Saulo Benchimol Bastos

# Document representations and its measurements applied to Finance

Brasília

2019

Saulo Benchimol Bastos

# Document representations and its measurements applied to Finance

Tese apresentada ao Programa de Pós-Gradução em Economia, do Departamento de Economia da Universidade de Brasília, como parte dos requisitos para a obtenção do título de Doutor em Economia.

Área de concentração: Economia Aplicada.

Universidade de Brasília – UnB

Faculdade de Economia, Administração e Contabilidade – FACE

Programa de Pós-Graduação em Economia

Orientador: Dr. Daniel Oliveira Cajueiro

Brasília

2019

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

BB327d

Bastos, Saulo Benchimol
Document representations and its measurements applied to Finance / Saulo Benchimol Bastos; orientador Daniel Oliveira Cajueiro. -- Brasília, 2019.
116 p.

Tese (Doutorado - Doutorado em Economia) -- Universidade de Brasília, 2019.

1. Análise de texto. 2. Representação de documentos. 3. Análise de sentimento. 4. Mercado financeiro. I. Cajueiro, Daniel Oliveira, orient. II. Título.

Saulo Benchimol Bastos

# Document representations and its measurements applied to Finance

Tese apresentada ao Programa de Pós-Gradução em Economia, do Departamento de Economia da Universidade de Brasília, como parte dos requisitos para a obtenção do título de Doutor em Economia.

Área de concentração: Economia Aplicada.

Trabalho aprovado. Brasília, 28 de junho de 2019:

---

**Dr. Daniel Oliveira Cajueiro** (Orientador)
Universidade de Brasília

---

**Dr. Herbert Kimura**
Universidade de Brasília

---

**Dr. José Guilherme de Lara Resende**
Universidade de Brasília

---

**Dr. Thiago Christiano Silva**
Banco Central do Brasil

Brasília
2019

# Acknowledgements

I thank my beautiful wife, Isabela, and my perfect son, Samuel, for being on my side. Without them, I would be nothing. They are my eternal motivation to always be a better person and to keep moving forward.

I thank my parents, Nilson and Grace, my anchors, my guides. I thank my brother, Nelio. I also thank my sister, Liliane, one of the greatest fighters of all times, my ultimate example that all problems are small.

I thank my academic advisor, Daniel, whose commitment to passing knowledge and to doing good surpasses any imaginable boundaries. It was a pleasure and an absolute honor to be his pupil.

I thank all professors for shaping my learning.

I thank all colleagues for sharing this journey with me.

I thank Thadeu Penna for the access to the cluster.

I thank all other people, who somehow helped on my path, but I do not mention explicitly.

*"All models are wrong, some are useful".*
*(George Box, 1976)*

# Resumo

Uma representação de documento é a descrição matemática de um texto. Aprender a representar informação é o passo inicial para uma extração automatizada de conhecimento. Reescrevemos a metodologia para extrair sentimento do texto, presente na literatura econômica, como um problema de recuperação de informação, possibilitando assim a aplicação de técnicas consagradas em ciência da computação. Mostramos que a escolha da ponderação adequada da matriz TF-IDF (frequência do termo-inverso da frequência do documento) e representações densas levam a resultados mais consistentes. Além disso, usamos documentos completos, em vez de versões filtradas com dicionários, como variáveis de séries temporais, o que só foi possível devido a representações densas. Propomos dois modelos para extrair sentimento do texto. Primeiro, um que aprende o vocabulário de acordo com movimentos em uma variável específica. Validamos nosso modelo usando o retorno *overnight* no mercado de ações; encontramos evidências de que o sentimento prevê retornos, que as notícias em $t-1$ têm o maior efeito sobre os retornos em $t$ e que a positividade ou negatividade de uma palavra depende do contexto. Segundo, usamos variáveis quantitativas e texto para criar um vetor de sentimento cujas coordenadas se relacionam entre si, em vez de simples números. Encontramos com sucesso estados opostos em um sentimento bidimensional, otimismo e pessimismo, cujas regressões em variáveis do mercado financeiro produzem resultados que têm amparo em teorias financeiras.

**Palavras-chave**: análise textual, representação de documentos, análise de sentimento, mercado financeiro.

# Abstract

A document representation is the mathematical description of a text. Learning how to represent information is the initial step towards an automated extraction of knowledge. We rewrite the methodology to extract sentiment from text, present in economic literature, as an information retrieval problem, thus enabling the application of consecrated techniques in computer science. We show that the choice of an adequate weighting scheme of the TF-IDF (Term-Frequency Inverse-Document-Frequency) matrix and dense representations leads to more consistent results. Also, we use whole documents, instead of filtered versions with dictionaries, as time series variables, which was only possible due to dense representations. We propose two models to extract sentiment from text. First, one that learns the vocabulary according to movements in a specific variable. We validate our model using the overnight return in the stock market; we find evidence that the sentiment predicts returns, that news in $t-1$ has the greatest effect on the overnight returns in $t$, and that the positivity or the negativity of a word depends on the context. Second, we use quantitative variables and text to create a sentiment vector whose coordinates relate to each other, instead of single numbers. We successfully find opposite states in a two dimensional sentiment, optimism and pessimism, whose regressions on financial market variables produce results that are supported by financial theories.

**Keywords**: text analysis, document representation, sentiment analysis, stock market.

# List of Figures

# List of Tables

# Contents

# Part I

# Introduction

# 1 Where does text analysis stand in the Finance literature?

## 1.1 Introduction

The current thesis explores textual analysis in Finance. Reading and understanding text is part of any qualitative analysis, e.g. fundamental analysis of the stock market or macro-economic factors. Mathematically represent information from text helped us to extract quantitative measures from a text.

Section 1.2 (Finance and the media) briefly presents some interesting results in the Finance literature.

Chapter 2 (Document representations and information measurements in time series) focus on document representations and its measurements. Text can be mathematically represented in may ways. Articles in Finance usually represent documents in sparse and high-dimensional spaces, but many results in computer science show that working in low and dense spaces has many advantages. Also, this chapter tries to bring together the literature in Finance and computer science regarding text analysis by formalizing key procedures used in articles (and yet are not seen elsewhere). Finally, it proposes some alternative methods to extract information from text.

Chapter 3 (Do prices absorb public information?) hypothesizes if it is possible to classify news into positive or negative using signals from the stock market, instead of humans. If so, it is possible to create a sentiment measurement based on learning the vocabulary that affects most returns. This sentiment corresponds to the expectations of future market movements, and it is used to verify if the market absorbs public information. We contrast positive and negative words from dictionaries and the learned vocabulary.

Chapter 4 (Modeling an analyst) creates a sentiment vector that incorporates past information on quantitative variables and news as well. A measure of optimism and pessimism is extracted directly from all available information, without the use of dictionaries.

## 1.2 Finance and the media

Articles in finance usually use the sentiment measure in a regression to understand the media influence over another variable. Li (2011) acknowledges that there are a lot of papers based on methodology and economic hypotheses, and most effort is directed to methodology, leaving hypotheses not well developed. Hopefully future research is likely to benefit more from developing hypotheses that are more closely tied to economic theories. Loughran and McDonald (2016) incisively state that the literature needs to be less centered on finding ways to apply off-the-shelf textual methods borrowed from highly evolved technologies in computational linguistics and instead be more motivated by hypotheses closely tied to economic theories. This is true, but I believe both need to walk together. State-of-the-art methods in computational linguistics might help find information where other methods could not find

before, and this could base an economic hypothesis. But the true reason, the economic hypothesis, should be the main motivation and never should be forgotten.

There are several approaches when it comes to the impact of the media on financial assets such as:

- Credit crisis in the banking sector;
- News' partiality of local newspapers, which impacts on the equity value of local companies;
- Asymmetry of information between domestic and foreign investors;
- Media coverage and the Investor Recognition Hypothesis;
- Sentiment and its impact on the stock market;
- How the market reacts to stale information;
- Event studies and tests for private information;
- Readability.

Next, I will briefly expose the main results of each of them so that it motivates this thesis and inspires future work.

### 1.2.1 Credit crisis in the banking sector

The health of the financial system depends on trust, which is the basis for financial institutions remaining solvent. Lack of confidence by inducing the belief that people will withdraw their deposits can lead to a panic situation. The massive withdrawal of deposits may cause a mismatch of assets and liabilities. The confidence on which the success of banking operations is based can be threatened by the media.

This section focuses on the work of Wisniewski and Lambe (2013), which proposes to study the dynamical relationship and quantify the intensity of speculative media and the performance of banking institutions.

The media data was extracted from the LexisNexis[1] database. The measure of pessimism in the media corresponds to the number of articles containing the following negative expressions in a monthly range: Credit Crunch; Financial Crisis; and Bank Failures.

The variation of the stock price of the banking institutions was represented by the FTSE[2] weighted banking index. The chosen period was from January 2005 to May 2010.

The Granger causality test, proposed by Granger (1969), was used to verify that the current climate of journalistic opinion can influence future values of actions of the banking sector. The media not only records the state of a particular economic reality, but also plays a key role in its creation.

A media-based stock trading strategy has also been built, which can be very useful for investors.

[1] <http://www.lexisnexis.com>
[2] <http://www.ftse.com>

### 1.2.2 News' partiality of local newspapers

Gurun and Butler (2012) study the value of companies against the positive coverage of local newspapers on local companies compared to the negative coverage of local media on non-local companies.

It has been found that the news published by local newspapers on local companies, on average, use fewer negative words when compared with news about companies that are not local. This happens for a few reasons. First due to the newspaper's target audience, they are most likely to be local, and therefore are employees of local companies and prefer to read good news about their companies. Second, because of advertising spending by local businesses, which account for a large portion of newspaper revenue, and therefore put the newspaper in a conflict of interest. The news content is not unbiased, it varies according to the interests of the source.

Newspapers not only show events, but also influence the perception of these events in those who read them. People tend to invest disproportionately in companies that are geographically closer to them. This result is associated directly with news more favorable to local companies. Local journalism, therefore, influences local investors.

### 1.2.3 Asymmetry of information between domestic and foreign investors

Dvorak (2005) shows the difference in profits of domestic and foreign investors caused by local news. The study is conducted in Indonesia, and shows that domestic investors have higher profits than foreign investors.

Global brokerage clients have higher profits on long-term operations than clients of local brokerage firms in Indonesia who do better in short-term operations. This suggests that local Indonesian clients have advantages because they have the information in real time, as well as not having the linguistic and cultural difficulty with the information. The clients of global brokers, who usually have more experience, end up choosing better long-term investments.

### 1.2.4 Media coverage and the Investor Recognition Hypothesis

Merton (1987) develops a capital market model considering incomplete information. Investors tend to invest in companies that have more information and require greater returns to compensate for situations of incomplete information.

Engelberg and Parsons (2011) show the behavior of investors exposed to different coverage of the same event by the media. They used the annoucements of the S&P 500 index[3]. Based on the postal codes of investors and the coverage provided by the local media, it was possible to predict the most traded stocks.

Fang and Peress (2009) quantify media coverage of a particular company. LexisNexis is used, which, for each company name, is a set of associated keywords. For example, "IBM" is associated with "IBM" and "International Business Machine". Search for articles related to each company, each article only being selected if its relevance is greater than 90%. The time series values of the coverage for that

[3] Standard & Poor's 500: stock market index based on the market capitalization of the 500 largest companies that hold common shares on the NYSE (New York Stock Exchange) or NASDAQ

particular company will be based on the total of published articles that were selected by the company in a given month. Specificity, media coverage is equal to the weighted sum of articles published about each company in each month, where weights are equal to the newspapers' circulation obtained from the Audit Bureau of Circulations. Stocks with low media coverage have higher returns than those with high media coverage. This can be explained by the hypothesis of investor recognition.

Solomon (2012) shows how media coverage affects price in response to news publications. Positive hedging increases investors' expectations for future returns, thereby raising share prices in the short term and reducing returns in the near future of earnings release.

### 1.2.5 Sentiment and its impact on the stock market

A sentiment analysis could be seen as an application of the classification problem, or, according to Batrinca and Treleaven (2015), the application of natural language processing, computational linguistics and text analytics to identify and extract subjective information in source materials. Other terms are used in the literature as well, such as (sentiment) polarity classification: $(i)$ binary classification task of labelling a document as expressing either an overall positive or an overall negative opinion Pang and Lee (2008) - positive, negative; $(ii)$ a multi-class text categorization Pang and Lee (2005) - positive, negative or neutral.

Liu (2010) defines levels of sentiment analysis: $(i)$ Document level, which classifies an entire document as expressing positive or negative sentiment; $(ii)$ Sentence level; and $(iii)$ Feature/aspect level (or entity/aspect analysis according to Feldman (2013)), which extracts sentiments relative to entities and/or their aspects.

There are many studies showing the relationship of the media and its impact and the stock market. Most of them find stronger results when using negative words to find a sentiment for the stock market. Not surprisingly, Baumeister et al. (2001) find that bad is more powerful than good, so negative information is processed more thoroughly than positive one.

Tetlock (2007) shows this quantitatively using the content of the Wall Street Journal. With an exaggerated media pessimism, it is possible to predict a negative pressure on prices, followed by a reversal to a fundamentalist analysis. High or low levels of pessimism result in a high trading volume. Moreover, negative words from the press can predict low profits from companies, that is, they capture aspects of fundamentalist analysis.

García (2013) shows that the predictability of stock returns using media content is best during times of economic recession. People are more sensitive to the news.

Tetlock, Saar-Tsechansky and Macskassy (2008) show that negative words in financial news can predict poor results for companies. They hypothize that words in the news are not redundant information, but contain fundamentalist aspects of the companies that are hard to capture. It is also shown that the stock market prices incorporate the information contained in the negative words with a delay.

Loughran, Mcdonald and Pragidis (2018) proposes a list of oil-related words to measure the information content of oil stories, and show that oil traders overreact to the content of widely-read

news articles. Phrases like output cut, production cut, shortage, and demand up in lagged news articles are associated with lower oil prices the following trading day.

### 1.2.6 How the market reacts to stale information

There are some interesting results regarding stale information as well. Tetlock (2011) tests whether stock market investors appropriately distinguish between new and old information about firms. Staleness of a news story was defined as its textual similarity to the previous ten stories about the same firm. He finds that individual overreact to stale information about publicly traded firms, leading to temporary movements in firms' stock prices. Individual investors trade more aggressively on news when it is stale, because return reversal is significantly larger in stocks with above-average individual investor trading activity, suggesting that individual investors sometimes fail to distinguish between old information and new information in news. Gropp and Kadareja (2012) propose to measure the effect of unobservable private information on volatility. They estimate the effect of a well-identified shock on the volatility of stock returns of European banks as a function of the quality of public information available about the banks. They find strong evidence that, as publicly available information becomes stale, volatility effects and its persistence increase, as private information of investors becomes more important.

### 1.2.7 Event studies and tests for private information

Fama (1991) in his review of market efficiency defines a few research areas, specifically: (i) event studies, which tries to understand how the ajustment of prices to public news; (ii) tests for private information, which inquires if any investors have private information that is not fully reflected in market prices.

Kaplanski and Levy (2010) study the effect of high-impact news, such as a plane crash, on the stock market. This kind of news affects mood and heightens people's anxiety. Anxious people become more pessimistic about future profitability, so they tend to take less risk. Anxiety affects decision making in investments. Consequently, the negative impact of this on the stock price can be verified a few days after the event.

Hendershott and Schurhoff (2015) show evidence that institutions are informed about the news because institutional trading occurs prior to the news announcement date.

### 1.2.8 Readability

Readability also is an interesting textual measurement. Li (2008) uses the Fog index to find that the annual reports of firms with lower earnings are harder to read. The Fog index is:

$$\text{Fog index} = 0.4 \times [(\text{words per sentence}) + (\text{percent of complex words})] \tag{1.1}$$

But other measurements are possible:

$$\text{Flesch-Kincaid grade level} = 0.39 \left( \frac{\text{total words}}{\text{total sentences}} \right) + 11.8 \left( \frac{\text{total syllables}}{\text{total words}} \right) - 15.59 \tag{1.2}$$

$$\text{Flesch Reading Ease} = 206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right) \tag{1.3}$$

The author also tested with these last two measures and got the same result.

## 1.3 The media in other economic themes

The media is starting to have a role in some major central banks' themes, such as monetary policy and financial stability.

Optimism or pessimism have aggregate effects, in the context of monetary policy. Hubert and Labondance (2017) create a central bank sentiment based on statements of FOMC[4] policymakers and central bank statements that follow monetary policy decision meetings. Their findings suggest that the central bank sentiment affects private interest rate expectations and macroeconomic dynamics. A positive shock to sentiment (i.e. optimism shocks) increase private interest rate expectations at 1-year maturity. Sentiment shocks affect inflation expectations, help predict the next policy decision, and have an effect on inflation and industrial production.

Correa et al. (2017) creates a finantial stability dictionary using on Central Banks' Financial Stability Reports (FSRs), which outputs positives and negatives words in the financial stability context. The dictionary is divided into the topics: banking, valuation, household, real estate, corporate, external and sovereign. Topic-specific FSS indexes are created, as an overall FSS index. The study envolves a panel of 35 countries for the sample period between 2005 and 2015. Their findings suggest that: bank-related indicators (SRISK-to-GDP ratio, bank CDS spreads, credit-to-GDP gap, and debt service ratio for private nonfinancial corporations) are contemporaneously correlated with the bank FSS index, therefore convey information about the health of this sector; the sentiment captured by the FSS index predicts financial cycle indicators related to credit, asset prices, and systemic risk; the sentiment deteriorates just prior to the start of banking crises.

The impact of the media on economic activity has also been studied. Shapiro, Sudhof and Wilson (2017) uses a proprietary predictive sentiment algorithm, and Sharpe, Sinha and Hollrah (2017) creates sentiment indexes based on a dictionary that quantifies the optimism and pessimism of the Greenbook[5] text. They both find that news sentiment indexes predict future economic activity, such as the federal funds rate, consumption, unemployment, GDP growth, inflation, industrial production, and the S&P500.

[4] Federal Open Market Committee. <https://www.federalreserve.gov/monetarypolicy/fomc.htm>.

[5] The Greenbook of the Federal Reserve Board of Governors (called the Greenbook for short) is a book with projections of various economic indicators for the economy of the United States produced by the Federal Reserve Board before each meeting of the Federal Open Market Committee.

# Part II

# Applications in Finance

# 2 Document representations and information measurements in time series

**Abstract**

This chapter offers a mathematical formalization of conventional techniques mentioned in the economic articles, approximating computational and economic literature, so that economic articles can use contemporary textual analysis techniques and alternative measurements of text. This opens several possibilities of measures to support economic theories, such as using consecrated TF-IDF weighting schemes and other similarity measurements and working with dense representations of documents. We explore two dense representations: the approximation of the term-document matrix using the singular value decomposition; and Paragraph Vectors. To test these new representations, we calculate the effect of negative news on stock returns. All approaches lead to better results than traditional formulas used in economic articles.
**Keywords**: Document representation, text analysis, economic textual indexes.

## 2.1 Introduction

Text analysis is the process of mathematically describe texts to extract information from text. In order to find meaningful measurements, we need to know and understand text representations.

The objectives of this article are to offer a mathematical formalization of conventional techniques mentioned in the economic literature, to present contemporary textual analysis techniques, and to show alternative measurements of text.

Text analysis in economic studies is different from text analysis in computational related studies because the problems are substantially different. Articles computationally oriented are concerned with the performance of the method in terms of accuracy and latency. On the other hand, economic articles concerns with the effect of text over economic variables.

The term-document matrix, which has weights associated with the terms (rows) that appear in each document (columns), represents text information in a collection of documents. Computer-related articles debate whether a weighting scheme in the term-document matrix retrieves documents correctly from queries, or how to create efficient algorithms to improve the response time. Usually, texts are the documents themselves.

For articles related to economics, it is best to manipulate the transposed term-document matrix, where documents are rows and terms are columns, because the primary use of the data is regressions. In an economic point of view, computational response time is not a problem if the information we use could support an economic hypothesis. We should choose the weighting scheme in the term-document matrix so that regressions provide coefficients that support an economic hypothesis and, preferably,

are statistically significant. And since we sample data by a time period, the text can be any set of documents that affect the studied variable. This means that economic studies usually need to aggregate documents so that their sampling matches the regression data, and that manipulations must be done to guarantee stationarity.

Loughran and McDonald (2016) provide an interesting survey on the literature in accounting and finance, but their main focus is presenting techniques and its applications of articles in the field. Our concern is to approximate the research in accounting and finance with the computer science literature, so we define sentiment extraction as an information retrieval problem. This unifies methods used throughout economic articles, that seem different but are mostly the same, and opens new possibilities for the application of other techniques. There is an enormous gap between what economic articles implement and its formalization. A mathematical formalization helps to understand what the researcher indeed does, considerably reducing any misinterpretation due to a textual description, and also helps to implement computer programs. So we formalize mathematically conventional techniques mentioned in the economic literature in a way they would appear in computer science articles. Also, the research in information representation advanced considerably. State-of-the-art methods consider words as vectors, called word embeddings, and can to provide syntactic and semantic information. And economic studies could explore these properties.

Articles related to economics usually follow a pattern to use a word list (such as a category in a dictionary) to extract information from texts. The information retrieval technique and the document representation affect the quality of the retrieval significantly. In fact, the appropriate choice of term weighting is as important as, and perhaps more important than, a complete and accurate compilation of the word list. These were the findings of Jegadeesh and Wu (2013). This is why the results of regressions of Loughran and McDonald (2011) were similar for both Harvard and Finance specific dictionary with the appropriate TF-IDF scheme. The TF-IDF weighting scheme provides different document representations for the vector space model. Many other possible representations may lead to better information extraction to a specific domain.

The economic literature uses a bag-of-words approach to measure document sentiment, which means that the position of words is irrelevant. State-of-the-art methods in machine learning represent documents as a list of word embeddings and consider the position of words, e.g. hierarchical recurrent neural networks, which decomposes documents into a list of sentences, and each sentence into a list of words. These algorithms usually require the skills of an experienced machine learning professional. We present a document representation using Paragraph Vectors, which is a dense representation of documents that benefits from the semantic relationship brought by word embeddings and also considers the position of words. We can generate these paragraph vectors, which is another document representation, by simply executing of the author's algorithm. It does not require any other skills from economists that they are already being using. So probably they are more adequate to the economic studies.

In the process of understanding word embeddings, we create our own notation to mathematically describe these objects, since it is not very clear in the literature.

Kearney and Liu (2014) present what economic articles are using for text analysis, comparing

and contrasting the various information sources, content analysis methods, and empirical models that have been used to date. This survey focuses on adequating computational methods to text analysis in economic theory. So it brings some information retrieval theory and rewrites textual sentiment in the economy according to that theory, so both approaches are aligned. However, that study does not unify the process of information extraction as an information retrieval problem, it does not include dense representations, it does not formalize the process of document aggregation, and it does not mention similarity measurements; we cover all of these in our article.

Finally, computational methods usually have a database used for checking. Since the database is well known, it is easy to calculate any quality metric, such as accuracy or the $F_1$ score. This is not usually the case for an economic hypothesis. It is important to have a solid economic hypothesis, and then find which method (or document representation) can support that theory.

This article is organized as follows. Section 2.3.1 (Document representations) reviews the most common and useful quantitative representations of a document and Section 2.5 (Notes on document preprocessing) mentions preprocessing techniques. Section 2.3.3 (Methods to extract information from documents) presents common information measurements used in Finance and propose alternatives. Section 2.4 (A new way to look at information measurements in economic articles) motivates the use of documents represented in dense spaces and proposes ways to measure their qualities. Section 2.5 (Notes on document preprocessing) briefly comments on document preprocessing. Section 2.6 (Data) details the data used in our results, which we show in Section 2.7 (Results: differences in negativity extracted from different document representations and stock market returns). Conclusions are in Section 2.8 (Conclusions).

## 2.2 Characterizing the document information extraction

The intent of this section is to characterize the process of extracting information from documents before reviewing in details document representations and its manipulations, so this levels and motivates the reader.

There are many ways to represent texts, also referred to as documents. A document is usually a column vector $\mathbf{d}_j \in \mathbb{R}^{N_f}$, whose dimension $N_f$ represent document features, which can be something meaningful of the real world (such as a word count) or not. Let $\mathbf{M} \in \mathbb{R}^{N_f \times N_D}$ be a feature-document matrix, defined as the concatenation by columns of all documents of a collection of size $N_D$. If $\mathbf{e}_j$ is the $j$-th unit column vector[1], then $\mathbf{d}_j = \mathbf{M}\mathbf{e}_j$. Each document is associated with a time $t$. If there is more than one document per time, aggregation is necessary, i.e. using the sum or the mean, and $\mathbf{d}^t \in \mathbb{R}^{N_f}$ will be the aggregate document in time $t$. The feature-time matrix $\mathbf{M}_t \in \mathbb{R}^{N_f \times T}$ is the documents aggregated by $T$ time periods concatenated by columns.

A query is a list of words that we are interested in evaluating according to our specific problem. It can be considered as a "short document" and is represented as the vector $\mathbf{q} \in \mathbb{R}^{N_f}$.

Let the information extracted from documents be denoted by $s_t$ and the information measure

[1] $\mathbf{e}_j$ is the $j$-th column of an identity matrix of size $N_D$

be $x_t^s$. We define the process of extracting information from documents as:

$$\mathbf{d}_j = f_R(d_j),\ \forall j \tag{2.1}$$

$$\mathbf{d}_t = f_A(\mathcal{D}_t) \tag{2.2}$$

$$s_t = f_{ext}(\mathbf{d}^t, \mathbf{d}^{t-1}, \ldots, \mathbf{d}^1, \mathbf{q}) \tag{2.3}$$

$$x_t^s = f_s(\mathbf{d}^t, \mathbf{d}^{t-1}, \ldots, \mathbf{d}^1, s_t) \tag{2.4}$$

where $d_j$ is the document (list of words) $j$ of a collection of documents, $f_R$ is a method that transforms text into a mathematical representation, $\mathcal{D}_t$ is the set of documents in the collection that are associated to time $t$, $f_A$ is a time aggregation method, $f_{ext}$ is a specific strategy to extract information and $f_s$ can be any function to prepare $s_t$ ready for regressions. For example, $f_{ext}$ could be a relative word count of words belonging to a dictionary; or, being more specific, if we want to extract how much negative information documents have, $\mathbf{d}^t$ has TF-IDF weights[2] of words that appeared in $t$ and $\mathbf{q}_{neg}$ has binary weights indicating the presence or absence of words belonging to the negative category of a dictionary, then $s_t = f_{ext}(\mathbf{d}^t, \mathbf{q}_{neg}) = <\mathbf{d}^t, \mathbf{q}_{neg}> / <\mathbf{d}^t, \mathbf{1}_{N_f}>$, which is the basic formula for extracting the negative tone from documents in many finance papers, such as Tetlock, Saar-Tsechansky and Macskassy (2008), Loughran and McDonald (2011), Solomon (2012), García (2013), Agarwal and Zhang (2014), Allee and DeAngelis (2015), or Tsai, Lu and Hung (2016). And $f_s$ can be any function necessary to prepare $s_t$ for the regressions, like guaranteeing stationarity; being more specific, $x_t^s = s_t$ or $x_t^s = (s_t - \bar{s}_t)/\sigma_{s_t}$.

The document information extraction problem is to choose a document representation of matrix $\mathbf{M}$, an aggregation strategy to map the collection of documents to time periods, an extraction function $f_{ext}$ based on information retrieval methods, and an adjustment procedure $f_s$ to prepare $s_t$ for regressions. We discuss these in the next section.

## 2.3 Extracting information from documents

This section focusses on the technique of extracting information from documents. Section 2.3.1 (Document representations) presents document representations, normally a column vector $\mathbf{d}_j \in \mathbb{R}^{N_f}$. The concatenated document vectors form the feature-document matrix $\mathbf{M} \in \mathbb{R}^{N_f \times N_D}$. Table 2.1 specifies what these matrices and their corresponding number of features for each document representation are.

Section 2.3.2 (Aggregation of documents by time) shows how to use the feature-document matrix ($\mathbf{M}$) to generate a feature-time matrix ($\mathbf{M}_t$). Section 2.3.3 (Methods to extract information from documents) presents methods to extract information from the documents, which is an operation that can be done with documents aggregated by time or not. Specially, it shows some functions $f_{ext}$ of

[2] TF-IDF weights means that the elements of the term-document matrix are a function of the Term Frequency (TF) and the Inverse of the Document Frequency (IDF), and ideia that will be discussed in Section 2.3.1.1 (Vector Space Model).

**Table 2.1.** Specification of the notation used in matrices, documents and number of features used in each document representation. Documents represented by the Vector Space Model with the TF-IDF weighting scheme are in Section 2.3.1.1 (Vector Space Model); documents represented by the singular value decomposition of the term-document matrix in Section 2.3.1.2 (Singular value decomposition); documents represented by the lexicon-based model in Section 2.3.1.3 (Lexicon-based model); documents as word embeddings list are in B Word embeddings list; and documents as word embeddings aggregation are in C Word embeddings aggregation; documents represented by paragraph vectors are in D Paragraph vectors; documents grouped by time are in Section 2.3.2 (Aggregation of documents by time).

| **Method** | **Feature-document matrix** $(\mathbf{M})$ | **Document** $(\mathbf{d}_j^t)$ | **Number of features** $(N_f)$ |
|---|---|---|---|
| Vector Space Model or TF-IDF | term-document, $\mathbf{M}_{\text{tfidf}}$ | $\mathbf{d}_{\text{tfidf}}$ | vocabulary size $(N_V)$ |
| TF-IDF-SVD | $k$-component-document, $\mathbf{M}_{\text{svd}(k,p)}$ | $\mathbf{d}_{\text{svd}(k,p)}$ | number of singular values $(k)$ |
| Lexicon-based model | category-document, $\mathbf{M}_a$ | $\mathbf{d}_a$ | number of terms in the category $(\lvert\mathbf{Q}_a\rvert)$ |
| Word embeddings list | list of $N_D$ document-embedding matrices $\mathbf{D}_j$ | $\mathbf{D}$ | $\mathbf{M}_W \times L_{d_j}$ |
| Word embeddings aggregation | embeddings-document, $\mathbf{M}_W$ | $\mathbf{d}_{\text{emb}(N_W)}$ | embedding size $(N_W)$ |
| Paragraph vectors (PV) | paragraph-document, $\mathbf{M}_{PV}$ | $\mathbf{d}_{\text{PV}(N_{PV})}$ | paragraph size $(N_{PV})$ |
| Time vectors | feature-time, $\mathbf{M}_t$ | $\mathbf{d}^t$ | Any above |

eq. (2.3), and also some methods to prepare the extracted information for regressions, which is the ajustment method $f_s$ of eq. (2.4).

### 2.3.1 Document representations

Text analysis started simply by counting words in a text, or its binary version, the presence or absence of each word. The vocabulary is all the words that appear in a collection of documents. In this context, document vectors are frequencies of each word in the vocabulary. But, just as the word frequency is important, the document frequency is as well. Assuming that a collection of documents exists, the document frequency of a word is the number of documents that a word appears. It gives a sense of term informativeness. This is why now each element of the document vector has a weight, associated to a word, that is a function of the word and document frequencies, called TF-IDF (Term Frequency - Inverse Document Frequency) weighting scheme. The document vectors concatenated in a matrix format is called the term-document matrix. We explain this idea in Section 2.3.1.1 (Vector Space Model).

When we map each word directly into an element of the document vector, sparsity is inevitable.

So when retrieving information from documents, some documents might not have any similarity to others if they do not share the same words. There are two ways to deal with this problem. The first is to use the approximated version of the term-document matrix, that is, make the singular value decomposition of the term-document matrix and work with the eigenvectors associated with the $s$ highest eigenvalues. This technique is also called Latent Semantic Indexing Model, and it maps each document into a lower dimensional space of concepts. We refer to it as TF-IDF-SVD, which we explain in Section 2.3.1.2 (Singular value decomposition). The second is to use groups of words instead of single words. Usually, a group is a category of a well-known dictionary, such as the General Inquirer's Harvard dictionary, and the term-document matrix collapses into these categories to form the category-document matrix. We explain this in Section 2.3.1.3 (Lexicon-based model).

The Vector Space Model and the Latent Semantic Indexing Model are part of the classical information retrieval theory, which Manning, Raghavan and Schütze (2008) and Baeza-Yates and Ribeiro-Neto (2008, section 3) explain in details. The lexicon-based model is used in many Finance articles but is not yet formalized so it cannot be easily reproduced.

A step forward towards text representation is to model each word as a vector, instead of an element of the document vector. Word vectors, also called word embeddings, are learned from texts and made possible to find similarities between words; therefore, a generalization of unseen words comes naturally. There are two ways to represent documents when working with word embeddings: (i) a list of word embeddings; (ii) aggregation of the word embeddings.

The list of word embeddings can be used directly in some machine learning algorithms. Machine learning, according to Goodfellow, Bengio and Courville (2016), is a ability that artificial intelligence (AI) systems have to acquire their own knowledge by extracting patterns from raw data, instead of relying on hard-coded knowledge. Kearney and Liu (2014) make a survey in the literature in Finance, specifying machine learning methods used in recent articles.

The aggregation of the word embeddings can be made by averaging all the words, or a group of words in a context window, in the document.

Instead of going from word embeddings to documents, it is also possible to do the other way around, i.e. to find document representations and get word embeddings as a byproduct. Paragraph vectors can represent any piece of text (sentences, paragraphs, or documents), and two representations are possible, one that considers words order and the bag-of-words. Word embeddings are discussed in A (Word embeddings), word embeddings lists are in B (Word embeddings list), the aggregation of word embeddings to documents is in C (Word embeddings aggregation), and paragraph vectors are in D (Paragraph vectors).

Now we will make definitions that will be used throughout the section:

- Term $w_i$ is a word, or a group of consecutive words, in a document identified by the unique index $i$;
- Vocabulary $\mathcal{V} = \{w_1, \ldots, w_i, \ldots, w_{N_V}\}$ is the set of all distinct terms in all documents, and $I_V = \{1, \ldots, N_V\}$ is the set of all term indexes.

- Document $d_j = [w_{i_1}, \ldots, w_{i_k}, \ldots, w_{i_{L_{d_j}}}]$ is a list of $L_{d_j}$ nonunique consecutive terms, $L_{d_j}$ is the length of the document, $1 \leq k \leq L_{d_j}$ and $i_k \in I_V$. The superscript $I$ in document $d_j$ means that the document is described by its terms' indexes, i.g. $d_j^I = [i_1, \ldots, i_k, \ldots, i_{L_{d_j}}]^T \in \mathbb{N}^{L_{d_j}}$. $\mathcal{V}^{d_j}$ is the vocabulary that appears in document $d_j$. $\mathcal{D} = \{d_1, \ldots, d_j, \ldots, d_{N_D}\}$ is the collection of all documents, where $N_D$ is the number of documents.
- Term frequency $\text{tf}_{i,j}$ is the frequency of ocurrence of term $w_i$ in document $d_j$.
- Document frequency $\text{df}_i$ is the number of documents in the collection that $w_i$ occurs in all the documents. It can be interpreted as an indicator of informativeness. A semantically focussed word will often occur several times in a document if it occurs at all. Semantically unfocussed words are spread out homogeneously over all documents.

Now we will present the models introduced previously.

#### 2.3.1.1 Vector Space Model

A term-document matrix $\mathbf{M}_{\text{tfidf}}$ is a $N_V \times N_D$ matrix that establishes a relation between a term in a document:

$$\begin{array}{c} \\ w_1 \\ w_2 \\ \vdots \\ w_{N_V} \end{array} \begin{array}{c} \begin{array}{cccc} d_1 & d_2 & & d_{N_D} \end{array} \\ \begin{bmatrix} \omega_{1,1} & \omega_{1,2} & \cdots & \omega_{1,N_D} \\ \omega_{2,1} & \omega_{2,2} & \cdots & \omega_{2,N_D} \\ \vdots & \vdots & \cdots & \vdots \\ \omega_{N_V,1} & \omega_{N_V,2} & \cdots & \omega_{N_V,N_D} \end{bmatrix} \end{array} = \mathbf{M}_{\text{tfidf}}$$

where each row is a term and each column is a document. The column vector $\mathbf{d}_j \in \mathbb{R}^{N_V}$, the $j$-th column of the term-document matrix, represents the document $d_j$ in a vector space. Following the characterization of Section 2.2 (Characterizing the document information extraction), $\mathbf{M} = \mathbf{M}_{\text{tfidf}}$ and the document dimension is $N_f = N_V$.

The weight $\omega_{i,j}$ characterizes term importance. Early papers used frequency or binary weights, and many studies show that one is no better than the other.Robertson (2004), motivated by the work of Jones (1972), shows that significant improvements are possible if the document frequency is also used. Nowadays, the weight $\omega_{i,j}$ is based on three parts, one related to the term frequency, the other to the document frequency, and the last a normalization factor:

$$\widetilde{\omega}_{i,j} = \begin{cases} f_{\text{tf}}(\text{tf}_{i,j}) \times f_{\text{idf}}(\text{df}_i) & \text{if } \text{tf}_{i,j} > 0 \\ 0 & \text{if } \text{tf}_{i,j} = 0 \end{cases} \tag{2.5}$$

$$\omega_{i,j} = \frac{\widetilde{\omega}_{i,j}}{\text{norm}_j} \tag{2.6}$$

where $f_{\text{tf}}(\text{tf}_{i,j})$ is the weight associated with the term frequency, $f_{\text{idf}}(\text{df}_i)$ is the weight associated with the document frequency, and $\text{norm}_j$ is a document length normalization factor to compensate undesired effects of long documents. Usually, normalizing documents improves the quality of retrieval, specially when documents have considerably different lengths. Table 2.2 shows some common choices for $f_{\text{tf}}$, $f_{\text{idf}}$ and $\text{norm}_j$. The term frequency (TF) and inverse document frequency (IDF) weighting scheme, called

TF-IDF, is one of the most popular in information retrieval. The mnemonic formed by the 3-tuple *ddd* (where each *d* represents $f_{\mathsf{tf}}$, $f_{\mathsf{idf}}$ and $\mathsf{norm}_j$, respectively) is called SMART notation, used by an early information retrieval system of Salton (1971), and is still used for denoting TF-IDF weighting variants in the vector space model.

**Table 2.2. Variants of TF-IDF weights**. This table shows examples of functions related to the term frequency ($f_{\mathsf{tf}}(\mathsf{tf}_{i,j})$), inverse of the document frequency ($f_{\mathsf{idf}}(\mathsf{df}_i)$), and normalizations. Extracted from Baeza-Yates and Ribeiro-Neto (2008, Tables 3.4 and 3.5, p. 73-74), Manning, Raghavan and Schütze (2008, Figure 6.15, p. 118) and Dumais (1991).

| | **Term frequency** | $f_{\mathsf{tf}}(\mathsf{tf}_{i,j})$ |
|---|---|---|
| b | binary | $\min\{\mathsf{tf}_{i,j}, 1\}$ |
| n | natural (raw frequency) | $\mathsf{tf}_{i,j}$ |
| a | augmented | $0.5 + 0.5\frac{\mathsf{tf}_{i,j}}{\max_{i'} \mathsf{tf}_{i',j}}$ |
| l | logarithm | $1 + \log_2(\mathsf{tf}_{i,j})$ |
| L | log average | $\frac{1 + \log_2(\mathsf{tf}_{i,j})}{1 + \log_2(\mathsf{avg}_{w_{i'} \in d_j} \mathsf{tf}_{i',j})}$ |
| | **Document frequency** | $f_{\mathsf{idf}}(\mathsf{df}_i)$ |
| n | no document frequency | $1$ |
| f | inverse frequency | $\log_2\left(\frac{N_D}{\mathsf{df}_i}\right)$ |
| t | inverse frequency | $1 + \log_2\left(\frac{N_D}{\mathsf{df}_i}\right)$ |
| s | inverse frequency smooth | $\log_2\left(1 + \frac{N_D}{\mathsf{df}_i}\right)$ |
| m | inverse frequency max | $\log_2\left(1 + \frac{\max_k \mathsf{df}_k}{\mathsf{df}_i}\right)$ |
| e | entropy | $1 - \sum_j \frac{p_{i,j} \log(p_{i,j})}{\log(N_D)}$ <br> $p_{i,j} = \frac{\mathsf{tf}_{i,j}}{\sum_j \mathsf{tf}_{i,j}}$ |
| p | probabilistic inverse frequency | $\max\left\{\log_2\left(\frac{N_D - \mathsf{df}_i}{\mathsf{df}_i}\right), 0\right\}$ |
| | **Normalization** | $\mathsf{norm}_j$ |
| n | no normalization | $1$ |
| c | cosine | $\sqrt{\sum_i^{N_V} \widetilde{\omega}_{i,j}^2}$ |
| w | word count | $\sum_i^{N_V} \mathsf{tf}_{i,j}$ |

Matrix $\mathbf{M}_{\mathsf{tfidf}}$ is sparse, so computational implementation should always use libraries for sparse

matrices[3]. An efficient way to calculate all weights, given by eq. (2.5), using only the raw frequency information, is to consider:

$$\begin{aligned} \widetilde{\mathbf{M}}_{\text{tfidf}} &= \text{diag}(f_{\text{idf}}(\mathbf{df}))\, f_{\text{tf}}(\mathbf{M}_{\text{tf}}) \\ \mathbf{M}_{\text{tfidf}} &= \widetilde{\mathbf{M}}_{\text{tfidf}}\, \text{diag}(\mathbf{n})^{-1} \end{aligned} \tag{2.7}$$

where $\mathbf{M}_{\text{tf}}$ is the term-document matrix using raw frequencies only, $f_{\text{tf}}$ is the function associated with the term frequencies applied to each element of $\mathbf{M}_{\text{tf}}$, $\mathbf{M}_1 \mathbf{1}_{N_D} = \mathbf{df} \in \mathbb{N}^{N_V}$ is the document frequencies of the terms, diag transforms the vector into a diagonal matrix, $\mathbf{1}_{N_D}$ is a column vector of ones of size $N_D$, $\mathbf{M}_1$ is the matrix $\mathbf{M}_{\text{tf}}$ with binary weights (or *bnn* weights in the SMART notation), $f_{\text{idf}}$ is the function associated with the document frequencies, and $\mathbf{n} \in \mathbb{R}^{N_D}$ is the normalization vector (the $j$-th component of $\mathbf{n}$ is the normalization factor of document $d_j$, or $\text{norm}_j$). For instance, for the word count normalization, $\mathbf{n} = \mathbf{M}_{\text{tf}}^T \mathbf{1}_{N_V}$.

#### 2.3.1.2 Singular value decomposition

Summarizing contents of documents and queries through a set of terms can give poor results, because: $(i)$ many unrelated documents might be a good match; or $(ii)$ relevant documents that are not indexed by any of the query keywords will not be retrieved. One way to look at a document is that it is a narrative that includes concepts, i.e. references to things of the world, and relations among them. Latent semantic indexing makes possible to match documents to a given query based on concept matching instead of term matching.

Furnas et al. (1988) proposed a latent semantic index model that maps each document and query into a dimensional space composed of concepts. It uses singular value decomposition of matrix $\mathbf{M}$. Let us decompose the term-document matrix into three components:

$$\mathbf{M}_{\text{tfidf}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{2.8}$$

where $\mathbf{U}$ and $\mathbf{V}^T$ are matrices of eigenvectors derived from term-term and document-document covariance matrices, and $\mathbf{\Sigma}$ is an $r \times r$ diagonal matrix of singular values where $r = \min(N_D, N_V)$ is the rank of $\mathbf{M}_{\text{tfidf}}$. Now consider that only the $k$ largest singular values of $\mathbf{\Sigma}$ are used along with their corresponding columns in $\mathbf{U}$ and $\mathbf{V}^T$:

$$\mathbf{M}_{\text{tfidf}(k)} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T \tag{2.9}$$

which is the reduced SVD or truncated SVD. Since the effect of small eigenvalues on matrix products is small, it seems plausible that the term-document matrix will not alter substantially after replacing these small eigenvalues by zero or dropping them.

Given that $\mathbf{U}$ and $\mathbf{V}$ are orthogonal, notice that $\mathbf{M}_{\text{tfidf}} \approx \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T \Rightarrow \mathbf{\Sigma}_k^{-1} \mathbf{U}_k^T \mathbf{M}_{\text{tfidf}(k)} \approx \mathbf{V}_k^T$, then the columns of $\mathbf{V}_k^T$ approximate $\mathbf{d}_j$ in the vector space that has $\mathbf{U}_k \mathbf{\Sigma}_k^{-1}$ as its basis set. Therefore, document $\hat{\mathbf{d}}_j = \mathbf{V}_k^T \mathbf{e}_j$ will the the document in this reduced space (the document dimension is $N_f = k$), and the $k$-component-document matrix will be the $k \times N_D$ matrix:

$$\mathbf{M}_{\text{svd}(k)} = \mathbf{V}_k^T \tag{2.10}$$

[3] In Python, the library SciPy manipulates sparse matrices using the module `scipy.sparse`. Also, it is possible to generate dummy variables using the sparse structure using the method `get_dummies` from Pandas.

The work of Caron (2001) shows that there are other basis for the $span(\mathbf{M}_{\text{tfidf}})$, which can improve the quality of retrieval. We can factor the matrix with the eigenvalues as $\mathbf{\Sigma}_k = \mathbf{\Sigma}_k^{-p/2}\mathbf{\Sigma}_k^{1+p/2}$. Since $\mathbf{M}_{\text{tfidf}} \approx \mathbf{U}_k\mathbf{\Sigma}_k^{-p/2}\mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T \Rightarrow \mathbf{\Sigma}_k^{p/2}\mathbf{U}_k^T\mathbf{M}_{\text{tfidf}} \approx \mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T$, then the columns of $\mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T$ approximate $\mathbf{d}_j$ in the vector space that has $\mathbf{U}_k\mathbf{\Sigma}_k^{p/2}$ as its basis set. In the same way as before, the $k$-component-document matrix will be the $k \times N_D$ matrix:

$$\mathbf{M}_{\text{svd}(k,p)} = \mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T \tag{2.11}$$

so $\mathbf{d}_{\text{svd}(k,p)} = \mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T\mathbf{e}_j$. The parameter $p$ affects all singular values, so we refer to it as the tuning parameter.

We define TF-IDF-SVD(k,p) as the document representation using the singular value decomposition of the term-document matrix using $k$ components and tuning parameter $p$.

#### 2.3.1.3 Lexicon-based model

The start point of a lexicon-based model is the frequency of the terms that belong to a category of the dictionary. Each category has a list of words, which reflects the category intent. For example, the negative category usually has words inducing negativity. Common dictionaries are: DictioSoftware, Harvard General Inquiry Dictionary[4], Opinion Lexicon[5], OpinionFinder[6], SentiWordNet[7], AFINN[8], NRC[9].

Some authors argue that dictionaries should be context-specific and researchers should always consider the original intent of the dictionary. This is why Loughran and McDonald (2011) created a finance dictionary[10], Correa et al. (2017) a finantial stability dictionary[11] using on Central Banks' Financial Stability Reports (FSRs), and Loughran, Mcdonald and Pragidis (2018) a dictionary related to oil.

Let $a$ be a category in a dictionary (such as negative, positive, etc), $\mathcal{Q}_a$ be the set of terms that belong to category $a$ of the dictionary:

$$\mathcal{Q}_a = \{w_i \in \mathcal{V} \mid w_i \text{ belongs to category } a \text{ of the dictionary}\} \tag{2.12}$$

$\mathbf{Q}_a \in \mathbb{R}^{N_V \times |\mathcal{Q}_a|}$ be a matrix such that each column indicates the presence of one term in the dictionary (the $i$-th element of the column is 1 if $w_i \in \mathcal{Q}_a$ and 0 otherwise). The category-document matrix $\mathbf{M}_a \in \mathbb{R}^{|\mathcal{Q}_a| \times N_D}$ is the $\mathbf{M}_{\text{tfidf}}$ matrix with only the rows corresponding to the words belonging to the category $a$:

$$\mathbf{M}_a = \mathbf{Q}_a^T\mathbf{M}_{\text{tfidf}} \tag{2.13}$$

Documents using this representation have the size $N_f = |\mathcal{Q}_a|$. It is also relevant to define the single column category vector of $\mathbf{Q}_a$ as $\mathbf{Q}_a\mathbf{1}_{|\mathcal{Q}_a|} = \mathbf{q}_a \in \mathbb{R}^{N_V}$ (the $i$-th element of $\mathbf{q}_a$ is one if $w_i \in \mathcal{Q}_a$

[4] General Inquirer. Available at: <http://www.wjh.harvard.edu/~inquirer/>.
[5] <https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>
[6] <http://mpqa.cs.pitt.edu/opinionfinder/>
[7] <http://sentiwordnet.isti.cnr.it/>
[8] <http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010>
[9] <http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm>
[10] Software Repository for Accounting and Finance. Available at: <https://sraf.nd.edu/textual-analysis/resources/>.
[11] <https://www.federalreserve.gov/econres/notes/ifdp-notes/constructing-a-dictionary-for-financial-stability-20170623.htm>

and zero otherwise), which we use as the query vector that extracts sentiment from documents. We explore the sentiment extraction in Section 2.3.3 (Methods to extract information from documents).

#### 2.3.1.4 Models based on word embeddings

The Vector Space Model and Singular value decomposition represent documents as bag-of-words, therefore the position of words is irrelevant and independence of words is implied. A single vector $\mathbf{d}_j \in \mathbb{R}^{N_f}$ represents a document. Each word is an atomic unit, represented by a single number. It is not possible to measure similarity between words.

A word embedding is a vector that represents a word. So, there are several features used to capture the essence of the word. This makes possible to check words similarities and also result in better performance for natural language algorithms.

The theory of Hinton, McClelland and Rumelhart (1986) introduced the concept of learning a distributed representation for symbolic data, which stated that "each entity is represented by a pattern of activity distributed over many computing elements, and each computing element is involved in representing many different entities". This abstract idea was implemented using neural networks, Rumelhart, Hinton and Williams (1986) used back-propagation errors can be used to learn representations and found out that the hidden units (which are not part of the input or the output) had interesting features of the task domain, and some regularities in the task could be caputured by these units.

A language model assigns probabilities to sequence of words. The N-gram model predicts the next word given the previous $N-1$ words, that is, estimates the probability function $P$ that $P(w_n|w_1,\ldots,w_{n-1})$. Manning and Schütze (1999, chapter 6) provides the basic theory on language models and N-grams. Bengio et al. (2003) successfully applied this idea of learning distributed representations of words to create a (neural probabilistic) language model, which significantly outperformed N-gram models.

Then several papers proposed ways to find word representations, or word embeddings. They are language models that find distributed representations of words. The most common models are in the Appendix, Section 2.9.2 (Word embedding algorithms).

Next, we will introduce the concept of word embeddings in A (Word embeddings), and explain a few key definitions. Then we will present three other document representations in B (Word embeddings list), C (Word embeddings aggregation) and D (Paragraph vectors).

#### A *Word embeddings*

Let $\mathbf{e}_i \in \mathbb{R}^{N_V}$ be the $i$-th vector of the standard basis associated to the term $w_i$[12], i.e. the $i$-th column of the identity matrix of size $N_V = |\mathcal{V}|$, and $\mathbf{v}_{w_i} \in \mathbb{R}^{N_W}$ its corresponding word embedding, where each coordinate of the $N_W$ dimension embedding represents one feature. An word embedding algorithm tries to find representations in a vector space for each word in the vocabulary so that vectors that are semantically related are closer than ones that are not.

[12] The vector $\mathbf{e}_i$ of the standard basis is also called a one-hot vector in the machine learning literature.

The embedding matrix $\mathbf{E} \in \mathbb{R}^{N_W \times N_V}$ is:

$$\mathbf{E} = \begin{bmatrix} \mathbf{v}_{w_1} & \cdots & \mathbf{v}_{w_{N_V}} \end{bmatrix} \tag{2.14}$$

and $\mathbf{e}_i$ is related to $\mathbf{v}_{w_i}$ by:

$$\mathbf{v}_{w_i} = \mathbf{E}\mathbf{e}_i \tag{2.15}$$

When dealing with word embeddings, we are usually interested in how much one word is related to the other. The relatedness measure uses a similarity function $sim : \mathbb{R}^{N_W} \times \mathbb{R}^{N_W} \to [0, 1]$. The most common similarity function the the cosine function:

$$sim_{cos}(\mathbf{v}_{w_i}, \mathbf{v}_{w_j}) = \cos(\mathbf{v}_{w_i}, \mathbf{v}_{w_j}) = \frac{< \mathbf{v}_{w_i}, \mathbf{v}_{w_j} >}{\left\|\mathbf{v}_{w_i}\right\| \left\|\mathbf{v}_{w_j}\right\|} \tag{2.16}$$

If we want to know if a word $w_i$ is semantically related to $w_j$, we need to order all the words by the similarity with $w_i$, and keep the most $\varphi$ similar ones. If $w_j$ belongs to the set of the most similar words of $w_i$, then we say that $w_i$ is semantically related to $w_j$, or $w_i \sim w_j$. For the notation definition, please refer to the appendix, Section 2.9.1 (Word embedding relatedness notation).

Word embedding vectors are good at word analogies as well, i.e. two pair of words $w_a : w_{a^*}$ and $w_b : w_{b^*}$ (e.g. "man:woman" and "king:queen") that share the relation "$w_a$ is to $w_{a^*}$ as $w_b$ is to $w_{b^*}$. It is hypothesized that $\mathbf{v}_{w_{a^*}} - \mathbf{v}_{w_a}$ is close to $\mathbf{v}_{w_{b^*}} - \mathbf{v}_{w_b}$. So if we let $w_k$ in eq. (2.47) be $w_{a^*} - w_a + w_b$, which means that we would be using the vector $\mathbf{v}_{w_k} = \mathbf{v}_{w_{a^*}} - \mathbf{v}_{w_a} + \mathbf{v}_{w_b}$, then we can find $w_{b^*}$ which is semantically related, that is, $w_{a^*} - w_a + w_b \sim w_{b^*}$. Common inferences in the literature are $Berlin - Germany + Paris \sim France$ or $King - Man + Woman \sim Queen$.

Levy and Goldberg (2014a) propose a different method for word-analogy, one that tries to amplify the differences between small quantities and reduce the differences between larger ones. So if $w_{a^*} - w_a + w_b \sim w_{b^*}$, then:

$$w_{b^*} = \arg \max_{w_k \in \mathcal{V} \setminus \{w_a, w_{a^*}, w_b\}} \frac{\widetilde{\cos}(\mathbf{v}_{w_k}, \mathbf{v}_{w_b})\, \widetilde{\cos}(\mathbf{v}_{w_k}, \mathbf{v}_{w_{a^*}})}{\widetilde{\cos}(\mathbf{v}_{w_k}, \mathbf{v}_{w_a}) + \varepsilon} \tag{2.17}$$

where we use the transformation $\widetilde{\cos}(\mathbf{v}_1, \mathbf{v}_2) = (\cos(\mathbf{v}_1, \mathbf{v}_2) + 1)/2$ for the similarity between two embeddings to guarantee that it will be non-negative, a requirement of eq. (2.17).

### B *Word embeddings list*

If $d_j^I = [i_1, \ldots, i_k, \ldots, i_{L_{d_j}}]^T \in \mathbb{N}^{L_{d_j}}$ is the document of length $L_{d_j}$ described by its terms' indexes, then the document represented as a list of word embeddings will be the matrix $\mathbf{D}_j \in \mathbb{R}^{N_W \times L_{d_j}}$:

$$\mathbf{D}_j = [\mathbf{v}_{w_{i_1}} \mid \cdots \mid \mathbf{v}_{w_{i_k}} \mid \cdots \mathbf{v}_{w_{L_{d_j}}}] \tag{2.18}$$

The list of word embeddings can be the input of several algorithms which taking into account the order of words in the document, such as Recurrent Neural Networks (RNNs), which are one type of neural networks specialized in processing sequential data, i.g., texts. Salehinejad et al. (2018) make a survey on recurrent neural networks. Current state-of-the-art methods include attention mechanisms in RNNs so that more important elements of the sequence have greater weights. Attention is used in many areas, i.g. Bahdanau, Cho and Bengio (2014) use in machine translation, Xu et al. (2015) use

in image caption generation. The review of Cho, Courville and Bengio (2015) didactically explain the topic. Yang et al. (2016) proposed an architectured based on hierarchical attention. i.e. attentions for each level of words and sentences belonging to a document.

A more simple approach would be just averaging over the embeddings, as explained bellow in Item C (Models based on word embeddings).

C *Word embeddings aggregation*

The first strategy to represent documents using word embeddings is by averaging its words embeddings. Let us define the average document vector $\bar{\mathbf{d}}_j \in \mathbb{R}^{N_W}$ as:

$$\bar{\mathbf{d}}_j = \frac{1}{L_{d_j}} \sum_{w_i \in d_j} \mathbf{v}_{w_i} \tag{2.19}$$

This approach could improve classifications results because word embeddings incorporate information on the context of words, but it still does not consider the order of the words. It usually works on small texts, such as the ones from Twitter. The document size will be $N_f = N_W$.

D *Paragraph vectors*

Another way to represent document vectors in a dense space is by using Paragraph Vectors (PV), proposed by Le and Mikolov (2014). A Paragraph Vector represents documents by a dense vector using word embeddings from variable length texts. The algorithm is trained to predict words in the paragraph (sentences or documents). The paragraph vector is concatenated with several word vectors from a paragraph and predicts the following word in the given context. The algorithm learns both word vectors and paragraph vectors, but paragraph vectors are unique among paragraphs, and word vectors are shared.

Before training, it is necessary to choose a few hyperparameters, such as the size of paragraph and word vectors, or the context size (how many words are used to predict the next word). The algorithm has two versions. The first is the Distributed Memory Model of Paragraph Vectors (PV-DM), which takes into consideration the word order, at least in a small context. The second is Distributed Bag of Words version of Paragraph Vector (PV-DBOW), without word ordering.

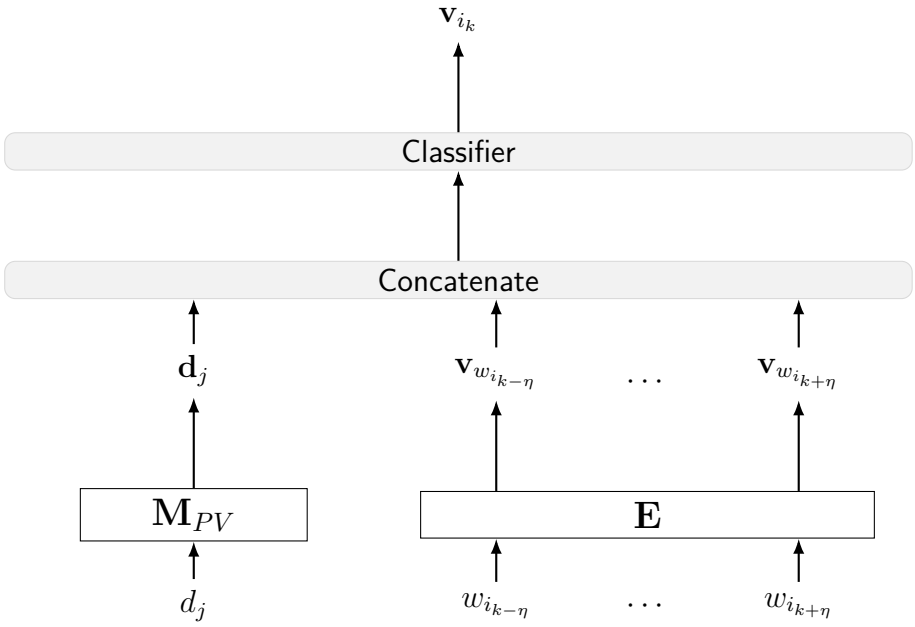
If $w_{i_1}, w_{i_2}, \ldots, w_{i_L}$ is a sequence of training words, the objective of word2vec, a word vector model proposed by Mikolov et al. (2013) that generates words embeddings only, is to maximize the average log probability:

$$\frac{1}{L} \sum_{k=\eta}^{L-\eta} \log p(w_{i_k} | C_\eta(w_{i_k})) \tag{2.20}$$

where $C_\eta(w_{i_k}) = [w_{i_{k-\eta}}, \ldots, w_{i_{k-1}}, w_{i_{k+1}}, \ldots, w_{i_{k+\eta}}]$ is the training context of size $\eta$. The Paragraph Vector algorithm is very similar to this idea, but instead of using word embeddings only, it uses also paragraph vectors. Figure 2.1 shows a schematic representation of the PV-DM algorithm. $\mathbf{M}_{PV}$ is a $N_{PV} \times N_D$ matrix and $\mathbf{E}$ is a $N_W \times N_V$ matrix, where $N_{PV}$ is the size of the paragraph or document, $N_D$ is the total of documents in the collection, $N_W$ is the word embedding size and $N_V$ is the vocabulary size. Each document $d_j$ is concatenated[13] with the word embeddings that belong to the context $C_\eta(w_{i_k})$,

[13] The authors of the Paragraph Vector algorithm propose that paragraph vectors and word embeddings are concatenated or averaged, but they adopt the concatenation method.

**Figure 2.1 – Paragraph vectors**. We show a schematic representation of the Distributed Memory Model of Paragraph Vectors (PV-DM) algorithm.




and then used to predict the word $w_{i_k}$ using a classifier. The paragraph vector can be thought of as another word, but it acts as a memory that remembers what is missing from the current context or the topic of the paragraph.

PV-DBOW ignores the context words in the input and forces the model to predict words randomly sampled from the paragraph in the output.

Documents represented by Paragraph Vectors are $\mathbf{d}_{\text{PV}(N_{PV})} = \mathbf{M}_{PV}\mathbf{e}_j$.

### 2.3.2 Aggregation of documents by time

When dealing with regressions, we collapse documents by day to align with dates samples. This collapse is not strictly necessary but a modeling decision regarding specific problems.

Each document $d_j^t$ is associated with a time period $t$, and there are $1, \ldots, T$ time periods. Let $\mathbf{K} = [\delta_1 \mid \cdots \mid \delta_{N_D}]^T$ be a $N_D \times T$ matrix of concatenated dummy variables, where each $\delta_j$ is a $T$ sized dummy equal to 1 if $d_j^t$ is associated to $t$ and 0 otherwise. The term-time matrix $\mathbf{M}^t_{\text{tfidf}} \in \mathbb{R}^{N_V \times T}$ is the term-document matrix with weights grouped by time. Term frequencies weights could be grouped by the time period using the:

- sum:

$$\mathbf{M}^t_{\text{tf}} = \mathbf{M}_{\text{tf}}\mathbf{K} \tag{2.21}$$

- average[14]:

$$\mathbf{M}^t_{\text{tf}} = \mathbf{M}_{\text{tf}}\mathbf{K}(\mathbf{K}^T\mathbf{K})^{-1} \tag{2.22}$$

[14] Since $\mathbf{K}$ has dummies, $\mathbf{K}^T\mathbf{K}$ will be a diagonal matrix whose elements in the diagonal will correspond to the total of documents in each time $t$.

We can derive document frequencies from: $\mathbf{M}_{\text{tf}}$, the original term-document matrix, if we want to focus on the original word distribution; or $\mathbf{M}_{\text{tf}}^{t}$, the term-time matrix considering term frequencies only, if we want to focus on the new relation of words. And now document frequencies, $\mathbf{df}_t$, dependent on time $t$. So if we include more documents to the collection, e.g. documents associated to $t+1$ for an out-of-sample prediction, we could use $\mathbf{df}_t$ or their new document frequencies based on $\mathbf{df}_{t+1}$. If we update document frequencies, normalizing documents by its norm, or applying standardization, becomes crucial to compensate for the variation of the distribution of words in documents over time. It is not of our knowledge if the literature has explored these issues and their effect on the quality of representations.

Using these term and document frequencies, TF-IDF weights follow eq. (2.5), applying the same functions $f_{\text{tf}}$, $f_{\text{idf}}$ and $\text{norm}_j$ of Table 2.2. And the resulting matrix $\mathbf{M}_t^{\text{tfidf}}$ will be the aggregated matrix considering the Vector Space Model.

For the aggregated version using the singular value decomposition, as described in Section 2.3.1.2 (Singular value decomposition), $\mathbf{M}_t^{\text{tfidf}}$ will be decomposed using eq. (2.8), then $k$ components will be selected, and documents will be given by (2.11).

Representation involving word embeddings, $\mathbf{M}_W$ and $\mathbf{M}_{PV}$, have real features, negative and positive. Therefore should be aggregated using the average, as in eq. (2.22).

### 2.3.3 Methods to extract information from documents

This section focus on methods to extract information from the documents, functions $f_{ext}$, eq. (2.3), and $f_s$, eq. (2.4). Articles in economy usually try to capture information from documents and study the impact in some variables. This information is usually based on a list of words, which does not need to be a category from a pre-defined dictionary, such as negative or uncertain category, but can also be any group of words that we want to study, i.g. words related to corruption.

We explore the tone, which measures the quantity of information that a document has from a word list. First, we review some basics in information retrieval, then we present what current methods in economic papers do.

#### 2.3.3.1 Basic theory in information retrieval

In the information retrieval theory, commonly, documents are compared with a query, which is a vector that contains the information of interest. The TF-IDF weighting scheme can be different for documents and queries. We describe the weighting setting by the mnemonic *ddd* · *qqq*, where each letter follows the SMART notation in Table 2.2. Common TF-IDF weighting schemes are in Table 2.3. For other combinations, please see Salton and Buckley (1988) and Baeza-Yates and Ribeiro-Neto (2008, Table 3.6, p. 74).

The comparison between documents and queries is provided by a scoring function, which is used to rank documents in order of similarity with the query. If $\mathbf{d}_j \in \mathbf{R}^{N_f}$ is a document and $\mathbf{q} \in \mathbf{R}^{N_f}$ is a query, then the degree of similarity is a function $sim : \mathbb{R}^{N_f} \times \mathbb{R}^{N_f} \to \mathbb{R}$. Salton (1989, Table 10.1, p. 318) proposes four measures of similarity:

**Table 2.3. Common TF-IDF weighting schemes**. *ddd · qqq* is the SMART notation for the document and query vectors, according to Table 2.2. The vector norm normalization was omitted for clarity. We extract these suggestions from Baeza-Yates and Ribeiro-Neto (2008, Table 3.6, p. 74), Salton and Buckley (1988) and Dumais (1991).

| Weighting system | Document term weight | Query term weight |
|---|---|---|
| *nfc · afc* | $\mathsf{tf}_{i,j} \log_2 \left(\frac{N_D}{\mathsf{df}_i}\right)$ | $\left(0.5 + 0.5\frac{\mathsf{tf}_{i,q}}{\max_{i'} \mathsf{tf}_{i',q}}\right) \log_2 \left(\frac{N_D}{\mathsf{df}_i}\right)$ |
| *lfc · lfc* | $(1 + \log_2 (\mathsf{tf}_{i,j})) \log_2 \left(\frac{N_D}{\mathsf{df}_i}\right)$ | $(1 + \log_2 (\mathsf{tf}_{i,q})) \log_2 \left(\frac{N_D}{\mathsf{df}_i}\right)$ |
| *lec · lec* | $(1 + \log_2 (\mathsf{tf}_{i,j})) \left(1 - \sum_j \frac{p_{i,j} \log p_{i,j}}{\log N_D}\right)$ | $(1 + \log_2 (\mathsf{tf}_{i,q})) \left(1 - \frac{p_{i,q} \log p_{i,q}}{\log N_D}\right)$ |

1. **Inner product**:
$$sim(\mathbf{d}_j, \mathbf{q}) = <\mathbf{d}_j, \mathbf{q}> \tag{2.23}$$

2. **Dice coefficient**:
$$sim(\mathbf{d}_j, \mathbf{q}) = \frac{2 <\mathbf{d}_j, \mathbf{q}>}{\|\mathbf{d}_j\|_2^2 + \|\mathbf{q}\|_2^2} \tag{2.24}$$

3. **Cosine coefficient**:
$$sim(\mathbf{d}_j, \mathbf{q}) = \frac{<\mathbf{d}_j, \mathbf{q}>}{\|\mathbf{d}_j\|_2 \|\mathbf{q}\|_2} = \left\langle \frac{\mathbf{d}_j}{\|\mathbf{d}_j\|_2}, \frac{\mathbf{q}}{\|\mathbf{q}\|_2} \right\rangle \tag{2.25}$$

4. **Jaccard coefficient**:
$$sim(\mathbf{d}_j, \mathbf{q}) = \frac{<\mathbf{d}_j, \mathbf{q}>}{\|\mathbf{d}_j\|_2^2 + \|\mathbf{q}\|_2^2 - <\mathbf{d}_j, \mathbf{q}>} \tag{2.26}$$

The inner product and the cosine are the most common scoring functions. But since the cosine is the inner product with normalized vectors, and the vector normalization can be incorporated into the weights, we will adopt the inner product as our similarity function.

When using the singular value decomposition of the term-document matrix, Section 2.3.1.2 (Singular value decomposition), we need to map the query vector $\mathbf{q}_{\mathsf{tfidf}} \in \mathbb{R}^{N_V}$ in the new space. The remapped query vector is $\mathbf{q}_{\mathsf{svd}(k,p)} \in \mathbb{R}^k$ and is given by:

$$\mathbf{q}_{\mathsf{svd}(k,p)} = \mathbf{\Sigma}_k^{p/2} \mathbf{U}_k^T \mathbf{q}_{\mathsf{tfidf}} \tag{2.27}$$

where $\mathbf{\Sigma}_k$ and $\mathbf{U}_k$ are given by eq. (2.9).

The similarity between $\mathbf{d}_j = \mathbf{M}_{\mathsf{svd}(k,p)} \mathbf{e}_j$ and $\mathbf{q} = \mathbf{q}_{\mathsf{svd}(k,p)}$ can be done using the cosine similarity:

$$sim(\mathbf{d}_j, \mathbf{q}) = cos(\mathbf{d}_j, \mathbf{q}) = \frac{<\mathbf{d}_j, \mathbf{q}>}{\|\mathbf{d}_j\| \|\mathbf{q}\|} \tag{2.28}$$

#### 2.3.3.2 Impact of terms in the similarity index

The similarity index between a document $\mathbf{d}_j$ and a query $\mathbf{q}$ can be decompose by a $N_f$ sizes vector $\mathbf{sim}(\mathbf{d}_j, \mathbf{q}) \in \mathbb{R}^{N_f}$ so that we can analyse how much each component is contributing the the similarity index. In this case, the similarity index is $sim(\mathbf{d}_j, \mathbf{q}) = <\mathbf{sim}(\mathbf{d}_j, \mathbf{q}), \mathbf{1}_{N_f}>$.

When using the documents represented by the Vector Space Model, Section 2.3.1.1 (Vector Space Model), the $i$-th component of $\mathbf{d}_j$ corresponds to term $w_i$. So $\mathbf{sim}(\mathbf{d}_j, \mathbf{q})$ is:

$$\mathbf{sim}(\mathbf{d}_j, \mathbf{q}) = \text{diag}\,(\mathbf{q}) \left( \frac{\mathbf{d}_j}{\eta_j} \right) \tag{2.29}$$

$$\eta_j(\mathbf{d}_j, \mathbf{q}) = \begin{cases} 1 & \text{if similarity} = \text{inner product} \\ \frac{1}{2}(\left\|\mathbf{d}_j\right\|_2^2 + \left\|\mathbf{q}\right\|_2^2) & \text{if similarity} = \text{dice coefficient} \\ \left\|\mathbf{d}_j\right\|_2 \left\|\mathbf{q}\right\|_2 & \text{if similarity} = \text{cosine coefficient} \\ \left\|\mathbf{d}_j\right\|_2^2 + \left\|\mathbf{q}\right\|_2^2 - <\mathbf{d}_j, \mathbf{q}> & \text{if similarity} = \text{jaccard coefficient} \end{cases} \tag{2.30}$$

and the $i$-th component of $\mathbf{sim}(\mathbf{d}_j, \mathbf{q})$ corresponds directly to the relevance of $w_i$ in $sim(\mathbf{d}_j, \mathbf{q})$. But this impact is not directly measured in dense spaces, so it is necessary to make the necessary ajustments. Suppose $\mathbf{d}_j = \mathbf{d}_{\text{svd}(k,p)} = \mathbf{\Sigma}_k^{1+p/2} \mathbf{V}_k^T \mathbf{e}_j$ is the document coming from the approximation of the term-document matrix using the singular value decomposition, and that $\mathbf{q} = \mathbf{q}_{\text{svd}(k,p)} = \mathbf{\Sigma}_k^{p/2} \mathbf{U}_k^T \mathbf{q}_{\text{tfidf}}$ is the query in the dense space, and $\mathbf{q}_{\text{tfidf}}$ is the query in the TF-IDF weighting scheme. The impact of $w_i$ is the $i$-th term of:

$$\mathbf{sim}(\mathbf{d}_{\text{svd}(k,p)}, \mathbf{q}_{\text{svd}(k,p)}) = \text{diag}\left( \mathbf{U}_k \mathbf{\Sigma}_k^{-p/2} \mathbf{q}_{\text{svd}(k,p)} \right) \left( \frac{\mathbf{U}_k \mathbf{\Sigma}_k^{-p/2} \mathbf{d}_{\text{svd}(k,p)}}{\eta_j(\mathbf{d}_{\text{svd}(k,p)}, \mathbf{q}_{\text{svd}(k,p)})} \right) \tag{2.31}$$

where $\mathbf{U}_k \mathbf{\Sigma}_k^{-p/2} \mathbf{d}_{\text{svd}(k,p)}$ and $\mathbf{U}_k \mathbf{\Sigma}_k^{-p/2} \mathbf{q}_{\text{svd}(k,p)}$ are the document and query approximation in the high-dimensional space, respectively.

#### 2.3.3.3 Current methods

Methods used in economic papers deal with sparsity of the term-document matrix by using categories in a dictionary. Section 2.3.1.3 (Lexicon-based model) defined $\mathbf{q}_a \in \mathbb{N}^{N_V}$ as the binary query vector having ones on the indexes of terms which belong to the category $a$ and zero otherwise. The tone of category $a$ using proportional weights is the similarity index using the inner product, eq. (2.23), using weights *nnw* · *bnn*:

$$s_t^{\text{tone},a} = f_{ext}(\mathbf{d}_j^t, \mathbf{q}_a) = <\mathbf{d}_j^t, \mathbf{q}_a> = \sum_i^{N_V} \left( \frac{\text{tf}_{i,j}}{\sum_{i'}^{N_V} \text{tf}_{i',j}} \right) (\mathbf{q}_a)_i \tag{2.32}$$

where $(\mathbf{q}_a)_i$ is the $i$-th component of $\mathbf{q}_a$. $\mathbf{d}_j^t$ could be a single document or a group of aggregated documents. This defines $f_{ext}$ of eq. (2.3).

There is not a consense in what treatment should be used to prepare $s_t$ for regressions, or $f_s$ in eq. (2.4). This is something that needs to be done to the series according to the application and the data. Tetlock, Saar-Tsechansky and Macskassy (2008) studies the effect of negative information over individual firms' accounting earnings and stock returns. They use the proportion of words in eq. (2.32),

with negative category of the Harvard IV-4 psychosocial dictionary, then create a standardization based on the statistics over the prior calendar year:

$$x_t^{\text{tone},neg} = \frac{s_t^{\text{tone},neg} - \mu_{s_t^{\text{tone},neg}}}{\sigma_{s_t^{\text{tone},neg}}} \tag{2.33}$$

where $\mu_{s_t^{\text{tone},neg}}$ and $\sigma_{s_t^{\text{tone},neg}}$ is the mean and standard deviation of $s_t^{\text{tone},neg}$ over the prior calendar year. The article argues that $s_t^{\text{tone},a}$ might be non-stationary due to changes in coverage or style of news, which could affect the words distribution.

Brown and Tucker (2011) and Loughran and McDonald (2011) explored other TF-IDF weights, instead of weights with the raw term frequency only. The first used weights *nfn* · *nfn*, i.e. $f_{\text{tf}}(\text{tf}_{i,j}) = \text{tf}_{i,j}$ and $f_{\text{idf}}(\text{df}_i) = \log\left(\frac{N_D}{\text{df}_i}\right)$ for $\mathbf{d}_j^t$, and the cosine similarity, eq. (2.25), between documents. Specifically, they find the score of firm's current year MD&A and that for the previous year $s_t = f_s(\mathbf{d}_t, \mathbf{d}_{t-1})$, and then create a measurement based on the difference score $1 - s_t = x_t^s \in [0, 1]$. The second used weights *Lfn* · *bnn*, i.e. $f_{\text{tf}}(\text{tf}_{i,j}) = \frac{1 + \log_2(\text{tf}_{i,j})}{1 + \log_2(\text{avg}_{w_{i'} \in d_j} \text{tf}_{i',j})}$ and $f_{\text{idf}}(\text{df}_i) = \log\left(\frac{N_D}{\text{df}_i}\right)$ for $\mathbf{d}_j^t$, and the inner product similarity, eq. (2.23), between the document and the category query vector.

## 2.4 A new way to look at information measurements in economic articles

An emotion is a psychological state, such as fear or happiness. The sentiment is a mental attitude influenced by emotions, it connects primary emotions with action. The news may cause emotions in many individuals, leading to sentiment towards a specific topic. We want to capture sentiment on an aggregate level and capture its effects over economic variables.

So how do we know that our document representation actually captures the true aggregate effect of the sentiment? Suppose $\Omega$ is a set of all possible representations for documents and queries, e.g. TF-IDF with weights in Table 2.3, or its approximation TF-IDF-SVD(k,p) with $(k, p) \in \mathbb{N} \times \mathbb{R}$, or Paragraph Vectors with any size, and that $r \in \Omega$ is a representation that is able to convert a document $d_j$ and a query $q$, as a list of words, into vectors $\mathbf{d}_j, \mathbf{q} \in \mathbb{R}^{N_f}$. So the sentiment is:

$$s_t = E\left[S_t | d_j, q\right] = \sum_{r \in \Omega} p_r \cdot sim(\mathbf{d}_j, \mathbf{q}) \tag{2.34}$$

where $S_t$ the sentiment treated as a random variable, $p_r$ is the probability that representation $r$ captures the true sentiment, and $sim$ is a similarity function, eqs. (2.23) to (2.26). There are two problems with this approach: the set of all representations is an infinite set and the probabilities are not observable. So researchers usually choose a single representation and hope that it leads to a sentiment close to the true aggregate value, with the least possible bias. And he or she is more likely to succeed if he or she chooses the appropriate document representation.

We motivate the use of dense representations in Section 2.4.1 (Why dense representations?). Then we propose methods to choose dense representations in Section 2.4.2 (Choosing a good document representation in dense spaces) and new measurements in Section 2.4.3 (Proposed measurements).

### 2.4.1 Why dense representations?

Text analysis problems have high dimension because the vocabulary size is usually big, ranging from 100 thousand to 2 million terms. Economic articles usually have much fewer observations than that. So if documents of the term-document matrix were used in regressions, there would be more variables than observations and, therefore, infinite solutions. This is one of the reasons why sparsity is bad for regressions.

The first approach to fight sparsity is to use dictionaries in the term-document matrix, which extracts information related to a category in a dictionary, such as negative or positive, and collapses the whole vocabulary into a single variable. Changing the matrix weights from raw frequencies only to functions of the term and document frequencies, and some normalization as well, may lead to more accurate and statistically significant regressions. The information extraction power, and consequently the success of these regressions, depend on the quality of the selected words of the dictionary and the TF-IDF weighting scheme. Words that do not belong to the dictionary have no effect on the information being analyzed.

Another approach to fight sparsity it to use an aggressive feature selection. Usually, the reduced term-document matrix feeds a classifier, which then outputs the information used in regressions. The vocabulary reduction usually works well and improves results, but overly aggressive feature selection can degrade classification performance.

In both of these attempts, the document key concepts or its main context is not taken into consideration. Only the exact terms used in the pruned vocabulary have an effect over the information extraction.

Representing documents in dense spaces could benefit the quality of the information being extracted in many ways. First, it avoids sparsity and reduces the document dimension. Second, terms that are alike tend to share some common characteristics, and this alikeness can be captured by a similarity function on the document's features.

We propose to use two dense space representations, the singular value decomposition of the term-document matrix and Paragraph Vectors. We will clarify this abstract idea with the simple example provided in Table 2.4. We want to show that it is possible to capture the effect of a certain group of words, even if they are not present in the document. Suppose we want to extract positive information on our collection of documents, and that the positive category is $\mathcal{Q}_{\text{positive}} = \{\text{good}\}$. The collection of documents is given by Panel A, which has only three documents: (i) one associating technology to the positive word *good*; (ii) one associating a company to technology; and (iii) another that is about the same company, but with some unuseful information. The term-document matrix with weights *Lfn* is in Panel B. Its corresponding approximation using the singular value decomposition, eq. (2.9), with $k = 3$ components is in Panels C. The binary query vector is $\mathbf{q}_{\text{positive}} = (0, 1, 0, 0, 0, 0)^T$, which sets to one the index associated to the term *good*, and zero otherwise. If we used any traditional similarity measure, such as the one proposed by Loughran and McDonald (2011), documents with weights *Lfn* · *bnn* and the inner product similarity, eq. (2.23), then the only document that could caputure the similarity with the positive category would be document 1, because the word *good* is present in its text. Documents 2

**Table 2.4. Example of the representation of documents in dense space.** Panel A defines the documents in the collection. Panel B shows the term-document matrix with weights *lfn*. Panel C show the singular value decomposition of the term-document matrix, eq. (2.9), using $k = 3$ components and $p = 1$. Panel D shows the similarities with the positive category composed by term *good*. The query vector is $\mathbf{q}_{\text{tfidf}} = (0, 1.5849, 0, 0, 0, 0)^T$, which is $\mathbf{q}_{\text{positive}} = (0, 1, 0, 0, 0, 0)^T$ with weights *lfn*. The similarity function $s_{\text{sparse}}$ is the inner product given by eq. (2.23), and $\mathbf{d}_{\text{tfidf}} = \mathbf{M}_{\text{tfidf}}\,\mathbf{e}_j$ (the $j$ index was dropped for notation clarity). The relevance of term $w_i$ is the $i$-th element of the vector of eq. (2.29). The similarity function $s_{\text{dense}}$ is given by eq. (2.28), $\mathbf{d}_{\text{svd}(k,p)} = \mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T\,\mathbf{e}_j$, $\mathbf{q}_{\text{svd}(k,p)} = \mathbf{\Sigma}_k^{p/2}\mathbf{U}_k^T\mathbf{q}_{\text{tfidf}}$, and $p = 1$. The relevance of term $w_i$ is the $i$-th element of the vector of eq. (2.31). Only terms with strictly positive values are considered relevant.

**Panel A: Documents**

| Id | Text |
|---|---|
| 1 | technology good |
| 2 | company1 releases technology |
| 3 | useless information company1 |

**Panel B: Term-document matrix**

$$\mathbf{M}_{\text{tfidf}} = \begin{bmatrix} 0 & 0.5850 & 0.5850 \\ 1.5850 & 0 & 0 \\ 0 & 0 & 1.5850 \\ 0 & 1.5850 & 0 \\ 0.5850 & 0.5850 & 0 \\ 0 & 0 & 1.5850 \end{bmatrix} \begin{matrix} \text{company1} \\ \text{good} \\ \text{information} \\ \text{releases} \\ \text{technology} \\ \text{useless} \end{matrix}$$

**Panel C: Singular value decomposition of the term-document matrix**

$$\mathbf{M}_{\text{tfidf}}^{(k=3,p=1)} = \underbrace{\begin{bmatrix} -0.2871 & -0.2176 & 0.1723 \\ -0.0141 & -0.4751 & -0.8159 \\ -0.6724 & 0.1217 & -0.0667 \\ -0.1056 & -0.7111 & 0.5337 \\ -0.0442 & -0.4378 & -0.1042 \\ -0.6724 & 0.1217 & -0.0667 \end{bmatrix}}_{\mathbf{U}_3} \times \underbrace{\text{diag}\left(\begin{bmatrix} 2.3281 \\ 1.8348 \\ 1.6219 \end{bmatrix}\right)}_{\mathbf{\Sigma}_3} \times \underbrace{\begin{bmatrix} -0.0207 & -0.1552 & -0.9877 \\ -0.5499 & -0.8232 & 0.1409 \\ -0.8349 & 0.5461 & -0.0683 \end{bmatrix}}_{\mathbf{V}_3^T} \approx \mathbf{M}_{\text{tfidf}}$$

**Panel D: Similarities with the term *good***

| | $s_{\text{sparse}} = <\mathbf{d}_{\text{tfidf}}, \mathbf{q}_{\text{tfidf}}>$ | | $s_{\text{dense}} = \frac{<\mathbf{d}_{\text{svd}(k,p)}, \mathbf{q}_{\text{svd}(k,p)}>}{\left\|\mathbf{d}_{\text{svd}(k,p)}\right\|\left\|\mathbf{q}_{\text{svd}(k,p)}\right\|}$ | |
|---|---|---|---|---|
| **Document** | **Similarity** | **Relevant terms** | **Similarity** | **Relevant terms** |
| 1 | 1.5849 | good | 0.9931 | good, technology |
| 2 | 0.0000 | | 0.0532 | technology |
| 3 | 0.0000 | | -0.0007 | information, useless |

and 3 would have zero similarity. The only relevant terms that have an impact over the similarity are the ones that belong to the dictionary's category. If the similarity measure were to be calculated using the dense space vector, eq. (2.28), documents 1 and 2 would be strongly associated with the positive

category and other terms would have an impact over the index. If we use eq. (2.31) to measure the impact of the term $w_i$ on $s_{\text{dense}}$, we find that *good* and *technology* have an impact on document 1 and *technology* has an impact on document 2, when we are analysing positiveness using a dictionary; since document 3 is related to something other than technology and has not positive terms, it is weakly related with the positive dictionary. A sense of positiveness was passed from *good* to *technology* in document 1, which was caputure in document 2; this means that the dense representation was able to caputure that *good* and *technology* share some alikeness. The content of Panel D shows that.

Dense representations need to be tuned. It is necessary to find an appropriate dimension and other specific parameters to have a good representation. But what is a good representation?

In documents using the vector space model, a good representation implies that queries are able to retrieve related document correctly, which also implies that it is able to capture the sentiment provided by the category in the dictionary. The TF-IDF weighting scheme has a major impact on the quality of retrieval, i.e. see Salton and Buckley (1988). Loughran and McDonald (2011) used a more appropriate weighting scheme and found that regressions provided by the generic Harvard dictionary and the specific finance dictionary produce similar results. Jegadeesh and Wu (2013) even argues that the weighting scheme could be more important than the dictionary itself. We propose to start with consecrated TF-IDF weights, such as the ones in Table 2.3, for documents and queries.

In dense representations, this could not be any different. For the latent semantic index model, Section 2.3.1.2 (Singular value decomposition), if the TF-IDF weighting scheme and the number of components $k$ of the singular value decomposition are chosen adequately, they provide significant improvements in the quality of retrieval; these are the findings of Dumais (1991). Also, the parameter $p$ in eq. (2.11) needs to be calibrated correctly. For paragraph vectors, D Paragraph vectors in Section 2.3.1.4 (Models based on word embeddings), different document representations are generated by varying the the paragraph size $N_{PV}$, the window size, the algorithm version (PV-DM or PV-DBOW), and the number of negative samples.

In information retrieval articles, we evaluate the quality of retrieval, and consequently the quality of document representation, using the accuracy metric, which compares the documents in the retrieval with the supposedly correct documents that should have been retrieved. But in economic articles, extracting information from text does not have any parameter for comparison. In fact, according to Wiebe et al. (2001), the assessment of sentiment in written text is inevitably subjective and subject to considerable disagreement. So how can we know if a document is a good representation in economic articles? We will address this matter in Section 2.4.2 (Choosing a good document representation in dense spaces) bellow.

Finally, once we are able to find a good document representation, we propose new sentiment extraction methods in Section 2.4.3 (Proposed measurements).

### 2.4.2 Choosing a good document representation in dense spaces

The approximation of the term-document matrix using the singular value decomposition gives the lowest Frobenius error $\left\|\mathbf{M}_{\text{tfidf}} - \mathbf{M}_{\text{tfidf}(k)}\right\|_F$ possible. But it is still necessary to evaluate if the

approximation works with the specific data. One way to do this is first select key terms that are related to the problem under analysis. For example, if the regression data and documents are related to the stock market, then possible key terms would be stock symbols and company names; alternatively, the dictionary terms themselves could be used for this purpose.

Let $\mathcal{Q}^{\text{key}}$ be the set of key terms, $w_i \in \mathcal{Q}^{\text{key}}$ be one term in this set, and $\mathbf{q}_{\text{tfidf}}^{w_i} \in \mathbb{R}^{N_V}$ be a query vector whose $i$-th position is greater than zero and all other are zero. Let us define three quality measurements:

1. **Retrieval quality**

   Let the set of documents in which $w_i$ appears be:

   $$\mathcal{D}^{w_i} = \{d_j \in \mathcal{D} \mid <\mathbf{d}_{\text{tfidf}}, \mathbf{q}_{\text{tfidf}}^{w_i}> > 0, \text{ where } \mathbf{d}_{\text{tfidf}} = \mathbf{M}_{\text{tfidf}}\mathbf{e}_j\} \tag{2.35}$$

   where the represention of $\mathbf{d}_{\text{tfidf}}$ and $\mathbf{q}_{\text{tfidf}}^{w_i}$ is the vector space model. Common document and query weights are given by Table 2.3.

   Let $\widehat{\mathcal{D}}_{\gamma_d}^{w_i}$ be the $|\mathcal{D}^{w_i}| + \gamma_d$ documents with the highest similarity given by eq. (2.28) using a dense representation for documents and queries vectors:

   $$\begin{aligned}\widehat{\mathcal{D}}_{\gamma_d}^{w_i} = \{d_j \in \mathcal{D} \mid \quad & \cos(\mathbf{M}_{\text{dense}}\mathbf{e}_j, \mathbf{q}_{\text{dense}}^{w_i}) \text{ is in the } |\mathcal{D}^{w_i}| + \gamma_d \text{ highest} \\ & \cos(\mathbf{M}_{\text{dense}}\mathbf{e}_{j'}, \mathbf{q}_{\text{dense}}^{w_i}),\ \forall j' \in \{1, \ldots, N_D\}\}\end{aligned} \tag{2.36}$$

   where denses document $\mathbf{d}_{\text{dense}} = \mathbf{M}_{\text{dense}}\mathbf{e}_j$ and query $\mathbf{q}_{\text{dense}}^{w_i}$ are: $\mathbf{d}_{\text{dense}} = \mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T\mathbf{e}_j$ and $\mathbf{q}_{\text{dense}}^{w_i} = \mathbf{\Sigma}_k^{p/2}\mathbf{U}_k^T\mathbf{q}_{\text{tfidf}}^{w_i}$, if the representation is TF-IDF-SVD; and $\mathbf{d}_{\text{dense}} = \mathbf{M}_{\text{PV}}\mathbf{e}_j$ and $\mathbf{q}_{\text{dense}}^{w_i} = \mathbf{f}_{\text{PV}}([w_i])$, if the representation is PV. The parameter $\gamma_d \geq 0$ relaxes the restriction of imposing a fully correct mapping between sparse and dense spaces, which could be handy if we want to use concepts instead of actual terms.

   The retrieval quality is:

   $$R_{\text{retrieval}}^{w_i} = \frac{|\widehat{\mathcal{D}}_{\gamma_d}^{w_i} \cap \mathcal{D}^{w_i}|}{|\mathcal{D}^{w_i}|} \tag{2.37}$$

   Good quality means that the number of documents that term $w_i$ appears is equal, or approximately equal, to the number of documents with the highest similarity in the dense space, i.g. $\mathcal{D}^{w_i} \cap \widehat{\mathcal{D}}_{\gamma_d}^{w_i} = \mathcal{D}^{w_i}$, or $R_{\text{retrieval}}^{w_i} = 1$, and $\gamma_d$ be the lowest as possible.

2. **Term relevance quality**

   For each document $d_j \in \mathcal{D}^{w_i}$, $w_i$ should be among the most relevant terms in eq. (2.31). If $\gamma_t^j$ is the position of $w_i$ among terms of $d_j$ sorted in decreasing order by relevance, $\gamma_t^j$ should be close to one for all documents. $\gamma_t^j \geq 1$ does not necessarily need to be equal to one for all documents, since we are using concepts instead of actual terms, and it is possible that other terms related to $w_i$ be more relevant in a specific document. Calculating $\gamma_t^j$ for every document $j$ can be very computationally expensive. So we define $\gamma_t$ as the position of $w_i$ among terms of $\sum_{d_j \in \mathcal{D}^{w_i}} \mathbf{sim}(\mathbf{d}_{\text{dense}}, \mathbf{q}_{\text{dense}}^{w_i})$ sorted in decreasing order by relevance. The term relevance quality is:

   $$R_{\text{term}}^{w_i} = \gamma_t \tag{2.38}$$

3. **Term semantics quality**

We want to measure if word embeddings, from models based on word embeddings, are semantically related to its group. For example, if $\mathcal{Q}^{\text{key}}$ is a set of stocks, we hope that every stock is semantically related to at least another stock in the set. $R^{w_i}_{\text{semantic},\varphi}$ indicates if $w_i$ is semantically related to at least one other word $w_j \in \mathcal{Q}^{\text{key}}$, $i \neq j$:

$$R^{w_i}_{\text{semantic},\varphi} = \begin{cases} 1 & \text{if } \exists w_j \in \mathcal{Q}^{\text{key}}, i \neq j, \text{such that } w_i \sim_\varphi w_j \\ 0 & \text{otherwise} \end{cases} \tag{2.39}$$

where $\varphi$ are the number of most similar terms in the vocabulary considered for semantic similarity, as we explain in Section 2.9.1 (Word embedding relatedness notation).

Then, the following quality indexes:

$$\bar{R}^{\text{key}}_{\text{retrieval}} = \frac{1}{|\mathcal{Q}^{\text{key}}|} \sum_{w_i \in \mathcal{Q}^{\text{key}}} R^{w_i}_{\text{retrieval}} \tag{2.40}$$

$$\bar{R}^{\text{key}}_{\text{term}} = \frac{1}{|\mathcal{Q}^{\text{key}}|} \sum_{w_i \in \mathcal{Q}^{\text{key}}} R^{w_i}_{\text{term}} \tag{2.41}$$

$$\bar{R}^{\text{key}}_{\text{semantic},\varphi} = \frac{1}{|\mathcal{Q}^{\text{key}}|} \sum_{w_i \in \mathcal{Q}^{\text{key}}} R^{w_i}_{\text{semantic},\varphi} \tag{2.42}$$

should help choosing a good document representation in dense spaces.

### 2.4.3 Proposed measurements

Since sentiment measurements using in economics are basically an information retrieval problem, initially we propose making improvements by: (i) using consecrated TF-IDF weighting schemes, such as the ones in Table 2.3; (ii) exploring other similarity measurements, instead of only using the inner product, such as the cosine, dice and jaccard, eqs. (2.25) to (2.26).

We also propose measuring sentiment in dense spaces, one based in the tone or the intensity of the words in a pre-defined word list, and other based in the similarity of the documents and that list.

The intensity of the words in a pre-defined word list measures how much information from a category is present in a document. It is similar to what current methods used in in economic articles do because it is a measurement that uses actual terms in the dictionary only. But since we are dealing with dense spaces, it also has the benefit of including some synonyms and related words. The tone or the intensity of the words will be:

$$tone = \left\| \mathbf{d}^a \right\|_p \tag{2.43}$$

where $\mathbf{d}^a$ is a dense space representation of the category vector, and $\left\| \cdot \right\|_p$ is the $p$-norm. If we are using the approximation of the term-document matrix by SVD, and $\mathbf{d}_{\text{tfidf}} \in \mathbb{R}^{N_V}$ is the document using the TF-IDF weighting scheme, $\mathbf{q}_a \in \mathbb{R}^{N_V}$ is the query vector with binary weights, $\mathbf{U}_k$, $\mathbf{\Sigma}_k$, $\mathbf{V}_k$ are matrices of the singular value decomposition of the term-document matrix using $k$ components and refinement parameter $p$, then $\mathbf{d}^a_{\text{tfidf}} = \text{diag}(\mathbf{q}_a)\mathbf{d}_{\text{tfidf}}$ will be the document with only the terms in the category,

$\mathbf{d}^a_{\text{svd}(k,p)} = \mathbf{\Sigma}_k^{p/2}\mathbf{U}_k^T\mathbf{d}^a_{\text{tfidf}}$ will be its representation in the dense space using SVD, and $\mathbf{d}^a = \mathbf{d}^a_{\text{svd}(k,p)}$. If we are using paragraph vectors, $\mathbf{d}^a = \mathbf{f}_{DBOW}(q_a)$.

Alternatively, we could measure not only how intense words belonging to a category appear in the text, but how much these words have an effect on the document. We can achieve that by finding the similarity between the document and query vector using the dense representation. If the dense space refers to the SVD decomposition, then $\hat{\mathbf{d}}_j = \mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T\mathbf{e}_j$ and $\hat{\mathbf{q}} = \mathbf{q}^T\mathbf{U}_k\mathbf{\Sigma}_k^{p/2}$. If the dense space refers to paragraph vectors, then $\hat{\mathbf{d}}_j = \hat{\mathbf{d}}_j^{DBOW}$ and $\hat{\mathbf{q}} = \mathbf{f}_{DBOW}(q)$. The similarity between the document and query is given by the similarity measurements, eqs. (2.23) to (2.26), which indicates how much the document is related to the category.

The advantage of these measurements is that they do not depend on specific words, they capture the essence in the concept space.

## 2.5 Notes on document preprocessing

### 2.5.1 Elimination of Stopwords

Stopwords is a list of terms that do not aggregate any useful information to the document, such as articles and connectives. The elimination usually makes algorithms achieve a better performance.

### 2.5.2 Stemming

Stemming refers to a process that reduces words to their common gramatical root.

Porter (1980) provides the most popular stemming algorithm for the English language. Even though there are adaptations made in Porter's algorithm for several languages, the stemming algorithm proposed by Orengo and Huyck (2001)[15], fits better for the Portuguese language.

Overly aggressive stemming can easily degrade classification performance.

### 2.5.3 Feature Selection or Dimensionality Reduction

Feature selection is a technique to reduce the dimensionality of the vocabulary, reduce the size of the feature space by selecting a subset of all features to represent the documents. Limiting the vocabulary is a common practice when building classifiers, since a smaller feature space can ease computational processing and also reduce overfitting, according to Baeza-Yates and Ribeiro-Neto (2008).

Even though selecting words using dictionaries is rather easy, Li (2011) and Kearney and Liu (2014) mentions the disadvantages of this method. There are a lot of methods to limit the vocabulary, such as the the term frequency, document frequency, mutual information, information gain, and chi-square. Antweiler and Frank (2006) use information gain of each word and create a dictionary based in the ones with the largest information gain to feed the Naive Bayes algorithm.

[15] The implementation of this algorithm in Python is provided by the class `RSLPStemmer` of the NLTK package. The documentation is available at <https://www.nltk.org/_modules/nltk/stem/rslp.html>.

We will make the feature selection using the document frequency, since it is simple and, according to the results of Yang and Pedersen (1997), it has the efficiency equivalent to more sofisticated methods (such as information gain, mutual information and $\chi^2$ statistic) for a reduction factor of approximately 10.

The document frequency $\text{df}_i$ is the number of documents in the collection that the term $w_i$ occurs in all the documents. When using feature selection using the document frequency, we set a threshold $\xi$, and all terms $w_i$ for which $\text{df}_i \geq \xi$ are retained, otherwise they are discarded. We will refer to this type of feature selection as $\text{DF}(P_k)$, where $P_k$ is the percentile used to make the cut and $\xi$ is its value.

## 2.6 Data

This article studies dense representation of documents and regress these representation for testing purposes. There are two sources of data, one that provides text data and the other that provides quantitative data.

The Economatica[16] database provide quantitative data relative to the stock market. The only variables used are: (i) ajusted stock daily return $r_t$; (ii) detrended log volume, $vlm_t$ , which we use the rolling average of the past 60 days of log volume to detrend log of daily volume, the methodology based on Campbell, Grossman and Wang (1993); (iii) stock volatility, $vol_t$, which we fit a GARCH(1,1) model for each stock return, that is, we estimate a model with constant mean $r_t = \mu + \epsilon_t$, and time-varying volatility $\sigma^2_{t+1} = \omega + \alpha_1\epsilon^2_t + \beta_1\sigma^2_t$, where $\sigma^2_t \equiv \text{var}(\epsilon_t)$.

Documents are news from newspapers, each news is a document. News from 1st January 2012 to 2nd July 2018, were downloaded from two online newspapers: $(i)$ *Valor Econômico*[17], sections *Finance*, *Companies* and *Politics*; and $(ii)$ *Folha de S.Paulo*[18], sections *Market*, *Power* and *World*. All news are written in Portuguese.

Tokenization is the process of removing undesired information from text (such as commas, hyphens and periods), standardize terms (i.e. putting all words in lowercase and removing accents), and splitting the text so that is becomes a list of terms. Since we are interested in stock codes and company names to study the quality of document representations, we created a specific tokenization process, described in Section 2.9.3 (Tokenization process). Essentially we have a special list of terms to differentiate letter cases and accents and also find compound words. In this way, we can separate company names from regular words.

We removed the stopwords from the documents. The stopwords removal needs to be done using the lowercased original term in the text. The stopwords used were the ones provided by the NLTK Python package[19], except the stopword "não" (*no* or *not* in English), which we kept because of its importance in any sentence, since it has the capacity to change a sentence meaning. The list

[16] Economatica. Available at: <https://economatica.com/>.
[17] Valor Econômico. Available at: <http://www.valor.com.br/>.
[18] Folha de S.Paulo. Available at: <http://www.folha.uol.com.br/>.
[19] Natural Language Toolkit. Available at <http://www.nltk.org/>.

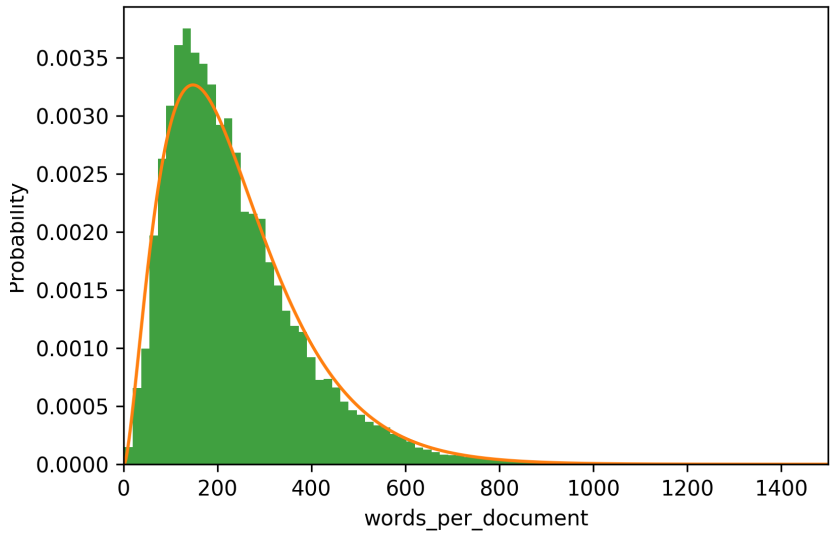
**Table 2.5. Database versions information**. News are from online newspapers *Valor Econômico* and *Folha de S.Paulo*. The tokenization procedure is in Section 2.9.3 (Tokenization process).

| Version | Stemming | #Terms | $\Gamma$ $(\alpha, \beta^{-1})$ | Words per news cut limits | |
|---|---|---|---|---|---|
| | | | | Lower | Upper |
| 1 | No | 350,258 | (2.597, 91.73) | 28 | 520 |
| 2 | Yes | 217,054 | | | |

of stopwords is in Table 2.13, Section 2.9.4 (Stopwords). Then we constructed two tokens databases versions (Table 2.5), with or without stemming.

Even though Porter (1980) provides the the most popular stemming algorithm, we used the stemming algorithm proposed by Orengo and Huyck (2001), because it is more fit to the Portuguese language, since it results in less understemming and overstemming errors than the Portuguese version of Porter's Algorithm. The stemming process needs to happen to the original term in the text, because accents are important. For example, the stemmed term "ganharão" (*will win* in English) is "ganh", but the stemmed term "ganharao" is "ganhara". So, when stemming terms with the same radical, such as "ganharam" (*won* in English), the result is also "ganh". Therefore, the term without accent gives an inaccurate stemmed term. Tokens in the special word list (such as company names and stock codes) were kept without stemming.

**Figure 2.2 – Words per document gamma distributions**. News with stopwords removed, $\Gamma(\alpha = 2.597, \beta^{-1} = 91.73)$.




News were filtered according to a few criteria. First, news that are too short or too long were removed. It was manually verified that short news usually have some video embedded in the web page or are just a simple announcement. Long news usually are interviews with too much information, and that information is usually present over several other shorter news. Since too short news do not bring any information and long news make training much more difficult in any algorithm (parameter matrices are dense and much bigger), they were removed. The words count per news approximately followed a

**Table 2.6. Impact after applying filters on the original sample**. The cut limits of the short and long news removal are shown in Table 2.5. The query $\mathcal{Q}^S$ used to filter the news was composed by the 50 most frequent stocks in the database and their corresponding companies (appendix 2.9.5, Table 2.14).

| Source/Filter | Sample size (#news) | words per news | | |
|---|---|---|---|---|
| | | Mean | Std | Max |
| All news | 402,013 | 239.3 | 194.0 | 52,994 |
| Short and long news removal | 378,384 | 215.3 | 110.6 | 520 |
| Filter using portifolio query | 90,188 | 242.9 | 117.8 | 520 |

gamma distribution $\Gamma(\alpha, \beta)$, see Figure 2.2. We decided to cut the distribution in approximately 1% in the lower tail and 5% in the upper tail. The $\Gamma$ parameters and cut limits are shown in Table 2.5. Second, news were filtered according a query related to the portifolio $S$. The portifolio $S$ is the set of the 50 most frequent stocks in the database (appendix 2.9.5, Table 2.14). The vocabulary $\mathcal{Q}^{s_k}$ is composed by the stock code and company names. For example, $\mathcal{Q}^{\text{PETR4}} = \{\text{PETR4}, \text{Petrobras}\}$. The vocabulary $\mathcal{Q}^S = \cup_{s_k \in S} \mathcal{Q}^{s_k}$ was used in the filter, news that did not have any terms in $\mathcal{Q}^S$ were removed. Table 2.6 shows the result after applying each of these filters.

## 2.7 Results: differences in negativity extracted from different document representations and stock market returns

Economic papers use document representations to investigate a certain economic phenomenon. Thus we will evaluate the quality of document representations by investigating the effect of negative news to stock market returns. Each news, considered as a document, has a publication date. To align the news dates with returns, the date associated with each news is set using a cutoff of the B3 closing time, which is 5 pm in standard time in Brazil and 6 pm during daylight saving time. News published after the closing date are given the following trading date. Pessimism was measured using the category "Negative" from the Finance dictionary provided by Loughran and McDonald (2011). Since the words are in English, they were translated to the Portuguese Language using Google Translator[20]. Each word in English is translated into several words in Portuguese, and each translated word is classified as suggested, common, uncommon or rare. Only the suggested and common translations were kept.

We will try to capture the negative effect of the media in stock returns by regressing:

$$r_t = \alpha_0 + \boldsymbol{\alpha} \cdot \mathcal{L}^5(s_t^{neg}) + \boldsymbol{\beta} \cdot \mathcal{L}^5(r_t) + \boldsymbol{\gamma} \cdot \mathcal{L}^5(vlm_t) + \boldsymbol{\delta} \cdot \mathcal{L}^5(vol_t) + \boldsymbol{\lambda} \cdot dum_{t-1} + \varepsilon_t \tag{2.44}$$

where $\mathcal{L}^n(x_t) = [x_{t-1}, \ldots, x_{t-n}]$, $dum$ are dummy variables (days of the week and january), and $s_t^{neg}$ is the negative sentiment measurement. This regression was exaustly studied by Tetlock (2007). $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_5)$ are the coefficients of $(s_{t-1}^{neg}, \ldots, s_{t-5}^{neg})$. We assume that negative news have a negative impact on stock returns of the following trading day. Tetlock (2007) verified this hypothesis using data from the New York Stock Exchange (NYSE). This implies that $\alpha_1$, the coefficient of $s_{t-1}^{neg}$, should be

[20] <https://translate.google.com>

**Table 2.7. The effect of negative news in different document representations using the vector space**. This tables shows the coefficient $\alpha_1$ of $s_{t-1}^{neg}$, eq. (2.44), with one, two or three asterisks if its corresponding p-value is inferior to 0.10, 0.05 and 0.01, respectively, and the corresponding p-value of the joint hypothesis $H_0 : \alpha_1 = \cdots \alpha_5 = 0$ in parenthesis for different combinations of TF-IDF weights and similarity measurements. The similarity measurements are the inner product, cosine, dice and jaccard, eqs. (2.23) to (2.26). The TF-IDF weights are the ones proposed by Tetlock, Saar-Tsechansky and Macskassy (2008) and Loughran and McDonald (2011) and in Table 2.3. For the TF-IDF weight *nnw* · *bnn*, the standardization of eq. (2.33) was applied after the similarity.

| **TF-IDF weights** | **Similarity measurement** | | | |
|---|---|---|---|---|
| | **Inner product** | **Cosine** | **Dice** | **Jaccard** |
| *nnw* · *bnn* with standardization | 0.000142 | -0.000231 | -0.000340* | -0.000342* |
| after similarity | (0.0525) | (0.7303) | (0.2786) | (0.2725) |
| *Lfn* · *bnn* | 0.000002 | -0.004777 | -0.014444 | -0.027555 |
| | (0.1740) | (0.4497) | (0.4220) | (0.4244) |
| *nfc* · *afc* | -0.015992 | -0.015992 | -0.015992 | -0.030910 |
| | (0.5908) | (0.5908) | (0.5908) | (0.5882) |
| *lfc* · *lfc* | -0.007760 | -0.007760 | -0.007760 | -0.013890 |
| | (0.3791) | (0.3791) | (0.3791) | (0.3711) |
| *lec* · *lec* | 0.000428 | 0.000428 | 0.000428 | 0.002075 |
| | (0.7021) | (0.7021) | (0.7021) | (0.6819) |

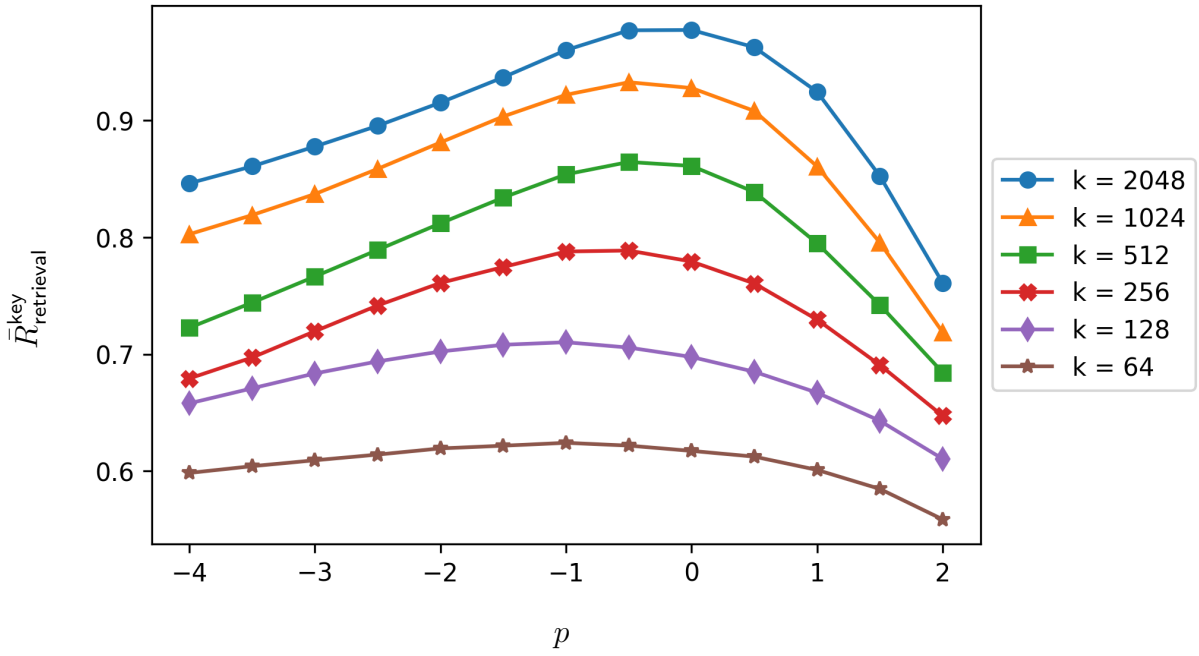
negative because the stock market in $t$ should absorb the increase of negatively charged words in $t-1$, thus resulting in a downward pressure over stock prices.

The representation of documents using the singular value decomposition of the term-document matrix has three main choices: (i) the TF-IDF weights scheme; (ii) the number of components $k$; (iii) the tuning parameter $p$.

First, we study the choice of TF-IDF weights and similarity measurements on the stock market absorption of negative news. The TF-IDF weights are *nnw* · *bnn*, *Lfn* · *bnn* and the ones in Table 2.3. Tetlock, Saar-Tsechansky and Macskassy (2008) introduced the first with the standardization of eq. (2.33) after inner product similarity; the second was explored by Loughran and McDonald (2011) with the inner product similarity; the others are consecrated weights in the Computer Science literature. The similarity measurements are the inner product, cosine, dice and jaccard, eqs. (2.23) to (2.26). Table 2.7 shows the coefficient $\alpha_1$ of $s_{t-1}^{neg}$, eq. (2.44), the corresponding p-value of the hypothesis $H_0 : \alpha_1 = 0$, and the joint hypothesis $H_0 : \alpha_1 = \cdots = \alpha_5 = 0$ for each each combination.

The inner product using vectors with TF-IDF weights *nnw* · *bnn* and *Lfn* · *bnn* cannot capture the effect of negative news on stock returns. This misfortune was probably due to normalization, which was none or insufficient to correct the differences in document lengths. All other similarities, rather than the inner product, provided negative coefficients, supporting this assumption. Since weights *nfc* · *afc* and *lfc* · *lfc* are already normalized by the vector norm, they provide a negative coefficient in any similarity measurement. Weights *lec* · *lec* did not provide negative coefficients in any similarity measure, probably due to the document frequency factor that is not appropriate to capture word informativeness. So the evidence indicates that there are two essential requirements to capture the effect more consistently:

**Figure 2.3 – Choosing a good document representation for the singular value decomposition of the term-document matrix**. We try to evaluate the quality of the singular value decomposition of the term-document matrix for different number of components $k$ and $p$. $\mathcal{Q}^{\text{key}}$ contains stocks and company names; for each $w_i \in \mathcal{Q}^{\text{key}}$, the quality measurements $R^{w_i}_{\text{retrieval}}$ and $R^{w_i}_{\text{term}}$ are given by eq. (2.37) and (2.38), respectively.




normalization involving the document norm (if weights do not normalize documents with their norm, then the similarity should use some normalization rather than the simple inner product); the inverse document frequency factor, $\log_2(N_D/\text{df}_i)$. Henceforth, we use weights *lfc* · *lfn* due to its commonness and support in information retrieval theory.

Using dense representations starts by defining the parameters which can characterize documents satisfactorily. We used the quality indexes defined in Section 2.4.2 (Choosing a good document representation in dense spaces). The key terms $\mathcal{Q}^{\text{key}}$ are the stock codes and company names, used to filter the news, detailed Table 2.14 in Section 2.9.5 (50 most frequent stocks).We calculated $\bar{R}^{\text{key}}_{\text{retrieval}}$, eq. (2.40), for $k \in \{64, 128, 256, 512, 1024\}$ for the news with and without stemming; for each $k$, we tested a few values of the tuning parameter $p$ in the range $[-4, 2]$. Figure 2.3 shows the result for the stemmed news. The best index value, $\max_p \bar{R}^{\text{key}}_{\text{retrieval}}$, and its corresponding argument for each $k$ are in Table 2.8.

Figure 2.3 provides a few interesting conclusions. The default tuning parameter is $p = -2$, meaning that the $\mathbf{\Sigma}_k$ matrix was not dismembered. For every parameter tuple $(k, p = -2)$, where $k$ is the number of components of the singular value decomposition and $p$ is the tuning parameter, seems like that exists another tuple $(k', p')$ such that $p' > p$ and $k' < k$ and equal $\bar{R}^{\text{key}}_{\text{retrieval}}$. For instance, $(k = 2048, p = -2)$ provides approximately the same $\bar{R}^{\text{key}}_{\text{retrieval}}$ as $(k = 1024, p = -0.25)$. Therefore, the tuning parameter $p$ makes possible to find representations with the same quality, but with fewer components. This suggests having computational ease, and probably better regressions. Furthermore, for each quality index calculated for the document representation with $k$ components, exists a global

**Table 2.8. Retrieval quality of the SVD document representation**. $k$ is the number of components used in the singular value decomposition of the term-document matrix, eq. (2.9), and frequency selection refers to the vocabulary reduction, where DF$(P_k)$ means feature selection using the $P_k$ percentile document frequency. Each cell has the best measure $\max_p \left(\bar{R}^{\text{key}}_{\text{retrieval}}\right)$, eq. (2.40), and its argument, the tuning parameter $p$, in parenthesis. The best results for each $k$ are in bold.

| $k$ | Without stemming | | | With stemming | | |
|---|---|---|---|---|---|---|
| | No FS | DF$(50th)$ | DF$(75th)$ | No FS | DF$(50th)$ | DF$(75th)$ |
| 64 | 59.61% ($p$ = -2.00) | 59.62% ($p$ = -2.00) | 59.60% ($p$ = -2.00) | 62.45% ($p$ = -1.00) | 62.42% ($p$ = -1.00) | **62.45%** ($p$ = -0.75) |
| 128 | 70.38% ($p$ = -1.25) | 70.37% ($p$ = -1.25) | 70.50% ($p$ = -1.25) | 71.03% ($p$ = -1.00) | 71.03% ($p$ = -1.00) | **71.06%** ($p$ = -1.00) |
| 256 | 77.70% ($p$ = -0.50) | 77.73% ($p$ = -0.75) | 78.01% ($p$ = -0.75) | 78.92% ($p$ = -0.75) | 78.92% ($p$ = -0.75) | **78.94%** ($p$ = -0.75) |
| 512 | 86.50% ($p$ = -0.25) | 86.54% ($p$ = -0.25) | 86.39% ($p$ = -0.50) | 86.47% ($p$ = -0.50) | 86.45% ($p$ = -0.50) | **86.80%** ($p$ = -0.50) |
| 1024 | 93.12% ($p$ = -0.50) | 93.14% ($p$ = -0.50) | 92.46% ($p$ = -0.50) | 93.25% ($p$ = -0.50) | **93.27%** ($p$ = -0.50) | 92.32% ($p$ = -0.25) |
| 2048 | 97.84% ($p$ = -0.25) | 97.87% ($p$ = -0.25) | 97.49% ($p$ = -0.25) | 97.88% ($p$ = -0.25) | **97.91%** ($p$ = -0.25) | 97.57% ($p$ = -0.25) |

maximum for $\bar{R}^{\text{key}}_{\text{retrieval}}$, which occurs at $p^* = \arg\max_p \left(\bar{R}^{\text{key}}_{\text{retrieval}}\right)$. Table 2.8 shows what these values are, i.e. the best index value, $\max_p \bar{R}^{\text{key}}_{\text{retrieval}}$, and its corresponding argument $p^*$ for each number of components $k$. The best indexes for each $k$ do not vary substantially. The effect of truncating the approximation of the term-document matrix by selecting only $k$ components has the effect to discard zeros or values close to zero, which is leveling documents with and without feature selection, and with and without stemming. Indexes for the documents with stemming provide a slightly better performance. Hence we will use stemming from now on.

Since dense representations of documents work on concepts, instead of words, we study the impact of the term "Gerdau", Brazilian steel company, on documents which the term appears using eq. (2.31) with the cosine similarity. We check what are the 15 most relevant terms for $(k, p) \in \{(64, -1), (256, -0.75), (1024, -0.5)\}$ with TF-IDF weights *lfn* · *lfn*. The results are in Table 2.9. The term relevance quality, eq. (2.38), is $R^{w_i}_{\text{term}} = 5$ for $(k, p) = (64, -1)$ and 1 for $(k, p) \in \{(256, -0.75), (1024, -0.5)\}$, which means that 256 components are enough to correctly identify the term "Gerdau" as the main term, since it was the only term in the query. Panel A shows what these terms are. Interestingly, terms are actually related to Gerdau, and we can identify a few topics: related companies (Usiminas, Companhia Siderúrgica Nacional, Ternium) and products (steel, ore); stock market (IBovespa, GGBR4, USIM5, VALE3); Gerdau's corruption scandal (Carf, Zelotes operation).

We use the semantic quality, eq. (2.42), to study the effect of the embedding size in the

**Table 2.9. Impact of the term "Gerdau" using the dense representations on documents which the term appears.**. The TF-IDF weighting scheme is *lfn* · *lfn*, the singular value decomposition uses $(k, p) \in \{(64, -1), (256, -0.75), (1024, -0.5)\}$, and the relevance of terms is given by eq. (2.31) using the cosine similarity. Panel A shows the 15 most relevant terms, ordered by relevance. Panel B shows a few interpretations for a few terms.

**Panel A: Related terms**

| k | p | Terms |
|---|---|---|
| 64 | -1 | aço, siderúrg, usiminas, companhia_siderurgica_nacional, Gerdau, tonel, ibovesp, ternium, [-]%, comarc, steel, ltd, nippon, lamin, minéri, usin, export, var, açõ, recu |
| 256 | -0.75 | Gerdau, aço, siderúrg, metalúrg, ibovesp, usiminas, carf, johannpet, pn, andré, zelot, açõ, empres, siderurg, sub, jorg, conselh, long, lamin, recu |
| 1024 | -0.5 | Gerdau, metalúrg, aço, siderúrg, johannpet, pn, andré, jorg, carf, zelot, ggbr4, siderurg, metalurgica_gerdau, usim5, sub, vale3, prefer, gaúch, grup, miner |

**Panel B: Interpretation**

| Stemmed terms | Interpretation |
|---|---|
| Gerdau | Gerdau is a steel company |
| metalurgica_gerdau | Metalúrgica Gerdau is a holding company that controls Gerdau |
| [-]% | Negative percentage |
| andré, jorg, johannpet | André Bier Gerdau Johannpeter is the son of Jorge Gerdau Johannpeter |
| aço, steel | Steel |
| açõ | Stocks |
| carf | Conselho Administrativo de Recursos Fiscais (The Administrative Council of Tax Appeals) |
| comarc | judicial district |
| companhia_siderurgica_nacional | Companhia Siderúrgica Nacional, the largest steel industry in Brazil and Latin America |
| conselh | Council |
| empres | Company |
| export | Export |
| ggbr4 | Gerdau SA Preference Shares |
| grup | Group |
| ibovesp | Bovespa Index is the most important indicator of the average performance of the shares traded on B3 |
| lamin | laminate |
| ltd | Ltda |
| metalúrg | metallurgical |
| miner, minéri | ore, mining |
| prefer, pn | Preference stocks |
| siderurg, siderúrg | steel industry, steelworks |
| ternium | Ternium, the largest steel producer in Latin America and a partner of Usiminas |
| tonel | Ton |
| usiminas | Usiminas is a steel company |
| usim5 | Usiminas preference share |
| vale3 | Vale preference share |
| zelot | Operation Zelotes was set off by the Federal Police of Brazil, aiming to investigate a corruption scheme in the Tax Appeals Board (Carf) |

representation of documents using Paragraph Vectors. We check the quality for groups of all stocks and companies (and not just the ones in portifolio $S$ in Table 2.14, appendix 2.9.5), $\mathcal{Q}^{\text{stock}}$ and $\mathcal{Q}^{\text{company}}$, respectively, using embedding sizes $N_W \in \{64, 128, 256, 512, 1024\}$. The generation process of Paragraph Vector used a window of size 10. The embedding size $N_W$ is the paragraph vector size $N_{PV}$, since paragraph vectors are an aggregation of word embeddings. We test different similarity measures, used in eq. (2.47) in preference relations, to check if the result is completly dependent of the measure:

1. Cosine similarity, given by eq. (2.16);
2. Similarity based on the Minkowski distance:

$$\text{dist}^{(p)}_{mink}(\mathbf{v}_{w_i}, \mathbf{v}_{w_j}) = \left( \sum_{k=1}^{N_W} \left| (\mathbf{v}_{w_i})_k - (\mathbf{v}_{w_j})_k \right|^p \right)^{1/p} \tag{2.45}$$

$$sim^{(p)}_{mink}(\mathbf{v}_{w_i}, \mathbf{v}_{w_j}) = e^{-\text{dist}^{(p)}_{mink}(\mathbf{v}_{w_i}, \mathbf{v}_{w_j})/\theta_{mink}} \tag{2.46}$$

where $(\mathbf{v}_{w_i})_k$ is the $k$-th coordenate of the embedding vector $\mathbf{v}_{w_i} \in \mathbb{R}^{N_W}$, $p \geq 1$, and $\theta_{mink} = \max_{w \in \mathcal{V}} \|\mathbf{v}_w\|_p$. We chose the function $sim(x) = e^{-dist(x)/\theta}$, $\theta > 0$, to convert distance to similarity because: $(i)$ $dist(x = 0) \to sim(x) = 1$; $(ii)$ $dist(x = \infty) \to sim(x) = 0$; and $(iii)$ $sim$ is strictly decreasing. $\theta$ is just a normalization factor.

We use $\varphi = 10$ most similar terms for the embeddings similarity, eq. (2.52). Results are in Table 2.10.

Using the Minkowski similarity to verify semantic relationships is inappropriate. The variation of the quality indexes $\bar{R}^{\text{stock}}_{\text{semantic},\varphi}$ and $\bar{R}^{\text{company}}_{\text{semantic},\varphi}$ seems disproportional, that is, $\bar{R}^{\text{stock}}_{\text{semantic},\varphi}$ dropped from 57.76%-69.83% when $N_W = 64$ to 2.59%-4.31% when $N_W = 1024$ while $\bar{R}^{\text{company}}_{\text{semantic},\varphi}$ dropped from 78.24%-82.64% to 31.05%-34.47% on the same embedding sizes. Increasing the embedding size should not drastically diminish the quality index, as happens in $\bar{R}^{\text{stock}}_{\text{semantic},\varphi}$ for $N_W = 1024$. The cosine similarity seems more appropriate for the task. $\bar{R}^{\text{stock}}_{\text{semantic},\varphi}$ achieved its maximum when $N_W = 512$ and $\bar{R}^{\text{company}}_{\text{semantic},\varphi}$ achieved its maximum when $N_W \in \{64, 128\}$.

We show examples of semantic similarity using the cosine and inferences in Table 2.11. Word embeddings are from Paragraph Vectors and have dimension $N_W = 64$. We use the cosine and analogy similarities and analise the most 5 similar terms to each word. The word embeddings can correctly uncover semantic relation from words. For example, $vale3 \sim vale5$ or $bbdc4 \sim bbdc3$, which are stock symbols from the same company, or other similar words are stock symbols as well. All other word similarities also expose semantic relations. The analogy "vale3 is to Vale as petr4 is to *Petrobras*" revels that other words envolved are also related to Petrobras, and the first two words in the analogy "itub3 is to itau_unibanco as bbdc3 is to *Bradesco*" are banks. Therefore, there is evidence that word embeddings from Paragraph Vectors are able to capture semantic relationships between words. If Paragraph Vectors documents are a good summarization from these relationships, then they could be a strong candidate to extract similarities between texts.

Next, we evaluate the dense representation measurements of Section 2.4.3 (Proposed measurements): (a) intensity, eq. (2.43), with $p \in \{1, 2, 3, 10, \infty\}$, which we will refer as $int_p$; (b) similarity, eqs. (2.23) to

**Table 2.10. Semantic quality of embeddings in Paragraph Vectors**. We show embeddings from Paragraph Vectors of sizes $N_W \in \{64, 128, 256, 512, 1024\}$, generated with a window of size 10. The similarity measurements to verify semantics, as in the preference relation of eq. (2.47), are the cosine similarity $sim_{cos}$, eq. (2.16), and the Minkowski similarity, eq. (2.46). We calculate the semantic quality using eq. (2.42) for all stocks and companies (and not just the ones in portifolio $S$ in Table 2.14, appendix 2.9.5), $\mathcal{Q}^{\text{stock}}$ and $\mathcal{Q}^{\text{company}}$, which we refer as $\bar{R}^{\text{stock}}_{\text{semantic},\varphi}$ and $\bar{R}^{\text{company}}_{\text{semantic},\varphi}$, respectively. We use $\varphi = 10$ most similar terms for the embeddings similarity, eq. (2.52).

| $N_W$ | $sim_{cos}$ | $sim^{(1)}_{mink}$ | $sim^{(2)}_{mink}$ | $sim^{(3)}_{mink}$ | $sim^{(10)}_{mink}$ | |
|---|---|---|---|---|---|---|
| 64 | 88.79% | 64.66% | 68.97% | 69.83% | 57.76% | $\leftarrow \bar{R}^{\text{stock}}_{\text{semantic},\varphi}$ |
| | 82.64% | 79.22% | 82.64% | 82.40% | 78.24% | $\leftarrow \bar{R}^{\text{company}}_{\text{semantic},\varphi}$ |
| 128 | 87.07% | 46.55% | 54.31% | 53.45% | 44.83% | |
| | 82.64% | 82.15% | 82.89% | 82.64% | 76.77% | |
| 256 | 86.21% | 25.86% | 27.59% | 25.86% | 26.72% | |
| | 82.40% | 78.73% | 78.48% | 76.53% | 69.19% | |
| 512 | 92.24% | 8.62% | 11.21% | 9.48% | 7.76% | |
| | 77.26% | 59.17% | 59.17% | 58.19% | 54.77% | |
| 1024 | 85.34% | 3.45% | 2.59% | 3.45% | 4.31% | |
| | 65.53% | 34.23% | 34.47% | 32.52% | 31.05% | |

(2.26), which we will refer as $sim_{ip}$, $sim_{cos}$, $sim_{dice}$ and $sim_{jac}$. The text input will be with stemming and DF$(50th)$ feature selection, since this configuration provided a better document representation quality. The TF-IDF-SVD(k,p) representation used $(k, p) \in \{(64, -1), (1024, -0.5)\}$ with TF-IDF weighting schemes *Lfc* · *Lfc* and *lfc* · *lfc*, time aggregation by sum, eq. (2.21), and document frequencies derived from the original term-document matrix, before the SVD transformation. We altered *Lfn* · *bnn*, explored earliear in Table 2.7, to *Lfc* · *Lfc* because results suggest that the cosine normalization is beneficial for the sentiment measurement. The PV representation used vectors with size $N_{PV} = \{64, 1024\}$, and time aggregation by mean, eq. (2.22). The coefficients of stock returns regression, eq. (2.44), were also studied. Results are in Table 2.12.

All coefficients $\alpha_1$, associated to $s^{neg}_{t-1}$, are negative, regardless of document size or measurement. Increasing the document size leads to more coefficients that are individually significant at 10% level. Also, we notice a decrease of the $p$-value of the joint hypothesis when we compare each measurement using 1024 components with the 64 components one. Therefore, a very low truncation of the term-document matrix or representation for paragraph vectors cannot represent well documents, so increasing the document size has a beneficial effect on the sentiment. We obtained similar results for the TF-IDF-SVD using weighting schemes *lsc* · *lsc* and *ltc* · *ltc* but we omitted them in Table 2.12. We base our further comparions with the 1024 document size only.

The intensity measurement, eq. (2.43), provides better results for $0.5 \leq p \leq 3$, that is, it reaches a p-value minimum in the jointly statistical significance and produces more coefficients individually

**Table 2.11. Examples of similarities and inferences using embeddings generated by Paragraph Vectors**. We generated Paragraph Vectors with embeddings of size $N_W = 64$ and window $\eta = 10$. We use the cosine similarity, eq. (2.16), to compare word embeddings and the analogy similarity, eq. (2.17), for analogies. We show the set of the $\varphi = 5$ most similar terms to each term $w_k$, or $\mathcal{W}^{\varphi}_{w_k}$, eq. (2.50). For each stemmed word, we indicate one unstemmed word and its respective translation in parenthesis.

| **Example** | **Term** $(w_k)$ | $\varphi = 5$ **most similar terms** |
|---|---|---|
| Stock | vale3 | petr3, usim5, petr4, vale5, csna3 |
| | bbdc4 | bbdc3, bbas3, ITUB4, usim5, petr4 |
| Economy | invest (investimento - *investment*) | aport (aporte - *contribution*), negóci (negócio - *business*), capt (captação - *captation*), desembols (desembolso - *disbursement*), aloc (alocação - *allocation*) |
| Politics | congress (congresso - *congress*) | sen (senado - *senate*), parl (parlamento - *parliament*), govern (governo - *government*), assemble (assembleia - *assembly*), comiss (comissão - *committee*) |
| Criminal | crim (crime - *crime*) | delit (delito - *offense*), infr (infração - *infringement*), atroc (atrocidade - *atrocity*), estelionat (estelionato - *stelionate*), ilegal (ilegal - *illegal*) |
| | corrupç (corrupção - *corruption*) | fraud (fraude - *fraud*), roubalh (roubalheira - *theft*), malfeit (malfeitos - *evildoing/malefaction*), difamatór (difamatório - *defamatory*), crim (crime - *crime*) |
| Analogy | vale3 is to Vale as petr4 is to ? | PetroRio, usiminas, hrt_participacoes_em_petroleo, Petrobras, ogx_petroleo_e_gas |
| | itub3 is to itau_unibanco as bbdc3 is to ? | banco_fibra, Santander, Bradesco, corre, J.P._Morgan |

significant. The cosine measurement, eq. (2.23), provides better results for the same reasons.

Now we compare TF-IDF-SVD and PV representations, both with sizes 1024. Both representations have similar joint hypothesis but TF-IDF-SVD (PV) has more coefficients individually significant for the intensity (similarity) measurement. A reasonable explanation is that TF-IDF-SVD is more efficient to measure the presence of terms since it is a bag-of-words, and the intensity measurement is just another way to measure the presence of the word list. PV is more efficient in the semantics since the generation process considers the relation of words in a context.

## 2.8 Conclusions

This chapter defined the sentiment analysis problem used in economics in terms of a information retrieval problem, which enabled many well known TF-IDF weighting strategies in computer science. These common weighting schemes and similarity measurement provided better results in the task of finding the effect of negative news on daily stock returns. Also, they provided better dense representation for the approximating of the term-document matrix using the singular value decomposition.

Dense representations worked in sentiment analysis problems, which the negative category of the Finance dictionary proposed by Loughran and McDonald (2011) was translated using Google translator. An inadequate document representation may lead to undesired results, such as negative news having a positive impact on stock returns. But if an adequate document representation is chosen, the results

**Table 2.12. The effect of negative news in different dense representations**. This tables shows the coefficient $\alpha_1$ of $s_{t-1}^{neg}$, eq. (2.44), with one, two or three asterisks if its corresponding p-value is inferior to 0.10, 0.05 and 0.05, respectively, and the corresponding p-value of the joint hypothesis $H_0 : \alpha_1 = \cdots \alpha_5 = 0$ in parenthesis for different dense representations and setiment measurements. The setiment measurements are: (a) $int_p$, eq. (2.43), with $p \in \{1, 2, 3, 10, \infty\}$; (b) $sim_{cos}$, $sim_{dice}$ and $sim_{jac}$, eqs. (2.25) to (2.26). The dense representations are TF-IDF-SVD, using TF-IDF weights *ddd* · *qqq* and approximation of $k$ components and tuning parameter $p$, and PV with size $N_{PV}$.

| | **TF-IDF-SVD** (*ddd* · *qqq* / $(k, p)$) | | | | **PV** ($N_{PV}$) | |
|---|---|---|---|---|---|---|
| | *Lfc* · *Lfc* | *Lfc* · *Lfc* | *lfc* · *lfc* | *lfc* · *lfc* | | |
| **Measurement** | $(64, -1)$ | $(1024, -0.5)$ | $(64, -1)$ | $(1024, -0.5)$ | 64 | 1024 |
| $int_1$ | -0.0024 | -0.0004** | -0.000041 | -0.00000050** | -0.000016 | -0.000002 |
| | (0.6166) | (0.1726) | (0.6732) | (0.1785) | (0.2563) | (0.2074) |
| $int_{0.5}$ | -0.000041 | -0.00000050** | -0.0162 | -0.0108** | -0.00000031* | -0.000000002 |
| | (0.6732) | (0.1785) | (0.5762) | (0.1698) | (0.2229) | (0.2012) |
| $int_2$ | -0.0162 | -0.0108** | -0.0256 | -0.0281** | -0.0001 | -0.0001 |
| | (0.5762) | (0.1698) | (0.5755) | (0.1806) | (0.3234) | (0.2039) |
| $int_3$ | -0.0256 | -0.0281** | -0.0133 | -0.0564 | -0.0001 | -0.0001 |
| | (0.5755) | (0.1806) | (0.6344) | (0.2345) | (0.3805) | (0.1925) |
| $int_{10}$ | -0.0133 | -0.0564 | -0.0095 | -0.0538 | -0.0001 | -0.0004 |
| | (0.6344) | (0.2345) | (0.6557) | (0.2663) | (0.5622) | (0.1737) |
| $int_{inf}$ | -0.0095 | -0.0538 | 0.0005 | -0.0015 | -0.0001 | -0.0004 |
| | (0.6557) | (0.2663) | (0.4703) | (0.1070) | (0.6372) | (0.2177) |
| $sim_{cos}$ | 0.0005 | -0.0015 | 0.1601 | -0.0400 | -0.0004 | -0.0093* |
| | (0.4703) | (0.1070) | (0.4987) | (0.0807) | (0.6420) | (0.1146) |
| $sim_{dice}$ | 0.0004 | -0.0007 | 0.0008 | -0.0005 | -0.0018 | -0.0428* |
| | (0.4410) | (0.1208) | (0.4252) | (0.1098) | (0.8478) | (0.1266) |
| $sim_{jac}$ | 0.0008 | -0.0005 | -0.0004** | -0.0042 | -0.0037 | -0.0849* |
| | (0.4252) | (0.1098) | (0.1726) | (0.4791) | (0.8483) | (0.1315) |

support the economic hypothesis (that negative news have a negative impact on stock returns) in any measurement, differentiating only in the coefficient amplitude and statistical significance.

Dense representation could correctly extract information from news using dictionaries. So, in theory, it is possible to use the whole vector instead of a filtered version from dictionary, and even lagged versions of documents, because it would be computationally viable. This will be explored in Chapter 3 (Do prices absorb public information?) and Chapter 4 (Modeling an analyst).

## 2.9 Appendix

### 2.9.1 Word embedding relatedness notation

Let $\succ_{w_k} \subseteq \mathcal{V}\backslash w_k \times \mathcal{V}\backslash w_k$ be the binary relation:

$$\succ_{w_k} = \left\{(w_i, w_j) \in \mathcal{V}\backslash w_k \times \mathcal{V}\backslash w_k \;\mid\; sim(\mathbf{v}_{w_k}, \mathbf{v}_{w_i}) > sim(\mathbf{v}_{w_k}, \mathbf{v}_{w_j})\right\} \tag{2.47}$$

The binary relation $\succ_{w_k}$ is a weak order[21]. And the upper contour set of $w_i$, denoted by $\succ_{w_k}(w_i)$, and the lower contour set of $w_i$, denoted by $\prec_{w_k}(w_i)$, are:

$$\succ_{w_k}(w_i) = \{w_j \in \mathcal{V}\backslash w_k \mid w_j \succ_{w_k} w_i\} \tag{2.48}$$

$$\prec_{w_k}(w_i) = \{w_j \in \mathcal{V}\backslash w_k \mid w_i \succ_{w_k} w_j\} \tag{2.49}$$

Let $\mathcal{W}^{\varphi}_{w_k}$ be the set of the $\varphi$ most similar terms of $w_k$:

$$\mathcal{W}^{\varphi}_{w_k} = \{\succ_{w_k}(w^*) \mid w^* \in \mathcal{V}\backslash w_k \text{ and } |\succ_{w_k}(w^*)| = \varphi\} \tag{2.50}$$

Notice that $\mathcal{W}^{1}_{w_k}$ is the maximal:

$$\text{MAX}(\mathcal{V}\backslash w_k, \succ_{w_k}) = \{w_i \in \mathcal{V}\backslash w_k \;\mid\; \nexists w_j \in \mathcal{V}\backslash w_k \text{ such that } w_j \succ_{w_k} w_i\} \tag{2.51}$$

Now we can define the binary relation that defines how words are semantically related:

$$\sim_{\varphi} = \left\{(w_i, w_j) \in \mathcal{V} \times \mathcal{V} \;\mid\; w_j \in \mathcal{W}^{\varphi}_{w_i}\right\} \tag{2.52}$$

So $w_i$ is semantically related to $w_j$, then $w_j$ belongs to $\mathcal{W}^{\varphi}_{w_i}$ and we write $w_i \sim_{\varphi} w_j$.

For example, suppose we are analysing the word *love* and that $adore \succ_{love} like$ because $sim(\mathbf{v}_{love}, \mathbf{v}_{adore}) > sim(\mathbf{v}_{love}, \mathbf{v}_{like})$. If $\mathcal{W}^{2}_{love} = \{adore, like\}$, then we can say that *love* is semantically related to *adore* and *like* by $love \sim_2 adore$ and $love \sim_2 like$. But if we considered $\mathcal{W}^{1}_{love}$, then *love* would not be semantically related to *like* ($love \nsim_1 like$).

The parameter $\varphi$ can be dropped from $\sim_{\varphi}$ for notation simplicity when it defined explicitly elsewhere.

### 2.9.2 Word embedding algorithms

The most popular algorithms that can generate word embeddings from text are:

- Singular value decomposition using the shifted positive PMI (Pointwise Mutual Information) matrix, proposed by Levy and Goldberg (2014b).

[21] A binary relation $R$ on $X$ is a weak order if it is complete and transitive. It is complete if $\forall x, y \in X$, $xRy$ or $yRx$. It is transitive if $\forall x, y, z \in X$, $xRy$ or $yRz$ imply $xRz$.

- `word2vec`[22], proposed by Mikolov et al. (2013);
- GloVe[23], proposed by Pennington, Socher and Manning (2014); and
- `fastText`[24], proposed by Joulin et al. (2016).

All these algorithms have some design and hyperparameter choices. Different vectors can be generated if you change the context size or the minimum frequency that a term appears in all the documents. In case of word2vec, the number of negative examples $\kappa$ or the threshold $\nu$ influence as well. In GloVe, there's $\alpha$ or $x_{max}$. Levy, Goldberg and Dagan (2015) discuss this issue and show that these choices impact directly the performance gains of word embeddings. Liu, Liu and Chen (2017) propose a method for fine tuning because their findings suggested that synonyms and antonyms often locate in similar contexts.

### 2.9.3 Tokenization process

The next step was the tokenization process, which normally ignores cases and accents, i.e. puts all text in lowercase and remove accents, and separates words when a blank space or a punctuation is found. Therefore, each term is one single word. This was the standard rule used for all terms.

Since this is very restrictive (company names are mostly compound words), and analysing each news manually is impracticable, we added a few rules in the tokenization step to improve term matching. These rules were implemented using a word list with all terms that needed a special treatment somehow, like compound words. Each term in this list has four attributes and an optional synomym. These attributes are: $(i)$ ignore case; $(ii)$ ignore accents; $(iii)$ apperance condition; and $(iv)$ first word in sentence. With this strategy, it is possible to differentiate a lot of terms and improve term matching.

Attributes $i$ and $ii$ are the mostly used by far. For example, the case-sensitive term "Vale" could correctly identify the company Vale, and ignore matches having the lowercase verb "vale" (*worth*), like "O investimento não vale a pena" (*The investment is not worth it* in English). The term "Banco do Brasil" could identify the company Banco do Brasil, instead of having three terms "banco", "do" and "Brasil"; or the term "Petrobras PN" could correctly identify the stock PETR4, instead of the company Petrobras.

Attribute $iii$ means that a term is only allowed if another term is present at the document; i.g. the term "BB" replaced by the synomym "Banco do Brasil" if "Banco do Brasil" is present in the document, otherwise it would be considered by the original meaning, like the BB credit rating[25].

Attribute $iv$ makes possible to ignore words that are a match if they are the first word in the sentence. Consider the company "Viver Incorporadora" and its case sentitive short version "Viver". If a

[22] The original word2vec code was available at <https://code.google.com/archive/p/word2vec/>, but some download links are broken. The corret download link is <https://storage.googleapis.com/google-code-archive-source/v2/code.google.com/word2vec/source-archive.zip>. Alternatively, it is possible to generate the vectors using the Gensim library for Python, available at <https://radimrehurek.com/gensim/>.

[23] <https://nlp.stanford.edu/projects/glove/>

[24] <https://fasttext.cc/>

[25] 'Parte de um processo normal', diz Meirelles sobre rating da S&P. Valor Econômico. 10th February 2017. Available at <https://www.valor.com.br/financas/4865864/parte-de-um-processo-normal-diz-meirelles-sobre-rating-da-sp>

match of the term "Viver" is found as the first word in the sentence, which does not correspond to "Viver Incorporadora" and the term is not allowed to be the first in a sentence, the term "Viver" would go through the standard tokenization and become the lowercase verb "viver" (*live*). With this rule, sentences like "Viver apenas do hoje"[26] (*Live for today* in English) would correctly consider the term "viver" instead of the company "Viver Incorporadora".

In this context, our tokenization process was:

1. Replace all characters with blank spaces, except for letters, numbers, dots, hyphens and %. If the hyphen was followed by a letter, it was kept; otherwise, it was replaced by a blank space as well.
2. Search the text for special terms, according to the rules detailed previously, and keep the text positions of all special terms found.
3. Split the text, where it is not inside a special word area, using the blank space symbol. Every element of the resulting list is a potential token.
4. Initialize a sentence counter to one.
5. For every potential token:
    a) If the last character is a dot, and the preceding character is not a capital letter (which often happens with initials), remove the dot and increment the sentence counter at the end of this step.
    b) If it does not have any letter, check if it is a percentage. If it is a percentage (the last character is %), the token should be [+]% in case of a positive percentage and [-]% in case of a negative one (the first character is a hyphen). Otherwise, the potential token is discarded.
    c) If it has only letters and dots, keep the token in lowercase and remove accents.
    d) If the potential token does not match any of the previous rules, discard it.

For every token that was not discarded, it was kept: (i) the index in the text; (ii) the sentence which the term belongs; (iii) the original term in the text; (iv) if the token was in the special list; and (v) the token.

### 2.9.4 Stopwords

The stopwords are from the Natural Language Toolkit[27], and a public Python library called `stopwords-pt`[28].

[26] O mercado de capitais já projeta um futuro melhor? Valor Econômico. 9th June 2015. Available at <http://www.valor.com.br/financas/4084648/o-mercado-de-capitais-ja-projeta-um-futuro-melhor>

[27] Natural Language Toolkit. Available at <http://www.nltk.org/>.

[28] Stopwords Portuguese (PT). Available at <https://www.npmjs.com/package/stopwords-pt>.

**Table 2.13. Stopwords**. Terms were merged from two sources: (i) the Natural Language Toolkit; (ii) the Python library `stopwords-pt`. The stopword *não* (*no* in English) was ignored.

| Stopword | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| a | como | dizem | está | houveremos | mês | para | quanto | suas | toda |
| acerca | comprido | dizer | estás | houveria | na | parece | quarta | são | todas |
| adeus | conhecido | do | estávamos | houveriam | nada | parte | quarto | sétima | todo |
| agora | conselho | dois | estão | houvermos | nao | partir | quatro | sétimo | todos |
| ainda | contra | dos | eu | houverá | naquela | paucas | que | só | trabalhar |
| alem | contudo | doze | exemplo | houverão | naquelas | pegar | quem | tal | trabalho |
| algmas | corrente | duas | falta | houveríamos | naquele | pela | quer | talvez | treze |
| algo | cuja | durante | fará | houvesse | naqueles | pelas | quereis | tambem | três |
| algumas | cujas | dá | favor | houvessem | nas | pelo | querem | também | tu |
| alguns | cujo | dão | faz | houvéramos | nem | pelos | queremas | tanta | tua |
| ali | cujos | dúvida | fazeis | houvéssemos | nenhuma | perante | queres | tantas | tuas |
| além | custa | e | fazem | há | nessa | perto | quero | tanto | tudo |
| ambas | cá | ela | fazemos | hão | nessas | pessoas | questão | tarde | tão |
| ambos | da | elas | fazer | iniciar | nesse | pode | quieto | te | tém |
| ano | daquela | ele | fazes | inicio | nesses | podem | quinta | tem | têm |
| anos | daquelas | eles | fazia | ir | nesta | poder | quinto | temos | tínhamos |
| antes | daquele | em | faço | irá | nestas | poderá | quinze | tempo | um |
| ao | daqueles | embora | fez | isso | neste | podia | quáis | tendes | uma |
| aonde | dar | enquanto | fim | ista | nestes | pois | quê | tenha | umas |
| aos | das | entao | final | iste | no | ponto | relação | tenham | uns |
| apenas | de | entre | foi | isto | noite | pontos | sabe | tenhamos | usa |
| apoio | debaixo | então | fomos | já | nome | por | sabem | tenho | usar |
| apontar | dela | era | for | lado | nos | porque | saber | tens | vai |
| apos | delas | eram | fora | lhe | nossa | porquê | se | tentar | vais |
| após | dele | essa | foram | lhes | nossas | portanto | segunda | tentaram | valor |
| aquela | deles | essas | forem | ligado | nosso | posição | segundo | tente | veja |
| aquelas | demais | esse | forma | local | nossos | possivelmente | sei | tentei | vem |
| aquele | dentro | esses | formos | logo | nova | posso | seis | ter | vens |
| aqueles | depois | esta | fosse | longe | novas | possível | seja | terceira | ver |
| aqui | desde | estado | fossem | lugar | nove | pouca | sejam | terceiro | verdade |
| aquilo | desligado | estamos | foste | lá | novo | pouco | sejamos | terei | verdadeiro |
| as | dessa | estar | fostes | maior | novos | poucos | sem | teremos | vez |
| assim | dessas | estará | fui | maioria | num | povo | sempre | teria | vezes |
| através | desse | estas | fôramos | maiorias | numa | primeira | sendo | teriam | viagem |
| atrás | desses | estava | fôssemos | mais | numas | primeiras | ser | terá | vindo |
| até | desta | estavam | geral | mal | nunca | primeiro | serei | terão | vinte |
| aí | destas | este | grande | mas | nuns | primeiros | seremos | teríamos | você |
| baixo | deste | esteja | grandes | me | nível | promeiro | seria | teu | vocês |
| bastante | destes | estejam | grupo | mediante | nós | propios | seriam | teus | vos |
| bem | deve | estejamos | ha | meio | número | proprio | será | teve | vossa |
| boa | devem | estes | haja | menor | o | própria | serão | tinha | vossas |
| boas | deverá | esteve | hajam | menos | obra | próprias | seríamos | tinham | vosso |
| bom | dez | estive | hajamos | meses | obrigada | próprio | sete | tipo | vossos |
| bons | dezanove | estivemos | havemos | mesma | obrigado | próprios | seu | tive | vários |
| breve | dezasseis | estiver | havia | mesmas | oitava | próxima | seus | tivemos | vão |
| cada | dezassete | estivera | hei | mesmo | oitavo | próximas | sexta | tiver | vêm |
| caminho | dezoito | estiveram | hoje | mesmos | oito | próximo | sexto | tivera | vós |
| catorze | dia | estiverem | hora | meu | onde | próximos | sim | tiveram | zero |
| cedo | diante | estivermos | horas | meus | ontem | puderam | sistema | tiverem | à |
| cento | direita | estivesse | houve | mil | onze | pôde | sob | tivermos | às |
| certamente | dispoe | estivessem | houvemos | minha | os | põe | sobre | tivesse | área |
| certeza | dispoem | estiveste | houver | minhas | ou | põem | sois | tivessem | é |
| cima | diversa | estivestes | houvera | momento | outra | quais | somente | tiveste | éramos |
| cinco | diversas | estivéramos | houveram | muito | outras | qual | somos | tivestes | és |
| coisa | diversos | estivéssemos | houverei | muitos | outro | qualquer | sou | tivéramos | último |
| com | diz | estou | houverem | máximo | outros | quando | sua | tivéssemos | |

### 2.9.5 50 most frequent stocks

**Table 2.14. Portifolio $S$, 50 most frequent stocks in the news database and its company.** The number of news consider the documents in which any term in $\mathcal{Q}^{s_k}$ appears, i.e., the stock code or the company name.

| Stock | Freq | Company | #News |
|---|---|---|---|
| PETR4 | 4475 | Petrobras | 38986 |
| VALE5 | 4120 | Vale | 16956 |
| ITUB4 | 2314 | Itaú Unibanco | 10416 |
| BBDC4 | 1984 | Bradesco | 10983 |
| VALE3 | 1330 | Vale | 16956 |
| OGXP3 | 1276 | Ogpar | 1590 |
| PETR3 | 1110 | Petrobras | 38986 |
| USIM5 | 1076 | Usiminas | 3354 |
| BBAS3 | 1074 | Banco Do Brasil | 10746 |
| GGBR4 | 838 | Gerdau | 3248 |
| CSNA3 | 651 | Companhia Siderúrgica Nacional | 3181 |
| ELET3 | 560 | Eletrobras | 5487 |
| FIBR3 | 455 | Fibria | 1917 |
| ELET6 | 442 | Eletrobras | 5487 |
| BBDC3 | 442 | Bradesco | 10983 |
| GOLL4 | 432 | Gol Linhas Aéreas | 3668 |
| JBSS3 | 416 | Jbs | 6213 |
| OIBR4 | 411 | Oi | 5457 |
| CMIG4 | 399 | Cemig | 2736 |
| MRFG3 | 396 | Marfrig | 1333 |
| MMXM3 | 393 | Mmx | 1467 |
| BRAP4 | 352 | Bradespar | 704 |
| EMBR3 | 348 | Embraer | 3095 |
| GFSA3 | 347 | Gafisa | 1070 |
| B3SA3 | 346 | B3 | 6100 |

| Stock | Freq. | Company | #News |
|---|---|---|---|
| PDGR3 | 344 | Pdg Realty | 959 |
| MRVE3 | 342 | Mrv Engenharia | 1072 |
| OIBR3 | 338 | Oi | 5457 |
| SUZB5 | 306 | Suzano | 2014 |
| BRKM5 | 286 | Braskem | 2104 |
| RSID3 | 277 | Rossi Residencial | 630 |
| CESP6 | 270 | Cesp | 1313 |
| BTOW3 | 263 | B2W | 854 |
| BISA3 | 258 | Brookfield Incorporações | 1106 |
| KROT3 | 249 | Kroton | 1408 |
| CPLE6 | 243 | Copel | 1255 |
| USIM3 | 233 | Usiminas | 3354 |
| NATU3 | 206 | Natura | 1294 |
| TIMP3 | 196 | Tim | 3972 |
| CIEL3 | 188 | Cielo | 1209 |
| LIGT3 | 181 | Light | 1282 |
| SBSP3 | 176 | Sabesp | 1618 |
| HGTX3 | 175 | Companhia Hering | 173 |
| PCAR4 | 172 | Pao De Acucar | 2497 |
| CSAN3 | 160 | Cosan | 1126 |
| RENT3 | 158 | Localiza | 572 |
| BRFS3 | 156 | Brf | 1706 |
| SMLS3 | 148 | Smiles | 673 |
| ESTC3 | 148 | Estacio | 322 |
| CCRO3 | 139 | Ccr | 1281 |

# 3 Do prices absorb public information?

**Abstract**

This chapter proposes to create a sentiment measurement based on a learned vocabulary, instead of predefined dictionaries, by using the overnight return to classify news into positive or negative. We learn the vocabulary on small sliding in-sample windows and then evaluate our sentiment with learned vocabulary in out-of-sample regressions. We find that public information do influence returns. Returns in $t$ are mostly influenced by news in $t-1$, but the influence of other past news is not zero. Also, long in-sampling windows are bad for training (the excess of information makes it difficult to learn a vocabulary) and that one year is the ideal size for learning. Additionally, we analyse terms that have the most and the least impact on the sentiment based on the learned vocabulary, which we refer as positive and negative terms, and find that the positivity or negativity is relative to the context.

**Keywords**: stock market, text analysis, learned vocabulary, sentiment

## 3.1 Introduction

News helps a reader to create a perception of reality. So, in some way, it influences readers to form an opinion and take a particular action based on their beliefs of what might happen. Investors from the stock market gather many sources of information, such as public news, and buy or sell stocks according to a decision based on profit maximization and risk. The aggregated effect of the absorption of all sources of information converges to stock prices movements.

In this paper, we propose to learn the vocabulary found in public information that has the most effect on future returns. The vocabulary learning does not depend on any manual classification of news. We use the learned vocabulary to build a sentiment measurement and evaluate its significance on stock market regressions. Also, we show that the positivity or negativity of a word depends on the context.

Text analysis gives us the tools to translate qualitative information (text from the news) into quantitative measures. Batrinca and Treleaven (2015) explain that sentiment analysis is the application of natural language processing, computational linguistics and text analytics to identify and extract subjective information in source materials. There are many methods to find sentiment from text, which are usually divided into unsupervised or supervised models.

Unsupervised methods extract information directly from the text. A common approach is to use word lists in dictionaries, such as the Harvard General Inquiry Dictionary[1], that measures the intersection between the word list and the text under analysis. The benefit of using dictionaries is that they facilitate the process of extracting information from text, since they drastically reduce the vocabulary to the words that belong to the dictionary, and that words chosen comes from psychosocial studies and, therefore,

[1] General Inquirer. Available at: <http://www.wjh.harvard.edu/~inquirer/>.

might capture a certain sentiment. This approach is very common among economic articles. Tetlock, Saar-Tsechansky and Macskassy (2008) use negative word count to measure the negativity of the media and predict individual firms' accounting earnings and stock returns. Loughran and McDonald (2011) use information retrieval techniques in Finance text, such as the TF-IDF scheme with weight normalization, to compare the effect of using unspecialized dictionaries and their own Finance Dictionary[2] on stock market variables, such as returns, trading volume, return volatility.

Another way to extract sentiment from the stock market is to use a supervised method, which we know the correct value for each classification in advance. We use this previous knowledge of the data to train and to calibrate the parameters of a model. Finance papers manually classify news to get training data for the classifier, e.g. Antweiler and Frank (2004), Antweiler and Frank (2006), Li (2010) and Huang, Zang and Zheng (2014) use the Naive Bayes classifier, trained with manually classified documents, to extract information in the analysis. Hendershott and Schurhoff (2015),Allen, McAleer and Singh (2015) and Heston and Sinha (2017) used Thomson Reuters News Analytics (TRNA), which is a black-box news sentiment classifier based on neural networks; but this method was also trained using specialists knowledge.

Manually classify news into negative or positive is very work demanding and subjective to multiple interpretations. So if the researcher does not work for a big company such as Thomson Reuters, it is very likely that his or her sample will be small and biased to the view of one person only. This implies that the supervised algorithm cannot benefit from the improvements in the generalization capacity brought by big data. So what if we could classify news into positive or negative using signals from the stock market instead of humans?

According to the efficient market hypothesis, prices fully reflect all available information, which implies that returns are unpredictable from past returns or other past variables, and the best forecast of a return is its historical mean. If the market is perfect, according to Black (1971), "the price adjusts so rapidly as the information becomes available" that it would not be possible to make profits. Furthermore, if we consider some information set of all available information, Campbell, Lo and MacKinlay (1997, p. 20) assert that the market is efficient with respect to that information set if prices would be unaffected by revealing that information to all participants.

Fama (1991) states that this strong version definition is undoubtedly false, due to ambiguity about information and trading costs; returns are predictable from past returns. Several empirical works show that the media influences prices and other stock market variables. Tetlock (2007) measures pessimism using the Wall Street Journal column and finds that that high media pessimism predicts downward pressure on market prices followed by a reversion to fundamentals, and unusually high or low pessimism predicts high market trading volume. Peress (2014) studies newspaper strikes in several countries and finds that the trading volume, the dispersion of stock returns and their intraday volatility are reduced on strike days, which leads to the conclusion that newspapers propagate news from the previous day.

Therefore, if media influences trading activity and prices absorb the information provided by the news (at least partially), then theoretically we can use future prices (or returns) to classify news. We

[2] Software Repository for Accounting and Finance. Available at: <https://sraf.nd.edu/textual-analysis/resources/>.

show that this is possible using the overnight returns. In our research, we did not find any substantial result using daily returns whatsoever. The night period, or the moment after the stock exchange closes and when it opens in the following day, might be the time that the investors take the news into account.

We build a sentiment measure based on public information that tries to capture the investors belief of what the market movement will be. This measurement is based on a learned vocabulary that uses the overnight return to classify news into negative or positive. Each news is actually an aggregation of past news for each stock. Learning involves finding the parameters related to the vocabulary and the weight used to average past documents. We learn the vocabulary and the document weight in small sliding in-sample windows. Then we evaluate our sentiment with learned vocabulary in out-of-sample regressions, and analyse the weight used to average past documents and terms used in selected time periods. We also analyse what words are negative and positive on periods which the trends of the preference share (PETR4) of Petrobras is up and down.

First, Section 3.2 (Background) presents some theoretical background regarding information retrieval. Then we detail the procedure of finding the sentiment measure based on learned vocabulary in Section 3.3 (Predicting stock market movements using learned dictionary). The description of our data is in Section 3.4 (Data) and results are in Section 3.5 (Results). Conclusions are in Section 3.6 (Conclusion).

## 3.2 Background

Extracting sentiment from documents starts by choosing the document representation. A common representation uses the vector space model, where each document is represented by a vector whose coordinates are associated to terms[3]. Concatenated document vectors form a term-document matrix, $\mathbf{M}^{\text{tfidf}}$, which establishes a relation between a term in a document. The weight $\omega_{i,j}$ of this matrix characterizes term importance, and is calculated according to a rule that considers two factors, one related to the term frequency (TF) and the other related to the inverse document frequency (IDF), which we call TF-IDF weighing scheme, and a normalization. That is:

$$\widetilde{\omega}_{i,j} = \begin{cases} f_{\text{tf}}(\text{tf}_{i,j}) \times f_{\text{idf}}(\text{df}_i) & \text{if } \text{tf}_{i,j} > 0 \\ 0 & \text{if } \text{tf}_{i,j} = 0 \end{cases} \tag{3.1}$$

$$\omega_{i,j} = \frac{\widetilde{\omega}_{i,j}}{\text{norm}_j} \tag{3.2}$$

where $\text{tf}_{i,j}$ is the term frequency of term $w_i$ in document $d_j$, $\text{df}_i$ is the document frequency of term $w_i$, $f_{\text{tf}}(\text{tf}_{i,j})$ is the weight associated with the term frequency of term $i$ in document $j$, $f_{\text{idf}}(\text{df}_i)$ is the weight associated with the document frequency of term $i$, and $\text{norm}_j$ is a document length normalization factor to compensate undesired effects of long documents. There are several weighting schemes:

[3] According to the information retrieval literature, a term is a word or group of consecutive words.

- suggested by Baeza-Yates and Ribeiro-Neto (2008, Table 3.6, p. 74), one recommended TF-IDF weighting scheme:

$$\widetilde{\omega}_{i,j} = (1 + \log(\mathsf{tf}_{i,j})) \log\left(\frac{N_D}{\mathsf{df}_i}\right) \tag{3.3}$$

- suggested by Loughran and McDonald (2011), used in many economic articles:

$$\widetilde{\omega}_{i,j} = \frac{(1 + \log(\mathsf{tf}_{i,j}))}{(1 + \log(\mathsf{avg}_{i'}\ \mathsf{tf}_{i',j}))} \log\left(\frac{N_D}{\mathsf{df}_i}\right) \tag{3.4}$$

where $\mathsf{avg}_{i'}\ \mathsf{tf}_{i',j}$ is the average of terms in the document $j$, $N_V$ is the size of the vocabulary and $N_D$ is the total of documents. As for the document length normalization factor, the most common is the document norm, $\mathsf{norm}_j = \|\mathbf{d}_j\| = \sqrt{\sum_i^{N_V} \widetilde{\omega}_{i,j}^2}$, but Tetlock, Saar-Tsechansky and Macskassy (2008) and García (2013) use the total of words, $\mathsf{norm}_j = \|\mathbf{d}_j\| = \sum_i^{N_V} \mathsf{tf}_{i,j}$.

Economic and finance articles usually capture sentiment from documents using the inner product similarity and some treatment:

$$sent_t^{sim} = \mathsf{sim}(\mathbf{d}_j^{\mathsf{tfidf}}, \mathbf{q}^{\mathsf{tfidf}}) = <\mathbf{d}_j^{\mathsf{tfidf}}, \mathbf{q}^{\mathsf{tfidf}}> \tag{3.5}$$

$$sent_t = \mathsf{treat}(sent_t^{sim}) \tag{3.6}$$

where $\mathbf{d}_j^{\mathsf{tfidf}} \in \mathbb{R}^{N_V}$ is the $j$-column of $\mathbf{M}^{\mathsf{tfidf}}$, which uses the TF-IDF weighting scheme, and $\mathbf{q}^{\mathsf{tfidf}} \in \mathbb{R}^{N_V}$ is a query vector whose terms come from a dictionary. One example of treatment commonly used is the standardization (z-score).

Chapter 2 (Document representations and information measurements in time series) shows other representations, such as dense vectors. Representing documents in dense spaces avoids sparsity and reduces the document dimension. And, instead of working with terms, it works with concepts or topics derived directly from the data, so similarity measurements can capture the context of terms since terms that are alike tend to share some common characteristics. The cosine is a common similarity function:

$$\mathsf{sim}(\mathbf{d}_j, \mathbf{q}) = \frac{<\mathbf{d}_j, \mathbf{q}>}{\|\mathbf{d}_j\|_2 \|\mathbf{q}\|_2} = \left\langle \frac{\mathbf{d}_j}{\|\mathbf{d}_j\|_2}, \frac{\mathbf{q}}{\|\mathbf{q}\|_2} \right\rangle \tag{3.7}$$

Two dense representations are: (i) Singular value decomposition of the term-document matrix; (ii) Paragraph vectors, or PV.

Singular value decomposition (SVD) of the term-document matrix is also called latent semantic index model in computer science, as in Furnas et al. (1988). It maps each document and query into a dimensional space composed of concepts. If $\mathbf{M}^{\mathsf{tfidf}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the SVD of matrix $\mathbf{M}^{\mathsf{tfidf}}$, we can construct a rank $k$ approximation of $\mathbf{M}^{\mathsf{tfidf}}$ by selecting the first $k$ columns of $\mathbf{U}$ and $\mathbf{V}$ and the first $k$ diagonal elements of $\mathbf{\Sigma}$:

$$\mathbf{M}^{\mathsf{tfidf}(k)} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T \tag{3.8}$$

where $k$ is the number of components used to approximate $\mathbf{M}^{\text{tfidf}}$.

Caron (2001) notices that $\mathbf{M}^{\text{tfidf}} \approx \mathbf{U}_k\mathbf{\Sigma}_k^{-p/2}\mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T \Rightarrow \mathbf{\Sigma}_k^{p/2}\mathbf{U}_k^T\mathbf{M}_{\text{tfidf}} \approx \mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T$, so $\mathbf{U}_k\mathbf{\Sigma}_k^{p/2}$ is a basis for $span(\mathbf{M}^{\text{tfidf}})$ and the columns of $\mathbf{M}^{\text{svd}(k,p)} = \mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T$ approximate $\mathbf{d}^{\text{tfidf}}$. This decomposition improves the quality of retrieval. From now on, we define SVD-TF-IDF(k,p) as the document representation using the singular value decomposition of the term-document matrix using $k$ components and tuning parameter $p$.

Paragraph Vectors (PV), proposed by Le and Mikolov (2014), represent documents by a dense vector using word embeddings from variable length texts. The algorithm is trained to predict words in the paragraph (sentences or documents) according the the context. Each paragraph vector has size $k$, a parameter chosed at training. The algorithm has two versions. The first is the Distributed Memory Model of Paragraph Vectors (PV-DM), which takes into consideration the word order, at least in a small context. The second is Distributed Bag of Words version of Paragraph Vector (PV-DBOW), without word ordering. Even though each of them perform really well for most taks, it is strongly recommended that both vectors should be concatenated, which we will name PV-DM+DBOW. The paragraph vector $\hat{\mathbf{d}}_j^{PV} = \mathbf{f}_{PV}(d_j)$ will be the document generated using the PV-DM or the PV-DBOW version of the algorithm and $\mathbf{f}_{PV}$ is the trained model that can map $d_j$ to $\hat{\mathbf{d}}_j^{PV}$. More details please refer to Item D (Models based on word embeddings).

## 3.3 Predicting stock market movements using learned dictionary

Predicting a stock market variable is a difficult problem. A stock market movement is a direction in which a stock market variable moves, i.e., if the variable goes up or down. So there are only two possible movements: up, when the variable becomes positive; or down, when it is negative. Let us denote the stock market signal $s_t$ as the stock market variable, e.g. daily or overnight returns, or the volume first difference. Then the stock market movement is $m_t$:

$$m_t = \begin{cases} 1 & \text{if } s_t \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{3.9}$$

which could be interpreted as the target variable (or class label) used in classifiers.

We want to find a function $\widetilde{m}_t = f_m(\mathbf{d}^{t-1}, \ldots, \mathbf{d}^{t-\tau}; \mathbf{q}; \boldsymbol{\theta})$, where $\mathbf{d}^t$ is a document in time $t$, $\mathbf{q}$ is a query vector and $\boldsymbol{\theta}$ represents other parameters, that is as close as $m_t$ as possible, i.e. that solves:

$$\min_{\mathbf{q}} \frac{1}{T-\tau} \sum_{t=\tau+1}^{T} \{m_t - \widetilde{m}_t\}^2 \tag{3.10}$$

The function $f_m$ could come from a classifier, such as the Naive Bayes or the Random Forest. The problem in representing documents with TF-IDF is that document and query vectors have high dimension and are sparse, and this may lead to a scenery with more variables than samples. That is why some researchers use feature selection to filter terms before, which actually improves the classifier performance as long as the feature selection is not extremely aggressive. Nevertheless, information is still missed, since terms are removed from the vocabulary. Another approach would be to use dense representation of documents, such as the SVD-TF-IDF or PV.

The sentiment measurement, eq. (3.5), used in economic articles rely on a query vector derived from categories (e.g. "negative" or "uncertain") from dictionaries. Sometimes it captures sentiment successfully in regressions because its coefficient is statistically significant. But this is not a general rule. In fact, Li (2011) already has cited several articles that cannot find statistically significant regressions and has pointed the deficiencies of dictionary approaches: a dictionary might not contain specific terms of a certain topic, such as finance; the context of terms is ignored[4]; any prior knowledge that researchers may have is ignored; and it is not as natural as statistics. A reasonable question would be: is there a query vector which could capture sentiment and does not rely on dictionaries? Specifically, is it the problem in eq. (3.10) feasible and does it produce consistent and statistically significant regressions?

Our baseline hypothesis is that a variable in time $t$ is influenced by previous documents, and that the more recent the document is, higher should be its influence:

$$\begin{aligned}\widetilde{m}_t(\mathbf{d}^{t-1},\ldots,\mathbf{d}^{t-\tau};q_0,\mathbf{q}) &= q_0+\sum_{s=1}^{\tau}\theta_s \text{sim}(\mathbf{d}^{t-s},\mathbf{q})\\ &= q_0+\sum_{s=1}^{\tau}\theta_s\left\langle\frac{\mathbf{d}^{t-s}}{\left\|\mathbf{d}^{t-s}\right\|_2},\frac{\mathbf{q}}{\left\|\mathbf{q}\right\|_2}\right\rangle\\ &= q_0+\left\langle\sum_{s=1}^{\tau}\theta_s\frac{\mathbf{d}^{t-s}}{\left\|\mathbf{d}^{t-s}\right\|_2},\frac{\mathbf{q}}{\left\|\mathbf{q}\right\|_2}\right\rangle\end{aligned} \tag{3.11}$$

where $q_0$ is a constant, $\theta_1\geq\cdots\geq\theta_\tau\geq 0$ and $\sum_{s=1}^{\tau}\theta_s=1$. The similarity is given by eq. (3.7). We propose the following definition for $\theta_s$:

$$\theta_s(\rho)=\frac{s^{-\rho}}{\sum_{r=1}^{\tau}r^{-\rho}},\ \rho\geq 0 \tag{3.12}$$

Notice that $\rho=0\Rightarrow\theta_s=1/\tau$, $\forall s$, and $\rho\to\infty\Rightarrow\theta_1\to 1$. Finally, the learned query vector $\mathbf{q}$ will be the solution of the minimization:

$$\begin{aligned}\min_{q_0;\mathbf{q};\rho}\ & \frac{1}{T-\tau}\sum_{t=\tau+1}^{T}\left\{m_t-\left(q_0+\left\langle\bar{\mathbf{d}}_{(\tau,\rho)}^{t-1},\mathbf{q}\right\rangle\right)\right\}^2\\ \text{s.t.}\ & \left\|\mathbf{q}\right\|_2=1\\ & \rho\geq 0\end{aligned} \tag{3.13}$$

where $\bar{\mathbf{d}}_{(\tau,\rho)}^{t-1}$ is the weighted average document:

$$\bar{\mathbf{d}}_{(\tau,\rho)}^{t-1}=\sum_{s=1}^{\tau}\theta_s(\rho)\frac{\mathbf{d}^{t-s}}{\left\|\mathbf{d}^{t-s}\right\|_2} \tag{3.14}$$

So, given a set of documents $\mathbf{d}^{t-1},\ldots,\mathbf{d}^{t-\tau}$, the learned query vector $\mathbf{q}$ is used to extract information from these past news and tries to predict what will be the future movement. Therefore, it acts as a sentiment or a belief if the future movement will be positive or not. Specifically, if we denote $b_{t-1}$ as the sentiment in $t-1$ of what the movement will be in $t$ as:

$$b_{t-1}=\left\langle\bar{\mathbf{d}}_{(\tau,\rho)}^{t-1},\mathbf{q}\right\rangle-q_0 \tag{3.15}$$

[4] A common problem in Natural Language Processing is word disambiguation. Many words have several meanings. Thus, if words are out of context, there is ambiguity interpreting them.

then the sentiment it will be positive if $s_{t-1} \geq 0$, and negative otherwise. Additionally, we can use the learned query vector $\mathbf{q}$ to analyse what words have the most impact on the sentiment. In case of the SVD-TF-IDF representation, it is necessary to remap the dense document to the TF-IDF sparse representation by multiplying by $\mathbf{U}_k\mathbf{\Sigma}_k^{-p/2}$, so the impact of each term is its corresponding component of:

$$\mathbf{sim} = \text{diag}(\mathbf{U}_k\mathbf{\Sigma}_k^{-p/2}\mathbf{q})(\mathbf{U}_k\mathbf{\Sigma}_k^{-p/2}\bar{\mathbf{d}}_{(\tau,\rho)}^{t-1}) \tag{3.16}$$

where $\mathbf{sim}$ is the similarity vector that measures the impact of each term in the TF-IDF sparse space.

We estimate parameters in eq. (3.13) using an in-sample period, then we evaluate the performance of the model using an out-of-sample period. The in-sample period must be right-sized, it cannot be short enough so that it is unable to estimate or gives inaccurate estimates of the parameters, but it also cannot be long enough so that old news becomes entirely different from recent news and do not add up. The same goes for the out-of-sample period, news used for prediction would be utterly unrelated to the ones used in the in-sample period if the out-of-sample period is too long, or could bring unnecessary computational harassment if the period is too short.

We test our method using a sliding window strategy. Let $W_{(t_i,t_f)}$ be a time window:

$$W_{(t_i,t_f)} = \{t \in \{1,\ldots,T\} \mid (t \geq t_i) \text{ and } (t \leq t_f)\} \tag{3.17}$$

First, we define in- and out-of-sample window sizes, $n_{in}$ and $n_{out}$, respectively, and the number of lags $\tau$ used to aggregate documents. Next we use the in-sample window $W^{in}_{(t,t+n_{in}-1)}$ to estimate parameters $\rho$, $q_0$ and the query vector $\mathbf{q}$, and the out-of-sample window $W^{out}_{(t+n_{in},t+n_{in}+n_{out}-1)}$ to evaluate the performance of the model. We use $W^{in}$ and $W^{out}$ for estimation and prediction, respectively, then slide the in-sample window by $n_{out}$, and redo the procedure. We show a pseudo-code of these steps in Algorithm 1.

**Algorithm 1 Estimating parameters for the learned vocabulary sentiment**. $n_{in}$ and $n_{out}$ are in- and out-of-sample window sizes, respectively, $T$ is the total of times steps, $\mathbf{M}^{\text{svd}(k,p)}$ are dense documents using the singular value decomposition of the term-document matrix using $k$ components and tuning parameter $p$, $\mathbf{m}$ is the movement associated to each document, and $\tau$ is the number of lags when averaging documents. $\mathbf{x}[W]$ is the subset of $\mathbf{x}$ that belongs to the window $W$.

**procedure** GENERATE THE LEARNED SENTIMENT($n_{in}, n_{out}, T, \mathbf{M}^{\text{svd}(k,p)}, \mathbf{m}, \tau$)
  $t \leftarrow 1$
  **while** $t + n_{in} + n_{out} \leq T$ **do** ▷ There is enough data
    $W_{in} \leftarrow$ window between $t$ and $t + n_{in} - 1$ ▷ In-sample window
    $q_0, \mathbf{q}, \rho \leftarrow$ Arguments of the minimization of eq. (3.13) using $\mathbf{M}^{\text{svd}(k,p)}[W_{in}]$, $\mathbf{m}[W_{in}]$ and $\tau$

    $W_{out} \leftarrow$ window between $t + n_{in}$ and $t + n_{in} + n_{out} - 1$ ▷ Out-of-sample window
    $\mathbf{b}_{out} \leftarrow$ sentiment of eq. (3.15) using $\mathbf{M}^{\text{svd}(k,p)}[W_{out}], q_0, \mathbf{q}, \rho, \tau$

    Save $\mathbf{b}_{out}, q_0, \mathbf{q}, \rho$ with their association to the window $W_{out}$
    $t \leftarrow t + n_{out}$ ▷ Slide window
  **end while**
**end procedure**

## 3.4 Data

We have two sources of data: quantitative variables and news. The Economatica[5] database provide quantitative data relative to the stock market. The only variables used are: (i) stock daily return $r_t$; (ii) stock overnight return $o_t$; (iii) detrended log volume, $vlm_t$; (iv) stock volatility, $vol_t$. Stock returns are $r_t = \log\left(p_t^c/p_{t-1}^c\right)$ and $o_t = \log\left(p_t^o/p_{t-1}^c\right)$, where $p_t^c$ and $p_t^o$ are the closing and opening price in time $t$. For the detrended log volume, we use the rolling average of the past 60 days of log volume to detrend log of daily volume, the methodology based on Campbell, Grossman and Wang (1993). We use the detrended squared return residuals to proxy for past volatility, a strategy also used by Tetlock (2007), which we demean the daily return to obtain the residual, square this residual, and then subtract the past 60-day moving average of the squared residual.

News are from two online newspapers: $(i)$ *Valor Econômico*[6], sections *Finance*, *Companies* and *Politics*; and $(ii)$ *Folha de S.Paulo*[7], sections *Market*, *Power* and *World*. The time period is from 1st January 2012 to 2nd July 2018. All news are written in Portuguese. Stopwords were removed from the news. Short and long news were removed, i.e. the gamma distribution using the words per document was cut using approximately 1% in the lower tail and 5% in the upper tail. Also, only the news whose stocks were the most 50 frequent in the database were kept. This procedure is detailed in Section 2.6 (Data).

Considering the 50 stocks chosen (appendix 2.9.5, Table 2.14) and the trading period from $1^{th}$ January 2012 to $7^{th}$ July 2018, there are over 46,000 samples, uniquely identified by the trading date and stock. We sample the stock market data by trading date and stock. We aggregate news in each sample since there is multiple news per day and stock. We concatenate all the news belonging to a stock and trading date into a big news, and then we apply the TF-IDF weighing scheme considering the raw frequencies in this big news and document frequencies from the original term-document matrix. We use the original document frequencies because it is an indicator of informativeness, so we want to maintain its original word distribution.

The amount of data in regressions vary with the in-sample window. The first in-sample period is discarded because we only use out-of-sample data in our regressions. Table 3.1 details some information of in- and out-of-sample periods. The proportion positive and negative samples is approximately constant on periods, roughly 60% of the samples have positive overnight return.

## 3.5 Results

We study the prediction of the overnight return and the effect of the sentiment generated by a learned dictionary by running the linear regression:

$$o_t = \alpha_0 + \boldsymbol{\alpha} \cdot \mathcal{L}^s(o_t) + \boldsymbol{\beta} \cdot \mathcal{L}^s(b_t) + \boldsymbol{\gamma} \cdot \mathcal{L}^s(\mathbf{x}_t) + \boldsymbol{\delta} \cdot \mathbf{d}_t + \varepsilon_t \tag{3.18}$$

[5] <https://economatica.com/>
[6] Valor Econômico. Available at: <http://www.valor.com.br/>.
[7] Folha de S.Paulo. Available at: <http://www.folha.uol.com.br/>.

**Table 3.1. Regression sample size for different in- and out-of-sample periods.** The data has 46,000 samples. Regressions only consider the out-of-sample period, so the first in-sample period is discarded. The in-sample window size, $n_{in}$, is in days. We show the out-of-sample size that we considered in regressions, along with statistics of how much positive and negative overnight return is present in the data. Columns $\#(o_t \geq 0)$ and $\#(o_t < 0)$ correspond do positive and negative overnight returns, respectively.

| **In-sample window** | | **Out-of-sample period** | | | |
|---|---|---|---|---|---|
| **Period** | $n_{in}$ | **Period** | **Size** | $\#(o_t \geq 0)$ | $\#(o_t < 0)$ |
| 2-jan-2012 to 9-jan-2013 | 252 | 10-jan-2013 to 2-jul-2018 | 37,703 | 22,677 | 14,989 |
| | | | | *60.1%* | *39.8%* |
| 2-jan-2012 to 15-jan-2014 | 504 | 16-jan-2014 to 2-jul-2018 | 30,230 | 18,355 | 11,838 |
| | | | | *60.7%* | *39.2%* |
| 2-jan-2012 to 21-jan-2015 | 756 | 22-jan-2015 to 2-jul-2018 | 22,903 | 13,993 | 8,873 |
| | | | | *61.1%* | *38.7%* |

where $\mathcal{L}^s(y_t)$ denotes an $s$-lag operator[8], $o_t$ denotes the overnight return, and $\mathbf{x}_t$ and $\mathbf{d}_t$ are other variables and dummies, respectively. We evaluate our model performance using two regressions: (A) use the trading volume $vlm_t$ and the volatility $vol_t$ for the set of variables $\mathbf{x}_t$, and days of the week dummies and a dummy to indicate if the month is January for $\mathbf{d}_t$, inspired by Tetlock (2007); (B) $\mathbf{x}_t = (o_t^2)$ and $\mathbf{d}_t$ using days of the week dummies, inspired by García (2013). Both regressions use $s = 5$ lags.

If news causes the movement of the future overnight return, and the sentiment proxies the belief that news may cause that movement, then the coefficient of $b_{t-1}$ should be positive.

The problem of creating a sentiment that is based on a learned vocabulary, eq. (3.15), involves several choices: the document representations, including its hyperparameters tuning; the sizes of the in- and out-of-sample windows; and the number of lags $\tau$ to include in the documents weighted average. We found the solution of the minimization problem of eq. (3.13) using several configurations of parameters. We represent documents using the singular value decomposition of the term-document matrix. We test two variations, a low dimension with parameters $(k, p) = (128, -1)$, and a high dimension with parameters $(k, p) = (1024, -0.5)$. Chapter 2 (Document representations and information measurements in time series) shows that the high dimensional approximation can mostly capture the essence of the original term-document matrix. In- and out-of-sample windows should be right sized, so we test $n_{in} \in \{252, 504, 756\}$ days (1, 2 and 3 years) and $n_{out} \in \{21, 63, 126, 252\}$ days (1, 3, 6 and 12 months). For each combination of these parameters, we use lags $\tau \in \{1, 2, 3\}$. We evaluate our sentiment $b_t$, eq. (3.15), by analyzing the coefficient of $b_{t-1}$ in the regression of (3.18) using the sets of variables (A) and (B) detailed just below. The results of these regressions are summarized in Table 3.2. Each element of the table corresponds to the average of $\beta_1$ over the regressions using $\tau \in \{1, 2, 3\}$ lags for $b_t$, and on the side of each value we put one, two or three asterisks corresponding to the maximum $p$-value at 10%, 5% and 1% level of the hypothesis $H_0 : \beta_1 = 0$ on these regressions.

Regressions using the high dimensional approximation provide more statistically significant coefficients than the low approximation. $(k, p) = (128, -1)$ results in $7 \times 3 = 21$ significant $\beta_1$'s at 10% level while $(k, p) = (1024, -0.5)$ results in $12 \times 3 = 36$ significant coefficients, considering all

[8] $\mathcal{L}^s(y_t) = \{y_{t-1}, \ldots, y_{t-s}\}$.

**Table 3.2. Simulation performance in regressions**. We show averages of the first component of the coefficient $\boldsymbol{\beta}$, associated to $b_{t-1}$ which we refer as $\beta_1$, in regressions of eq. (3.18). We vary: (i) the number of components ($k$) of the approximation of the term-document matrix; (ii) the size of the in-sample window, $n_{in}$; and (iii) the size of the out-of-sample window, $n_{out}$. For each of these combination of parameters, we regress using sets (A) and (B) of variables $\mathbf{x}_t$ and dummies $\mathbf{d}_t$ with the sentiment $b_t$ given by eq. (3.15). Each value in this table represents the average of $\beta_1$ using $\tau = \{1, 2, 3\}$ for the sentiment. On the side of each coefficient average we put one, two or three asterisks corresponding to the maximum $p$-value at 10%, 5% and 1% level of the hypothesis $H_0 : \beta_1 = 0$ on these regressions.

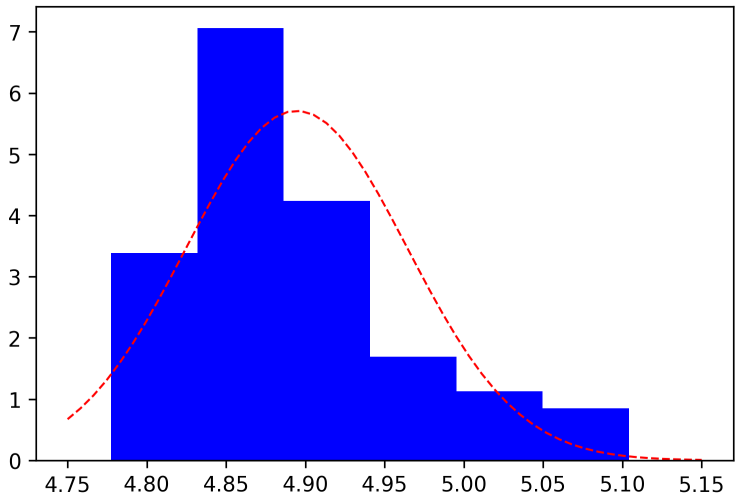
| **Regression** | $k$ | $n_{in}$ | $n_{out}$ | | | |
|---|---|---|---|---|---|---|
| | | | 21 | 63 | 126 | 252 |
| A | 128 | 252 | 0.0037*** | 0.0034*** | 0.0029** | 0.0030 |
| | | 504 | 0.0027* | 0.0020 | 0.0015 | 0.0022 |
| | | 756 | 0.0022 | 0.0016 | 0.0010 | 0.0004 |
| | 1024 | 252 | 0.0089*** | 0.0085*** | 0.0100*** | 0.0095*** |
| | | 504 | 0.0046* | 0.0049* | 0.0053* | 0.0061** |
| | | 756 | 0.0018 | 0.0010 | 0.0017 | −0.0003 |
| B | 128 | 252 | 0.0033*** | 0.0031** | 0.0025** | 0.0025 |
| | | 504 | 0.0025 | 0.0018 | 0.0011 | 0.0018 |
| | | 756 | 0.0019 | 0.0013 | 0.0007 | 0.0001 |
| | 1024 | 252 | 0.0079*** | 0.0077*** | 0.0090*** | 0.0083*** |
| | | 504 | 0.0038 | 0.0041 | 0.0041 | 0.0047 |
| | | 756 | 0.0011 | 0.0002 | 0.0006 | −0.0017 |

lags for $\tau$. Lower in-sample period implies in better regressions, when $n_{in} = 252$ (1 year) almost every set of parameters gives a significant $\beta_1$. The statistical power increases for documents using $k = 1024$ components because all regressions using sets (A) and (B) of variables provide statistically significant coefficients of $b_{t-1}$ at the level of 1%. Fewer components mean having the capacity to retain less information; this is why only smaller out-of-sample periods ($n_{out} \leq 126$) produce significant $\beta_1$'s. On the other hand, more components mean that the document representation can retain information in a way that it can predict $n_{out} = 252$ days ahead (the same size as $n_{in}$). Since the learned query vector $\mathbf{q}$ has more parameters, it can memorize more topics that are important in a broader range of situations described in the news. Following the same line of thinking, we conclude that smaller out-of-sample windows are better for regressions; $n_{out} = 21$ days (or 1 month) provided more significant coefficients across different values of $k$ and $n_{in}$. The coefficients for $(k, p) = (1024, -0.5)$, $n_{in} = 252$ are approximately the same different values of $n_{out}$ and sets of regression variables.

Given the context just described, we choose the document representation with $(k, p) = (1024, -0.5)$ and windows $n_{in} = 252$ and $n_{out} = 21$ to be our basic setting hereafter. And we now investigate the effect of the number of lags $\tau$ on regressions. Specifically, how important are recent news for predicting future movements? Is there an ideal number of lags for documents when predicting future movements?

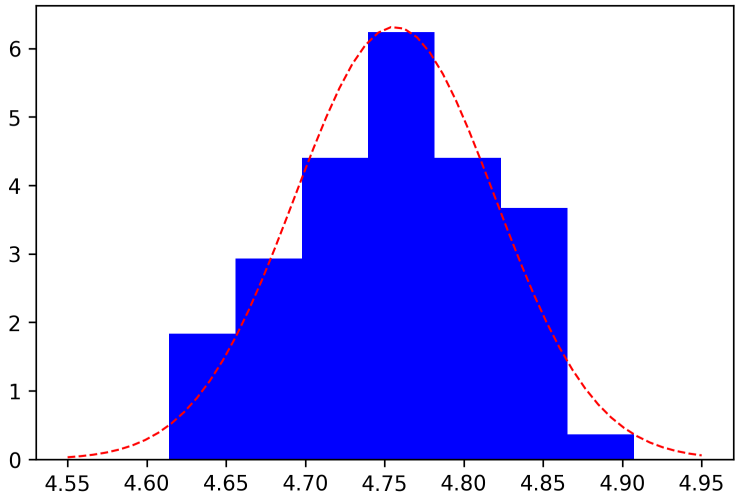
The parameter $\rho$ gives the importance of each weight in the average of past documents. It is an argument in the minimization given by eq. (3.13). We show in Figure 3.1 the distribution of $\rho$ considering each minimization solution for the several slided out-of-sample windows. Item (a) shows the

**Figure 3.1 – How important are recent news to the overnight return**? Distribution of the parameter $\rho$ of eq. (3.12), which represents how much weight is given to recent documents, in simulations using documents with parameters $(k, p) = (1024, -0.5)$, in-sampling windows of size $n_{in} = 252$ days, out-of-sample windows of size $n_{in} = 63$ days. There are 45,337 samples, from $2^{nd}$ January 2012 to $2^{nd}$ July 2018. The histogram in bars is the actual data from the values of $\rho$ obtained in each out-of-sample windows, and the dashed line is the fitted normal distribution. Item $(a)$ is the simulation using $\tau = 2$ lags and gives $\mu = 4.89$ and $\sigma = 0.07$; item $(b)$ is the simulation using $\tau = 3$ and gives $\mu = 4.76$, $\sigma = 0.06$.




(a) $\tau = 2$

| $\rho = 4.89$ | |
|---|---|
| $s$ | $\theta_s(\rho)$ |
| 1 | 96.8 % |
| 2 | 3.2% |



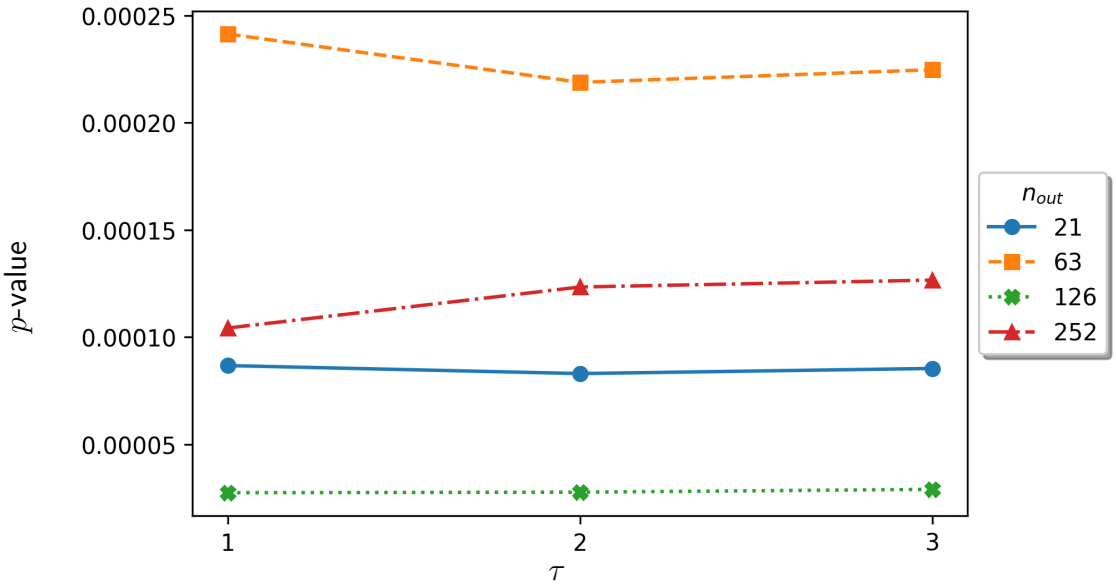

(b) $\tau = 3$

| $\rho = 4.76$ | |
|---|---|
| $s$ | $\theta_s(\rho)$ |
| 1 | 96.0 % |
| 2 | 3.5% |
| 3 | 0.5% |

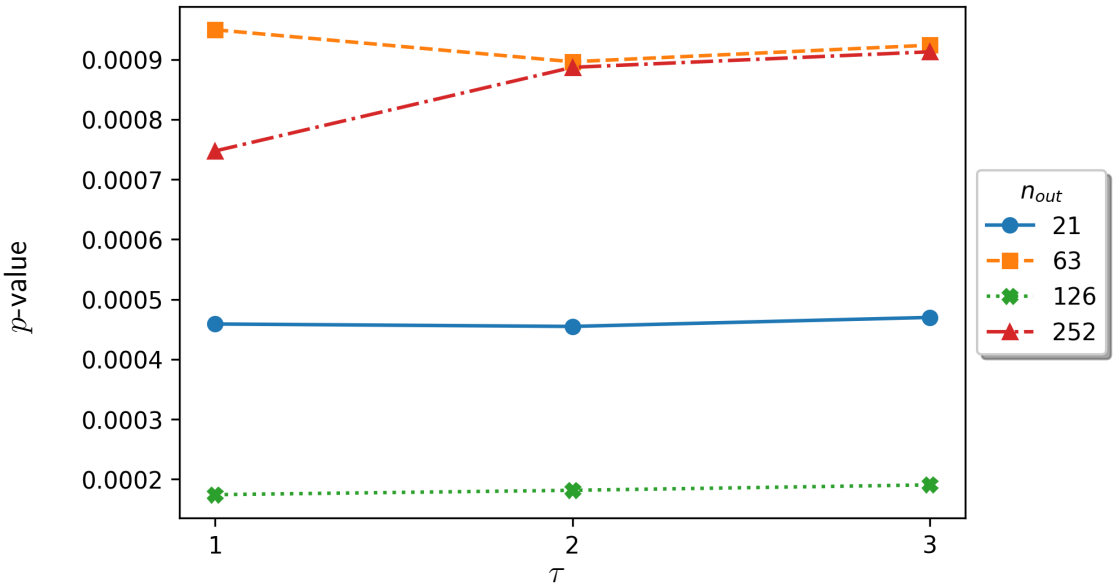
distribution of $\tau = 2$ lags and (b) shows the distribution of $\tau = 3$ lags. Since $E[\rho] \neq 0$, news are not weighted equally. The weights of recent news are greater than past news, which implies that they are more important. And since $E[\rho] < +\infty$, we infer that not only $t-1$ is important, but other lags are important as well. The document in $t-1$ is the most important for the effect in $t$ and is responsible for approximately 96% of the averaged document.

Weights of documents in $t-1$ are greater than the ones associated to other past documents. So we investigate if there exists an ideal number of lags for documents when predicting future movements

**Figure 3.2 – How many lags results in the best prediction**? $p$-values of the coefficient of $b_{t-1}$ in the regression given by eq. (3.18) using documents with parameters $(k, p) = (1024, -0.5)$ and in-sampling windows of size $n_{in} = 252$ days. Item (a) refers to the set (A) of variables, while item (b) refers to the set (B).




(a) Regression (A)




(b) Regression (B)

or if lags $\tau > 1$ can be discarded. We plot in Figure 3.2 the $p$-values of the coefficient of $b_{t-1}$ in the regression given by eq. (3.18). The smallest $p$-values vary according to the out-of-sample window, it is either $\tau = 1$ or $\tau = 2$. So we choose $\tau = 2$ for our further results[9].

We showed so far that exists parameters that can successfully capture statistical significance of the sentiment $b_t$, which is a belief of what future movement the overnight return will be. The movement sentiment captures the impact of words in public news on the future overnight return. Ideally, this effect should be positive because it indicates that the movement expectation and reality are positively

[9] Choosing $\tau = 1$ does not alter any of the conclusions.

correlated. Table 3.3 shows regressions of eq. (3.18) using sets of variables (A) and (B), described just below the equation. We use two measurements for the sentiment: one dictionary based on eq. (3.5) with weights given by eq. (3.4); and other based on the learned vocabulary given by the sentiment $b_t$ in eq. (3.15). We use positive words from the finance dictionary, from Loughran and McDonald (2011), because we want to verify if positive words capture the effect of positivity. We calculate the learned query vectors using documents with parameters $(k, p, \tau) = (1024, -0.5, 2)$, in-sampling windows of size $n_{in} = 252$ days and out-of-sampling windows of size $n_{out} = 21$ days.

The only statistically significant at the 10% level is the coefficient of the sentiment extracted from the words in the dictionary is the one related to $t-2$, which shares the same sign of the coefficient of the sentiment using the learned vocabulary. Even though the coefficient of $t-1$ is negative, its confidence interval has a positive interval. All other coefficients of the dictionary sentiment are not significant. They are not jointly significant either. The sentiment using the learned vocabulary fits the data better, regressions produce coefficients of the sentiment jointly significant at 1% level, and mostly individually significant at 1% level. Both set of variables have similar values, but set (A) has one more significant coefficient associated to $b_{t-3}$. This means that there is evidence that the sentiment using learned vocabulary is associated in some way with future overnight returns.

In our research, we applied the same methodology to the daily return instead of the overnight return. Very few regressions were statistically significant and, even so, were not consistent with the coefficient signs. This suggests that the Brazilian stock market is not perfect, but prices adjust at most of one day as public information becomes available.

It is also interesting to look at the effect of returns and other economic variables on the content of the public news. If the learned vocabulary sentiment is a reasonable measure that captures public news essence, then market variables from the recent past may predict the values of the sentiment. We use the equation below to verify this hypothesis:

$$b_t = \alpha_0 + \boldsymbol{\alpha} \cdot \mathcal{L}^s(o_t) + \boldsymbol{\beta} \cdot \mathcal{L}^s(b_t) + \boldsymbol{\gamma} \cdot \mathcal{L}^s(\mathbf{x}_t) + \boldsymbol{\delta} \cdot \mathbf{d}_t + \varepsilon_t \tag{3.19}$$

where $\mathbf{x}_t$ is the same sets of variables (A) and (B) as before. Results are in Table 3.4. The dictionary sentiment using set (A) of variables is the only that does not produce a positive coefficient for $o_{t-1}$, all other regressions have a positive coefficient for $o_{t-1}$. This means that an increase in $t-1$ returns can predict more optimism or more belief that prices will follow the trend to go up.

The learned vocabulary sentiment captures the effect of the positive feedback trading of Long J Bradford (1990). Rational speculators receive good news and trade on this news, stimulating buying by positive feedback traders tomorrow.

Since the sentiment movement comes mostly from a learned query vector $\mathbf{q}$, it is relevant to check what words positively or negatively influence these movements. For such a task, we use the documents related to the preference share (PETR4) of Petrobras, a Brazilian corporation in the petroleum industry. We select two periods, one preceding a drop in the stock price, which we will refer as the pessimistic period, and other preceding an increase in the stock price, which we will refer as the optimistic period. Figure 3.3 shows the opening price, the overnight return (acting as the stock market

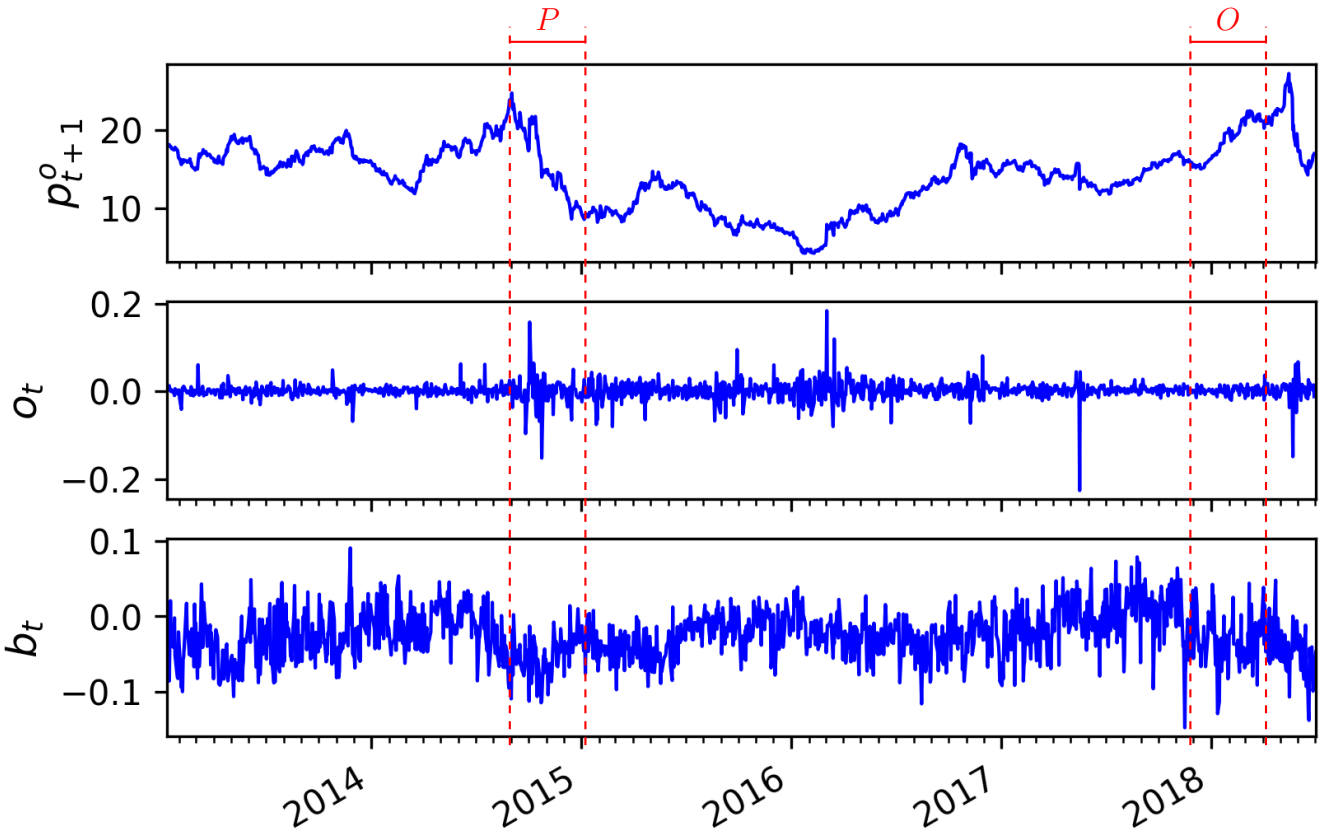
**Table 3.3. Predicting overnight returns using learned sentiment.** We regress the overnight return $o_t$ using eq. (3.18). The only regressand is the overnight return. Dictionary columns calculate the sentiment $b_t$ using the dictionary approach, eq. (3.5) with weights given by eq. (3.4), with the positive words from the finance dictionary. Learned vocabulary columns calculate the sentiment $b_t$ in eq. (3.15) using documents with parameters $(k, p, \tau) = (1024, -0.5, 2)$, in-sampling windows of size $n_{in} = 252$ days and out-of-sampling windows of size $n_{out} = 21$ days. Columns (A) and (B) refers to the set of variables described just below eq. (3.18). We only show all coefficients of $\boldsymbol{\beta}$, other coefficients we only show the coefficient corresponding to the first lag. One, two or three asterisks means that the p-value of the $H_0$ : coefficient $= 0$ is inferior to 0.10, 0.05 and 0.01, respectively. The estimates use a 37,437 sample over $16^{th}$ January 2013 to $2^{nd}$ June 2018, which is the of-the-sample period.

| | Dictionary | | Learned vocabulary | |
|---|---|---|---|---|
| | (A) | (B) | (A) | (B) |
| const | 0.0018*** | 0.0015*** | 0.0021*** | 0.0017*** |
| | (0.0003) | (0.0003) | (0.0002) | (0.0002) |
| $o_{t-1}$ | 0.0037 | 0.0304*** | 0.0007 | 0.0283*** |
| | (0.0053) | (0.0048) | (0.0053) | (0.0048) |
| ⋮ | | | | |
| $b_{t-1}$ | -0.0031 | -0.0001 | 0.0089*** | 0.0080*** |
| | (0.0046) | (0.0045) | (0.0023) | (0.0023) |
| $b_{t-2}$ | -0.0077* | -0.0087* | -0.0018 | -0.0029 |
| | (0.0047) | (0.0046) | (0.0023) | (0.0023) |
| $b_{t-3}$ | 0.0036 | 0.0021 | 0.0039* | 0.0034 |
| | (0.0047) | (0.0046) | (0.0023) | (0.0023) |
| $b_{t-4}$ | 0.0068 | 0.0063 | 0.0082*** | 0.0076*** |
| | (0.0047) | (0.0046) | (0.0023) | (0.0023) |
| $b_{t-5}$ | -0.0027 | -0.0031 | 0.0065*** | 0.0062*** |
| | (0.0045) | (0.0045) | (0.0023) | (0.0023) |
| $vlm_{t-1}$ | 0.0011*** | | 0.0011*** | |
| | (0.0002) | | (0.0002) | |
| ⋮ | | | | |
| $vol_{t-1}$ | 0.0318*** | | 0.0321*** | |
| | (0.0027) | | (0.0027) | |
| ⋮ | | | | |
| $o_{t-1}^2$ | | 0.2625*** | | 0.2558*** |
| | | (0.0272) | | (0.0272) |
| ⋮ | | | | |
| $R^2$ | 0.0088 | 0.0081 | 0.0114 | 0.0101 |
| F | 13.35 | 15.99 | 17.33 | 20.05 |
| p-value | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $H_0 : \beta_1 = \cdots = \beta_5 = 0$ | | | | |
| $F$ statistics | 1.16 | 1.11 | 20.85 | 16.39 |
| $p$-value | 0.32 | 0.35 | 0.00 | 0.00 |

**Table 3.4. Predicting sentiment using overnight returns.** We regress the sentiment $b_t$ using eq. (3.19). The only regressand is the sentiment. Dictionary columns calculate the sentiment $b_t$ using the dictionary approach, eq. (3.5) with weights given by eq. (3.4), with the positive words from the finance dictionary. Learned vocabulary columns calculate the sentiment $b_t$ in eq. (3.15) using documents with parameters $(k, p, \tau) = (1024, -0.5, 2)$, in-sampling windows of size $n_{in} = 252$ days and out-of-sampling windows of size $n_{out} = 21$ days. Columns (A) and (B) refers to the set of variables described just below eq. (3.18). We only show all coefficients of $\beta$, other coefficients we only show the coefficient corresponding to the first lag. One, two or three asterisks means that the p-value of the $H_0$ : coefficient $= 0$ is inferior to 0.10, 0.05 and 0.01, respectively. The estimates use a 37,437 sample over $16^{th}$ January 2013 to $2^{nd}$ June 2018, which is the of-the-sample period.

| | Dictionary | | Learned vocabulary | |
|---|---|---|---|---|
| | (A) | (B) | (A) | (B) |
| const | 0.0102*** | 0.0100*** | -0.0040*** | -0.0042*** |
| | (0.0004) | (0.0004) | (0.0004) | (0.0004) |
| $o_{t-1}$ | -0.0147** | 0.0135** | 0.0461*** | 0.0348*** |
| | (0.0060) | (0.0054) | (0.0120) | (0.0108) |
| $o_{t-2}$ | 0.0051 | 0.0006 | 0.0769*** | 0.0578*** |
| | (0.0061) | (0.0056) | (0.0123) | (0.0111) |
| $o_{t-3}$ | -0.0003 | -0.0054 | 0.0484*** | 0.0500*** |
| | (0.0062) | (0.0057) | (0.0125) | (0.0115) |
| $o_{t-4}$ | -0.0045 | -0.0122** | 0.0382*** | 0.0370*** |
| | (0.0064) | (0.0059) | (0.0129) | (0.0118) |
| $o_{t-5}$ | -0.0115* | -0.0209*** | 0.0018 | 0.0012 |
| | (0.0064) | (0.0059) | (0.0129) | (0.0118) |
| $b_{t-1}$ | 0.2473*** | 0.2551*** | 0.2675*** | 0.2675*** |
| | (0.0051) | (0.0051) | (0.0051) | (0.0051) |
| ⋮ | | | | |
| $vlm_{t-1}$ | 0.0035*** | | -0.0009** | |
| | (0.0002) | | (0.0004) | |
| ⋮ | | | | |
| $vol_{t-1}$ | 0.0271*** | | -0.0049 | |
| | (0.0031) | | (0.0061) | |
| ⋮ | | | | |
| $o_{t-1}^2$ | | 0.1365*** | | 0.0573 |
| | | (0.0307) | | (0.0614) |
| ⋮ | | | | |
| $R^2$ | 0.3904 | 0.3836 | 0.3049 | 0.3045 |
| F | 957.90 | 1224.95 | 656.02 | 861.74 |
| p-value | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $H_0 : \alpha_1 = \cdots = \alpha_5 = 0$ | | | | |
| $F$ statistics | 2.14 | 4.84 | 15.83 | 13.94 |
| $p$-value | 0.06 | 0.00 | 0.00 | 0.00 |

**Figure 3.3 – Opening price, overnight return and movement sentiment series**. Out-of-sample period from $10^{th}$ January 2013 to $2^{nd}$ July 2018. We detach two regions from the graph: region $(P)$, from $1^{st}$ September 2014 to $31^{st}$ December 2014, the pessimistic period in which prices drop; and region $(O)$, from $1^{st}$ December 2017 to $31^{st}$ March 2018, the the optimistic period in which there is an increase in prices.




signal) and the sentiment; we cut off the pessimistic and optimistic period, which we denote (P) and (O), respectively.

The movement sentiment acts as a belief that the stock signal movement, the overnight return, will increase in the following day. Each element of the inner product $\left\langle \bar{\mathbf{d}}^{t-1}_{(\tau,\rho)}, \mathbf{q} \right\rangle$ contributes to a portion of the belief. Adding documents in the dense space is equivalent of adding them in the TF-IDF sparse space[10]. So we add all documents in the pessimistic (optimistic) period, remap the aggregated documents and the query vector obtained from the minimization problem from the dense space to the TF-IDF sparse one, calculate the similarity vector using eq. (3.16), then analyse 100 terms that have the smallest (highest) value. Table 3.5 and Table 3.6 show the result graphically and for some selected terms.

Usually, sentiment is measured taking into consideration the amount of terms in a dictionary. We translated the positive and the negative words from the Harvard IV-4 psychosocial dictionary from English to Portuguese using Google Translator[11]. Each word in English is translated into several words in Portuguese, and each translated word is classified as suggested, common, uncommon or rare. We

[10] Just notice that $\sum_i \mathbf{d}^i_{\text{tfidf}} = \sum_i \mathbf{U}_k \mathbf{\Sigma}_k^{-p/2} \mathbf{d}^i_{\text{svd}(k,p)} = \mathbf{U}_k \mathbf{\Sigma}_k^{-p/2} \left( \sum_i \mathbf{d}^i_{\text{svd}(k,p)} \right)$.

[11] Google Translator. Available at: <https://translate.google.com/>.

only kept the suggested and common words, then we applied stemming, which resulted in 1,514 positive and 1,902 negative words. We verified that 19 negative words from the dictionary were in the 100 least important terms in the pessimistic period and that 22 positive words from the dictionary were in the 100 most important terms in the optimistic period[12]. Since the vocabulary has 42,910 words, and the negative and positive words are roughly 4% of the total of words, it exists an intersection between the words from the learned query vector and the words chosen from a psychological context.

A common source of misclassification of words into positive or negative is the negation. Negating a word changes its polarity. But sometimes an isolated word from the dictionary might not add much information to a sentiment measurement. For instance, consider the negative word *raise*[13] from the Harvard dictionary. Raising taxes has a negative impact on the stock market, but increasing investments has a positive one. This distortion is not only due to the automated translation but to the context in which the word belongs. This is why not only the words in the dictionary are important but the context as well. Since we learn query vectors in small out-of-sample windows, we are learning the context that matters in these windows.

Table 3.5 shows words that absorbed a negative effect of the overnight return. The learned vocabulary captured some important words that give context to the news, such as stock expectations, option contracts, energy related topics (such as the companies Eletrobras, Usiminas and Petrobras), oil refinery or the pre-salt exploration. On the selected negative period, there was a lot of legislation debate regarding the pre-salt exploration and production, a topic which was frequently brought up by a few presidential candidates. This is why there are a lot of government and legislative terms, such as Eduardo Cunha, the former Chamber Deputies president, and also convict by crimes of corruption and money laundering. As to the economy, inflation started to haunt again and was consistently increasing. Operation Car Wash, a criminal investigation carried out by the Federal Police, began in March 2014 and had put Petrobras in the center of government corruption; important terms from this extremely relevant event are present, such as Janot, Prosecutor-General of the Federal Prosecution Service (MPF), or Youssef, a black market money dealer who was arrested as a consequence of the investigations.

Table 3.6 shows words that absorbed a positive effect of the overnight return. The first two rows in the table of selected words imply in gain of productivity, increase in refinery capacity, integrity of the company, barrel export, which all definitely carry positivity. But the relation between other words and positive stock returns is somehow not obvious. Partially because they are mainly due to the Brazilian scenery, but also because bad information is processed more thoroughly than good, which impact our memory (see Baumeister et al. (2001)). On the selected time period, there was a debate on gasoline prices, government taxation and inflation control. Also, inflation was much more controled[14]. There are

[12] If we increase the number of terms kept from the similarity vector, we maintain approximately 20% of negative and positive words from the Harvard dictionary.

[13] *Raise* and *increase* are both translated to *aumentar* in Portuguese, which stemmed becomes *aument*.

[14] Inflação pelo IPCA desacelera em novembro, aponta IBGE. Available at: <https://www.valor.com.br/brasil/5221999/inflacao-pelo-ipca-desacelera-em-novembro-aponta-ibge>.

**Table 3.5. Negative words that affect the overnight return of Petrobras**. We sum the averaged documents, eq. (3.14), over each out-of-sample window, remap documents and the query vector obtained from the minimization problem, and calculate the similarity vector using eq. (3.16). The similarity vector is summed over the pessimistic period, detached in Figure 3.3, from $1^{st}$ September 2014 to $31^{st}$ December 2014. Panel A shows the word cloud of the 100 terms that contribute the least to the movement sentiment. Panel B details a few selected words from the cloud.

**Panel A: Word cloud**


**Panel B: Selected words from the word cloud**

| Category | Terms with stemming | Translated terms without stemming |
|---|---|---|
| Market | açã papel expect contrat | stock expectation contract |
| Energy | energ elétr usin hidrelétr Eletrobras usiminas Petrobras estatal camp petróle refin pré sal | energy electric plant hydroelectric Eletrobrás Usiminas Petrobras state-owned field oil refinery pre-salt |
| Economy | econom tax básic jur cort invest indústr tribut | economy basic interest rate cuts investments industry taxes |
| Government | govern minist ministr reform congress parlament eduard cunh câm deput sen lei plen comiss vot | government department officer reform congress parliament Eduardo Cunha chamber deputies senate law plenary commission voting |
| Justice | crim investig tribun penal procur janot mpf del lav jat youssef | crime investigation court criminal attorney Janot MPF delation Lava Jato Youssef |

many news that mention Petrobras and BNDES[15,16,17] (Brazilian Development Bank), impeachment[18,19]

[15] Juro baixo leva empresas a quitar dívida com BNDES. Available at: <https://www.valor.com.br/financas/5386141/juro-baixo-leva-empresas-quitar-divida-com-bndes>.

[16] CPI do BNDES no Senado aprova parecer sem pedidos de indiciamento. Available at: <https://g1.globo.com/politica/noticia/cpi-do-bndes-aprova-parecer-sem-pedidos-de-indiciamento.ghtml>.

[17] TCU abre investigação sobre bônus para funcionários de estatais. Available at: <https://noticias.uol.com.br/politica/ultimas-noticias/2018/01/31/bonus-para-diretores-de-estatais-entra-na-mira-do-tcu.htm>.

[18] 'Seria covardia não ser candidato', diz Temer a revista. Available at: <https://www1.folha.uol.com.br/poder/2018/03/seria-covardia-nao-ser-candidato-diz-temer-a-revista.shtml>.

[19] Lula rechaça radicalismo e prega Estado forte. Available at: <https://www.valor.com.br/politica/5234325/lula-rechaca-radicalismo-e-prega-estado-forte>.

**Table 3.6. Positive words that affect the overnight return of Petrobras**. We sum the averaged documents, eq. (3.14), over each out-of-sample window, remap documents and the query vector obtained from the minimization problem, and calculate the similarity vector using eq. (3.16). The similarity vector is summed over the pessimistic period, detached in Figure 3.3, from $1^{st}$ December 2017 to $31^{st}$ March 2018. Panel A shows the word cloud of the 100 terms that contribute the most to the movement sentiment. Panel B details a few selected words from the cloud.

**Panel A: Word cloud**


**Panel B: Selected words from the word cloud**

| Category | Terms with stemming | Translated terms without stemming |
|---|---|---|
| Productivity | ganh produt capac refin integr | gain productivity capacity refinery integrity |
| Oil | export barril internac Petrobras | export barrel international Petrobras |
| Economy | impost govern rentabil infl real | taxes government rentability inflation Real |
| Politics | BNDES impeachment camarg | BNDES impeachment Camargo |
| Justice | JBS joesley del premi | JBS Joesley plea bargain |
| | pf polici investig dol albert youssef | PF (Federal Police) investigation doleiro (black-market money dealer) Albert Youssef |

(mostly mentioning the impeachment of former president Dilma Rousseff) and Camargo Correa[20]. The connection between the company JBS, which was involved in corruption, is not obvious, but there are highly postive news[21] to the IBovespa index after the plea bargain of JBS owner Joesley Batista. Also, Operation Car Wash carried out by the Federal Police of Brazil, which developed greatly after plea

[20] Ex-vice-presidente da Camargo Corrêa tem prisão preventiva decretada. Available at: <https://www1.folha.uol.com.br/colunas/monicabergamo/2018/03/ex-vice-presidente-da-camargo-correia-tem-prisao-preventiva-decretada.shtml>.

[21] Varejistas disparam até 16% em 5 pregões; JBS sobe 86% desde maio e apaga perdas causadas pela delação. Available at: <https://www.infomoney.com.br/mercados/acoes-e-indices/noticia/7167460/varejistas-disparam-ate-pregoes-jbs-sobe-desde-maio-apaga-perdas>.

bargain of Alberto Youssef, was mentioned in news[22,23,24] that positively impacted the overnight return.

The word *real*[25] is positive in the Harvard dictionary. But real is also the the official currency of Brazil. Therefore, translating words blindly may lead to distortions.

Finally, the word polarity depends on the context because words may have a different meaning due to context or time of its use. Petrobras was a negative word during in 2014 but became positive in 2017. Interestingly, that also happened to *plea bargain* and *Youssef*. The beginning of the investigation of Operation Car Wash, which put Petrobras at the center of corruption in 2014, led to events such as the arrest of Youssef and had terribly negative repercussions for the company. From the plea bargain of Youssef, as well as changes in the management of Petrobras, the investigations indicated restitution of devious money and had the effect of returning the image of the company.

## 3.6 Conclusion

We proposed the creation of a sentiment measurement based on learned vocabulary. We used public news and the overnight return to find the vocabulary. Regressions envolving the learned vocabulary sentiment suggest that public news influence overnight returns. We found that news in $t-1$ have the most impact in returns in $t$, but the influence of other past news is not negligible.

The polarity of words can change in different contexts. Also, using the dictionary approach with translated words may lead to distortions, so letting the text dictate what is positive or negative seems like a right approach.

[22] Lava Jato devolve R$ 654 milhões de uma vez à Petrobras. Available at: <https://www1.folha.uol.com.br/poder/2017/12/1941381-lava-jato-devolve-r-654-milhoes-de-uma-vez-a-petrobras.shtml>.

[23] Ex-policial que carregava malas de dinheiro para Youssef é preso. Available at: <https://www.valor.com.br/politica/5297075/ex-policial-que-carregava-malas-de-dinheiro-para-youssef-e-preso>.

[24] Operação Lava Jato completa quatro anos. Available at: <https://www.valor.com.br/politica/5391411/operacao-lava-jato-completa-quatro-anos>.

[25] Real can be translated the same in Portuguese, or can be translated to *verdadeiro*, which means truthful.

# 4 Modeling an analyst

**Abstract**

Sentiment analysis is a process that identifies, extracts, measures and study subjective information from text, such as affective states, usually by contrasting a text to a dictionary that relates words to a particular subjective category (for example, "pessimism" or "uncertain"). We propose to generate a sentiment vector from text and quantitative variables, so each component of the vector would bring a different degree of subjectivity, without the use of any dictionary. Since quantitative and qualitative variables are used all together as time variables, we call this model "the analyst model". Using data from B3 (Brazilian Stock Exchange) and two online newspapers (*Valor Econômico* and *Folha de S.Paulo*), we create a two dimension sentiment to study the effect of two opposite states (optimism and pessimism) on stock returns, trading volume and volatility. The pessimism of our model produces similar results from previous works, which gives us support to use the optimism sentiment in other analysis.

**Keywords**: stock market, market analyst, text analysis, sentiment

## 4.1 Introduction

In this paper, we propose to model sentiment as a vector using time variables and documents as well. Then we test our model using data from the stock market and online news.

An analyst interprets relevant variables and news, and is able to form a perception based on the information that is available. This perception acts as an input to emotions that causes a certain action. Emotions are directly related in decision making. For example, Gino, Brooks and Schweitzer (2012) find that anxiety motivates individuals to seek and use advice, even if it is bad.

A qualitative analysis extracted from the news is part of any analyst's life, e.g. fundamental analysis of the stock market or macro-economic factors. Mathematically represent information from text helps us to extract quantitative measures from a text, which we know as textual analysis. One example is sentiment extracted from the news, which classifies news into positive or negative using words from a dictionary. Studies, such as the ones conducted by Tetlock (2007), Tetlock (2010) and Loughran and McDonald (2011), show that market sentiment is correlated with stocks.

The representation of documents impacts directly in the quality of classification. One of the most popular forms to mathematize text is to create a term-document matrix, which relates a term in a document. The weights of this matrix are calculated using term frequencies (TF), the inverse of document frequencies (IDF), and some kind of normalization. According to Li (2010), statistical methods should be prefered over word categorization one; Loughran and McDonald (2015) show that this may lead to a sentiment misclassification. Nevertheless, the process of extracting sentiment using a list of words from a dictionary (such as pessimism, negative or weak) is still mostly used. Loughran and

McDonald (2011) show two solutions to correct this misclassification: (i) choose an adequate weighting scheme, with term and document frequencies; or (ii) use a dictionary with terms related to the text, e.g. a finance dictionary with terms more adequate to finance text. If the correct weighting scheme is chosen (one that considers term and document frequencies, instead of term frequencies only), the dictionary becomes less important because the weights incorporate the word informativeness.

Another matter regarding dictionaries is that words are not absolute. First, the meaning of some words in a dictionary could be captured by synonyms which are not in the dictionary. Second, dictionaries available are mostly in English, so translating them to a different language could distort the meaning of the words; probably this could be another source of misclassification. Third, the context of words could distort its meaning, so a positive word in a certain context could be negative in a different one, e.g. the word *high* would be positive in *high stock return* but be negative in *high inflation*. These issues could be mitigated if documents were represented in a dense space because they would be formed by concepts or ideas, instead of the actual words. This will be discussed later on.

Many articles in finance use supervised methods to create a sentiment measurement, which is a single valued variable. Antweiler and Frank (2004), Antweiler and Frank (2006), Li (2010) and Huang, Zang and Zheng (2014) use the Naive Bayes classifier to extract information, while Hendershott and Schurhoff (2015) and Heston and Sinha (2017) used Thomson Reuters News Analtyics (TRNA)[1], a black-box news sentiment classifier based on neural networks. The problem with this approach is that sentiment is treated as binary, such as negative/positive or good/bad, usually between 0 and 1. So if a sentiment is measured as a score of the classifier, and negativity is measured by this value, then positiveness would be one minus this score. By construction, negative and positive sentiments would sum to one, so these variables could not participate in the same regression (this would cause perfect multicollinearity), and the coefficients in individual regressions would be equal with opposite signs[2]. Since sentiments are perfectly correlated, it is not possible to extract the true value of positiveness or negativity. Therefore, we propose to use a sentiment vector, where each component corresponds to a true sentiment state (such as positive or negative).

A similar issue, going to a single value representation to a vectorized one, has been present in information theory before. Each column in the term-document matrix represents a document, and a word is a single number in this column vector. Advances in the area make possible to represent each word as a vector. Each component of a word vector corresponds to a feature and might even have a semantic or grammatical interpretation, as noted by Turian, Ratinov and Bengio (2010). The new representation of words as vectors unveiled several relationships between words and led to better performance in algorithms. Examples of algorithms to generate word vectors from text are: Word2vec, proposed by Mikolov et al. (2013), and GloVe, proposed by proposed by Pennington, Socher and Manning (2014).

Lastly, we propose a procedure to generate a sentiment vector that uses quantitative variables and text as an input and puts in perspective a relevant variable, the reason of the analysis. We believe

[1] Thomson Reuters News Analtyics: sentiment analysis, relevance, novelty. White paper. Available at <https://sircaknowledgebase.force.com/s/article/Thomson-Reuters-News-Analtyics-TRNA>.

[2] Here is an example to clarify this fact. Let $s_t^{\text{neg}} \in [0,1]$ be the score provided by a classifier algorithm, and $s_t^{\text{pos}} + s_t^{\text{neg}} = 1$. If we regress a variable $y_t$ on $s_t^{\text{neg}}$, since $y_t = \beta_0 + \beta_{\text{neg}} s_t^{\text{neg}} = \beta_0 + \beta_{\text{neg}}(1 - s_t^{\text{pos}}) = (\beta_0 + \beta_{\text{neg}}) - \beta_{\text{neg}} s_t^{\text{pos}}$, the coefficients of $s_t^{\text{pos}}$ and $s_t^{\text{neg}}$ have the same absolute value and opposite signs.

that this would simulate the process of analysis because it is the basis of the perception construction of an analyst. So each component of the sentiment vector gives meaningful interpretations. Studies usually explore the information in quantitative variables, or these added to a single-valued sentiment. Time delays are applied to variables and sentiment. As far as we know, we have not seen any model using quantitative variables and documents, with time delays applied to the document vector, to derive a vector of sentiments. Therefore, the model tries to capture all available information, quantitative and qualitative. Hence, it models an analyst.

Many articles regress pessimism, measured by words from dictionaries, on stock market variables, such as returns and volume. Some try to use positive words instead of negative ones, but the statistical significance in regressions is much weaker. This is why Tetlock (2007) only used words associated to negativity (pessimism, negative and weak categories from the Harvard Dictionary) on his regressions. We generate a two dimension sentiment, representing the opposite states optimism and pessimism, compare the result of pessimism with the previous work for validation, and show the unprecedented result of optimism.

Section 4.2 (Background) reviews some key themes in the literature used throughout this chapter, Section 4.3 (The analyst model) presents the analyst model, Section 4.4 (Analysing the stock market using learned sentiments) shows the results of the analyst model in the brazilian stock market, and Section 4.5 (Conclusions) closes the chapter.

## 4.2 Background

Columns of the term-document matrix represents documents as vectors such that each element corresponds to a term in the vocabulary. This representation typically is high dimensional and very sparse. So if we used these representations as parameters in a model without any treatment, likely we would have more variables than samples. One way to fight sparsity is to use dense representations. This will be addressed in Section 4.2.1 (Document representations).

Next, Section 4.2.2 (Sentiment in economic articles) reviews how sentiments are extracted from texts. Essentially, the inner product by the document vector and the query vector representing the word list from a dictionary, and a further transformation.

Lastly, Section 4.2.3 (Pessimism and the stock market) focus on the theory of the media and the stock market, which is used to interpret the results.

### 4.2.1 Document representations

One of the most popular forms to mathematize text is to create a term-document matrix, in which rows correspond to words or terms, and columns correspond to documents. The weights of this matrix are calculated using term frequencies (TF), the inverse of document frequencies (IDF), and some kind of normalization, hence the name TF-IDF matrix. So, if $\mathbf{M}_{\text{tfidf}}$ is the $N_V \times N_D$ term-document matrix, where $N_V$ is the size of the vocabulary and $N_D$ is the number of documents, then the weight

$\omega_{i,j}$ is:

$$\widetilde{\omega}_{i,j} = \begin{cases} f_{\text{tf}}(\text{tf}_{i,j}) \times f_{\text{idf}}(\text{df}_i) & \text{if } \text{tf}_{i,j} > 0 \\ 0 & \text{if } \text{tf}_{i,j} = 0 \end{cases} \tag{4.1}$$

$$\omega_{i,j} = \frac{\widetilde{\omega}_{i,j}}{\text{norm}_j} \tag{4.2}$$

where $f_{\text{tf}}(\text{tf}_{i,j})$ is the weight associated with the term frequency, $\text{tf}_{i,j}$ is the raw term frequency of term $i$ in document $j$, $f_{\text{idf}}(\text{df}_i)$ is the weight associated with the document frequency, $\text{df}_i$ is the document frequency of term $i$ following the collection of documents[3], and $\text{norm}_j$ is a document length normalization factor to compensate undesired effects of long documents. There are several weighting scheme; a common choice is $\omega_{i,j} = (1 + \log(\text{tf}_{i,j})) \times \left(\log \frac{N_D}{\text{df}_i}\right)$. For other consecrated TF-IDF weights, see Salton and Buckley (1988), Manning, Raghavan and Schütze (2008), Baeza-Yates and Ribeiro-Neto (2008).

Documents represented by the term-document matrix are relatively easy to implement. But they are sparse and have high dimension. The vocabulary usually has a lot of terms, ranging from 100 thousand to 2 million terms, so using all document features as regressors would be unlikely to succeed because regressions usually use data sampled by day. There would not be enough observations. If those regressions also considered features of delayed documents, we would bet that finding valid regressions would be nearly impossible.

One way to fight sparsity is using dimensionality reduction by term selection, or just feature selection. Traditional information retrieval methods include selecting terms by the document frequency, mutual information, information gain, and chi-square statistics. A direct aplication of this technique would be to create a dictionaries. Antweiler and Frank (2006) use the information gain of each word and create a dictionary based in the ones with the largest information gain, which is later used to feed the Naive Bayes algorithm. Another approach, mostly used in economic articles, is to select terms based on a pre-defined dictionary, such as the Harvard General Inquiry Dictionary[4] or the Finance dictionary[5] provided by Loughran and McDonald (2011). These dictionaries count on the specialty and experience of the authors. Only the terms in the dictionary are selected.

Measuring sentiment using word lists from dictionaries filters the vocabulary drastically, but it leads to a few drawbacks. Even though dictionaries are created by specialists, leaving most of the terms out of the analysis implies that information is being left out. Also, the sentiment is only measured in documents which the words from a dictionary category are present. Thus, concepts or synonyms to those list of words are not considered. If documents were represented in dense spaces, they would reduce their dimension and still keep their information. Examples of dense representations are the approximation of the term-document matrix using the singular value decomposition and Paragraph Vectors.

The approximation of the term-document matrix using the singular value decomposition is named latent semantic index model in information retrieval. This methodology, proposed by Furnas et al. (1988), maps each document into a dimensional space composed of concepts. Alternatively, the approximation of the documents is also possible using Paragraph Vectors, an algorithm proposed by

[3] If $\mathbf{M^1}$ is the matrix $\mathbf{M}^{\text{tfidf}}$ with binary weights, then $\text{df}_i = \mathbf{M^1 1}_{N_D}$, where $\mathbf{1}_{N_D}$ is a column vector of ones of size $N_D$.
[4] General Inquirer. Available at: <http://www.wjh.harvard.edu/~inquirer/>
[5] Software Repository for Accounting and Finance. Available at: <https://sraf.nd.edu/>.

Le and Mikolov (2014) that generate document vectors from a text corpus. A paragraph is defined as any variable length text, such as paragraphs, sentences or documents. Paragraph vectors are trained to predict words in the paragraph, and somehow generate good representation of documents. This chapter embraces the first approach[6], which we will call TF-IDF-SVD.

An important issue regarding document sampling is how to group documents in a time period so that they can be used in regressions. Documents are associated to a time period, and there could be many documents for each time period. We follow the grouping procedure of Chapter 2 (Document representations and information measurements in time series). Let $\mathbf{M}^{\text{tf}} \in \mathbb{R}^{N_V \times N_D}$ be the high-dimensional and sparse term-document matrix, where weights use term frequencies only. If $\mathbf{e}_j$ is the j-th unit column vector, then $\mathbf{d}_j^{\text{tf}} = \mathbf{M}^{\text{tf}}\mathbf{e}_j$ is the $j$-th document of the term-document matrix. Each document $\mathbf{d}_j^{\text{tf}}$ is associated with a time period $t$, and there are $1, \dots, T$ time periods. Let $\mathbf{K} = [\delta_1 \mid \cdots \mid \delta_{N_D}]^T$ be a $N_D \times T$ matrix of concatenated dummy variables, where each $\delta_j$ is a $T$ sized dummy equal to 1 if $\mathbf{d}_j^{\text{tf}}$ is associated to $t$ and 0 otherwise. Then the term-time matrix will be the term-document matrix with weights summed by the time period, i.e. $\mathbf{M}_t^{\text{tf}} = \mathbf{M}^{\text{tf}}\mathbf{K}$.

Then we apply the TF-IDF weighting scheme to $\mathbf{M}_t^{\text{tf}}$, resulting in the matrix $\mathbf{M}_t^{\text{tfidf}}$ whose weights are:

$$\omega_{i,t} = \begin{cases} (1 + \log(\text{tf}_{i,t})) \times \left(\log \frac{N_D}{\text{df}_i}\right) & \text{if } \text{tf}_{i,t} > 0 \\ 0 & \text{if } \text{tf}_{i,t} = 0 \end{cases} \tag{4.3}$$

where $\text{tf}_{i,t}$ is the raw term frequency of term $i$ in time $t$.

The singular value decomposition of the matrix is $\mathbf{M}_t^{\text{tf}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, and if we consider only the $k$ largest singular values of $\mathbf{\Sigma}$, along with their corresponding columns in $\mathbf{U}$ and $\mathbf{V}^T$, then the approximation of $\mathbf{M}_t^{\text{tfidf}}$ is $\widetilde{\mathbf{M}}_t^{\text{tfidf}} = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$. Since $\mathbf{M}_t^{\text{tfidf}} \approx \mathbf{U}_k\mathbf{\Sigma}_k^{-p/2}\mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T \Rightarrow \mathbf{\Sigma}_k^{p/2}\mathbf{U}_k^T\mathbf{M}_{\text{tfidf}} \approx \mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T$, the columns of $\mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T$ approximate the columns of $\mathbf{M}_t^{\text{tfidf}}$ in a space that has $\mathbf{U}_k\mathbf{\Sigma}_k^{p/2}$ as its basis set. So $\mathbf{M}^{\text{svd}(k,p)} = \mathbf{\Sigma}_k^{1+p/2}\mathbf{V}_k^T \in \mathbb{R}^{k \times T}$ will be our component-time matrix with documents being represented in a lower-dimension space, and $\mathbf{d}_t = \mathbf{M}^{\text{svd}(k,p)}\mathbf{e}_t \in \mathbb{R}^k$ will be the document in time $t$.

### 4.2.2 Sentiment in economic articles

Many articles in economics that extract sentiment from documents from the term-document matrix with TF-IDF weights and a predefined word lists from a dictionary. If $\mathbf{d}_t^{\text{tfidf}}, \mathbf{q}^{\text{tfidf}} \in \mathbb{R}^{N_V}$ is the document and the word list, respectively, then the basic sentiment measurement is the inner product $<\mathbf{d}_t^{\text{tfidf}}, \mathbf{q}^{\text{tfidf}}>$. Articles diverge in what weights are used and what treatment to make before regressions.

Tetlock, Saar-Tsechansky and Macskassy (2008) uses negative terms from the General Inquirer's Harvard IV-4 psychosocial dictionary; and documents weights are term frequencies normalized by the total of terms in the document. Then the measure is standardized with the mean and standard deviation of each previous year. Basically, the sentiment is based on proportional weights. Loughran and McDonald (2011) derive the sentiment measure using the full TF-IDF weighting scheme (with functions applied to term and document frequencies, and a weight normalization) and a specific finance dictionary.

[6] Our regressions using Paragraph Vectors gave the same results as the singular value decomposition of the term-document matrix, but with less statistical significance. Thus, conclusions are the same.

### 4.2.3 Pessimism and the stock market

The strong version of the market efficiency hypothesis assumes that security prices fully reflect all available information. This implies that the diffusion of every type of publicly available information takes place instantaneously among all investors and that investors act on the information as soon as it is received. So regressing sentiment variables on returns should not provide any statistical significant coefficients. Obviously, this assumption is false. Non-exhaustive reasons include ambiguity about information and trading costs. Merton (1987), Fama (1991) provide a nice discussion of this matter. Empirically, Tetlock (2007) finds statistical significance when regressing pessimism on the Daily Jones return. His study uses $n_c = 77$ categories from the Harvard IV-4 psychosocial dictionary, such as *pessimism*, *negative* and *weak*, as an input to find sentiment based on its categories; sentiments are the most important semantic components extracted from the principal components factor analysis.

So if the media is a proper channel to transmit information, it certainly has the power to influence the stock market activity. Antweiler and Frank (2004) studies Internet stock message boards, and find that stock messages help predict market volatility and that disagreement among the posted messages is associated with increased trading volume.

Moreover, the stock market activity is closely related to the psychological behavior of investors. Anxiety affects investment decisions, and consequently asset pricing, because people become more pessimistic about future profitability, so they tend to take less risk. Interestingly, Coval and Shumway (2001) use ambient noise level in the Chicago Board of Trade's 30-year Treasury Bond futures trading pit as a measure of anxiety. High levels of noise imply immediacy in trades. Therefore, a belief that the costs of trading will change. If traders perceive that the costs of trading might rise in the future, they have strong incentives to execute their trades immediately, therefore affecting the stock market, such as in volume and volatility.

Campbell, Grossman and Wang (1993) proposes a model based on risk-averse "market makers", who accommodate buying or selling pressure from "liquidity" or "noninformational" traders, and give other reasons why pessimism could be related to volume. For example, if price dropped due to an exogenous pressure by noninformational traders, market makers would have to buy stocks expecting higher return, which would be accompanied by high volume and be most likely reversed in the following days. Otherwise, expected return would not change, any pressure by noninformational traders must reveal itself in unusual volume, so we would expect a low volume. This is why this models suggest that a stock price decline is more likely to happen on a high-volume day.

Nevertheless, Tetlock, Saar-Tsechansky and Macskassy (2008) state that sentiment derived from negative words produce much stronger results than the ones derived from positive ones, probably because texts frequently negate positive words. Later on, García (2013) continued the research on sentiment and the stock market and successfully used positive words in regressions. Since both articles use the same basic measurement, word count normalized by the total of words, we attribute this phenomenon to the dictionary, the first use the Harvard dictionary and the latter use the specific finance dictionary.

## 4.3 The analyst model

Let $\mathcal{L}^{p_i}(x_t^i) = [x_{t-1}^i, \ldots, x_{t-p_i}^i]^T$ be a vector with $p_i$ time delays for the time variable $x_t^i$, $\mathbf{z}_t^x = (\mathcal{L}^{p_1}(x_t^1)^T, \ldots, \mathcal{L}^{p_n}(x_t^n)^T) \in \mathbb{R}^{n_x}$ be $n$ concatenated time variables with their respective time delays, $\mathbf{d}_t \in \mathbb{R}^k$ be a document with dense space representation associated to time $t$, and $\mathbf{z}_t^d = \mathcal{L}^{p_{n+1}}(\mathbf{d}_t)$ be a concatenated vector of documents in different time periods. There are $1, \ldots, T$ time periods.

The model uses quantitative variables ($\mathbf{z}_t^x$) and text ($\mathbf{z}_t^d$) to analyse a specific variable of interest. This variable, $y_t$, is the reason of the analysis. So the purpose of the analyst model is to find a vector of meaningful sentiments that can associate $\mathbf{z}_t^x$ and $\mathbf{z}_t^d$ to $y_t$. Essentially, it tries to find a representation for a sentiment by mapping the input, $\left[\mathbf{z}_t^x, \mathbf{z}_t^d\right]$, to a vector that relates to the variable of interest $y_t$.

Let $\mathbf{s}_t \in [0,1]^{n_s}$ be the sentiment vector. Each coordinate in $\mathbf{s}_t$ represents a feature that could be considered a mood, or a perception towards a particular topic, regarding $y_t$. The model is given by:

$$\mathbf{s}_{t-1} = f_s(\mathbf{z}_t^x, \mathbf{z}_t^d; \boldsymbol{\theta}_s) \tag{4.4}$$

where $\boldsymbol{\theta}_s$ are the model parameters and $f_s$ is a function that translates time variables and documents to the sentiment vector $\mathbf{s}_{t-1}$. These parameters are obtained by the minimization:

$$\min_{\boldsymbol{\theta}_s, \boldsymbol{\theta}_y} \sum_{t=1}^{T} [y_t - f_y(\mathbf{s}_{t-1}; \boldsymbol{\theta}_y)]^2 \tag{4.5}$$

where $f_y$ is a function that predicts the desired variable $y_t$ using the sentiment vector $\mathbf{s}_{t-1}$ and $\boldsymbol{\theta}_y$ are the parameters of $f_y$.

The function $f_s$ is defined by the following steps:

i. **project the time variables and low-dimensional documents into a even lower sentiment space**.

First we consider two sentiment vectors separately, one from variables and other from documents. We map each input, $\mathbf{z}_t^x$ and $\mathbf{z}_t^d$, to a new representation of size $n_s$. We do this by making a linear projection of the input vectors:

$$\mathbf{p}_{t-1}^x = \mathbf{W}_x \, \mathbf{z}_t^x \tag{4.6}$$

$$\mathbf{p}_{t-1}^d = \mathbf{W}_d \, \mathbf{z}_t^d \tag{4.7}$$

where $\mathbf{W}_x \in \mathbb{R}^{n_s \times n_x}$ and $\mathbf{W}_d \in \mathbb{R}^{n_s \times n_d}$. Elements of $\mathbf{W}_d$ are set to zero with a probability $\zeta$ because an analyst is always subject to leave some information out or decide to ignore some information based on his beliefs. The probability $\zeta$ is also refered as a dropout parameter in machine learning articles, and according to Srivastava et al. (2014), it is a simple way to prevent overfitting. Now $\mathbf{p}_{t-1}^x, \mathbf{p}_{t-1}^d \in \mathbb{R}^{n_s}$.

ii. **apply a bounded function to enforce the same scale to quantitative variable and documents**.

The linear projections $\mathbf{p}_{t-1}^x$ and $\mathbf{p}_{t-1}^d$ have the downside that they might not be centered and might be in completely different scales, so each could have a disproportional effect if summed. Thus

we add a bias term to each linear projection to guarantee the best centering position, and then applied to a bounded function. The result is the scaled sentiments $\mathbf{q}^x_{t-1}$ and $\mathbf{q}^d_{t-1}$. Mathematically, we have that each coordinate or feature in $\mathbf{p}^x_{t-1}$ and $\mathbf{p}^d_{t-1}$ will be bounded between -1 and 1:

$$\mathbf{q}^x_{t-1} = \tanh(\mathbf{p}^x_{t-1} + \mathbf{b}_x) \tag{4.8}$$

$$\mathbf{q}^d_{t-1} = \tanh(\mathbf{p}^d_{t-1} + \mathbf{b}_d) \tag{4.9}$$

where $\mathbf{b}_x, \mathbf{b}_d \in \mathbb{R}^{n_s}$ are bias parameters, $\mathbf{q}^x_{t-1} \in [-1, 1]^{n_s}$ is the sentiment associated to variables, $\mathbf{q}^d_{t-1} \in [-1, 1]^{n_s}$ is the sentiment associated to documents, and tanh is the hyperbolic tangent:

$$\tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1} \tag{4.10}$$

that we apply element-wise. Now $\mathbf{q}^x_{t-1}$ and $\mathbf{q}^d_{t-1}$ have the same scale.

iii. **make sentiments relative to each other**.

Sentiments should be relative, that is, its value should consider other sentiments in a given context. We consider that $S_t$ is a random variable taking on values $i \in \{0, \ldots, n_s\}$, $i$ is the index of a sentiment state, $\mathbf{q}_{t-1} = \mathbf{q}^x_{t-1} + \mathbf{q}^d_{t-1}$ a set of conditioning variables, and $\mathbf{s}_{t-1} \in [0, 1]^{n_s}$ a sentiment vector with probabilities that state $i$ happens. Then the $i$-th coordinate of $\mathbf{s}_{t-1}$ is:

$$(\mathbf{s}_{t-1})_i = P(S_t = i | \mathbf{q}_{t-1}) = \frac{\exp\left((\mathbf{q}_{t-1})_i\right)}{1 + \sum_{j=1}^{n_s} \exp\left((\mathbf{q}_{t-1})_j\right)}, \; i = 1, \ldots, n_s \tag{4.11}$$

which is the multinomial logit model.

We define the generic equation eq. (4.4) as the eq. (4.11). So the model parameters are $\boldsymbol{\theta}_s = (\mathbf{W}_x, \mathbf{W}_d, \mathbf{b}_x, \mathbf{b}_d)$. And, lastly, we define $f_y$ as:

$$f_y(\mathbf{s}_{t-1}; \boldsymbol{\theta}_y) = \alpha_0 + \boldsymbol{\alpha}^T \mathbf{s}_{t-1} \tag{4.12}$$

so the parameters are $\boldsymbol{\theta}_y = (\alpha_0, \boldsymbol{\alpha}) \in \mathbb{R}^{n_s+1}$.

## 4.4 Analysing the stock market using learned sentiments

We have two distinct data sources, one related to quantitative variables and the other related to documents. The stock market quantitative variables were obtained using the Economatica[7] database. All variables used in this article are detailed in Section 4.6.1 (Variable Definitions). Documents are news from online newspapers *Valor Econômico* and *Folha de S.Paulo*, from 1st January 2012 to 2nd July 2018. The preparation procedure is described in section Section 2.6 (Data). We used the database without stopwords and with stemming.

Our study does not use any predefined word lists provided by dictionary categories since sentiments are learned from the data. We will use the analyst model to generate sentiments of size

[7] Economatica. Available at <http://economatica.com/>.

$n_s = 2$. Sentiment features cannot be directly mapped to sentiments in the real world. But if we choose the sentiment dimension $n_s = 2$, hopefully the results will provide two opposite sentiment features, such as pessimism and optimism. The variables of $\mathbf{z}_t^x$ in eq. (4.4) are the intraday return $r_t^i$ and the $\beta_t$ coefficient with 5 time period lags, $\mathcal{L}^5(r_t^i)$ and $\mathcal{L}^5(\beta_t)$, respectively. Our target variable $y_t$ is the overnight return $r_t^o$. The minimization problem, eq. (4.5), was solved using the Adam optimizer, proposed by Kingma and Ba (2014), with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$, the default values suggested by the algorithm's authors, and learning rate $\alpha = 0.001$. The code was implemented using the Python[8] programming language. The library used to solve the minimization problem was Tensorflow[9].

This work is divided in two distinct parts. The first varies some parameters of the analyst model so that the robustness of the results are tested. Document are one of many inputs of the analyst model. So we evaluate what is the appropriate document representation, or, specifically, the number of components $k \in \{256, 1024\}$ and the tuning parameter $p$ of the singular value decomposition for the approximation of the term-document matrix. Considering the results of Chapter 2 (Document representations and information measurements in time series), $(k, p) = (1024, -0.5)$ is able to retain more information of the original term-document matrix, which leads to better retrieving power and more statistically significant results, than $(k, p) = (256, -0.75)$. We also study the influence of the probability of the analyst to discard information, that is, the parameter $\zeta$ corresponding to the probability to set values in $\mathbf{W}_d$ to zero in eq. (4.7) during training.

The second part focus on other stock market regressions, which uses a vector autoregressive (VAR) framework to estimate relationships between the learned sentiments and returns, volume and volatility. The independent variables of our regressions always contains daily returns and sentiments with lags, $\mathcal{L}^\tau(r_t)$ and $\mathcal{L}^\tau((\mathbf{s}_t)_i)$, and other control variables, $\mathbf{x}_t$, detailed in Table 4.1. $(\mathbf{s}_t)_i$ is the $i$-th component of the $n_s$ sized sentiment vector $\mathbf{s}_t$. We have three sets of control variables: set (A) just use dummy variables for days of the week; set (B) use the daily return squared with lags and dummies for days of the week, which we took from García (2013); nd set (C), we use the trading volume ($vlm_t$) and volatility ($vol_t$) with lags, and dummies for days of the week and January, as in Tetlock (2007). We use $\tau = 5$ lags for our regressions.

**Table 4.1. Set of variables used in regressions**. All variables used in this article are detailed in Section 4.6.1 (Variable Definitions). We use $\tau = 5$ lags in regressions.

| Set | Variables $(\mathbf{x}_t)$ | Description |
|---|---|---|
| A | $\mathbf{dum}_{week}$ | Dummies for days of the week |
| B | $\mathcal{L}^\tau(r_t^2)$, $\mathbf{dum}_{t-1}^{week}$ | Daily return squared ($r_t^2$), dummies for days of the week |
| C | $\mathcal{L}^\tau(vlm_t)$, $\mathcal{L}^\tau(vol_t)$, $\mathbf{dum}_{t-1}^{week}$, $dum_{t-1}^{jan}$ | Trading volume ($vlm_t$), volatility ($vol_t$), dummies for days of the week and January |

We now study the size and quality of the document representation (parameters $k$ and $p$ of the singular value decomposition) and the influence of the probability of the analyst to discard

[8] Python. Available at <http://www.python.org/>.

[9] Tensorflow, an open source machine learning framework for everyone. Available at <http://www.tensorflow.org/>.

information, that is, the parameter $\zeta$ corresponding to the probability to set values in $\mathbf{W}_d$ to zero in eq. (4.7) during training. We use the document representation TF-IDF-SVD with parameters $\{(k,p)\} \in \{(64,-1),(1024,-0.5)\}$ and probabilities $\zeta \in \{0.0,0.1,0.2,0.3,0.4,0.5,0.6\}$. We simulate the analyst model with these parameters for a sentiment vector of size $n_s = 2$, calculate the sentiment vector $\mathbf{s}_t$ and then regress:

$$r_t = \alpha_{10} + \boldsymbol{\alpha}_1 \cdot \mathcal{L}^5((\mathbf{s}_t)_i) + \boldsymbol{\beta}_1 \cdot \mathcal{L}^5(r_t) + \boldsymbol{\gamma}_1 \cdot \mathbf{x}_t + \varepsilon_{1t} \tag{4.13}$$

with the set of parameters (C). For notation clarity, $(\boldsymbol{\alpha}_1^j)$ is coefficient of $(\mathbf{s}_{t-j})_i$, or the $i$-th component of sentiment $\mathbf{s}_{t-j}$. The dependent variable is the stock return because we want to study the effect of the parameters over the target variable in eq. (4.5), since it is more likely to be directly related to the sentiment vector. Table 4.2 plots the first component of $\boldsymbol{\alpha}_1$, the coefficient of $(\mathbf{s}_{t-1})_i$, in eq. (4.13). Since the coefficient of $(\mathbf{s}_{t-1})_1$ ($(\mathbf{s}_{t-1})_2$) is positive (negative), it has a positive (negative) effect on the stock return; hence it will be referred to optimism (pessimism) sentiment. The coefficients of the pessimism sentiment are greater than the coefficients of optimism sentiment in absolute values, which means that the negativity somehow is higher than the positiveness. This is consistent with the literature in psychology, Baumeister et al. (2001) and Rozin and Royzman (2001) show the higher impact of negative information than positive one.

We notice two effects caused by the number of components $k$ of the document representation. First, the representation with $(k,p) = (1024,-0.5)$ captures a higher effect of the sentiment (the absolute value of the coefficient of $(\mathbf{s}_{t-1})_i$ and has more statistically significant coefficients than the representation with $(k,p) = (64,-1)$. Probably because an approximation of the term-document matrix using fewer components is unable to capture the whole document essence. Second, the sentiment tends to zero as the probability of the analyst to discard information ($\zeta$) increases. As the analyst discards more information, his or her perception captured by the sentiment diminishes. We find that $\zeta = 0.1$ is the probability value that maximizes the positive and negative sentiment effects because its absolute value achieves the highest value and the regression results in more statistically significant coefficients at 10% level for the sentiments. This suggests that it is beneficial for the analyst to discard information in order to have a more balanced sentiment, reducing the bias toward a certain sentiment.

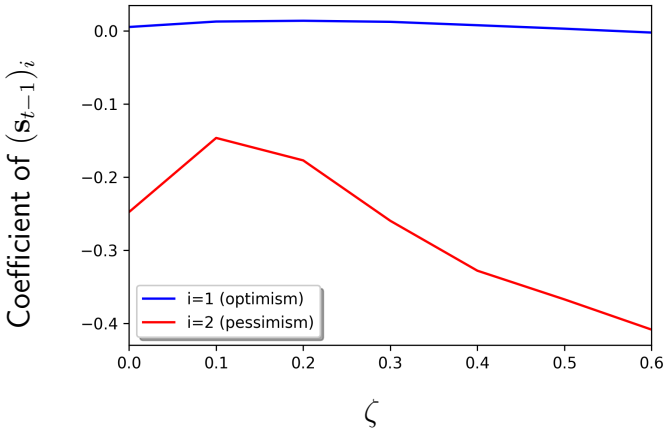
From now on, our basic representation will be the TF-IDF-SVD with $k = 1024$ components and tuning parameter $p = -0.5$. And the analyst model will use the probability of the analyst to discard information $\zeta = 0.1$.

Now we regress the stock return, eq. (4.13), and show the results in Table 4.3. The optimism and pessimism sentiments are individually statistically significant at least at 1% level through lags $t-1$ and $t-2$, and mostly significant at 10% level through lags $t-3$ and $t-4$. The joint hypothesis $H_0 : \boldsymbol{\alpha}_1^1 = \cdots = \boldsymbol{\alpha}_1^5 = 0$ is strongly rejected in all regressions ($p$-values approximately zero for the optimism and pessimism sentiments), which implies that the sentiment factors are associated in someway with future returns. So optimism (pessimism) exerts statistically and economically significant positive (negative) influence on future returns, and the effect of pessimism is approximately $|(\mathbf{s}_{t-1})_2|/|(\mathbf{s}_{t-1})_1| = 6.81 \pm 0.42$ times higher than optimism.

Sentiment so far studied is extracted from news. Public information may cause investors to re-evaluate stock prices. If news does not contain any fundamentals, there is no reason for the expected

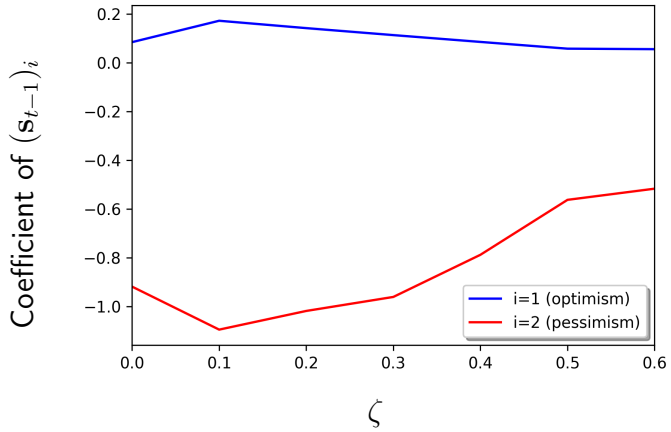
**Table 4.2. Sentiment coefficients versus the probability of the analyst discard information**. We try to evaluate how the probability of the analyst to discard information ($\zeta$) impacts the regression coefficient of $(\mathbf{s}_{t-1})_i$, eq. (4.13), using the set of parameters (C). We use the document representation TF-IDF-SVD with parameters $\{(k,p)\} \in \{(64,-1),(1024,-0.5)\}$ and sentiment vector of size $n_s = 2$, and analyse the variation of coefficients $(\mathbf{s}_{t-1})_1$ and $(\mathbf{s}_{t-1})_2$, referred to optimism and pessimism, respectively. Figures on the left of each Panel plot $(\mathbf{s}_{t-1})_i$ against $\zeta$. Tables on the right specify the values used in the graphic, where $(\mathbf{s}_{t-1})_i$ is the coefficient value and $\#ss$ is the number of statistically significant coefficients at 1% of $(\mathbf{s}_{t-1})_i$ to $(\mathbf{s}_{t-5})_i$.

**Panel A: TF-IDF-SVD with $(k,p) = (64,-1)$**




| $\zeta$ | optimism ($i=1$) | | pessimism ($i=2$) | |
|---|---|---|---|---|
| | $(\mathbf{s}_{t-1})_i$ | $\#ss$ | $(\mathbf{s}_{t-1})_i$ | $\#ss$ |
| 0.0 | 0.0051 | 1 | -0.2478 | 3 |
| 0.1 | 0.0126 | 2 | -0.1466 | 2 |
| 0.2 | 0.0136 | 3 | -0.1773 | 2 |
| 0.3 | 0.0122 | 2 | -0.2599 | 1 |
| 0.4 | 0.0075 | 2 | -0.3281 | 1 |
| 0.5 | 0.0028 | 2 | -0.3673 | 1 |
| 0.6 | -0.0025 | 2 | -0.4087 | 1 |

**Panel B: TF-IDF-SVD with $(k,p) = (1024,-0.5)$)**




| $\zeta$ | optimism ($i=1$) | | pessimism ($i=2$) | |
|---|---|---|---|---|
| | $(\mathbf{s}_{t-1})_i$ | $\#ss$ | $(\mathbf{s}_{t-1})_i$ | $\#ss$ |
| 0.0 | 0.0838 | 3 | -0.9191 | 5 |
| 0.1 | 0.1715 | 4 | -1.0954 | 3 |
| 0.2 | 0.1413 | 3 | -1.0188 | 2 |
| 0.3 | 0.1127 | 3 | -0.9614 | 2 |
| 0.4 | 0.0847 | 2 | -0.7885 | 2 |
| 0.5 | 0.0570 | 1 | -0.5631 | 3 |
| 0.6 | 0.0550 | 2 | -0.5173 | 3 |

return on the stock market to change. If noninformational traders sell stocks for exogenous reasons, market makers (risk-averse utility maximizers) are willing to accommodate the selling pressure, so price changes will tend to be reversed. This suggests that the market perception, captured through sentiments, should be reversed as well. The hypothesis of no reversal, $H_0 : \sum_{j=2}^{5} \boldsymbol{\alpha}_1^j = 0$, is strongly rejected for both optimism and pessimism sentiments, so the magnitude of the reversal in lags two through five is significantly different from zero at the 1% level. We also reject the null $H_0 : \sum_{j=1}^{5} \boldsymbol{\alpha}_1^j = 0$, that is, the reversal of lags 1 to 5 does not surpass the initial effect of optimism (pessimism) because the sum of coefficients on the five lags is statistically different from zero.

**Table 4.3. Predicting stock returns using a sentiment vector of size $n_s = 2$.** The document representation used was the approximation of the term-document matrix, weights specified in eq. (4.3), using 1024 components of the singular value decomposition and tuning parameter $p = -0.5$. The probability of the analyst to discard information is $\zeta = 0.1$. The analyst model considers sentiments of size $n_s = 2$ with lags of 5 time periods. The dependent variable in each regression, given by eq. (4.13), is the ajusted stock return $r_t$. We omit the coefficients of dummy variables. One, two or three asterisks means that the p-value of the $H_0 : \boldsymbol{\alpha}_1^j = 0$ is inferior to 0.10, 0.05 and 0.01, respectively. The definition of all variables used in the regressions are in Section 4.6.1 (Variable Definitions). The estimates use a 44,969 sample over 2012 to 2018.

| **Explanatory variable** | Optimism $(i = 1)$ | | | Pessimism $(i = 2)$ | | |
|---|---|---|---|---|---|---|
| | (A) | (B) | (C) | (A) | (B) | (C) |
| const | -0.0138*** | -0.0124*** | -0.0172*** | 0.0950*** | 0.1026*** | 0.0969*** |
| $r_{t-1}$ | 0.0480*** | 0.0246*** | 0.0568*** | 0.0395*** | 0.0075 | 0.0497*** |
| $r_{t-2}$ | 0.0596*** | 0.0567*** | 0.0621*** | -0.0190*** | -0.0226*** | -0.0236*** |
| $r_{t-3}$ | 0.0369*** | 0.0335*** | 0.0304*** | -0.0029 | -0.0049 | -0.0144*** |
| $r_{t-4}$ | -0.0266*** | -0.0261*** | -0.0211*** | -0.0022 | -0.0036 | -0.0005 |
| $r_{t-5}$ | 0.0313*** | 0.0295*** | 0.0390*** | 0.0065 | 0.0040 | 0.0049 |
| $(\mathbf{s}_{t-1})_i$ | 0.1627*** | 0.1545*** | 0.1716*** | -1.1097*** | -1.1155*** | -1.0954*** |
| $(\mathbf{s}_{t-2})_i$ | -0.0333*** | -0.0365*** | -0.0297*** | 0.0219 | -0.0275 | 0.0239 |
| $(\mathbf{s}_{t-3})_i$ | -0.0317*** | -0.0284*** | -0.0233*** | -0.0229 | -0.0272 | -0.0351* |
| $(\mathbf{s}_{t-4})_i$ | -0.0254*** | -0.0261*** | -0.0152*** | 0.1753*** | 0.1609*** | 0.1559*** |
| $(\mathbf{s}_{t-5})_i$ | -0.0061 | -0.0060 | 0.0002 | 0.0620*** | 0.0628*** | 0.0560*** |
| $r_{t-1}^2$ | | 0.3821*** | | | 0.4612*** | |
| $r_{t-2}^2$ | | -0.1050*** | | | -0.0372* | |
| $r_{t-3}^2$ | | 0.0023 | | | 0.0078 | |
| $r_{t-4}^2$ | | -0.0199 | | | -0.0138 | |
| $r_{t-5}^2$ | | -0.0243 | | | 0.0030 | |
| $vlm_{t-1}$ | | | 0.00004 | | | 0.0003 |
| $vlm_{t-2}$ | | | -0.0001 | | | 0.000001 |
| $vlm_{t-3}$ | | | 0.0013*** | | | 0.0011*** |
| $vlm_{t-4}$ | | | -0.0004 | | | -0.0003 |
| $vlm_{t-5}$ | | | 0.0001 | | | 0.0001 |
| $vol_{t-1}$ | | | -0.1571*** | | | -0.0258 |
| $vol_{t-2}$ | | | 0.1735*** | | | 0.1013** |
| $vol_{t-3}$ | | | -0.0759 | | | -0.0968* |
| $vol_{t-4}$ | | | -0.0015 | | | 0.0656 |
| $vol_{t-5}$ | | | -0.0779** | | | -0.0254 |
| $R^2$ | 0.0239 | 0.0312 | 0.0277 | 0.0658 | 0.0763 | 0.0701 |
| $F$-statistic | 78.67 | 76.23 | 47.82 | 226.14 | 195.29 | 126.43 |
| $p$-value | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $H_0 : \boldsymbol{\alpha}_1^1 = \cdots = \boldsymbol{\alpha}_1^5 = 0$ | | | | | | |
| $F$-statistic | 175.53 | 154.53 | 174.43 | 586.45 | 600.37 | 564.50 |
| $p$-value | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $H_0 : \sum_{j=2}^5 \boldsymbol{\alpha}_1^j = 0$ | | | | | | |
| $\sum_{j=2}^5 \boldsymbol{\alpha}_1^j$ | -0.0965 | -0.0969 | -0.0679 | 0.2362 | 0.1689 | 0.2007 |
| $F$-statistic | 218.38 | 214.13 | 85.21 | 37.11 | 18.94 | 25.93 |
| $H_0 : \sum_{j=1}^5 \boldsymbol{\alpha}_1^j = 0$ | | | | | | |
| $\sum_{j=1}^5 \boldsymbol{\alpha}_1^j$ | 0.0661 | 0.0575 | 0.1036 | -0.8733 | -0.9465 | -0.8947 |
| $F$-statistic | 145.52 | 99.82 | 212.28 | 430.51 | 504.84 | 438.84 |

Optimism (pessimism) statistically influences stock returns, so not all information is already incorporated into prices. There is an initial increase (decrease) in returns due to optimism (pessimism), and this is not followed by a complete return reversal. The upward (downward) price pressure by the optimistic (pessimistic) investor sentiment is not temporary, which could be signaling a change investors risk aversion. So, indeed, the news contains some information about fundamentals. These results are different from Tetlock (2007), who finds that the price pressure is only temporary because the return regression in his study suggest a complete statistically significant reversal.

We hypothesize two explanations for this divergence in results. First, a possible inadequacy in representing documents. Regarding our model, we could increase the number of components so that we could produce a regression with all five lags statistically significant, and then test the hypothesis. On the other hand, Tetlock (2007) generated a sentiment measurement based on a document representation with two not recommended approaches for the weights: use of raw frequencies only; and normalization by the total of terms in the document. Weights that use the factor related to the inverse of document frequency and normalization by the vector norm have superior results in information retrieval, therefore in capturing information from documents. Probably this explains why the author only captured two coefficients statistically significant at 5% level in the same regression. Either way, the choice of the document representation is subjective and hard to evaluate. Second, assuming that news has the power to change investor risk aversion, the elasticity of the Brazilian stock market is different from the U.S. stock market. Since Brazil is much more dependent on foreign investors, the news could encourage or scare these investors so that the effects of optimism or pessimism are more permanent or need more time to dissipate.

Alternatively, we regress returns on the sentiment change $\Delta(\mathbf{s}_{t-j})_i = (\mathbf{s}_{t-j})_i - (\mathbf{s}_{t-j-1})_i$ instead of the sentiment in eq. (4.13). Results are in Table 4.4. All coefficients related to changes in sentiment are statistically significant at 1% level. Coefficients of changes in optimism are all positive while coefficients of changes in pessimism are all negative. The effect of changes in pessimism in $t-1$ is approximately $|\Delta(\mathbf{s}_{t-1})_2|/|\Delta(\mathbf{s}_{t-1})_1| = 6.58 \pm 0.13$ times higher than changes in optimism. The effect of changes in sentiment slowly dissipates, that is, coefficients of $|\Delta(\mathbf{s}_{t-1})_i|$ is greater than $|\Delta(\mathbf{s}_{t-2})_i|$, and so on. And since we strongly reject the hypothesis $H_0 : |\Delta(\mathbf{s}_{t-5})_i| = 0$, we still observe the effects of changes in sentiments after 5 lags.

Now we shift our analysis to other market variables, volume and volatility, which are closely related. Daigler and Wiley (1999) find that there exists a positive volatility-volume relation driven by the general public[10], and that clearing members and floor traders often decrease volatility. Uninformed traders who cannot differentiate liquidity demand from fundamental value change increase volatility. Coval and Shumway (2001) use the noise level as a proxy for the degree of anxiety, and authors defend that the eagerness to trade immediately may be most pronounced when traders have current positions that are costly to maintain. To study these hypotheses, we regress:

$$vlm_t = \alpha_{20} + \boldsymbol{\alpha}_2 \cdot \mathcal{L}^5(\Delta(\mathbf{s}_t)_i) + \boldsymbol{\beta}_2 \cdot \mathcal{L}^5(r_t) + \boldsymbol{\gamma}_2 \cdot \mathbf{x}_t + \varepsilon_{2t} \tag{4.14}$$

[10] We refer as general public as a group of traders who are distant from the trading floor and therefore without precise information on order floor.

**Table 4.4. Predicting stock returns using the change in the sentiment vector of size $n_s = 2$.** The document representation used was the approximation of the term-document matrix, weights specified in eq. (4.3), using 1024 components of the singular value decomposition and tuning parameter $p = -0.5$. The probability of the analyst to discard information is $\zeta = 0.1$. The analyst model considers sentiments of size $n_s = 2$ with lags of 5 time periods. The dependent variable in each regression, given by eq. (4.13), is the ajusted stock return $r_t$; we use the sentiment change $\Delta(\mathbf{s}_{t-j})_i = (\mathbf{s}_{t-j})_i - (\mathbf{s}_{t-j-1})_i$ instead of the sentiment. We omit the coefficients of dummy variables. One, two or three asterisks means that the p-value of the $H_0 : \boldsymbol{\alpha}_1^j = 0$ is inferior to 0.10, 0.05 and 0.01, respectively. The definition of all variables used in the regressions are in Section 4.6.1 (Variable Definitions). The estimates use a 44,969 sample over 2012 to 2018.

| **Explanatory variable** | Optimism $(i = 1)$ | | | Pessimism $(i = 2)$ | | |
|---|---|---|---|---|---|---|
| | (A) | (B) | (C) | (A) | (B) | (C) |
| const | -0.0008** | -0.0012*** | -0.0008 | -0.0011*** | -0.0014*** | -0.0009* |
| $r_{t-1}$ | 0.0516*** | 0.0261*** | 0.0624*** | 0.0480*** | 0.0185*** | 0.0591*** |
| $r_{t-2}$ | 0.0562*** | 0.0529*** | 0.0549*** | -0.0076* | -0.0094** | -0.0099** |
| $r_{t-3}$ | 0.0307*** | 0.0271*** | 0.0178*** | 0.0087* | 0.0085* | -0.0008 |
| $r_{t-4}$ | -0.0295*** | -0.0296*** | -0.0278*** | 0.0085* | 0.0086* | 0.0120** |
| $r_{t-5}$ | 0.0277*** | 0.0260*** | 0.0283*** | 0.0189*** | 0.0180*** | 0.0186*** |
| $\Delta(\mathbf{s}_{t-1})_i$ | 0.1486*** | 0.1426*** | 0.1453*** | -0.9637*** | -0.9583*** | -0.9475*** |
| $\Delta(\mathbf{s}_{t-2})_i$ | 0.1045*** | 0.0965*** | 0.0972*** | -0.7998*** | -0.8307*** | -0.7781*** |
| $\Delta(\mathbf{s}_{t-3})_i$ | 0.0670*** | 0.0637*** | 0.0607*** | -0.6786*** | -0.7015*** | -0.6625*** |
| $\Delta(\mathbf{s}_{t-4})_i$ | 0.0358*** | 0.0334*** | 0.0339*** | -0.3610*** | -0.3857*** | -0.3578*** |
| $\Delta(\mathbf{s}_{t-5})_i$ | 0.0266*** | 0.0263*** | 0.0240*** | -0.1629*** | -0.1758*** | -0.1596*** |
| $r_{t-1}^2$ | | 0.4069*** | | | 0.4348*** | |
| $r_{t-2}^2$ | | -0.0800*** | | | -0.0598*** | |
| $r_{t-3}^2$ | | 0.0248 | | | -0.0084 | |
| $r_{t-4}^2$ | | -0.0005 | | | -0.0224 | |
| $r_{t-5}^2$ | | 0.0050 | | | -0.0054 | |
| $vlm_{t-1}$ | | | 0.0001 | | | 0.0002 |
| $vlm_{t-2}$ | | | -0.0002 | | | 0.0001 |
| $vlm_{t-3}$ | | | 0.0012*** | | | 0.0012*** |
| $vlm_{t-4}$ | | | -0.0005 | | | -0.0003 |
| $vlm_{t-5}$ | | | -0.0001 | | | 0.0001 |
| $vol_{t-1}$ | | | -0.1117*** | | | -0.0736** |
| $vol_{t-2}$ | | | 0.1999*** | | | 0.0868* |
| $vol_{t-3}$ | | | -0.0628 | | | -0.0954* |
| $vol_{t-4}$ | | | 0.0283 | | | 0.0678 |
| $vol_{t-5}$ | | | -0.0540 | | | 0.0082 |
| N | 44,911 | 44,911 | 41,896 | 44,911 | 44,911 | 41,896 |
| $R^2$ | 0.0215 | 0.0297 | 0.0234 | 0.0583 | 0.0676 | 0.0619 |
| $F$-statistic | 70.30 | 72.44 | 40.09 | 198.71 | 171.36 | 110.48 |
| $p$-value | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $H_0 : \boldsymbol{\alpha}_1^1 = \cdots = \boldsymbol{\alpha}_1^5 = 0$ | | | | | | |
| $F$-statistic | 152.24 | 140.44 | 136.11 | 510.05 | 510.85 | 485.43 |
| $p$-value | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

$$vol_t = \alpha_{30} + \boldsymbol{\alpha}_3 \cdot \mathcal{L}^5((\mathbf{s}_t)_i) + \boldsymbol{\beta}_3 \cdot \mathcal{L}^5(r_t) + \boldsymbol{\gamma}_3 \cdot \mathbf{x}_t + \varepsilon_{3t} \tag{4.15}$$

where $\Delta(\mathbf{s}_{t-j})_i = (\mathbf{s}_{t-j})_i - (\mathbf{s}_{t-j-1})_i$ is the change of sentiment $i$ from $t-j-1$ to $t-j$.

Table 4.5 and Table 4.6 show the coefficients of regressions in eqs. (4.14) and (4.15), which estimates the impact of sentiment on trading volume and volatility, respectively. We analyse both regressions simultaneously. We estimate not only the effect of pessimism, but also of optimism. There is evidence that both optimism and pessimism are related to volume and volatility because coefficients $\boldsymbol{\alpha}_2$ and $\boldsymbol{\alpha}_3$ are jointly significant at least at 1% level. We observe that the coefficients of pessimism are usually greater than its corresponding optimism coefficient in absolute value, i.e. $|\Delta(\mathbf{s}_{t-j})_i|$ is usually greater for pessimism for $j \in \{1, \ldots, 5\}$ and for each set of control variables.

The coefficient of pessimism in $t-1$, $\Delta(\mathbf{s}_{t-1})_2$, is positive and statistically significant at 1% level for the volume regression, suggesting that pessimism is as a proxy for trading costs. However, pessimism in $t-1$ is negative and statistically significant at least at 5% level for the volatility regression, suggesting that traders are informed. Trades in B3 are mostly from foreign investors, institutional investors and financial institutions, not the general public[11], supposedly informed investors who have relatively homogeneous beliefs. Since they have knowledge of the market and the fundamental characteristics of assets, they buy and sell within a small range of prices around the fair value of the asset.

As for optimism, since the coefficient of $(\mathbf{s}_{t-1})_1$ is positive and statistically significant at 1% level, optimism is also a proxy for trading costs and stimulates traders to buy. But since optimism is positive and significant at least at 10% level in the volatility regression, noise traders start to dominate the market and bring excess volatility. Probably because good news in Brazil are difficult to interpret or trust when new "information" is revealed, resulting in a wider dispersion of beliefs. To clarify, we can think of this as positive-feedback trading. According to Long J Bradford (1990), if rational speculators early buying triggers positive-feedback trading, then an increase in the number of forward-looking speculators can increase volatility about fundamentals. Coefficients of the optimism sentiment in $t-1$ are positive and statistically significant at 1% level. So good news, or optimism, cause rational speculators to trade and stimulate buying by positive feedback traders tomorrow, hence increasing volatility.

Lastly, we measured the volatility using the GARCH model. Perhaps if we used another volatility, such as the implicit one, we might have observed a different outcome.

## 4.5 Conclusions

We hypothesized a model inspired in a market analyst, using simultaneously quantitative variables and text, to create a vectorized sentiment with meaningful coordinates. Sentiment was generated without the use of any pre-defined dictionary.

Not all information is incorporated in prices because regressing stock returns on pessimism (optimism) was statistically significant. This negative (positive) influence was followed by an incomplete

[11] Individual investors were responsible for approximately 9% of the trades in 2019, according to the B3 official site at <http://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/consultas/mercado-a-vista/participacao-dos-investidores/volume-total/>.

**Table 4.5. Predicting volume using stock returns**. The document representation used was the approximation of the term-document matrix, weights specified in eq. (4.3), using 1024 components of the singular value decomposition and tuning parameter $p = -0.5$. The probability of the analyst to discard information is $\zeta = 0.1$. The analyst model considers sentiments of size $n_s = 2$ with lags of 5 time periods. The dependent variable in each regression, given by eq. (4.14), is the volume $vlm_t$. We only show the coefficients $\boldsymbol{\alpha}_2$ of $\mathcal{L}^5((\mathbf{s}_t)_i)$. One, two or three asterisks means that the p-value of the $H_0 : \boldsymbol{\alpha}_1^j = 0$ is inferior to 0.10, 0.05 and 0.01, respectively. The definition of all variables used in the regressions are in Section 4.6.1 (Variable Definitions). The estimates over 40 thousand samples over 2012 to 2018.

| **Explanatory variable** | Optimism $(i = 1)$ | | | Pessimism $(i = 2)$ | | |
|---|---|---|---|---|---|---|
| | (A) | (B) | (C) | (A) | (B) | (C) |
| const | 0.0311*** | 0.0137** | 0.0664*** | 0.0305*** | 0.0123** | 0.0651*** |
| $r_{t-1}$ | 1.7091*** | 1.1202*** | 1.0436*** | 1.9235*** | 1.2655*** | 1.1156*** |
| $r_{t-2}$ | 1.8874*** | 1.6852*** | 0.7043*** | 1.3750*** | 1.2043*** | 0.4082*** |
| $r_{t-3}$ | 1.6755*** | 1.4851*** | 0.4847*** | 1.0790*** | 0.9629*** | 0.2097*** |
| $r_{t-4}$ | 1.1726*** | 1.0372*** | 0.3318*** | 1.0470*** | 0.9410*** | 0.3296*** |
| $r_{t-5}$ | 1.3405*** | 1.2343*** | 0.2671*** | 0.9223*** | 0.8581*** | 0.0669 |
| $\Delta(\mathbf{s}_{t-1})_i$ | 1.7209*** | 1.5539*** | 0.8881*** | 1.9003*** | 2.1266*** | 2.5636*** |
| $\Delta(\mathbf{s}_{t-2})_i$ | 1.7224*** | 1.4407*** | 0.5920*** | 3.0725*** | 2.6172*** | 2.4085*** |
| $\Delta(\mathbf{s}_{t-3})_i$ | 1.7851*** | 1.5384*** | 0.6556*** | 4.0026*** | 3.5754*** | 2.5901*** |
| $\Delta(\mathbf{s}_{t-4})_i$ | 1.3117*** | 1.1307*** | 0.4239*** | 3.2571*** | 2.7442*** | 1.5495*** |
| $\Delta(\mathbf{s}_{t-5})_i$ | 0.7887*** | 0.7105*** | 0.3177*** | 1.7672*** | 1.4255*** | 0.8460*** |
| $r_{t-1}^2$ | | 9.4144*** | | | 9.7422*** | |
| $r_{t-2}^2$ | | 2.6350*** | | | 3.0417*** | |
| $r_{t-3}^2$ | | 2.0397*** | | | 2.3184*** | |
| $r_{t-4}^2$ | | 1.1727*** | | | 1.1324*** | |
| $r_{t-5}^2$ | | 0.4059 | | | 0.2353 | |
| $vlm_{t-1}$ | | | 0.3922*** | | | 0.3948*** |
| $vlm_{t-2}$ | | | 0.1077*** | | | 0.1074*** |
| $vlm_{t-3}$ | | | 0.0644*** | | | 0.0666*** |
| $vlm_{t-4}$ | | | 0.0522*** | | | 0.0520*** |
| $vlm_{t-5}$ | | | 0.0670*** | | | 0.0672*** |
| $vol_{t-1}$ | | | 0.2427 | | | 0.5569 |
| $vol_{t-2}$ | | | -1.6958** | | | -1.5820** |
| $vol_{t-3}$ | | | -0.0921 | | | -0.2171 |
| $vol_{t-4}$ | | | 0.3456 | | | 0.3251 |
| $vol_{t-5}$ | | | 0.6998 | | | 0.4526 |
| N | 44,911 | 44,911 | 41,896 | 44,911 | 44,911 | 41,896 |
| $R^2$ | 0.0588 | 0.0791 | 0.3337 | 0.0499 | 0.0724 | 0.3329 |
| $F$-statistic | 200.34 | 203.06 | 838.85 | 168.43 | 184.51 | 835.93 |
| $p$-value | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $H_0 : \boldsymbol{\alpha}_2^1 = \cdots = \boldsymbol{\alpha}_2^5 = 0$ | | | | | | |
| $F$-statistic | 103.63 | 81.87 | 28.81 | 18.58 | 16.36 | 19.05 |
| $p$-value | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

**Table 4.6. Predicting volatility using stock returns**. The document representation used was the approximation of the term-document matrix, weights specified in eq. (4.3), using 1024 components of the singular value decomposition and tuning parameter $p = -0.5$. The probability of the analyst to discard information is $\zeta = 0.1$. The analyst model considers sentiments of size $n_s = 2$ with lags of 5 time periods. The dependent variable in each regression, given by eq. (4.15), is the volatility $vol_t$. We only show the coefficients $\boldsymbol{\alpha}_3$ of $\mathcal{L}^5((\mathbf{s}_t)_i)$. One, two or three asterisks means that the p-value of the $H_0 : \boldsymbol{\alpha}_1^j = 0$ is inferior to 0.10, 0.05 and 0.01, respectively. The definition of all variables used in the regressions are in Section 4.6.1 (Variable Definitions). The estimates use 41,896 samples over 2012 to 2018.

| **Explanatory variable** | Optimism $(i = 1)$ | | | Pessimism $(i = 2)$ | | |
|---|---|---|---|---|---|---|
| | (A) | (B) | (C) | (A) | (B) | (C) |
| const | 0.0286*** | 0.0260*** | 0.0012*** | 0.0286*** | 0.0260*** | 0.0011*** |
| $r_{t-1}$ | 0.0285*** | -0.0103*** | 0.0202*** | 0.0306*** | -0.0104*** | 0.0227*** |
| $r_{t-2}$ | 0.0179*** | 0.0014 | -0.0032*** | 0.0147*** | 0.0002 | -0.0035*** |
| $r_{t-3}$ | 0.0189*** | -0.0014 | -0.0020*** | 0.0130*** | -0.0024 | -0.0035*** |
| $r_{t-4}$ | 0.0148*** | -0.0025 | -0.0016** | 0.0116*** | -0.0020 | -0.0025*** |
| $r_{t-5}$ | 0.0143*** | 0.0004 | -0.0014** | 0.0100*** | 0.0001 | -0.0018*** |
| $\Delta(\mathbf{s}_{t-1})_i$ | 0.0118*** | 0.0033* | 0.0017** | -0.0446*** | -0.0154** | -0.0094*** |
| $\Delta(\mathbf{s}_{t-2})_i$ | 0.0216*** | 0.0037* | 0.0058*** | -0.0156 | -0.0153* | 0.0353*** |
| $\Delta(\mathbf{s}_{t-3})_i$ | 0.0230*** | 0.0024 | 0.0019** | -0.0066 | -0.0100 | 0.0276*** |
| $\Delta(\mathbf{s}_{t-4})_i$ | 0.0190*** | 0.0021 | 0.0013* | 0.0105 | -0.0077 | 0.0302*** |
| $\Delta(\mathbf{s}_{t-5})_i$ | 0.0108*** | 0.0005 | 0.0005 | 0.0138 | -0.0008 | 0.0172*** |
| $r_{t-1}^2$ | | 0.6274*** | | | 0.6285*** | |
| $r_{t-2}^2$ | | 0.4868*** | | | 0.4873*** | |
| $r_{t-3}^2$ | | 0.4263*** | | | 0.4263*** | |
| $r_{t-4}^2$ | | 0.3919*** | | | 0.3909*** | |
| $r_{t-5}^2$ | | 0.3289*** | | | 0.3285*** | |
| $vlm_{t-1}$ | | | 0.0026*** | | | 0.0026*** |
| $vlm_{t-2}$ | | | -0.0007*** | | | -0.0007*** |
| $vlm_{t-3}$ | | | -0.0003*** | | | -0.0003*** |
| $vlm_{t-4}$ | | | -0.0005*** | | | -0.0005*** |
| $vlm_{t-5}$ | | | -0.0005*** | | | -0.0005*** |
| $vol_{t-1}$ | | | 0.7215*** | | | 0.7228*** |
| $vol_{t-2}$ | | | 0.0903*** | | | 0.0896*** |
| $vol_{t-3}$ | | | 0.0732*** | | | 0.0755*** |
| $vol_{t-4}$ | | | 0.0446*** | | | 0.0453*** |
| $vol_{t-5}$ | | | 0.0438*** | | | 0.0406*** |
| N | 41,896 | 41,896 | 41,896 | 41,896 | 41,896 | 41,896 |
| $R^2$ | 0.0114 | 0.3694 | 0.9058 | 0.0101 | 0.3694 | 0.9063 |
| $F$-statistic | 34.63 | 1291.11 | 16099.55 | 30.57 | 1291.25 | 16200.43 |
| $p$-value | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $H_0 : \boldsymbol{\alpha}_3^1 = \cdots = \boldsymbol{\alpha}_3^5 = 0$ | | | | | | |
| $F$-statistic | 18.78 | 0.82 | 12.02 | 7.51 | 1.15 | 59.62 |
| $p$-value | 0.0000 | 0.5307 | 0.0000 | 0.0000 | 0.3289 | 0.0000 |

return reversal, signaling that the price pressure was not temporary. News have the power to influence investors risk aversion in B3 (Brazilian Stock Exchange). Since this result was different from Tetlock

(2007), we wonder if it is due to the model or it just captures an specific stock market phenomenon. A document representation can drastically change results in a regression, but its choice is somewhat subjective. It is also a fact that the stock market from Brazil and the U.S. are substantially different, so the Brazilian stock market can be more susceptible to the under- or over-reaction of foreign investors to news.

Sentiment also influences trading volume and act as a proxy for trading costs. We observe that increases in both optimism and pessimism changes of sentiment cause trading volume to increase, so sentiment captured by news stimulate traders to buy or sell. This phenomenon accompanies a decrease in volatility for pessimism, explained by the presence of informed traders who are mostly present in B3 Stock Exchange, and an increase in volatility, explained by dispersion of beliefs of news and positive feedback trading.

## 4.6 Appendix

### 4.6.1 Variable Definitions

| Variable | Description |
|---|---|
| $p_t^c$ | Ajusted closing price. |
| $p_t^o$ | Ajusted opening price. |
| $r_t$ | Stock daily return, or close-to-close return, calculated as $\log p_t^c - \log p_{t-1}^c$. |
| $r_t^i$ | Stock intraday return, calculated as $\log p_t^c - \log p_t^o$. |
| $r_t^o$ | Stock overnight return, calculated as $\log p_t^o - \log p_{t-1}^c$. |
| $vlm_t$ | Detrended log volume, which we use the rolling average of the past 60 days of log volume to detrend log of daily volume, the methodology based on Campbell, Grossman and Wang (1993). |
| $vol_t$ | Stock volatility, we fit a GARCH(1,1) model for each stock return, that is, we estimate a model with constant mean $r_t = \mu + \epsilon_t$, and time-varying volatility $\sigma_{t+1}^2 = \omega + \alpha_1 \epsilon_t^2 + \beta_1 \sigma_t^2$, where $\sigma_t^2 \equiv \text{var}(\epsilon_t)$. |
| $\beta_t$ | Beta coefficient, which is the slope of the regression of $r_t$ on $r_t^m$, where $r_t^m$ is the market returns (the ibovespa index) |
| $\mathbf{d}_t$ | Document vector represented by the approximation of the term-document matrix, weights given by eq. (4.3), using $k$ components of the singular value decomposition and tuning parameter $p$. |
| $\mathbf{s}_t$ | Sentiment vector of size $n_s$, where $\mathbf{s}_t = (s_{t-1}^1, \ldots, s_{t-1}^{n_s})$. |
| $\mathbf{dum}^{week}$ | Dummy variables corresponding to days of the week |
| $dum^{jan}$ | Dummy variable that indicates if the trading date is in January |

# Part III

# Final

# 5 Closure

## 5.1 Conclusions

This dissertation contributed to the application of text analysis in finance.

Chapter 2 (Document representations and information measurements in time series) focused on document representations and their measures. It brought closer the literature in finance and computer science by defining the problem of extracting sentiment from the text as an information retrieval problem. By doing so, it enabled many possibilities, such as representing documents in lower and dense spaces and the application of different similarity formulas. Since dense spaces work with concepts instead of actual words, it seems more appropriate to work with translated dictionaries. In fact, translated dictionaries could successfully find the effect of negativity in a stock market regression using the representation and measurements proposed.

The other chapters are a direct application of dense representations, which are low dimensional compared to the traditional sparse TF-IDF matrix. A dense representation has advantage of truncation of the least important information, instead of excluding most of the vocabulary like when using dictionaries.

Chapter 3 (Do prices absorb public information?) proposed to create a sentiment that learned the most important vocabulary to impact the future overnight return. The news associated to a stock negotiated in B3 (Brazilian Stock Exchange) is the aggregation of past news in which the stock or its company appears. Since news are low dimensional, we used them directly as time variables. We found that the overnight return in $t$ is affected mostly by documents in $t-1$, but the effect of other past news is not negligible. We found a statistically significant effect by regressing the learned vocabulary sentiment on the overnight return, and vice-versa. We also showed that the polarity of a word depends on the context; that is, if the sentiment is a perception of investors, a word can positively or negatively influence investors in different periods.

Chapter 4 (Modeling an analyst) built a sentiment vector using news and quantitative variables from the stock market. The sentiment vector with two dimensions produced two opposite sentiment measures, optimism and pessimism. Regressions of several stock market variables on these sentiments were statistically significant and found support on the Finance literature.

## 5.2 Bibliography

AGARWAL, V. Y. S. C. S.; ZHANG, W. The information value of credit rating action reports: A textual analysis. *Management Science*, v. 62, n. 8, p. 2218–2240, 2014. Available at: <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.2015.2243?journalCode=mnsc>. Cited on page 32.

ALLEE, K. D.; DEANGELIS, M. D. The structure of voluntary disclosure narratives: Evidence from tone dispersion. *Journal of Accounting Research*, v. 53, n. 2, p. 241–274, 2015. Available at: <http://onlinelibrary.wiley.com/doi/10.1111/1475-679X.12072/abstract>. Cited on page 32.

ALLEN, D. E.; MCALEER, M.; SINGH, A. K. *Daily Market News Sentiment and Stock Prices*. [S.l.], 2015. Available at: <https://ideas.repec.org/p/ucm/doicae/1511.html>. Cited on page 70.

ANTWEILER, W.; FRANK, M. Do us stock markets typically overreact to corporate news stories? 08 2006. Available at: <http://dx.doi.org/10.2139/ssrn.878091>. Cited 4 times on page(s) 52, 70, 90, and 92.

ANTWEILER, W.; FRANK, M. Z. Is all that talk just noise? the information content of internet stock message boards. *The Journal of Finance*, [American Finance Association, Wiley], v. 59, n. 3, p. 1259–1294, 2004. ISSN 00221082, 15406261. Available at: <http://www.jstor.org/stable/3694736>. Cited 3 times on page(s) 70, 90, and 94.

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval*. 2nd. ed. USA: Addison-Wesley Publishing Company, 2008. ISBN 9780321416919. Cited 7 times on page(s) 34, 36, 43, 44, 52, 72, and 92.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. Available at: <http://arxiv.org/abs/1409.0473>. Cited on page 40.

BATRINCA, B.; TRELEAVEN, P. C. Social media analytics: a survey of techniques, tools and platforms. *AI & SOCIETY*, v. 30, n. 1, p. 89–116, Feb 2015. ISSN 1435-5655. Available at: <https://doi.org/10.1007/s00146-014-0549-4>. Cited 2 times on page(s) 24 and 69.

BAUMEISTER, R.; BRATSLAVSKY, E.; FINKENAUER, C.; VOHS, K. Bad is stronger than good. *Review of General Psychology*, American Psychological Association, v. 5, n. 4, p. 323–370, 1 2001. ISSN 1089-2680. Cited 3 times on page(s) 24, 85, and 98.

BENGIO, Y.; DUCHARME, R.; VINCENT, P.; JANVIN, C. A neural probabilistic language model. *Journal of Machine Learning Research*, JMLR.org, v. 3, p. 1137–1155, mar. 2003. ISSN 1532-4435. Available at: <http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf>. Cited on page 39.

BLACK, F. Toward a fully automated stock exchange, part i. *Financial Analysts Journal*, v. 27, n. 4, p. 28–35, 1971. Available at: <https://doi.org/10.2469/faj.v27.n4.28>. Cited on page 70.

BROWN, S. V.; TUCKER, J. W. Large-sample evidence on firms' year-over-year md&a modifications. *Journal of Accounting Research*, v. 49, n. 2, p. 309–346, 2011. Available at: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-679X.2010.00396.x>. Cited on page 46.

CAMPBELL, J. Y.; GROSSMAN, S. J.; WANG, J. Trading volume and serial correlation in stock returns. *Quarterly Journal of Economics*, v. 108, n. 4, p. 905–939, 1993. Cited 4 times on page(s) 53, 76, 94, and 106.

CAMPBELL, J. Y.; LO, A. W.; MACKINLAY, A. C. *The Econometrics of Financial Markets*. [S.l.]: Princeton University Press, 1997. Cited on page 70.

CARON, J. Computational information retrieval. In: BERRY, M. W. (Ed.). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. cap. Experiments with LSA Scoring: Optimal Rank and Basis, p. 157–169. ISBN 0-89871-500-8. Available at: <http://dl.acm.org/citation.cfm?id=762544.762556>. Cited 2 times on page(s) 38 and 73.

CHO, K.; COURVILLE, A. C.; BENGIO, Y. Describing multimedia content using attention-based encoder-decoder networks. *CoRR*, abs/1507.01053, 2015. Available at: <http://arxiv.org/abs/1507.01053>. Cited on page 41.

CORREA, R.; GARUD, K.; LONDONO, J. M.; MISLANG, N. *Sentiment in Central Banks' Financial Stability Reports*. [S.l.], 2017. Available at: <https://ideas.repec.org/p/fip/fedgif/1203.html>. Cited 2 times on page(s) 26 and 38.

COVAL, J. D.; SHUMWAY, T. Is sound just noise? *The Journal of Finance*, v. 56, n. 5, p. 1887–1910, 2001. Available at: <https://onlinelibrary.wiley.com/doi/abs/10.1111/0022-1082.00393>. Cited 2 times on page(s) 94 and 101.

DAIGLER, R. T.; WILEY, M. K. The Impact of Trader Type on the Futures Volatility-Volume Relation. *Journal of Finance*, v. 54, n. 6, p. 2297–2316, December 1999. Available at: <https://ideas.repec.org/a/bla/jfinan/v54y1999i6p2297-2316.html>. Cited on page 101.

DUMAIS, S. T. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers*, v. 23, n. 2, p. 229–236, Jun 1991. ISSN 1532-5970. Available at: <https://doi.org/10.3758/BF03203370>. Cited 3 times on page(s) 36, 44, and 49.

DVORAK, T. Do domestic investors have an information advantage? evidence from indonesia. *The Journal of Finance*, Blackwell Publishing, v. 60, n. 2, p. 817–839, 2005. ISSN 1540-6261. Available at: <http://dx.doi.org/10.1111/j.1540-6261.2005.00747.x>. Cited on page 23.

ENGELBERG, J. E.; PARSONS, C. A. The causal impact of media in financial markets. *The Journal of Finance*, Blackwell Publishing Inc, v. 66, n. 1, p. 67–97, 2011. ISSN 1540-6261. Available at: <http://dx.doi.org/10.1111/j.1540-6261.2010.01626.x>. Cited on page 23.

FAMA, E. Efficient capital markets: Ii. *Journal of Finance*, v. 46, n. 5, p. 1575–617, 1991. Available at: <https://EconPapers.repec.org/RePEc:bla:jfinan:v:46:y:1991:i:5:p:1575-617>. Cited 3 times on page(s) 25, 70, and 94.

FANG, L.; PERESS, J. Media coverage and the cross-section of stock returns. *The Journal of Finance*, Blackwell Publishing Inc, v. 64, n. 5, p. 2023–2052, 2009. ISSN 1540-6261. Available at: <http://dx.doi.org/10.1111/j.1540-6261.2009.01493.x>. Cited on page 23.

FELDMAN, R. Techniques and applications for sentiment analysis. *Commun. ACM*, ACM, New York, NY, USA, v. 56, n. 4, p. 82–89, apr. 2013. ISSN 0001-0782. Available at: <http://doi.acm.org/10.1145/2436256.2436274>. Cited on page 24.

FURNAS, G. W.; DEERWESTER, S.; DUMAIS, S. T.; LANDAUER, T. K.; HARSHMAN, R. A.; STREETER, L. A.; LOCHBAUM, K. E. Information retrieval using a singular value decomposition model of latent semantic structure. In: *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 1988. (SIGIR '88), p. 465–480. ISBN 2-7061-0309-4. Available at: <http://doi.acm.org/10.1145/62437.62487>. Cited 3 times on page(s) 37, 72, and 92.

GARCíA, D. Sentiment during recessions. *The Journal of Finance*, v. 68, n. 3, p. 1267–1300, 2013. ISSN 1540-6261. Available at: <http://dx.doi.org/10.1111/jofi.12027>. Cited 6 times on page(s) 24, 32, 72, 77, 94, and 97.

GINO, F.; BROOKS, A. W.; SCHWEITZER, M. E. Anxiety, advice, and the ability to discern: Feeling anxious motivates individuals to seek and use advice. *Journal of Personality and Social Psychology*, v. 102, n. 3, p. 132–140, 2012. Available at: <http://www.hbs.edu/faculty/Publication%20Files/gino_brooks_schweitzer_jpsp_2012_fd79893e-9f44-4a69-9460-848527d2d598.pdf>. Cited on page 89.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Cited on page 34.

GRANGER, C. W. J. Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. *Econometrica, Econometric Society*, v. 37, n. 3, p. 424–38, July 1969. Available at: <http://webber.physik.uni-freiburg.de/~jeti/studenten_seminar/stud_sem_SS_09/grangercausality.pdf>. Cited on page 22.

GROPP, R.; KADAREJA, A. Stale information, shocks, and volatility. *Journal of Money, Credit and Banking*, v. 44, n. 6, p. 1117–1149, 2012. Available at: <https://EconPapers.repec.org/RePEc:mcb:jmoncb:v:44:y:2012:i:6:p:1117-1149>. Cited on page 25.

GURUN, U. G.; BUTLER, A. W. Don't believe the hype: Local media slant, local advertising, and firm value. *The Journal of Finance*, Blackwell Publishing Inc, v. 67, n. 2, p. 561–598, 2012. ISSN 1540-6261. Available at: <http://dx.doi.org/10.1111/j.1540-6261.2012.01725.x>. Cited on page 23.

HENDERSHOTT, D. L. T.; SCHURHOFF, N. Are institutions informed about news? *Journal of Financial Economics*, v. 117, n. 2, p. 249–287, 2015. Available at: <http://faculty.haas.berkeley.edu/hender/HLS_IOF_News.pdf>. Cited 3 times on page(s) 25, 70, and 90.

HESTON, S. L.; SINHA, N. R. News vs. sentiment: Predicting stock returns from news stories. *Financial Analysts Journal*, v. 73, n. 3, p. 67–83, 2017. Available at: <https://doi.org/10.2469/faj.v73.n3.3>. Cited 2 times on page(s) 70 and 90.

HINTON, G. E.; MCCLELLAND, J. L.; RUMELHART, D. E. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. In: RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, C. P. R. (Ed.). Cambridge, MA, USA: MIT Press, 1986. cap. Distributed Representations, p. 77–109. ISBN 0-262-68053-X. Available at: <http://dl.acm.org/citation.cfm?id=104279.104287>. Cited on page 39.

HUANG, A. H.; ZANG, A. Y.; ZHENG, R. Evidence on the information content of text in analyst reports. *The Accounting Review*, v. 89, n. 6, p. 2151–2180, 2014. Available at: <https://doi.org/10.2308/accr-50833>. Cited 2 times on page(s) 70 and 90.

HUBERT, P.; LABONDANCE, F. *Central bank sentiment and policy expectations*. [S.l.], 2017. Available at: <https://ideas.repec.org/p/boe/boeewp/0648.html>. Cited on page 26.

JEGADEESH, N.; WU, D. Word power: A new approach for content analysis. *Journal of Financial Economics*, v. 110, n. 3, p. 712 – 729, 2013. ISSN 0304-405X. Available at: <http://www.sciencedirect.com/science/article/pii/S0304405X13002328>. Cited 2 times on page(s) 30 and 49.

JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, v. 28, p. 11–21, 1972. Cited on page 35.

JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; MIKOLOV, T. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016. Available at: <https://arxiv.org/abs/1607.01759>. Cited on page 65.

KAPLANSKI, G.; LEVY, H. Sentiment and stock prices: The case of aviation disasters. *Journal of Financial Economics*, v. 95, n. 2, p. 174 – 201, 2010. ISSN 0304-405X. Available at: <http://www.sciencedirect.com/science/article/pii/S0304405X09002086>. Cited on page 25.

KEARNEY, C.; LIU, S. Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis*, v. 33, p. 171 – 185, 2014. ISSN 1057-5219. Available at: <http://www.sciencedirect.com/science/article/pii/S1057521914000295>. Cited 3 times on page(s) 30, 34, and 52.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. Available at: <http://arxiv.org/abs/1412.6980>. Cited on page 97.

LE, Q. V.; MIKOLOV, T. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014. Available at: <http://arxiv.org/abs/1405.4053>. Cited 3 times on page(s) 41, 73, and 93.

LEVY, O.; GOLDBERG, Y. Linguistic regularities in sparse and explicit word representations. In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2014. p. 171–180. Available at: <http://www.aclweb.org/anthology/W14-1618>. Cited on page 40.

LEVY, O.; GOLDBERG, Y. Neural word embedding as implicit matrix factorization. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. Cambridge, MA, USA: MIT Press, 2014. (NIPS'14), p. 2177–2185. Available at: <http://dl.acm.org/citation.cfm?id=2969033.2969070>. Cited on page 64.

LEVY, O.; GOLDBERG, Y.; DAGAN, I. Improving distributional similarity with lessons learned from word embeddings. *TACL*, v. 3, p. 211–225, 2015. Available at: <https://levyomer.wordpress.com/2015/03/30/improving-distributional-similarity-with-lessons-learned-from-word-embeddings/>. Cited on page 65.

LI, F. Annual report readability, current earnings, and earnings persistence. *Journal of Accounting and Economics*, v. 45, n. 2-3, p. 221–247, 2008. Available at: <http://www.sciencedirect.com/science/article/pii/S0165410108000141>. Cited on page 25.

LI, F. The information content of forward-looking statements in corporate filings-a naïve bayesian machine learning approach. *Journal of Accounting Research*, v. 48, n. 5, p. 1049–1102, 2010. Available at: <https://EconPapers.repec.org/RePEc:bla:joares:v:48:y:2010:i:5:p:1049-1102>. Cited 3 times on page(s) 70, 89, and 90.

LI, F. Textual analysis of corporate disclosures: A survey of the literature. v. 29, 02 2011. Cited 3 times on page(s) 21, 52, and 74.

LIU, B. Sentiment analysis and subjectivity. In: *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca*. [S.l.: s.n.], 2010. Cited on page 24.

LIU, J.; LIU, Z.; CHEN, H. Revisit word embeddings with semantic lexicons for modeling lexical contrast. In: *2017 IEEE International Conference on Big Knowledge (ICBK)*. [S.l.: s.n.], 2017. p. 72–79. Cited on page 65.

LONG J BRADFORD, e. a. D. Positive feedback investment strategies and destabilizing rational speculation. *Journal of Finance*, v. 45, n. 2, p. 379–95, 1990. Available at: <https://EconPapers.repec.org/RePEc:bla:jfinan:v:45:y:1990:i:2:p:379-95>. Cited 2 times on page(s) 81 and 103.

LOUGHRAN, T.; MCDONALD, B. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *Journal of Finance*, v. 66, n. 1, p. 35–65, 2011. Available at: <http://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.2010.01625.x/abstract>. Cited 16 times on page(s) 30, 32, 38, 46, 47, 49, 55, 56, 62, 70, 72, 81, 89, 90, 92, and 93.

LOUGHRAN, T.; MCDONALD, B. The use of word lists in textual analysis. *Journal of Behavioral Finance*, Routledge, v. 16, n. 1, p. 1–11, 2015. Available at: <https://doi.org/10.1080/15427560.2015.1000335>. Cited on page 89.

LOUGHRAN, T.; MCDONALD, B. Textual analysis in accounting and finance: A survey. *Journal of Accounting Research*, v. 54, n. 4, p. 1187–1230, 2016. Available at: <http://onlinelibrary.wiley.com/doi/10.1111/1475-679X.12123/abstract>. Cited 2 times on page(s) 21 and 30.

LOUGHRAN, T.; MCDONALD, B.; PRAGIDIS, I. Assimilation of oil news into prices. *SSRN Electronic Journal*, 01 2018. Available at: <http://dx.doi.org/10.2139/ssrn.3074808>. Cited 2 times on page(s) 24 and 38.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. ISBN 0521865719, 9780521865715. Cited 3 times on page(s) 34, 36, and 92.

MANNING, C. D.; SCHüTZE, H. *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999. ISBN 9780262133609. Cited on page 39.

MERTON, R. C. A simple model of capital market equilibrium with incomplete information. *The Journal of Finance*, Blackwell Publishing Ltd, v. 42, n. 3, p. 483–510, 1987. ISSN 1540-6261. Available at: <http://dx.doi.org/10.1111/j.1540-6261.1987.tb04565.x>. Cited 2 times on page(s) 23 and 94.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G.; DEAN, J. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013. Available at: <http://arxiv.org/abs/1310.4546>. Cited 3 times on page(s) 41, 65, and 90.

ORENGO, V. M.; HUYCK, C. A stemming algorithm for the portuguese language. In: *Proceedings Eighth Symposium on String Processing and Information Retrieval*. [s.n.], 2001. p. 186–193. Available at: <https://pdfs.semanticscholar.org/e9d9/ea5fc73013ff9d408b95c744d668896eb31b.pdf>. Cited 2 times on page(s) 52 and 54.

PANG, B.; LEE, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: *Proceedings of ACL*. [S.l.: s.n.], 2005. p. 115–124. Cited on page 24.

PANG, B.; LEE, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, Now Publishers Inc., Hanover, MA, USA, v. 2, n. 1-2, p. 1–135, jan. 2008. ISSN 1554-0669. Available at: <http://dx.doi.org/10.1561/1500000011>. Cited on page 24.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. v. 14, p. 1532–1543, 01 2014. Available at: <https://nlp.stanford.edu/pubs/glove.pdf>. Cited 2 times on page(s) 65 and 90.

PERESS, J. The media and the diffusion of information in financial markets: Evidence from newspaper strikes. *The Journal of Finance*, v. 69, n. 5, p. 2007–2043, 2014. Available at: <https://onlinelibrary.wiley.com/doi/abs/10.1111/jofi.12179>. Cited on page 70.

PORTER, M. F. An algorithm for suffix stripping. *Program*, MCB UP Ltd, v. 14, n. 3, p. 130–137, 1980. Available at: <http://stp.lingfil.uu.se/~marie/undervisning/textanalys16/porter.pdf>. Cited 2 times on page(s) 52 and 54.

ROBERTSON, S. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, v. 60, p. 503–520, 2004. Cited on page 35.

ROZIN, P.; ROYZMAN, E. B. Negativity bias, negativity dominance, and contagion. *Personality and Social Psychology Review*, v. 5, n. 4, p. 296–320, 2001. Available at: <http://dx.doi.org/10.1207/S15327957PSPR0504_2>. Cited on page 98.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, p. 533–536, 10 1986. Cited on page 39.

SALEHINEJAD, H.; BAARBE, J.; SANKAR, S.; BARFETT, J.; COLAK, E.; VALAEE, S. Recent advances in recurrent neural networks. *CoRR*, abs/1801.01078, 2018. Available at: <http://arxiv.org/abs/1801.01078>. Cited on page 40.

SALTON, G. *The SMART Retrieval System – Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1971. Cited on page 36.

SALTON, G. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0-201-12227-8. Cited on page 43.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 24, n. 5, p. 513–523, aug. 1988. ISSN 0306-4573. Available at: <http://pmcnamee.net/744/papers/SaltonBuckley.pdf>. Cited 4 times on page(s) 43, 44, 49, and 92.

SHAPIRO, A. H.; SUDHOF, M.; WILSON, D. J. *Measuring News Sentiment*. [S.l.], 2017. Available at: <https://ideas.repec.org/p/fip/fedfwp/2017-01.html>. Cited on page 26.

SHARPE, S. A.; SINHA, N. R.; HOLLRAH, C. A. *What's the Story? A New Perspective on the Value of Economic Forecasts*. [S.l.], 2017. Available at: <https://ideas.repec.org/p/fip/fedgfe/2017-107.html>. Cited on page 26.

SOLOMON, D. H. Selective publicity and stock prices. *The Journal of Finance*, Blackwell Publishing Inc, v. 67, n. 2, p. 599–638, 2012. ISSN 1540-6261. Available at: <http://dx.doi.org/10.1111/j.1540-6261.2012.01726.x>. Cited 2 times on page(s) 24 and 32.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, v. 15, p. 1929–1958, 2014. Available at: <http://jmlr.org/papers/v15/srivastava14a.html>. Cited on page 95.

TETLOCK, P. C. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, Blackwell Publishing Inc, v. 62, n. 3, p. 1139–1168, 2007. ISSN 1540-6261. Available at: <http://dx.doi.org/10.1111/j.1540-6261.2007.01232.x>. Cited 11 times on page(s) 24, 55, 70, 76, 77, 89, 91, 94, 97, 101, and 106.

TETLOCK, P. C. Does public financial news resolve asymmetric information? *Review of Financial Studies*, v. 23, n. 9, p. 3520–3557, 2010. Available at: <https://academic.oup.com/rfs/article-abstract/23/9/3520/1671631>. Cited on page 89.

TETLOCK, P. C. All the news that's fit to reprint: Do investors react to stale information? *The Review of Financial Studies*, v. 24, n. 5, p. 1481–1512, 2011. Available at: <http://dx.doi.org/10.1093/rfs/hhq141>. Cited on page 25.

TETLOCK, P. C.; SAAR-TSECHANSKY, M.; MACSKASSY, S. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, Blackwell Publishing Inc, v. 63, n. 3, p. 1437–1467, 2008. ISSN 1540-6261. Available at: <http://dx.doi.org/10.1111/j.1540-6261.2008.01362.x>. Cited 8 times on page(s) 24, 32, 45, 56, 70, 72, 93, and 94.

TSAI, F.-T.; LU, H.-M.; HUNG, M.-W. The impact of news articles and corporate disclosure on credit risk valuation. *Journal of Banking & Finance*, v. 68, p. 100 – 116, 2016. ISSN 0378–4266. Available at: <http://www.sciencedirect.com/science/article/pii/S0378426616300231>. Cited on page 32.

TURIAN, J.; RATINOV, L.; BENGIO, Y. Word representations: A simple and general method for semi-supervised learning. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (ACL '10), p. 384–394. Available at: <https://www.aclweb.org/anthology/P10-1040>. Cited on page 90.

WIEBE, J.; BRUCE, R.; BELL, M.; MARTIN, M.; WILSON, T. A corpus study of evaluative and speculative language. In: *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2001. (SIGDIAL '01), p. 1–10. Available at: <http://dx.doi.org/10.3115/1118078.1118104>. Cited on page 49.

WISNIEWSKI, T. P.; LAMBE, B. The role of media in the credit crunch: The case of the banking sector. *Journal of Economic Behavior & Organization*, v. 85, n. 0, p. 163 – 175, 2013. ISSN 0167-2681. Financial Sector Performance and Risk. Available at: <http://www.sciencedirect.com/science/article/pii/S0167268111002617>. Cited on page 22.

XU, K.; BA, J.; KIROS, R.; CHO, K.; COURVILLE, A. C.; SALAKHUTDINOV, R.; ZEMEL, R. S.; BENGIO, Y. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015. Available at: <http://arxiv.org/abs/1502.03044>. Cited on page 40.

YANG, Y.; PEDERSEN, J. O. A comparative study on feature selection in text categorization. In: . [S.l.]: Morgan Kaufmann Publishers, 1997. v. 9, p. 412–420. Cited on page 53.

YANG, Z.; YANG, D.; DYER, C.; HE, X.; SMOLA, A. J.; HOVY, E. H. Hierarchical attention networks for document classification. In: *HLT-NAACL*. [s.n.], 2016. Available at: <https://www.cs.cmu.edu/~hovy/papers/16HLT-hierarchical-attention-networks.pdf>. Cited on page 41.