Dynamic non-parametric joint sentiment topic mixture model. Knowledge-Based Systems 82, 102–114. doi:10.1016/j.knosys.2015.02.021.

Gagolewski, M., 2020. **stringi**: Character string processing facilities. URL: https://CRAN.R-project.org/package=stringi. R package version 1.4.5.

Gandomi, A., Haider, M., 2015. Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management 35, 137–144. doi:10.1016/j.ijinfomgt.2014.10.007.

García, D., 2013. Sentiment during recessions. Journal of Finance 68, 1267–1300. doi:10.1111/jofi.12027.

Garz, M., 2014. Good news and bad news: Evidence of media bias in unemployment reports. Public Choice 161, 499–515. doi:10.1007/s11127-014-0182-2.

Gelper, S., Croux, C., 2010. On the construction of the European economic sentiment indicator. Oxford Bulletin of Economics and Statistics 72, 47–62. doi:10.1111/j.1468-0084.2009.00574.x.

Gelper, S., Peres, R., Eliashberg, J., 2018. Talk bursts: The role of spikes in pre-release word-of-mouth dynamics. Journal of Marketing Research 55, 801–817. doi:10.1177/0022243718817007.

Gentzkow, M., Kelly, B., Taddy, M., 2019a. Text as data. Journal of Economic Literature 57, 535–574. doi:10.1257/jel.20181020.

Gentzkow, M., Shapiro, J.M., 2010. What drives media slant? Evidence from U.S. daily newspapers. Econometrica 78, 35–71. doi:10.3982/ECTA7195.

Gentzkow, M., Shapiro, J.M., Taddy, M., 2019b. Measuring group differences in high-dimensional choices: Method and application to congressional speech. Econometrica 87, 1307–1340. doi:10.3982/ECTA16566.

Ghirelli, C., Pérez, J., Urtasun, A., 2019. A new economic policy uncertainty index for Spain. Economics Letters 182, 64–67. doi:10.1016/j.econlet.2019.05.021.

Ghysels, E., Sinko, A., Valkanov, R., 2007. MIDAS regressions: Further results and new directions. Econometrics Review 26, 53–90. doi:10.1080/07474930600972467.

Glanzer, M., Cunitz, A.R., 1966. Two storage mechanisms in free recall. Journal of Verbal Learning and Verbal Behavior 5, 351–360. doi:10.1016/S0022-5371(66)80044-0.

Glasserman, P., Mamaysky, H., 2019. Does unusual news forecast market stress? Journal of Financial and Quantitative Analysis 54, 1937–1974. doi:10.1017/S0022109019000127.

Grimmer, J., Stewart, B.M., 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. Political Analysis 21, 267–297. doi:10.1093/pan/mps028.

Grolemund, G., Wickham, H., 2011. Dates and times made easy with **lubridate**. Journal of Statistical Software 40, 1–25. doi:10.18637/jss.v040.i03.

Hamilton, W.L., Clark, K., Leskovec, J., Jurafsky, D., 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, ACM. pp. 595–605. doi:10.18653/v1/D16-1057.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., Ng, A.Y., 2014. Deep speech: Scaling up end-to-end speech recognition. URL: http://arxiv.org/abs/1412.5567. Working paper.

Hansen, P., Lunde, A., Nason, J., 2011. The model confidence set. Econometrica 79, 453–497. doi:10.3982/ECTA5771.

Hartzmark, S., Sussman, A., 2019. Do investors value sustainability? A natural experiment examining ranking and fund flows. Journal of Finance 74, 2789–2837. doi:10.1111/jofi.12841.

Hatzivassiloglou, V., McKeown, K.R., 1997. Predicting the semantic orientation of adjectives, in: Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, pp. 174–181. doi:10.3115/976909.979640.

He, Y., Lin, C., Gao, W., Wong, K.F., 2013. Dynamic joint sentiment-topic model. ACM Transactions on Intelligent Systems and Technology 5, 1–21. doi:10.1145/2542182.2542188.

Hendry, D.F., 1980. Econometrics – Alchemy or science? Economica 47, 387–406. doi:10.2307/2553385.

Henry, E., 2008. Are investors influenced by how earnings press releases are written? Journal of Business Communication 45, 363–407. doi:10.1177/0021943608319388.

Heston, S., Sinha, N., 2017. News vs. sentiment: Predicting stock returns from news stories. Financial Analysts Journal 73, 67–83. doi:10.2469/faj.v73.n3.3.

Hoerl, A., Kennard, R., 1970. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12, 55–67. doi:10.1080/00401706.1970.10488634.

Hofmann, T., 2001. Unsupervised learning by probabilistic latent semantic analysis. Machine Learning 42, 177–196. doi:10.1023/A:1007617005950.

Honnibal, M., Montani, I., 2017. **spaCy** 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. URL: https://spacy.io.

Hornik, K., 2016. **openNLP**: **Apache OpenNLP** tools interface. URL: https://CRAN.R-project.org/package=openNLP. R package version 0.2.6.

Hu, M., Liu, B., 2004a. Mining and summarizing customer reviews, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. doi:10.1145/1014052.1014073.

Hu, M., Liu, B., 2004b. Mining opinion features in customer reviews, in: Proceedings of the 19th National Conference on Artificial Intelligence, AAAI Press. pp. 755–760. URL: http://dl.acm.org/citation.cfm?id=1597148.1597269.

Huang, X., Teoh, S.H., Zhang, Y., 2014. Tone management. Accounting Review 89, 1083–1113. doi:10.2308/accr-50684.

Hubert, P., Labondance, F., 2018. Central bank sentiment. URL: https://www.nbp.pl/badania/seminaria/14xi2018.pdf. Working paper.

Hutto, C.J., Gilbert, E., 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text, in: Proceedings of the 8th International AAAI Conference on Weblogs and Social Media, pp. 216–225. URL: https://pdfs.semanticscholar.org/a6e4/a2532510369b8f55c68f049ff11a892fefeb.pdf.

Jegadeesh, N., Wu, D., 2013. Word power: A new approach for content analysis. Journal of Financial Economics 110, 712–729. doi:10.1016/j.jfineco.2013.08.018.

Jockers, M., 2017. **syuzhet**: Extract sentiment and plot arcs from text. URL: https://CRAN.R-project.org/package=syuzhet. R package version 1.0.4.

Joshi, M., Das, D., Gimpel, K., Smith, N., 2010. Movie reviews and revenues: An experiment in text regression, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 293–296. URL: http://dl.acm.org/citation.cfm?id=1857999.1858037.

Kalogeropoulos, A., 2018. Economic news and personal economic expectations. Mass Communication and Society 21, 248–265. doi:10.1080/15205436.2017.1403629.

Kanayama, H., Nasukawa, T., 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis, in: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 355–363. URL: https://www.aclweb.org/anthology/W06-1642.

Van de Kauter, M., Desmet, B., Hoste, V., 2015. The good, the bad and the implicit: A comprehensive approach to annotating explicit and implicit sentiment. Language Resources & Evaluation 49, 685–720. doi:10.1007/s10579-015-9297-4.

Kearney, C., Liu, S., 2014. Textual sentiment in finance: A survey of methods and models. International Review of Financial Analysis 33, 171–185. doi:10.1016/j.irfa.2014.02.006.

Kelly, B.T., Manela, A., Moreira, A., 2019. Text selection. doi:10.2139/ssrn.3491942. Working paper.

Keynes, J.M., 1936. The general theory of employment, interest, and money. Palgrave Macmillan, London, UK. doi:10.1007/978-3-319-70344-2.

Kräussl, R., Mirgorodskaya, E., 2017. Media, sentiment and market performance in the long run. European Journal of Finance 23, 1059–1082. doi:10.1080/1351847X.2016.1226188.

Krijthe, J., van der Maaten, L., 2018. **Rtsne**: T-distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation. URL: https://CRAN.R-project.org/package=Rtsne. R Package Version 0.15.

Kroese, L., Kok, S., Parleviet, J., 2015. Beleidsonzekerheid in Nederland. Economisch-Statistische Berichten (ESB) 100, 4715. URL: https://www.policyuncertainty.com/netherlands_monthly.html.

Kuhn, M., 2018. **caret**: Classification and regression training. URL: https://CRAN.R-project.org/package=caret.

Labille, K., Gauch, S., Alfarhood, S., 2017. Creating domain-specific sentiment lexicons via text mining, in: Proceedings of the 6th KDD Workshop on Issues of Sentiment Discovery and Opinion Mining, pp. 1–9. URL: https://sentic.net/wisdom2017labille.pdf.

Lacy, S., Watson, B.R., Riffe, D., Lovejoy, J., 2015. Issues and best practices in content analysis. Journalism & Mass Communication Quarterly 92, 791–811. doi:10.1177/1077699015607338.

Larsen, V.H., Thorsrud, L.A., 2018. Business cycle narratives. Technical Report 7468. CESifo. URL: https://ssrn.com/abstract=3338822.

Larsen, V.H., Thorsrud, L.A., 2019. The value of news for economic developments. Journal of Econometrics 210, 203–218. doi:10.1016/j.jeconom.2018.11.013.

Larsen, V.H., Thorsrud, L.A., Zhulanova, J., 2020. News-driven inflation expectations and information rigidities. doi:10.1016/j.jmoneco.2020.03.004. Forthcoming in Journal of Monetary Economics.

Leskovec, J., Rajaraman, A., Ullman, J., 2014. Mining of massive datasets. Cambridge University Press. chapter Finding similar items. pp. 72–134. doi:https://doi.org/10.1017/CBO9781139924801.

Lewis, C., Young, S., 2019. Fad or future? Automated analysis of financial text and its implications for corporate reporting. Accounting and Business Research 49, 587–615. doi:10.1080/00014788.2019.1611730.

Lin, C., He, Y., 2009. Joint sentiment/topic model for sentiment analysis, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM. pp. 375–384. doi:10.1145/1645953.1646003.

Liu, B., 2015. Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press. doi:10.1017/CBO9781139084789.

Liu, Y., Yu, X., Huang, X., An, A., 2009. Data mining for business applications. Springer. chapter Blog data mining: The predictive power of sentiments. pp. 183–195. doi:10.1007/978-0-387-79420-4_13.

Loria, S., 2019. **TextBlob**: Simplified text processing. URL: https://github.com/sloria/TextBlob.

Loughran, T., McDonald, B., 2011. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. Journal of Finance 66, 35–65. doi:10.1111/j.1540-6261.2010.01625.x.

Loughran, T., McDonald, B., 2014. Measuring readability in financial disclosures. Journal of Finance 69, 1643–1671. doi:10.1111/jofi.12162.

Loughran, T., McDonald, B., 2016. Textual analysis in accounting and finance: A survey. Journal of Accounting Research 54, 1187–1230. doi:10.1111/1475-679X.12123.

Lowry, D., 2008. Network TV news framing of good vs. bad economic news under Democrat and Republican presidents: A lexical analysis of political bias. Journalism & Mass Communication Quarterly 85, 483–498. doi:10.1177/107769900808500301.

Ludvigson, S.C., 2004. Consumer confidence and consumer spending. Journal of Economic Perspectives 18, 29–50. doi:10.1257/0895330041371222.

Lukeš, J., Søgaard, A., 2018. Sentiment analysis under temporal shift, in: Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, ACM. pp. 65–71. URL: https://www.aclweb.org/anthology/W18-6210.

Lütkepohl, 2017. Estimation of structural vector autoregressive models. Communications for Statistical Applications and Methods 24, 421–441. doi:10.5351/CSAM.2017.24.5.421.

Manela, A., Moreira, A., 2017. News implied volatility and disaster concerns. Journal of Financial Economics 123, 137–162. doi:10.1016/j.jfineco.2016.01.032.

Mayew, W.J., Venkatachalam, M., 2012. The power of voice: Managerial affective states and future firm performance. Journal of Finance 67, 1–43. doi:10.1111/j.1540-6261.2011.01705.x.

McCracken, M.W., Ng, S., 2016. FRED-MD: A monthly database for macroeconomic research. Journal of Business & Economic Statistics 34, 574–589. doi:10.1080/07350015.2015.1086655.

Mersmann, O., Ulrich, J., 2019. **microbenchmark**: Accurate timing functions. URL: https://CRAN.R-project.org/package=microbenchmark. R package version 1.4.7.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013. Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 3111–3119. URL: http://dl.acm.org/citation.cfm?id=2999792.2999959.

Miller, G.A., 1995. WordNet: A lexical database for English. Communications of the ACM 38, 39–41. doi:10.1145/219717.219748.

Mohammad, S., Salameh, M., Kiritchenko, S., 2016. How translation alters sentiment. Journal of Artificial Intelligence Research 55, 95–130. doi:10.1613/jair.4787.

Mohammad, S., Turney, P., 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon, in: Proceedings of the Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pp. 26–34. URL: http://dl.acm.org/citation.cfm?id=1860631.1860635.

Mohammad, S.M., Turney, P.D., 2013. Crowdsourcing a word-emotion association lexicon. Computational Intelligence 29, 436–465. URL: https://arxiv.org/pdf/1308.6297.pdf.

Moniz, A., 2016. Inferring the financial materiality of corporate social responsibility news. doi:10.2139/ssrn.2761905. Working paper.

Mullen, L., 2016. **textreuse**: Detect text reuse and document similarity. URL: https://CRAN.R-project.org/package=textreuse. R package version 0.1.4.

Munezero, M.D., Montero, C.S., Sutinen, E., Pajunen, J., 2014. Are they different? Affect, feeling, emotion, sentiment, and opinion detection in text. IEEE Transactions on Affective Computing 5, 101–111. doi:10.1109/TAFFC.2014.2317187.

Nimark, K.P., Pitschner, S., 2019. News media and delegated information choice. Journal of Economic Theory 181, 160–196. doi:10.1016/j.jet.2019.02.001.

Nowak, A., Smith, P., 2017. Textual analysis in real estate. Journal of Applied Econometrics 32, 896–918. doi:10.1002/jae.2550.

Pang, B., Lee, L., Vaithyanathan, S., 2002. Thumbs up? Sentiment classification using machine learning techniques, in: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, pp. 79–86. doi:10.3115/1118693.1118704.

Pástor, L., Veronesi, P., 2013. Political uncertainty and risk premia. Journal of Financial Economics 110, 520–545. doi:10.1016/j.jfineco.2013.08.007.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. **Scikit-learn**: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830. URL: http://www.jmlr.org/papers/v12/pedregosa11a.html.

Pennington, J., Socher, R., Manning, C., 2014. GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, ACM. pp. 1532–1543. doi:10.3115/v1/D14-1162.

Petropoulos Petalas, D., van Schie, H., Hendriks Vettehen, P., 2017. Forecasted economic change and the self-fulfilling prophecy in economic decision-making. PLoS ONE 12, e0174353. doi:10.1371/journal.pone.0174353.

Pham, H., Manzini, T., Liang, P.P., Poczos, B., 2018. Seq2Seq2Sentiment: Multimodal sequence to sequence models for sentiment analysis, in: Proceedings of the Grand Challenge and Workshop on Human Multimodal Language, ACM. pp. 53–63. URL: https://www.aclweb.org/anthology/W18-3308.

Picault, M., Renault, T., 2017. Words are not all created equal: A new measure of ECB communication. Journal of International Money and Finance 79, 136–156. doi:10.1016/j.jimonfin.2017.09.005.

Poria, S., Cambria, E., Howard, N., Huang, G.B., Hussain, A., 2016. Fusing audio, visual and textual clues for sentiment analysis from multimodal content. Neurocomputing 174, 50–59. doi:10.1016/j.neucom.2015.01.095.

Pröllochs, N., Feuerriegel, S., Neumann, D., 2015. Generating domain-specific dictionaries using Bayesian learning, in: Proceedings of the European Conference on Information Systems, pp. 1–14. doi:10.18151/7217444.

Qin, D., 2011. Rise of VAR modelling approach. Journal of Economic Surveys 25, 156–174. doi:10.1111/j.1467-6419.2010.00637.x.

Ravi, K., Ravi, V., 2015. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. Knowledge-Based Systems 89, 14–46. doi:10.1016/j.knosys.2015.06.015.

Remus, R., Quasthoff, U., Heyer, G., 2010. SentiWS – A publicly available German-language resource for sentiment analysis, in: Proceedings of the Seventh Conference on International Language Resources and Evaluation, European Languages Resources Association (ELRA). pp. 1168–1171. URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/490_Paper.pdf.

Ribeiro, F.N., Araújo, M., Gonçalves, P., Gonçalves, M.A., Benevenuto, F., 2016. SentiBench – A benchmark comparison of state-of-the-practice sentiment analysis methods. EPJ Data Science 5, 1–23. doi:10.1140/epjds/s13688-016-0085-1.

Ridout, T.N., Fowler, E.F., Searles, K., 2012. Exploring the validity of electronic newspaper databases. International Journal of Social Research Methodology 15, 451–466. doi:10.1080/13645579.2011.638221.

Riedl, A., Smeets, P., 2017. Why do investors hold socially responsible mutual funds? Journal of Finance 72, 2505–2550. doi:10.1111/jofi.12547.

Riffe, D., Lacy, S., Watson, B.R., Fico, F., 2019. Analyzing media messages: Using quantitative content analysis in research. Routledge. doi:10.4324/9780203551691.

Rinker, T., 2017. **qdap**: Quantitative discourse analysis package. URL: https://CRAN.R-project.org/package=qdap. R package version 2.3.0.

Rinker, T., 2018. **sentimentr**: Calculate text polarity sentiment. URL: https://CRAN.R-project.org/package=sentimentr. R package version 2.6.1.

Rinker, T., 2019. **lexicon**: Lexicon data. URL: https://CRAN.R-project.org/package=lexicon. R package version 1.2.1.

Roberts, M., Stewart, B., Tingley, D., 2019. **stm**: R package for structural topic models. Journal of Statistical Software 91, 1–40. doi:10.18637/jss.v091.i02.

Roberts, M.E., Stewart, B.M., Airoldi, E.M., 2016. A model of text for experimentation in the social sciences. Journal of the American Statistical Association 111, 988–1003. doi:10.1080/01621459.2016.1141684.

Rogers, J.L., Van Buskirk, A., Zechman, S.L.C., 2011. Disclosure tone and shareholder litigation. Accounting Review 86, 2155–2183. doi:10.2308/accr-10137.

Rousseeuw, P., Raymaekers, J., Hubert, M., 2018. A measure of directional outlyingness with applications to image data and video. Journal of Computational and Graphical Statistics 27, 345–359. doi:10.1080/10618600.2017.1366912.

Saleiro, P., Rodrigues, E., Soares, C., Oliveira, E., 2017. TexRep: A text mining framework for online reputation monitoring. New Generation Computation 35, 365–389. doi:10.1007/s00354-017-0021-3.

Saltzis, K., 2012. Breaking news online. Journalism Practice 6, 702–710. doi:10.1080/17512786.2012.667274.

Scheufele, D.A., Tewksbury, D., 2007. Framing, agenda setting, and priming: The evolution of three media effects models. Journal of Communication 57, 9–20. doi:10.1111/j.0021-9916.2007.00326.x.

Schmidt, D., 2019. **meanr**: Sentiment analysis scorer. URL: https://CRAN.R-project.org/package=meanr. R package version 0.1.2.

Selivanov, D., Wang, Q., 2018. **text2vec**: Modern text mining framework for R. URL: https://CRAN.R-project.org/package=text2vec. R package version 0.5.1.

Shapiro, A.H., Südhof, M., Wilson, D., 2018. Measuring news sentiment. Technical Report 2017-01. Federal Reserve Bank of San Francisco. doi:10.24148/wp2017-01.

Shiller, R.J., 2017. Narrative economics. American Economic Review 107, 967–1004. doi:10.1257/aer.107.4.967.

Silge, J., Robinson, D., 2016. **tidytext**: Text mining and analysis using tidy data principles in R. Journal of Open Source Software 1. doi:10.21105/joss.00037.

Soleymani, M., Garcia, D., Jou, B., Schuller, B., Chang, S.F., Pantic, M., 2017. A survey of multimodal sentiment analysis. Image and Vision Computing 65, 3–14. doi:10.1016/j.imavis.2017.08.003.

Soo, C.K., 2018. Quantifying sentiment with news media across local housing markets. Review of Financial Studies 31, 3689–3719. doi:10.1093/rfs/hhy036.

Spärck, J., 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28, 11–21. doi:10.1108/eb026526.

Stone, P.J., Dunphy, D.C., Smith, M.S., 1963. The General Inquirer: A computer approach to content analysis, in: Proceedings of the American Federation of Information Processing Societies spring joint computer conference, pp. 241–256. doi:10.1145/1461551.1461583.

Strapparava, C., Valitutti, A., 2004. WordNet-Affect: An affective extension of WordNet, in: Proceedings of the Fourth International Conference on Language Resources and Evaluation. URL: http://www.lrec-conf.org/proceedings/lrec2004/pdf/369.pdf.

Taboada, M., 2016. Sentiment analysis: An overview from linguistics. Annual Review of Linguistics 2, 325–347. doi:10.1146/annurev-linguistics-011415-040518.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M., 2011. Lexicon-based methods for sentiment analysis. Computational Linguistics 37, 267–307. doi:10.1162/COLI_a_00049.

Täckström, O., McDonald, R., 2011. Discovering fine-grained sentiment with latent variable structured prediction models, in: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (Eds.), Proceedings of the Advances in Information Retrieval, Springer. pp. 368–374. doi:10.1007/978-3-642-20161-5_37.

Taddy, M., 2013a. Measuring political sentiment on Twitter: Factor optimal design for multinomial inverse regression. Technometrics 55, 415–425. doi:10.1080/00401706.2013.778791.

Taddy, M., 2013b. Multinomial inverse regression for text analysis. Journal of the American Statistical Association 108, 755–770. doi:10.1080/01621459.2012.734168.

Taddy, M., 2015a. Distributed multinomial regression. Annals of Applied Statistics 9, 1394–1414. doi:10.1214/15-AOAS831.

Taddy, M., 2015b. Document classification by inversion of distributed language representations, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 45–49. doi:10.3115/v1/P15-2008.

Taddy, M., 2018. **textir**: Inverse regression for text analysis. URL: https://CRAN.R-project.org/package=textir.

Teoh, S.H., 2018. The promise and challenges of new datasets for accounting research. Accounting, Organizations and Society 68-69, 109–117. doi:10.1016/j.aos.2018.03.008.

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

Tetlock, P.C., Saar-Tsechansky, M., Macskassy, S., 2008. More than words: Quantifying language to measure firms' fundamentals. Journal of Finance 63, 1437–1467. doi:10.1111/j.1540-6261.2008.01362.x.

Theil, C.K., Štajner, S., Stuckenschmidt, H., 2018. Word embeddings-based uncertainty detection in financial disclosures, in: Proceedings of the First Workshop on Economics and Natural Language Processing, pp. 32–37. doi:10.18653/v1/W18-3104.

Thorsrud, L.A., 2020. Words are the new numbers: A newsy coincident index of the business cycle. Journal of Business & Economic Statistics 38, 393–409. doi:10.1080/07350015.2018.1506344.

Tibshirani, R., 1996. Regression shrinkage and selection via the LASSO. Journal of the Royal Statistical Society B 58, 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x.

Tibshirani, R., Taylor, J., 2012. Degrees of freedom in LASSO problems. The Annals of Statistics 4, 1198–1232. doi:10.1214/12-AOS1003.

Tobback, E., Naudts, H., Daelemans, W., Junqué de Fortuny, E., Martens, D., 2018. Belgian economic policy uncertainty index: Improvement through text mining. International Journal of Forecasting 34, 355–365. doi:10.1016/j.ijforecast.2016.08.006.

Turney, P., 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, pp. 417–424. doi:10.3115/1073083.1073153.

Tversky, A., Kahneman, D., 1981. The framing of decisions and the psychology of choice. Science 211, 453–458. doi:10.1126/science.7455683.

Ushey, K., 2018. **RcppRoll**: Efficient rolling / windowed operations. URL: https://CRAN.R-project.org/package=RcppRoll. R package version 0.3.0.

Varian, H.R., 2014. Big data: New tricks for econometrics. Journal of Economic Perspectives 28, 3–28. doi:10.1257/jep.28.2.3.

Wang, H., Divakaran, A., Vetro, A., Chang, S.F., Sun, H., 2003. Survey of compressed-domain features used in audio-visual indexing and analysis. Journal of Visual Communication and Image Representation 14, 150–183. doi:10.1016/S1047-3203(03)00019-1.

Wang, J., Shen, H.T., Song, J., Ji, J., 2014. Hashing for similarity search: A survey. URL: http://arxiv.org/abs/1408.2927. Working paper.

Weston, S., 2019. **foreach**: Provides foreach looping construct for R. URL: https://CRAN.R-project.org/package=foreach. R package version 1.4.7.

Wickham, H., 2016. **ggplot2**: Elegant graphics for data analysis. Springer-Verlag New York. URL: http://ggplot2.org.

Wickham, H., François, R., Henry, L., Müller, K., 2019. **dplyr**: A grammar of data manipulation. URL: https://CRAN.R-project.org/package=dplyr. R package version 0.8.3.

Wickham, H., Henry, L., 2019. **tidyr**: Tidy messy data. URL: https://CRAN.R-project.org/package=tidyr. R package version 1.0.0.

Wijffels, J., 2019. **udpipe**: A tokenization, parts of speech tagging, lemmatization and dependency parsing with the UDPipe NLP toolkit. URL: https://CRAN.R-project.org/package=udpipe. R package version 0.8.3.

Wijffels, J., 2020. **BTM**: Biterm topic models for short text. URL: https://CRAN.R-project.org/package=BTM. R package version 0.3.1.

Wischnewsky, A., Jansen, D.J., Neuenkirch, M., 2019. Financial stability and the Fed: Evidence from congressional hearings. Technical Report 633. De Nederlandsche Bank. URL: https://www.dnb.nl/binaries/Working%20paper%20No.%20633_tcm46-383881.pdf.

Yan, X., Guo, J., Lan, Y., Cheng, X., 2013. A biterm topic model for short texts, in: Proceedings of the 22nd international conference on World Wide Web, pp. 1445–1456. doi:10.1145/2488388.2488514.

Yogatama, D., Heilman, M., O'Connor, B., Dyer, C., Routledge, B., Smith, N., 2011. Predicting a scientific community's response to an article, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 594–604. URL: https://www.aclweb.org/anthology/D11-1055.

Young, L., Soroka, S., 2012. Affective news: The automated coding of sentiment in political texts. Political Communication 29, 205–231. doi:10.1080/10584609.2012.671234.

Yuan, M., Lin, Y., 2006. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B 68, 49–67. doi:10.1111/j.1467-9868.2005.00532.x.

Zeileis, A., Grothendieck, G., 2005. **zoo**: S3 infrastructure for regular and irregular time series. Journal of Statistical Software 14, 1–27. doi:10.18637/jss.v014.i06.

Zhang, M.L., Zhou, Z.H., 2014. A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering 26, 1819–1837. doi:10.1109/TKDE.2013.39.

Zhang, X., Zhao, J., LeCun, Y., 2015. Character-level convolutional networks for text classification, in: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 28. Curran Associates, Inc., pp. 649–657. URL: http://arxiv.org/abs/1509.01626.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B 67, 301–320. doi:10.1111/j.1467-9868.2005.00503.x.