UNIVERSITY OF BERGAMO

SCHOOL OF DOCTORAL STUDIES
DOCTORAL DEGREE IN ANALYTICS FOR ECONOMIC AND BUSINESS
XXXII CYCLE

Using Social Media Analytics for Alloy Steel Prices

Advisors:
Prof. Eugenio BRENTARI
& Prof. Paulo CORTEZ

Doctoral Thesis:
Paola ZOLA
Student ID: 1043440

Academic year 2018/19

Declaration of Authorship

I, Paola ZOLA, declare that this thesis titled, "Using Social Media Analytics for Alloy Steel Prices" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 3^{rd} September 2019

UNIVERSITY OF BERGAMO

Abstract

Department of Business Sciences, Economics and Quantitative Methods.

Doctor of Analytics for Economic and Business

Using Social Media Analytics for Alloy Steel Prices

by Paola ZOLA

The proposed PhD thesis was designed to research the statistical relation between alloy steel price time series and opinion of texts from social media. However, as the thesis shows, the specific case study (alloy steel prices) leads to further challenges that are not common for related works focused on stock prices or stock indexes. The peculiarity of the alloy steel market and the free and unstructured social media texts involved several issues that we tried to approach and solve with frameworks and tools specifically built for social media analytics. The thesis deals with different issues related to geoparsing, sentiment analysis, user reliability and financial disambiguation using approaches that merge statistic and computer science, generating novel solutions to existing social media analytics limitations.

Acknowledgements

This PhD. thesis is a summary of three years of study and work made possible by the Universities of Bergamo and Brescia. Thus, I want to thanks the support of the computational resources from the Big & Open Data Innovation Laboratory (BODaI-Lab), University of Brescia, granted by Fondazione Cariplo and Regione Lombardia.

Not less important was the support of my supervisors Prof. Eugenio Brentari and Prof. Maurizio Carpita, from Univeristy of Brescia, for the advise given and, especially, to having let me freely to choose the research topics, more suited to my research interests.

An immense thanks to my supervisor Prof. Paulo Cortez for having trusted in me, for all the knowledge, the support and the help that was provided to me during my visiting period at University of Minho, Portugal.

Thanks to all my PhD. and work colleagues from Italy and Portugal for the suggestions, advises and ideas shared during these years.

Finally, thanks to my family, Costantino, friends, colleagues and Professor that were with me during the achievements and defeats of these three intense years.

Contents

List of Figures

List of Tables

To my mum

Chapter 1

Introduction

One of the most controverse milestone in finance is represented by the theory of efficient markets (EMH) proposed by Malkiel and Fama (1970). According to this theory, the financial market is considered efficient when the prices of financial products quickly reflect any change in the available information on the market, preventing "free lunches". Actually, if each market's players have the same information set, all the deals would be conducted a fair value. Thus, the excess return should come from the noise traders, which is in contrast with the hypothesis of the rational man (Xing, Cambria, and Welsch, 2018). More recently, behavioural finance has come up with new theories that are compatible with the interactive nature of market and its actors, such as the Adaptive Market Hypothesis (AMH) proposed by Lo (2004). The AMH theory is based on what behaviouralists cite as counter examples to economic rationality, loss aversion, overconfidence, overreaction, mental accounting, and other behavioral biases are, in fact, consistent with an evolutionary model of individuals adapting to a changing environment via simple heuristics (Lo, 2004). Hence, the excess return can be referred to information asymmetry.

Knowing that information asymmetry determine excess returns, it is also crucial to analyze how investors reacts to information and news. The stock market history is full of passionate events: the Great Crash in 1929, the Tronics Boom in the '60s, the Go-Go Years in the late '60s, the Nifty bubble in the early 1970, the Black Monday crash in October 1987, the Internet bubble in 1990s, the Great recession in the 2000s and the recent European sovereign debt crisis from 2010. Each of these events involved a dramatic change in stock prices showing that the the assumption of financial *homo economicus* with unemotional behaviour is not longer justifiable. Following Baker and Wurgler (2007) nowadays the question is not whether investor sentiment affects stock prices, but, rather, how to measure investor sentiment and quantify its effects. Baker and Wurgler (2007) identify two type of approaches to investigate investor sentiments: the "bottom up" and the "top down" ones. The "bottom

up" approach uses biases in individual investor psychology, such as overconfidence, etc. to explain the individual investors reactions. However, as argued by Baker and Wurgler (2007) the "bottom up" models, focusing on a small sample of investors, lead to similar reduced form of variation over time in mass psychology. Thus, the "top down" approach focuses on the measurement of reduced-form, aggregate sentiment and traces its effects to market returns and stocks.
Initial analysis involving the study of investor sentiments used, as proxies of human behaviour, a series of both quantitative features (Trading Volume, Dividend Premium, IPO Volume, Option Implied Volatility) and qualitative features (Investor Mood and Investor Surveys). Focusing on qualitative approaches the Investor Surveys and Investor Mood might be related to the *Social Network perspective* to financial forecast (Bollen, Mao, and Zeng, 2011).

The *Social Network perspective* an interesting possibility to formalize the financial forecast tasks, other perspectives are related to: the *portfolio management, energy system* and the *connectionist perspective* (Xing, Cambria, and Welsch, 2018). The *Social Network perspective* derives from the early works in mathematics and later confirmed by experimental finance. Financial bubbles are generated by investors behaviour able to easy triplicate fundamental prices (Bao, Hommes, and Makarewicz, 2017) generating a serious question: how much of the financial prices depend on world economic scenario and markets and how much is the impact of mass sentiment?
To answer to this question a wide range of new studies merging computer science and finance knowledge have been proposed. The following Section 1.1 reports an overview of previous works that investigate the impact of social media sentiment analysis on financial market prices.

1.1 Sentiment analysis for financial markets prediction

In this Section a description and analysis of previous works focused on sentiment analysis for financial market is reported. "Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes. It represents a large problem space." (Liu, 2012).

The concept of "measuring sentiment" for financial prediction has taken

on different connotations over time. For example, some works refer to sentiment analysis as the measurement of quantitative features as: dividend premium, closed-end funds discounts, option implied volatility, option implied skewness (Gao and Süss, 2015). Other studies follow an approach that merge knowledge from computer science and finance to investigate the sentiment from textual sources. A common textual data source is represented by the *Thomson Reuters* datasets that have been widely researched for different type of asset forecasting (Lechthaler and Leinert, 2012; Feuerriegel and Neumann, 2013). Other studies analyzed the sentiment from Yahoo Finance message board (Nguyen, Shirai, and Velcin, 2015). More recently, the advent of Twitter has generated an increased sentiment analysis research attention for the financial domain. Twitter is one of the most used social media platform in the world, with around 100 million daily active users and an average of 500 million daily tweets[1]. Twitter, born with the idea of sharing quick and short texts, imposes a character limits, that nowadays is set to 280[2]. Tweets can include the so-called *hastag* denoted by the # symbol and that is a keyword or short phrase used to describe a topic or a theme. More recently, Twitter introduced also the *cashtag* denoted by the $ symbol, particularly adopted by the financial community[3]. As reported in Table 1.1 several studies focus on Twitter data (Pagolu et al., 2016; Bollen, Mao, and Zeng, 2011; Mudinas, Zhang, and Levene, 2019). Some studies analyzed also the impact of tweets sentiment jointly with other indicators such as the Search Volume Index (SVI) (Rao and Srivastava, 2013) or stocks trading volume (Oliveira, Cortez, and Areal, 2013). A different approach was is the study proposed by Preis, Moat, and Stanley (2013), which analyzed the relation between stock market index (Dow Jones Industrial Average) and Google Trends queries. The authors found that queries peak anticipate market collapses.

Focusing on the approaches designed to extract a quantitative score (sentiment) from textual data, one of the most used method is based on lexicon dictionaries. Lexicon dictionaries are ready made dictionary that assign a sentiment, either a numeric value, such as for the SentiWordNet Baccianella, Esuli, and Sebastiani (2010) or a qualitative mood (Mahmud, Nichols, and Drews, 2014), to a set of words. Lexicon dictionaries are appropriate when the data are not labelled, since they do not need any training. Oliveira, Cortez, and Areal (2013) compared five different popular lexicon to derive the sentiment and they also proposed two new lexicons: the Emoticons one, based on the analysis of emoticons in tweets, and a lexicon that merged the Emoticons with

[1] https://www.omnicoreagency.com/twitter-statistics/
[2] http://time.com/4958311/twitter-280-character-limit/
[3] http://www.newser.com/story/151173/twitter-introduces-the-cashtag.html

all the five popular databases. However, for the financial domain some researches also developed specific financial dictionaries as the Laughram & McDonald Finalncial sentiment (Pröllochs, Feuerriegel, and Neumann, 2015). In other cases, analysts can obtain the sentiment evaluation directly from the data provider such as Thomson Reuters MarketPsych Indices (Huang et al., 2018), or Thomson Reuters News (Lechthaler and Leinert, 2012). Feuerriegel and Neumann (2013) derive the sentiment from Thomson Reuters dataset computing different measures, such as Tetlock-Negative based on Harvard-IV dictionary, the Net-Optimism, the Tonality and the Bi-Normal Separation. The Net-Optimism is also used in Pröllochs, Feuerriegel, and Neumann (2015). Different from lexicon dictionaries are the approaches based on machine learning or deep learning. This family of approach need a set of labelled data and require a numerical depiction of the text's strings that can be computed by N-grams approaches or word embedding. Pagolu et al. (2016) examined both Word2Vec algorithm and N-grams depictions to classify sentences by sentiment. The authors investigated three machine learning approaches which are Logistic regression, Support vector machine and random forest. A more detailed explanation of machine learning and deep learning models for sentiment classification is given in Chapter 6.

As Table 1.1 shows in the column **asset class**, the most common financial asset evaluated in sentiment analysis is represented by stock indexes and, in particular, by the Dow Jones Industrial Average (DJIA) Index and its components. Also single stocks are often evaluated for sentiment analysis, while a very limited literature exists about commodity prices and, within our knowledge, no study has previously focused on alloys, such as steel or bronze. Moreover, comparing the asset class and the textual source, from Table 1.1, it is possible to notice that there is no research study that analyzed commodity prices using Twitter sentiment. In effect, only authoritative sources, such as Thompson Reuters, were previously used.

According to Xing, Cambria, and Welsch (2018) the reasons behind the preference of stocks, stocks indexes and currency as asset classes are three:

- lack of accessibility for many asset: this is typical problem if the target of the analysis relies on corporate financial statements;
- the nature of financial products: an example are treasury bonds which are based on term structure interest rates and thus, mass sentiment does not affect the pricing;
- transparency of stocks and currency markets: these markets usually have large capitalization and many participants, then it also easier to access to public information.

The Table 1.1 reports a chronological ordered summary of state-of-the-art studies about sentiment analysis in the financial domain, particularly focusing on studies which involve commodities as asset class. The first column identifies the financial asset class, while the column **Target** specifies the target domain. The column called **Textual Data Source** denotes the source for textual data and the column **Sentiment Computation** describes the approach used to define the sentiment over the textual dataset. The column **Model** reports the approach used in the studies reported in the Table 1.1.

TABLE 1.1: Summary of the related work for sentiment analysis in financial markets

Study	Asset[a] Class	Target[b]	Language[c]	Textual[d] Data Source	Sentiment[e] Computation	Model[f]	Time Period
Bollen, Mao, and Zeng (2011)	SI	DJIA	ENG	TW	OF, GPMOS	GC, SOFNN	2008
Lechthaler and Leinert (2012)	CF	CO	ENG	TR	TRNAD	SVAR	2003-2010
Feuerriegel and Neumann (2013)	CF	G, CO	ENG	TR	LDB	R	2003-2012
Oliveira, Cortez, and Areal (2013)	S	AMD, AMZ, DELL, EBAY, HP, GOOGL, IBM, INTEL, MSFT	ENG	TW	LDB	R	2012-2013
Preis, Moat, and Stanley (2013)	SI	DJIA	ENG	-	GT	C	2004-2011
Rao and Srivastava (2013)	SI, CF, F (+VIX)	DJIA, NASDAQ, CO, G, EUR/USD	ENG	TW, SVI	NLP Stanford	GC, R	2010-2011
Zheng2014	CF	CSCE, CBOT, CME KCBOT, COMEX	-	-	B&W	SpC, VAR-GARCH-M	1968-2010
Gao and Süss (2015)	CF	U.S. Ex•	-	-	Qnt Fea	R	1996-2013
Pröllochs, Feuerriegel, and Neumann (2015)	SI	TRD	ENG	RA	LDB	RB+HMM	2004-2011
Nguyen, Shirai, and Velcin (2015)	S	18S	ENG	YFMB	TF-IDF+SVM	Topic-Sentiment	2012-2013
Pagolu et al. (2016)	S	MSFT	ENG	TW	W2V, N-gram	LR, RF, SVM	2015-2016
Li et al. (2017)	CF	CO	ENG	TR	LDB	GC, LogR, SVM, DT, BPNN	2008-2014
Daniel, Neves, and Horta (2017)	S	DJIA stocks	ENG	TW	*	Sent-Event Detect	2013-2015
Maslyuk-Escobedo, Rotaru, and Dokumentov (2017)	CF	CO, NG, PR, GAS, HO	ENG	TRNA	LDB	CSI	2003-2014
Guo, Sun, and Qian (2017)	S	CSM	CHN	Xueqiu	LDB	TOP	2014-2015
Huang et al. (2018)	S, B, CF, F, H	SP500, HPI, 3-YGB, USD, TRC	ENG	TR	TRMI	GC, VAR	1998-2016
Mudinas, Zhang, and Levene (2019)	SI, S, F	DJIA, AAPL, GOOGL, HP, JPM, EUR/USD, GBP/USD	ENG	FT, TW, Reddit	LDB	GC	2011-2014

[a] B: bond, CF: commodity futures, F: forex, H: housing prices, S: stocks, SI: stock index, VIX: volatility index.
[b] 18S: 18 different stocks quoted on DIJA, 3-YGB: 3 years government bond, AAPL: Apple, AMD: Advanced Micro Devices, AMZ: Amazon, CBOT: CO: crude oil, COMEX: commodity Exchange, CME: Chicago Mercantile Exchange, CSM: Chinese stock market including SSE Index, biological medicine and real estate, CSCE: Coffee, Sugar and Cocoa Exchange, DELL: Dell Technologies, DJIA: Dow Jones Industrial Average, EBAY: E-bay, EUR/USD: forex euro US dollar, G: gold, GAS: gasoline, GOOGL: google, GBP/USD: forex Great British Pound and US dollar, HO: heating oil, HP: Hewlett-Packard, HPI: housing price index, IBM: International Business Machines Corporation, INTEL: Intel Corporation, KCBOT: Kansas City Board of Trade, MSFT: Microsoft, NG: natural gas, PR: propane, TRC: Thomson Reuters commodity prices, USD: US dollar currency, U.S. Ex: U.S. Exchange composed by eight commodity groups.
[c] CHN: Chinese, ENG: English.
[d] FT: Financial Times, TR: Thomson Reuters, TRNA: Thomson Reuters News Analytics Database, TW: Twitter, RA: regulatory announces, YFMB: Yahoo Finance message board.
[e] B&W: Baker and Wurgle's website, GPMOS: MoodStates, GT: Google Trends, LDB: lexicon databased, OF: OpinionFinder, Qnt Fea: quantitative features, TF-IDF+ SVM: Term Frequency- Inverse Document Frequency and support vector machines, TRMI: Thomson Reuters marketPsych indices, TRNAD: W2V: Word2Vec,
[f] BPNN: backpropagation neural network, C: correlation, CSI: cumulative sentiment index, DT: decision trees, GC: Granger causality, HMM: hidden Markov model, LogR: logit regression, LR: logistic regression, R: regression model, RB: rule based, RF: random forest, SOFNN: self organized fuzzy neural network, SpC: Spearman's rank-order correlation, SVAR: sentiment VAR, SVM: support vector machine, TOP: thermal optimal path, VAR: vector autoregressive model, VAR-GARCH-M: vector autoregressive model – generalized autoregressive conditional heteroskedasticity – mean.
•
* Different approaches: MySentimentAPI, TextBlob, Snetistrength, Affin

1.2 Thesis motivations

Following the analysis reported in the previous Section 1-1.1, this thesis was designed to cover the existing gap of microblog sentiment analysis for alloy prices. Specifically, the idea is to measure the correlation and the predictability of steel product prices using the Twitter's texts, which are abundant, freely

available and easy to collect. Steel industry and then steel products are not as "famous" and widespread topic as DJIA stocks, or crude oil, or gold, or forex. Nevertheless, the steel industry represents a supporting pillar for the economy and for the industrial sector. Steel is the fourth-most commonly used metal in the world. It is highly important to the global economy and trends in production can even be thought of as an indicator of the health of a country's economy[4]. Steel industry represents a great component for the domestic gross product for some countries, as India. In the United States the steel industries employee around 142,000 people and about 6.5 million Americans are employed by steel-consuming companies[5]. Thus, it is not surprising the great impact achieved by the recent introduction of Trump steel tariffs in 2018. The U.S. President aimed to increase the production and consumption of domestic steel, growing the number of employee in the steel industry and limiting the imports from foreign countries as Europe and China. In 2018, the countries with the higher steel production in 2018 were China, European Union and India that respectively produced 928.3, 168.2 and 106.5 million metric tons.

Steel prices, as opposite to other metals as gold or silver, are not unique. Several type of steel products exists (e.g., billets, scrap, hot rolled coils, sheets, rebars, stainless steel products), determining an wide heterogeneity of prices. Moreover, as opposite as gold or silver, steel prices does not have a word price list but, every country has internal unofficial lists. Only few steel products (e.g., billets) have a quotation (future) on London Stock Exchange. It follows that, predicting steel prices is more challenging than predicting common stocks, stocks index, currencies or commodities values. Several factors impact on steel prices, such as the row materials (e.g., iron ore), the energy costs (e.g., carbon), the dynamic of consuming sectors (e.g., constructions), politic acts (e.g., Trump tariffs) and all the other factors that commonly influence financial products, such as stocks and commodities.

Some previous works attempted to predict steel product prices, mainly by applying well know approach of time series predictions. Malanichev and Vorobyev (2011) studied about a regression model between metal roll prices, global steel capacities and the dynamics of steel production costs. Zola and Carpita (2016) computed autoregressive integrated moving average (ARIMA) models and its variants to predict Italian steel product prices. Trian (2013) explored gold equivalent for forecasting a series of steel product prices (e.g., billet, hot rolled coil and scrap steel) in pipeline projects. The authors investigated the possibility that steel prices depend on the gold price using a regression model.

[4] https://www.focus-economics.com/blog/steel-facts-commodity-explainer

[5] https://money.cnn.com/2018/03/07/news/companies/trump-tariffs-steel-jobs/index.html

The aim of this PhD. thesis is to investigate the impact of microblogs data on steel products prices. However, as above-mentioned, this is a non trivial task, which joint challenge related to the steel prices predictions and the selection of the right tweets. Within our knowledge, none studies have analyzed the correlation and the predictability of steel prices by using tweets as the informative source. In contrast with previous works, which evaluate the impact of tweets on stocks, stocks indexes, currencies and commodities, the steel domain involves major challenges due to:

- the variety of steel prices;
- the absence of cashtag or specific symbols (as ticker) to identify the messages strictly related to the steel alloy domain;
- the need to associate steel messages to a specific country due to the absence of word steel prices lists.

It follows that, to perform an accurate analysis to measure the correlation between tweets and steel prices several research steps need to be performed. The core of this thesis is composed by three chapters, as shown in Figure 1.1, which are three different research papers in which we tried to solve a specific problem to build an accurate tool to measure the correlation and predictability of tweets and steel prices. Chapter 2 gives an overview of the practical techniques adopted in all the following Chapters 3 – 4 – 5, which include Natural Language Processing and Data Mining. The Chapter 3 aims to solve the problem of country identification in order to assign the steel alloy tweet to the country and thus the relative prices lists. In fact, whenever in the tweet it is not specified the referring market (e.g., "steel hot rolled coils price down") it is important to implicitly infer to which country the tweet is related to, in order to select the correct price lists for the correlation analysis. The paper extracted from Chapter 3 has already been published on the *Decision Support Systems* Journal (Australian CORE A*, Scimago Q1 journal in Computer Science, Artificial Intelligence and Information Systems) (Zola, Cortez, and Carpita, 2019). Chapter 4 aims to disambiguate generic steel tweets (e.g., "incredible steel watch price") from the target alloy steel tweets proposing the Twitter Financial Disambiguation (TFD) task. In the same chapter we also propose a financial user expertise evaluation in order to overweight tweets originated by experts and authoritative sources. The Chapter 4 is a paper submitted and still under revision. Chapter 5 proposes a cross-source cross-domain sentiment classification in order to overcome the classical limitations of lexicon database sentiment analysis which are dependent on the original domain. Moreover, the proposed sentiment classification is easily extendable to different languages in order to be

able to compute the sentiment and thus, the correlation between steel prices and tweets written in different languages. The paper extracted from Chapter 5 has already been accepted and ready for publishing on *International Journal of Information Technology and Decision Making* (Scimago Q1 in Computer Science) (Zola et al., 2019). Finally, Chapter 6 ends the thesis highlighting the main conclusions of the proposed works and the future research directions.

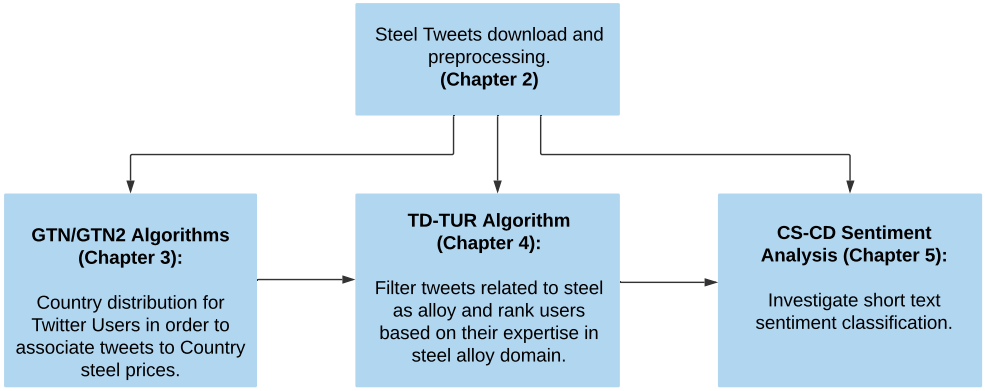

FIGURE 1.1: Schematic of the thesis.

Chapter 2

Tools

Before analysing the proposed approaches to overcome the different issues inherent to the text analysis for financial commodities/alloys prediction, this Chapter aims to provide the basic background about web scraping and text analysis.
The Chapter is composed by two sections: the Section 2.1 describes the tool used in the subsequent thesis chapters to gather and collect data from websites; Section 2.2 presents a brief introduction to Natural Language Processing.

2.1 Web scraping

The rapid grow of Word Wide web generated in the last decades a large amount of freely new data. From stocks prices, to product reviews, people opinions, clicks, etc. every day 2.5 quintillion bytes of data are created by each user[1]. It follows that it is impossible for a human collect and analyze all these giant amount of data, thus, machines are essential. Web scraping, data mining and social media analytic tools aim to automatically and recursively gather,collect and process data from the web.

Probably, the most known and easy approach to obtain web data is by the application program interface (API). The API generates a direct communication to the data related to a specific website, often generating a JSON or XML format output (Munzert et al., 2014). Some of the most famous documented APIs are the Twitter ones: the REST API and the *Streaming* API. In this work, to gather Twitter data (Chapter 3–4–5) the REST API has been used. The REST API offer access to the user's account, timeline, direct messages, friends and followers and in contrast with the *Streaming* API, the query response is not based on a random sample and it is only limited by communication costs and temporal bounds (Munzert et al., 2014). Thus, to avoid those limitation a *R*

[1] https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-da%ta-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read%/#50b5fdac60ba

script has been written to automatically fetch data every week, using the *TwitteR* package.

However, if Twitter APIs are rather common and easy to access, leading to a wide range of studies, others APIs are limited or forbidden for researcher (e.g., Tripadvisor API[2]). Thus, it is needed to develop specific scripts able to fetch data from websites as made in this thesis. Websites are written in a specific language: HyperText Markup Language (HTML), which is basically a plain text but, with a specific markup structure that makes its interpretation so powerful. The markup definitions are based on a set of specific token called *tags* that enclose part of the text, thus, part of the web page. However, beside the common HTML *tags*, each website has its own structure and also the design of the page might change during the time. Thus, for each specific source, a different *scraper* was built, mainly using the following *Python* modules:

- `requests`: it is an Apache2 Licensed HTTP library written in *Python* language and contains functions for requesting data across the web handing also some limitations as cookies or user agent. For example to collect data from *Amazon.com*, for the application in Chapter 5, we needed to fix an user agent to overcome the website limitation.

- `BeautifulSoup`: it is inspired by the poem of *Alice's in Wonderland* and it is a powerful module helping in organize the messy web pages' format (Mitchell, 2018). Similar to the `requests` module, it downloads the page content using either an HTML parser and a LXML one that has some likely property in dealing malformed HTML codes. In this thesis, `BeautifulSoup` has been widely used, for example to download news titles in Chapter 4, or in Chapter 5 for Tripadvisor.

- `Selenium`: it is a powerful web scraping tool and it works by automating browsers to load the website, retrieve the required data and even take screenshots, scrolling pages or press buttons (Mitchell, 2018). `Selenium` does not contain its web browser but it requires an integration with third-party browsers as Firefox or PhantomJS. The third-party browser is thus called by an API called WebDriver which load websites and can be used jointly with `BeautifulSoup`. Similar to `BeautifulSoup`, `Selenium` is based on CSS selectors that relies on the differentiation of HTML elements that might otherwise have the exact same markup in order to style them differently (Mitchell, 2018). CSS relies on identifying attributes that are readable from the HTML source page. However, a key advantage of the `Selenium` module is its ability to reach and act in every point of the websites, even scrolling infinite pages, as it was for Facebook website built

[2] https://developer-tripadvisor.com/content-api/request-api-access/

in AJAX format. In fact, after the Cambridge Analytica scandal [3], the free Facebook API service has been limited and to build a scraper able to gather Facebook public pages content `Selenium` was used.

Gathering raw data directly from websites involves several issues: query limits, IP (Internet Protocol) limits, internet connection problems, data storage and data quality. A central task in web scraping is detect relevant information from HTML, XML and other sources which are basically text documents presenting some systematic elements. Regular expressions identify a set of "rules" helpful in extracting information basically analysing textual strings and patterns. Depending on the programming language used, regular expressions might be slightly different. In this work, `regex` module in Python and `stringr` in R were adopted.

2.2 Natural language processing

Human brain and communication is mainly based on words. These words allow us to do many things, such as speak to other people and create written documents that can be shared. It is not by chance that children first learn words and only after they start counting and recognizing numbers. As opposite, computers system are based on a numerical depiction deriving on binary numerical system. Thus, for a machine the learning process is different from what occurs with humans, computers need a numerical vision of words to "understand" it.

Computational linguistic is the field of computer science in charge of leading machines capable to perform natural (human) language processing. The rise of Natural Language Processing (NLP) can be related to the 1950s with the Turing test developed by Alan Turing to evaluate the computational machine ability to exhibit intelligent behaviour equivalent to a human one (Moor, 2003). However, only from '70s, thanks to computational advances, NLP and computational linguistics started to assume a dominant role in computer science. The first works were mainly based on teaching to the machine to recognize patterns by hand-written rules (Greene and Rubin, 1971), while, from the '90s studies started to merge NLP with statistical models, in order to create models able to learn in a more unsupervised environment (Schütze and Singer,

[3] https://www.nytimes.com/2018/03/19/technology/facebook-cambridge-analytic%a-explained.html

1994; Borthwick and Grishman, 1999). Nowadays, NLP tasks are mainly approached by modern processing techniques, involving, often, machine learning (ML) and artificial intelligence (AI) algorithms (e.g., artificial neural networks) (Chen and Manning, 2014; Plank, Søgaard, and Goldberg, 2016). Moreover, a substantial difference that characterized modern ML and AI algorithm from previous rule-based and classical statistical approaches is the word depiction: words are not even more represented by characters sequences but by real numbers in multidimensional spaces via word embedding algorithms (Chapter 5).
The NLP area incorporates several tasks which can be grouped as:

- syntax analysis,
- semantic analysis, and
- speech analysis.

Syntax analysis involves methods that are fundamental in understanding words or sentences as sequence of characters. This class includes word segmentation, part-of-speech (POS) tagging, stemming, lemmatization and parsing. Word segmentation is related to identifies single word (token) in a document. It might be seen as an easy task, but, if we consider languages in which there are not whitespaces to split one word to others, such as chinese or japanese, this task assumes a higher complexity. POS tagging aims to identify the correct POS for each token in a document. Initial works were based on rules while more recent involves different complex algorithms. It might be seen as a static problem, but, the modern form of web 2.0 communication has open a new interesting challenge in this field. Stemming and lemmatization are often considered similar, however stemming is the process of reducing inflected words to their root form (as "close" for closes, closed, closing) while lemmatization aims to reduce the inflection giving the word lemma (e.g., "be" for will, is, are, were). Finally, parsing determines the parse tree (grammatical analysis) of a given sentence, reconstructing its meaning.

Semantic analysis aims to understand the concept and the meaning related to a document. This class is composed by several topics, such as the lexical semantics, the machine translation, natural language understanding, relationship extraction, named entity recognition (NER), which aims to identify specific named entities as locations, organizations or people names. Moreover, semantic analysis involves also the sentiment analysis (Chapter 5), topic detection and word sense disambiguation (Chapter 4), text summarization and question answering which is, probably, the most challenging task in NLP.

Speech analysis in NLP includes techniques that work on human speech, including speech segmentation, which aims to identify single tokens from a

vocal speech, speech recognition, which intents to give a textual representation of speech data and text to speech.

The next chapters of this thesis report social media analytics applications based on textual data, thus NLP techniques were necessary. In particular, we performed a classical syntax analysis and more complex semantic analysis in all the chapters, focusing on sentiment analysis (Chapter 5), introducing the concept of topic disambiguation (Chapter 4) and evaluating the location information contained in generic words (Chapter 3). Moreover, the textual sources in this thesis are from modern web 2.0 sites such as Amazon and Tripadvisor, but also from social networks, such as Facebook, and microblogs such as Twitter. These web 2.0 textual data introduced more complexity in NLP analysis due to the tendency of Internet users to write and communicate adopting the so-called cyberslang (Geană, 2018). Cyberslang refers to all those abbreviation, emoji, slang and modified words (e.g., "cooool") that people tend to use on web instead of traditional structured words. This tendency opened a new dynamic challenge for different NLP tasks (Zola and Golia, 2019), causing an increasing complexity for semantic NLP such as sentiment analysis.

Chapter 3

Geolocation of financial social media texts

Several Web and social media analytics require user geolocation data. Although Twitter is a powerful source for social media analytics, its user geolocation is a nontrivial task. This chapter presents an purely word distribution method for Twitter user country geolocation. In particular, we focus on the frequencies of tweet nouns and their statistical matches with Google Trends world country distributions (GTN method). Several experiments were conducted, using a recently created dataset of 744,830 tweets produced by 3,298 users from 54 countries and written in 48 languages. Overall, the proposed GTN approach is competitive when compared with a state-of-the-art world distribution geolocation method. To reduce the number of Google Trends queries, we also tested a machine learning variant (GTN2) that is capable of matching the GTN responses with an 80% accuracy while being much faster than GTN.

3.1 Problem specification

Due of the expansion of the Internet, Web and social media analytics are becoming a key element of many decision support systems. Modern Web platforms, such as Twitter and Google Trends (GT), provide valuable big data that are easy to collect. Twitter is an important microblogging service with approximately 330 million active users that generate opinionated texts [1]. Twitter sentiment analysis has been used to predict stock markets (Oliveira, Cortez, and Areal, 2016), political elections (Tumasjan et al., 2010), movie sales (Rui, Liu, and Whinston, 2013), and English Premier League soccer wins (Schumaker, Jarmoszko, and Labedz, 2016). GT is another relevant Web source, providing Google statistics of search terms across different world regions. GT data-based analytics were used to predict flu trends (Ginsberg et al., 2009), unemployment

[1] https://blog.hootsuite.com/twitter-statistics/

rates (Choi and Varian, 2009), consumer behavior (Choi and Varian, 2012), and the status of trending topics (Fang and Chen, 2016).

Several Web and social media analytics systems require user geographic location data. Examples include disaster early warning systems (Wu and Cui, 2018), property crime detection (Vomfell, Härdle, and Lessmann, 2018), event detection, epidemic dispersion, and news recommendations (Mahmud, Nichols, and Drews, 2014). However, estimating the current location of a user is a non-trivial task for several microblogging services. For example, Twitter allows users to add profile locations and geographically tag their tweets, but the percentage of geotagged tweets is low (Cheng, Caverlee, and Lee, 2010; Morstatter et al., 2013) and Twitter user profile location data is often unreliable (Hecht et al., 2011).

In this chapter, we present a novel statistical approach for country-level location detection of Twitter users. This geolocation is potentially valuable in several decision support system applications, allowing them to easily filter users from a specific country. For instance, it can be used in Twitter sentiment analysis related to country commodity prices, such as steel, silver, or cotton prices.

Our approach assumes that people tend to write about news, events, and so on, from the country to which they are more related. It follows that, even if a user lives in country A, she/he might be more interested in news or information linked to another country B, so the potential information held in the user's tweet is likely to refer to country B. Consider the following examples related to two tweets about steel production:

1. "chinese steel rebar production reach the maximum over a year"; and

2. "downhill price for steel beams".

Although it is clear for the first tweet example that the country of interest is China, for the second one it is not possible to link the information to a specific country. In contrast with the stock market domain, where easy identifiable cashtags [2] are common (for example, $AAPL for Apple stocks) (Oliveira, Cortez, and Areal, 2016), commodity country-specific tweets tend to be similar to the second tweet example: unstructured and without an obvious geographic term, hashtag, or cashtag. Moreover, these tweets are often written in English, so they could be related to any country's market. It follows that our approach aims to associate a tweet with a highly probable country context when such a geographic context is not explicitly known to assist in country-level Twitter analytics.

[2] https://techcrunch.com/2012/07/30/twitter-clickable-ticker-symbols/

To identify the unknown country, we analyze the word distribution of past user tweets. In contrast with previous studies that use specific geographical dictionaries, based on named-entity recognition (NER) modules (Lee et al., 2015), we consider generic nouns. As shown in 3.4.2, these nouns can incorporate geographic terms (like NER) but also non-geographic terms that are specific to a country. Examples of such nouns include "Brexit" (related to the United Kingdom), "Trump" (United States of America) and "cricket" (popular in Pakistan). In addition, because of cultural differences, there are nouns that are used in distinct countries with different frequencies (for example, "thanks" in 3.9) and such information can potentially aid in country discrimination. Moreover, non-English users can tweet in their native languages, and so non-English nouns (for example, "sono" and "stato" for Italy) can help in determining the country. To take advantage of this implicit information, we perform matching between frequent country-level GT and user tweet nouns (GTN). To the best of our knowledge, this is the first time that GT data has been used to detect geographical user information.

As a case study, we consider the steel production domain and recent Twitter data, which includes 744,830 tweets from 3,298 users. Following an empirical design science research approach (Arnott and Pervan, 2014), we show that our GTN model is competitive when compared with a state-of-the-art NER (Lee et al., 2015) (Table 3.4.1). To reduce the GT querying time, we also propose a GTN variant that uses machine learning (for example, deep multilayer perceptron and random forest) to learn the GT responses (3.4.3). Finally, we demonstrate the applicability of GTN to non-steel commodity domains using more recent Twitter data and a different but smaller sample of users (Section 3.4.4).

The contributions of the proposed approach include:

1. We perform a Twitter estimation of the most probable user country of interest when such explicit context is not known.

2. The estimation is based on generic nouns, retrieved from the user's historical tweets, which can include geographic words and other country-specific terms (including news, sports, religion, events, people, and native language nouns).

3. The proposed Google Trends nouns (GTN) method uses GT to solve a spatial detection task rather than a temporal task (as proposed in previous GT studies).

4. To reduce the GT query time, we proposed a second approach, termed GTN2, that uses machine learning.

5. We created a recent dataset related to the steel domain, which includes a conservative country estimate for 3,298 users, to empirically compare GTN with a state-of-the-art NER.

3.2 Related works

Several studies have investigated Web and social network user location estimation. Before the rise of social networks, the Internet protocol (IP) address was the main element used for Web geotagging (Buyukkokten et al., 1999). However, microblogs typically do not provide IP addresses. Moreover, the increasing use of virtual private networks (VPNs) reduces the reliability of IP address location.

Focusing on Twitter, user geographic estimation is a nontrivial task. Twitter location data can be directly retrieved by accessing geotagged tweets or user location field profiles. However, only a small fraction of tweets are geotagged. For example, the literature mentions low percentage values, varying from 0.42% (Cheng, Caverlee, and Lee, 2010) to 3.17% (Morstatter et al., 2013). While mobile devices are increasingly used, users often switch off global positioning system (GPS), for privacy reasons or to save battery consumption. Moreover, although Twitter users can add a geographic reference to their profiles, the field is free text and often unreliable locations are used (for example, “in your heart" or “everywhere"). Hecht et al. (2011) estimate that approximately 34% of Twitter users add nonrealistic text locations.

Table 3.1 summarizes the state-of-the-art research work on social network user location estimation, using chronological order and emphasizing the Twitter data source (**data source** column). There are three main types of social network user location estimation methods (**type** column):

1. Image recognition (IR): digital photos posted on social networks provide a vast amount of information, including location. For instance, Aulov and Halem (2012) studied the Deepwater horizon oil spill disaster in the Gulf of Mexico using Flickr photos and locating them to the desired area.

2. Friendship network (FN): the assumption is that the user’s location can be inferred by the locations of her/his friendship network. Examples of work that followed this assumption are Backstrom, Sun, and Marlow (2010), Davis Jr et al. (2011), and Rahimi, Cohn, and Baldwin (2015).

3. Word distribution (WD): related to our approach, it includes methods that are based on text analysis and word extraction. Some studies use existing NER modules, location indicative words (LIW), and gazetteers (geographic dictionaries) to extract locations from tweets (Lee et al., 2015).

Other studies are based on tweet word frequencies, proposing methods to filter local words (Cheng, Caverlee, and Lee, 2010; Dalvi, Kumar, and Pang, 2012; Ryoo and Moon, 2014).

Some studies complement the previous methods with the use of additional features (AF), such as the location field from the user profile metadata (Laylavi, Rajabifard, and Kalantari, 2016; Williams, Gray, and Dixon, 2017) or the tweeted time zone (Mahmud, Nichols, and Drews, 2014). Other studies combine the different types, such as: IR and WD (Crandall et al., 2009); WD and FN (Li et al., 2012; Minot et al., 2015; Rahimi et al., 2015; Rodrigues et al., 2016; Qian et al., 2017); and WD, FN, and AF (Williams, Gray, and Dixon, 2017).

The related works can also be characterized by the text **language**, location **target**, discrimination **level**, search **area** of interest, computational **algorithm**, evaluation method (**val.**), and **metric**. The type of language is often associated with the search area. In most cases, the messages are written in English. Regarding the target, while some studies focus on where the tweet was written (e.g., (Kinsella, Murdock, and O'Hare, 2011; Laylavi, Rajabifard, and Kalantari, 2016; Williams, Gray, and Dixon, 2017)), the majority try to detect the user's home location (e.g., (Cheng, Caverlee, and Lee, 2010; Li et al., 2012; Compton, Jurgens, and Allen, 2014; Rahimi et al., 2015; Qian et al., 2017)). As for the discrimination level, there are two main approaches: detecting larger regions (e.g., countries or states) or smaller regions (e.g., cities, landmarks, geographic coordinates, or postal codes). Some fine-grained level detection methods (e.g., geographic coordinates) are often associated with a specific geographic area and events, such as natural disasters or emergency responses (Aulov and Halem, 2012; Middleton, Middleton, and Modafferi, 2014; Laylavi, Rajabifard, and Kalantari, 2016; Avvenuti et al., 2018). The location level often affects the type of evaluation **metric** used. Large region discrimination methods tend to perform multiclass tasks, so common classification metrics (Witten et al., 2017) are often adopted (e.g., accuracy, precision, or recall). More diverse measures are used by the small region discrimination methods, including standard classification metrics (e.g., accuracy and precision), classification accuracy within a tolerance radius (Acc@R), or even regression metrics (e.g., root mean squared error). A wide variety of algorithms were adopted, including: approaches based on data frequency and statistics (e.g., information gain), generic machine learning models (e.g., neural network, support vector machine, or random forest), and specific geographic/Twitter-dependent methods (e.g., geocontext locator, geoparsing, or placemaker using tweet content). These algorithms were validated using either the simpler holdout (train and test split) or the more robust k-fold cross-validation.

The last row of Table 3.1 positions our work, which assumes a pure WD approach, a country-level detection, and multilingual tweets (mixed). The main

TABLE 3.1: Summary of the related work.

Study	Type[a]	Lang.[b]	Data Source[c]	Tar.[d]	Level[e]	Data Period[f]	User size[f]	Data size[f]	Val.[g]	Area[h]	Algorithm[i]	Metric[j]
Crandall et al. (2009)	IR,WD	EN	Flickr	F	SP	ND	307K	33M	ND	W	BC,SVM	Acc
Backstrom, Sun, and Marlow (2010)	FN	EN	TW	U	SP	ND	2.9M	ND	ND	USA	MLE	Acc@25mi
Cheng, Caverlee, and Lee (2010)	WD	EN	TW	U	CI	2009-10	1M	3M	*10*CV	USA	MLE	Acc@100ml
Davis Jr et al. (2011)	FN	PT	TW	T	CI	ND	25K	ND	*10*CV	BR	DFS	P
Kinsella, Murdock, and O'Hare (2011)	WD		TW	T	CO,SP	2010	7M	ND	5CV,HO	W	PM,KL,QL	Acc
Aulov and Halem (2012)	IR	EN	Flickr	F	SP	2010	ND	190	ND	MXG	GNOME	RMSE
Dalvi, Kumar, and Pang (2012)	WD	EN	TW	T	CI	2009-11	14M	200M	ND	USA	DM,LM	P,R
Li et al. (2012)	WD,FN	EN	TW	U	CI	2011	4.0M	ND	5CV	USA	UDI	Acc
Chang et al. (2012)	WD	EN	TW	U	CI	2009-10	136K	9M	HO	USA	GMM,LM, MLE	Acc
Compton, Jurgens, and Allen (2014)	FN	EN	TW	U	CI	2012-14	110M	ND	5CV,HO		TVM	ME (km)
Han, Cook, and Baldwin (2014)	WD	EN Mixed	TW	U	SP	2011-12	500K 1.4M	38M 12M	*10*CV	NA W	DFS	Acc@161km
Mahmud, Nichols, and Drews (2014)	WD,AF	EN	TW	U	SP	2011	10K	1M	*10*CV	USA	HE	Acc@100mi
Middleton, Middleton, and Modafferi (2014)	WD	EN,TR, IT,PT	TW	T	SP	2011-13	ND	1.5M	ND	USA	G	F1
Ryoo and Moon (2014)	WD	KR	TW	U	SP	2010-11	3.3M	615M	5CV	KR	PGM	Acc@10km
Minot et al. (2015)	FN,WD		TW	U	CI	2014	29K	7.0M	ND	AFR	SVM, CBF	Acc@10km
Lee et al. (2015)	WD	EN	TW	T	ST	2013-14	ND	113K	*10*CV	USA	SVM, BC,RF	R
Rahimi, Cohn, and Baldwin (2015)	FN	EN	TW TW	U	SP	2011-12	9.5K 450K 1.4M	380K 39M 12M	HO	USA USA W	LP	Acc@161km
Rahimi et al. (2015)	FN,WD	EN	TW TW TW	U	SP	2011-12	9.5K 450K 1.4M	380K 39M 12M	HO	USA USA W	LP	Acc@161km
Rodrigues et al. (2016)	FN,WD	PT	TW	U	CI	2010	12K	2M	*10*CV	BR	MM,BC, MRW	Acc
Kotzias, Lappas, and Gunopulos (2016)	FN	EN	TW	U	CI	2013	43K 40K 55K	1.9M 1.3M 1.5M	*10*CV	IR UK USA	LDA	P
Laylavi, Rajabifard, and Kalantari (2016)	WD,AF	EN	TW	T	SP	2015	2K	ND	ND	AUS	MELI	Acc@12.2km
Singh et al. (2017)	WD	EN HI	TW	T	SP	2015-16	ND	32K	ND	IN	MM	Acc
Williams, Gray, and Dixon (2017)	WD, FN,AF	EN	TW	T	SP	2016	15K	ND	ND	W	GCL	Acc@5km Acc@160km
Qian et al. (2017)	FN,WD	EN, ZH	TW TW Weibo FB	U	CO,CI	2011	1.5M 329K 1.0M 1K	ND	HO	W USA CH ND	NN	Acc
Avvenuti et al. (2018)	WD	EN, IT	TW	T	SP	2011-15	ND	9K 2K	ND	W IT	G	Acc
Rahimi, Cohn, and Baldwin (2018)	FN,WD	EN	TW TW TW	U	SP	2011-12	9.5K 450K 1.4M	380K 39M 12M	HO	USA USA W	NN DCCA	Acc@161km
This work	WD	Mixed	TW	U	CO	2017	49K	21M	*10*CV	W	GTN,GTN2	Acc, WF1

[a] image recognition (IR), friendship network (FN), word distribution (WD), additional features (AF).
[b] **Language**: Chinese (ZH), English (EN), Hindi (HI), Italian (IT), Korean (KR), Portuguese (PT), Turkish (TR); mixed: combination of multiple languages.
[c] Facebook (FB), Twitter (TW).
[d] **Target**: Flickr picture location (F), tweet location (T), user's home location (U).
[e] city (CI), country (CO), one of 50 states (ST), specific place (SP) from a region (e.g., coordinates, landmark or ZIP code).
[f] nondisclosed (ND), thousand (K), million (M); user and data size represent the initial collected values, before filtering.
[g] **Validation**: *n*-fold cross validation (*n*CV), hold out (HO), nondisclosed (ND).
[h] Africa (AFR), Australia (AUS), Brazil (BR), China (CH), India (IN), Ireland (IR), Italy (IT), Korea (KR), Mexican Gulf (MXG), nondisclosed (ND), North America (NA), United Kingdom (UK), United States of America (USA), World (W).
[i] Bayesian classifier (BC), consensus-based fusion (CBF), data frequency or statistic (DFS)-based, deep canonical correlation analysis (DCCA), distance model (DM), geoparsing-based (G), geocontext locator (GCL), Gaussian mixture model (GMM), general NOAA oil modeling environment (GNOME), Google Trends nouns (GTN), Google Trends nouns and machine learning (GTN2), hierarchical ensemble (HE), Kullback-Leibler (KL) divergence, label propagation (LP), language model (LM), latent Dirichlet allocation (LDA), Markov model (MM), maximum likelihood (MLE)-based, multi rank walk (MRW), multi-elemental location inference (MELI), neural network (NN), placemaker (PM) using tweet content, probabilistic generative model (PGM), query likelihood (QL), random forest (RF), support vector machine (SVM), total variation minimization (TVM), unified discriminative influence (UDI) model.
[j] accuracy (Acc), accuracy using a radius of R (Acc@R, R in miles (mi) or kilometers (km)), F1-score (F1), mean error (ME), precision (P), recall (R), root mean square error (RMSE), weight averaging F1-score (WF1).

novelty is the usage of generic nouns and GT source (the GTN method), as detailed in Section 3.3 and compared with a state-of-the-art WD method (Lee et al., 2015).

3.3 Proposed approach

3.3.1 Data description

Using automatic computational code (written in Python and R) and tools, we created a dataset with recent Twitter data to test the country geolocation methods. As an example in the decision support system application domain, we have targeted steel alloy. For the initial selection of users, we selected all tweets that included one of the keywords {"steel price", "steel industry", "steel production"}, from March to November 2017. These queries resulted in 138,484 tweets, related to 49,203 users. Only a tiny fraction of the tweets (192) were geotagged. In addition, only 33,886 users had a filled location profile field. We note that, in this work, retweets are treated in the same manner as common tweets, because retweets might be helpful in identifying the user's country of interest (e.g., retweets of a politician).

To set the ground truth, we designed a conservative procedure that discards a large number of users but is more reliable for comparing geolocation methods. The procedure is based on a strong double-source verification that considers both metadata (user profile location field) and LIW from historical user tweets. We considered the set of 33,886 users with some location profile data and retrieved up to a maximum of 3,200 past tweets for each user. We then used `OpenNLP` (Baldridge, 2005) and the `ggmap` R package (Kahle and Wickham, 2013) tools to extract LIW from the historical tweets (`OpenNLP`) and obtain the Google Maps country for each LIW (`ggmap`).The most frequent country, computed over the full set of LIW for a given user, was then compared with the metadata information. After removing country mismatches, including metadata with slang and nonrealistic locations, the final ground truth dataset contains 3,298 users and 744,830 tweets, representing an average of 226 tweets per user.

While all selected users have written at least one English term, from the set {"steel price", "steel industry", "steel production"}, the collected historical tweets were written by users from both native English speaking (e.g., Australia) and non-native English speaking (e.g., Spain) countries. Table 3.2 presents the percentage of tweets written in a specific language (**tweets** column) and the percentage of users per country (**users** column). Figure 3.1 plots these last values visually on a world map (the higher the percentage, the darker is the country color). The language values were obtained by using the `textcat`

R package (Feinerer et al., 2013). The majority of the tweets were written in English (66.2%), followed by the German (18.8%) and Catalan (4.4%) languages.

As for the countries, most users come from anglophone countries, such as United States of America (USA) (45.7%), United Kingdom (UK) (12.3%), and Australia (6.4%). As for the non-anglophone countries, most users are from India (27.1%), while other countries are much less prevalent (e.g., Germany with 0.5%). In total, the dataset contains tweets written in 48 languages and users from 54 countries.

TABLE 3.2: Dataset tweet languages and users per country.

Language	Tweets	Country	Users
English	66.2%	United States of America (USA)	45.7%
German	18.8%	India	27.1%
Catalan	4.4%	United Kingdom (UK)	12.3%
Danish	1.9%	Australia	6.4%
Nepali	1.3%	Canada	3.1%
Indonesian	1.1%	Germany	0.5%
Latin	0.9%	Pakistan	0.5%
Rumantsch	0.8%	South Africa	0.4%
Slovak	0.9%	China	0.3%
French	0.4%	France	0.3%
Esperanto	0.3%	Nigeria	0.3%
Swahili	0.3%	Spain	0.3%
Sanskrit	0.3%	Kenya	0.2%
Spanish	0.2%	Italy	0.2%
Romanian	0.2%	Mexico	0.2%
Swedish	0.2%	Finland	0.1%
Czech	0.2%	Ireland	0.1%
Malay	0.1%	Japan	0.1%
Hungarian	0.1%	Argentina	0.1%
Afrikaans	0.1%	Belgium	0.1%
Slovenian	0.1%	Brazil	0.1%
Dutch	0.1%	Colombia	0.1%
Tagalog	0.1%	Indonesia	0.1%
Basque	0.1%	Malaysia	0.1%
Others	0.6%	Others	1.2%

Only one state-of-the-art study performed a mixed-language tweet geolocation (Han, Cook, and Baldwin, 2014), as shown in Table 3.1. Our work does not separately consider datasets of tweets written in a specific language, because it is more trivial to identify the country when the language is distinctive of a nation (e.g., Japanese).

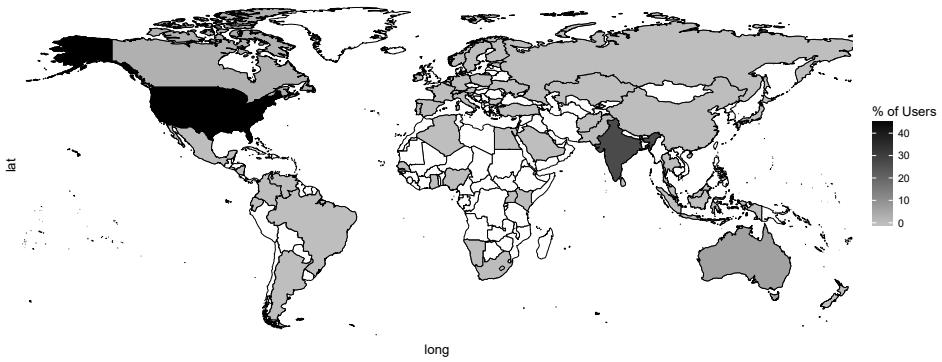

FIGURE 3.1: Percentage of users per country plotted on a world map.

Following the work of Han, Cook, and Baldwin (2014), we adopted a mixed language approach, which is more natural for the geolocation of countries, because Twitter is a multilingual platform. Nevertheless, the values in Table 3.2 reflect the steel domain scenario. Therefore, most of the tweets are written in English, which is a geographically widespread language that is more difficult to geolocate (Han, Cook, and Baldwin, 2014), making this dataset challenging and interesting for comparing purely WD methods.

3.3.2 Google Trends nouns

As explained above, the proposed GTN WD approach uses only tweet nouns, because we assume they are the most representative part of speech able to identify different countries.

For user u, the GTN approach works by first identifying the sequence of the most frequent nouns $\mathbf{n}_u = \langle n_1, n_2, ..., n_{l_u} \rangle$, in descending order and with a length of l_u elements. To obtain $\mathbf{n}_u$, the tweets are first preprocessed by transforming the text to lowercase and removing English stopwords. The `TextBlob` Python module is then used to extract noun phrases and then the nouns. We note that the `TextBlob` module is faster than other tools (Loria et al., 2014).

For each noun $n_i \in \mathbf{n}_u$, a GT query is executed by using the `Pytrends` Python module. To limit the number of queries, a fixed pruning threshold (p) is used, such that $l_u \leq p$ for all u users. The GT query result for noun n_i is a sequence with integer confidence scores for an alphabetic list of countries C with a length of $l_c = 250$. The scores range from 0 (lowest confidence) to

100 (highest confidence). Let $\mathbf{G}_u$ denote the GT confidence score matrix for user u with a size of $l_c \times l_u$, where each score is represented as $g_{c,i}$ for country $c \in C$ and the i-th most frequent noun. We test three strategies to weight the GT scores, resulting in the weighted confidence score matrix $\mathbf{S}_u$ ($l_c \times l_u$) with the elements $s_{c,i}$ (country, noun):

- equal weights (EQ): no weights are used, and so $s_{c,i} = g_{c,i}$.
- Internet usage (IU): weighted according to the fraction of Internet users for a specific country c (w_c) according to the World Bank statistics[3]:

$$\forall c \in C, \forall i \in \{1, ..., l_u\} : s_{c,i} = w_c g_{c,i} \tag{3.1}$$

- nouns frequency (NF): weighted according to the order of the nouns (more frequent nouns have stronger weights):

$$\forall c \in C, \forall i \in \{1, ..., l_u\} : s_{c,i} = w_i g_{c,i} \tag{3.2}$$

  where $w_i = (l_u - i + 1)/l_u$.

Once the confidence score is computed, we explore two statistical approaches to estimate the most probable country c_u for user u:

- join frequency (JF) – based on the highest score country when summing all noun scores:

$$c_u = \underset{c}{\text{argmax}}(\sum_{i=1}^{l_u} s_{c,i}) \tag{3.3}$$

- absolute frequency (AF) – selects the most common country (mode) when considering the highest score countries for all nouns:

$$c_u = Mode(argmax_c(s_{c,i}) \forall i \in \{1, ..., l_u\}) \tag{3.4}$$

  where *Mode* denotes the mode of a set.

3.3.3 Machine learning

In this work, we use machine learning for three different goals: to obtain the benchmark geolocation method outputs (for comparison purposes with GTN); to access the quality of the proposed GTN; and to mimic the GTN responses. For all three goals, the input features consist of the classical bag-of-words (BoW) (Goldberg, 2017), in a total of 24,269 unique nouns for the 3,298

[3] https://data.worldbank.org/indicator/IT.NET.USER.ZS

users considered. The classifier output is the geolocation country but the target values depend on the machine learning goal. The first goal is detailed in Section 3.3.4. The second goal is applied during the error analysis procedure (Ng, 2018), to verify whether the GTN errors are solvable by machine learning. The third goal, termed the GTN2 method here, is used to reduce the number of GT queries. Similarly to other Web query geolocation methods (for example, based on Google Maps), GTN requires a substantial computational effort because of the large number of GT requests. To solve this problem, we use GTN as an oracle, providing the target classification responses for the machine learning methods.

We explore four classification algorithms with powerful learning capabilities Hastie, Tibshirani, and Friedman (2008) and LeCun, Bengio, and Hinton (2015): bagging (BG), random forest (RF), support vector machine (SVM), and a deep learning multilayer perceptron (MLP).

Breiman's bagging or bootstrap aggregation algorithm (BG) trains t independent classifiers on a given training set by sampling, with replacement, instances from the training set. The essential idea is to average noise and avoid overfitting by using unbiased models that reduce the variance Hastie, Tibshirani, and Friedman (2008). Bagging is normally applied using decision trees as the individual weak learners, which corresponds to the BG model used in this work.

RF is a successful model that was proposed in 2001: it combines t decision trees based on bagging and random selection of input features Breiman (2001). RF tends to obtain good classification results even when using its default parameters and when no feature selection method is adopted Hastie, Tibshirani, and Friedman (2008). In a recent large comparison study, the RF classifier was ranked as the best classifier among 17 of the main machine learning types of algorithms Fernández-Delgado et al. (2014).

SVM are widely used in text classification (Joachims, 1998). The model is based on a maximized margin criterion (Wang and Xue, 2014). For binary classification, the SVM algorithm can compute the best separating hyperplane in a feature space, which is defined by a kernel transformation. In this work, we adopt the linear kernel, because it is very fast and works well with high-dimensional input features, which is the case with our nouns dataset. The model contains one hyperparameter (C) that controls the tradeoff between fitting the errors and obtaining a smooth decision boundary. Because we have 54 class labels, we used the one-vs-rest multiclass classification, which involves training a single classifier per class Bishop (2007).

Moreover, recent remarkable developments were proposed in the field of deep learning, leading to neural network architectures that obtained the best

results in diverse competitions (for example, computer vision and natural language processing) Goodfellow et al. (2016). Such success revived the popularity of the MLP neural model. In this work, we assume a modern MLP representation, also known as deep feedforward neural network LeCun, Bengio, and Hinton (2015), with three hidden layers (with h_1, h_2, and h_3 hidden nodes) that uses Goodfellow et al. (2016): the ReLU activation function on all hidden units, the Softmax function on the output layer, a dropout regularization, and early stopping (to reduce overfitting).

All classifiers are evaluated by using an external 10-fold cross-validation scheme, as explained in Section 3.3.5. For each of the 10 cross-validation iterations, the available data is divided into training data (90% of the instances) and test data (10%). The test data is used to measure the classification performance of the selected models. The training data is used to fit the machine learning models and to perform the hyperparameter selection. To reduce the bias towards a particular model Hand (2006), we apply the same hyperparameter selection procedure for BG, RF, SVM, and MLP. Using standard practice Hastie, Tibshirani, and Friedman (2008) and Ng (2018), the training data is further split into training and validation sets (internal holdout validation). The training set, with 80% of the training instances (0.8×0.9 =0.72% of all available data), is used to fit the classifier. The validation set, with the other 20% of the training data examples (0.18% of all data), is used to monitor the best generalization capability, in terms of global classification accuracy, associated with a hyperparameter or set of hyperparameter values. After selecting the hyperparameters, the machine learning model is retrained with all training data. To provide a fair comparison, we applied a grid search with 10 different hyperparameter combinations for each machine learning algorithm. For BG and RF, the number of trees ranged through $t \in \{50, 100, 150, 200, 250, 300, 500, 1000, 1500, 3000\}$. For SVM the C parameter was searched using $C \in \{0.01, 0.05, 0.1, 0.2, 0.5, 1, 5, 10, 50, 100\}$. For MLP, we tested ten different MLP models, which correspond to different combinations of numbers of hidden nodes and dropout values, as detailed in Table 3.3. The number of MLP inputs is large, because it includes all unique dataset nouns. Therefore, to reduce computational effort, and following what is suggested in Walczak and Cerpa (1999), the MLP combinations assume a decreasing hidden layer size structure, where $h_1 > h_2 > h_3$. The other parameters were set to their default values, as implemented using the `keras` and `sklearn` Python modules.

Because the country classes are unbalanced (for example, 45.7% of users are from the USA, while only 0.1% are from Brazil; see Table 3.2), we applied an oversampling procedure Batista, Prati, and Monard (2004) to all training sets of the machine learning algorithms. The goal is to improve classifier performance for the minority classes by performing random sampling, with repetition, such

TABLE 3.3: Different MLP models tested during the hyperparameter selection stage.

Model Number	Hidden layer size 1 (h_1)	Hidden layer size 2 (h_2)	Hidden layer size 3 (h_3)	Dropout
1	200	100	70	0.4
2	200	100	70	0.3
3	300	150	50	0.4
4	300	100	50	0.4
5	500	200	100	0.4
6	500	200	50	0.4
7	200	150	50	0.4
8	200	150	50	0.3
9	500	150	70	0.4
10	500	100	50	0.4

that the training set becomes balanced. We note that we did not consider undersampling because some classes are very rare, and so undersampling would lead to very small training sets. In addition, the test sets retain the original unbalanced class distribution.

3.3.4 Benchmark methods

For comparison purposes, we selected a recent WD geolocation benchmark method (BM) Lee et al. (2015) that can be simulated using similar procedures and tools already used in this research. The BM method first uses an NER tool (`Stanford CoreNLP`[4]) to extract geolocation terms. The terms are fed to Google Maps to obtain the geographic coordinates. When Google Maps does not return a single country, this is considered an ambiguous case, which is then estimated by using a machine learning algorithm: naive Bayes, SVM, or RF. Using only training data (the BoW approach), the algorithm is fitted to the subset of unambiguous cases and then used to predict all ambiguous cases, including those from the test data. Because RF achieved the best results in Lee et al. (2015), we adopt this learning classifier for BM. We also test a hybrid benchmark method (BM2), which works similarly to BM except that the ambiguous cases are estimated using GTN instead of the learning classifier (RF).

3.3.5 Evaluation

The created Twitter dataset is described in Section 5.3.1; it includes 3,298 users (instances) related to 54 countries. The input features consist of 24,269 unique

[4] https://stanfordnlp.github.io/CoreNLP/

nouns. The countries were identified by the ground truth procedure that is based on a conservative double-source verification, which considers both metadata (user profile location field) and LIW, given all historical tweets (744,830 messages). The Twitter user country geolocation is modeled as a multiclass task (with 54 output labels), and so common classification performance metrics are adopted. The confusion matrix maps predicted values to actual values. From this matrix, several multiclass performance measures can be computed. For a particular class c, we use Witten et al. (2017): accuracy$_c$ (Acc$_c$), precision$_c$, recall$_c$, and F1-score$_c$.

To obtain a single performance measure from the multiclass results, we adopt global accuracy (Acc), which is widely used in classification tasks. The F1-score is a more reliable measure when the data are unbalanced,which is true in our case (as shown in Table 3.2). Therefore, we also compute a single global F1-score by performing a weight averaging operation (WF1), in which each F1-score is weighted proportionally to the class frequency in the data. The evaluation metrics were computed using the `sklearn` module.

GTN is a statistical approach that does not require training data. Nevertheless, for comparison with the machine learning approaches (Section 3.12), we adopt the popular 10-fold cross-validation scheme (Section 3.2) in all comparison tests. The data are randomly split into ten equal-sized folds; then, using a rotation scheme, one fold is selected for testing and all of the others are used for training (if needed by the method). This results in 10 sets of predictions and desired values for each method. To aggregate the results, we average the $k = 10$ distinct classification performance results, and the statistical significance is obtained by applying the nonparametric Mann-Whitney test Hollander and Wolfe (1999).

3.4 Experimental evaluation

3.4.1 Google Trends nouns results

We conducted preliminary experiments with GTN, to tune the method. The preliminary experiments considered a random subset of our data related to 267 users (8%). Adopting the EQ and JF methods, we first tested distinct pruning threshold values, which were based on some noun distribution statistics (median, sixth percentile, third quartile, mean): $p \in \{112, 156, 298, 770\}$. The best results (with an accuracy of 76.0%) were achieved for $p = 298$, which was fixed. Using the same preliminary sample, we then compared different weighting methods for the country confidence scores and country classification, in a total of six GTN models (Section 3.4). The best classification results

were achieved by the first model, which uses EQ and JF, becoming the selected configuration for the GTN method.

TABLE 3.4: Comparison of different GTN weighting and country classification strategies (**bold** denotes best value).

Model	Score Weighting	Classification Strategy	Acc
1	**EQ**	**JF**	**76.0**
2	EQ	AF	73.0
3	IU	JF	56.6
4	IU	AF	40.1
5	NF	JF	75.3
6	NF	AF	45.3

The average 10-fold country geolocation results for GTN and benchmark methods are presented in Table 3.5. When analyzing both classification metrics, global accuracy (Acc) and weight-averaging F1-score (WF1), the comparison clearly favors GTN with respect to the state-of-the-art WD method (BM), showing a substantial difference (15.7 percentage points for Acc and 8.5 percentage points for WF1) that has statistical significance. The hybrid NER GTN method (BM2) provides better performance than BM, indicating that GTN handles the ambiguous cases better than RF. Nevertheless, GTN achieves the best overall results, with an improvement of 2.3 percentage points for Acc and 1.8 for WF1, although these are not statistically significant.

TABLE 3.5: Country geolocation results (in %, best dataset values in **bold**).

Metric	BM	BM2	GTN
Acc	64.9	78.3	**80.6**$^{\diamond}$
WF1	72.8	79.5	**81.3**$^{\diamond}$

$\diamond$ – Statistically significant under a pairwise comparison when compared with BM (p-value < 0.05).

3.4.2 Error analysis

To better understand the errors produced by GTN, we performed an error analysis Ng (2018), in which we manually inspected a total of 638 Twitter user accounts related to GTN country misclassification examples. 3.6 details the errors in terms of four main categories (**error type** column). There are 76 cases (11.9%) for which GTN provided the correct classification (error type A) when

the conservative ground truth method (Section 5.3.1) was wrong. These cases are mostly related to user metadata with ambiguous geolocation terms that can refer to more than one anglophone country (for example, "Newport" city can refer to USA or UK; see Table 3.7). We have recomputed the classification performance for GTN, BM, and BM2 by using the manually adjusted 76 "true" cases. The results obtained are presented in Table 3.8, which confirms that the "true" classification performance for GTN is actually higher than the results shown in Table 3.5. In fact, in Table 3.8 the GTN achieves an Acc of 83.0% and a WF1 of 83.4%. We particularly note that GTN statistically outperforms both benchmark methods (BM and BM2) when adjusted to the "true" values. A common GTN error (type B) is an anglophone country mismatch (32.0%, e.g., UK or Canada instead of USA). There are also some errors (type C, 3.1%) related to proximate countries when considering the location (e.g., Belgium and Netherlands) or language (e.g., Portugal and Brazil). Most GTN mismatches (type D, 53.0%) are related to other mismatches not included in the previous error types. Table 3.7 reports some examples of the A, B, C, and D error types. In the Table 3.7, the user name is omitted for privacy reasons.

TABLE 3.6: Error analysis for GTN.

Error type	**Number**	**Percentage**
Correct classification (A)	76	11.9
Anglophone mismatch (B)	204	32.0
Close country by language or location (C)	20	3.1
Other mismatches (D)	338	53.0
Total	638	100.0

TABLE 3.7: Examples of misclassified locations.

Error type	**Lang.**[a]	**Metadata location**	**Ground truth**	**GTN**	**Manual assessment**
A	EN	Newport	USA	UK	UK
A	EN	North East	USA	UK	UK
B	EN	Scotland	UK	USA	UK
C	NL	Mechelen	Belgium	Netherlands	Belgium
C	ES	Barcelona	Spain	Guatemala	Spain
C	EN	Suri	India	Bangladesh	India
C	PT	Portugal	Portugal	Brazil	Portugal
D	ES	Philadelphia	USA	Colombia	USA

Language: English (EN), Dutch (NL), Portuguese (PT), Spanish (ES).

TABLE 3.8: Country geolocation results for the adjusted ground truth (in %, best dataset values in **bold**).

Metric	BM	BM2	GTN
Acc	63.6	79.1	83.0$^{\diamond}$
WF1	71.6	80.1	83.4$^{\diamond}$

$\diamond$ – Statistically significant under a pairwise comparison when compared with BM and BM2 (p-value < 0.05).

To better exemplify how the nouns can be associated with countries, we present the distribution of the ten most frequent nouns used by the GTN method to identify the country. Table 3.9 is related to a sample of four anglophone countries (Australia, Canada, UK, and USA), while Table 3.10 shows the most frequent nouns for four examples of non-anglophone countries (Finland, Italy, Pakistan, and Singapore). To create the tables, we considered all nouns from all users that were correctly classified by the adjusted GTN model of Table 3.8. The respective classification accuracy (Acc) values for the selected country examples are: Australia – 80%, Canada – 32%, UK – 81%, USA – 94%, Finland – 75%, Italy – 100%, Pakistan – 74%, and Singapore – 100%.

Tables 3.9 and 3.10 show specific geographic terms that can be used to identify the country, working similarly to an NER tool. These include geographic nouns such as: "australia", "sydney", "canada", "scotland" (Table 3.9); and "finland", "oulu", "pakistan" (Table 3.10). GTN also benefits from language differences, as shown by the Italian examples of Table 3.10. However, even when considering the English language, there are also non-geographic terms (not used by NER) that do seem country specific and so can contribute added discrimination capability to GTN. For instance, "brexit" is associated with the UK, while "trump" is related to the USA. For Pakistan there are several other examples of country-specific terms, such as "maryamnsharif" (popular Pakistani politician), "cricket" (highly popular in the country), and "allah" (religion). A different interesting example is provided by the term "thanks", which is used in three anglophone countries (Canada, UK, USA) but with different frequencies (e.g., 0.46% in Canada vs 0.22% in USA). This might be because of cultural differences between countries. In contrast, there are other nouns that are often used with similar frequencies, such as "time" (0.39% for Canada and USA) and "year" (0.29% for Canada and 0.33% for USA). These generic nouns limit the GTN capability to discriminate between countries that use the same language, as shown by the anglophone errors of Table 3.6.

Following Table 3.6, we performed another error analysis step in which machine learning was used. We considered two machine learning error analysis setups:

TABLE 3.9: Most frequent nouns for four examples of anglophone countries.

Australia		**Canada**		**UK**		**USA**	
Word	Frequency	Word	Frequency	Word	Frequency	Word	Frequency
year	0.35%	canada	0.51%	time	0.46%	time	0.39%
time	0.31%	thanks	0.41%	people	0.42%	people	0.34%
people	0.30%	time	0.39%	news	0.34%	year	0.33%
australia	0.28%	year	0.29%	thanks	0.33%	news	0.26%
world	0.28%	business	0.29%	year	0.32%	trump	0.25%
news	0.26%	project	0.27%	work	0.29%	work	0.24%
work	0.24%	industry	0.27%	brexit	0.28%	world	0.23%
business	0.24%	news	0.24%	christmas	0.27%	life	0.22%
industry	0.22%	work	0.24%	scotland	0.26%	years	0.22%
sydney	0.21%	check	0.24%	government	0.24%	thanks	0.22%

TABLE 3.10: Most frequent nouns for four examples of non-anglophone countries.

Finland		**Italy**		**Pakistan**		**Singapore**	
Word	Frequency	Word	Frequency	Word	Frequency	Word	Frequency
congratulations	0.34%	sono	0.50%	pakistan	1.16%	china	0.62%
camp	0.22%	perch	0.40%	maryamnsharif	0.73%	steel	0.62%
finland	0.22%	anche	0.40%	people	0.58%	price	0.47%
business	0.22%	stato	0.30%	allah	0.58%	prices	0.47%
thesis	0.22%	grande	0.30%	world	0.44%	time	0.47%
time	0.22%	posso	0.30%	cricket	0.44%	year	0.47%
seminar	0.22%	prima	0.30%	pakistani	0.44%	data	0.47%
technology	0.22%	bella	0.30%	morning	0.44%	report	0.47%
oulun	0.22%	bello	0.30%	imran	0.44%	conference	0.47%
oulu	0.22%	alla	0.30%	army	0.44%	trade	0.47%

- **I** – The 204 misclassified user examples who live in anglophone countries (Table 3.6) are removed from the dataset and are always used as the same test set in the 10 iterations of the 10-fold procedure. The remaining dataset examples pass through a 10-fold validation, to generate 10 training sets and learning models that are tested on the same 204 test set cases.

- **II** – Similar to the previous setup, except that the fixed test set is composed of all $638 - 76 = 562$ "true" misclassified users (Table 3.6).

The machine learning models require a substantial computational effort because the nouns dataset is high-dimensional, with 24,269 features and 3,298 instances. To reduce the computational effort, the hyperparameter selection is first applied to the dataset, from Section 5.3.1. The best hyperparameters for each classifier are then fixed and used in the 10-fold evaluation of all machine learning comparisons (setups I and II and experiments of Section 3.4.3). The hyperparameter selection procedure uses a 10-fold validation. During each 10-fold iteration, the training data is split using an internal holdout (80%/20%).

For each learning algorithm, ten different models (described in Section 3.3.3) are trained. The best hyperparameter values are selected as the best 10-fold mean global accuracy (Acc) and this resulted in: BG – $t = 300$ trees, RF – $t = 150$ trees, SVM – $C = 0.01$, and MLP – model 7 of Table 3.3 ($h_1 = 200$, $h_2 = 150$, $h_3 = 50$, dropout=0.4).

The machine learning error analysis results are presented in Table 3.11. The obtained classification measure values (WF1 and Acc) range from 21% (setup I, Acc, and RF) to 50.8% (setup I, WF1, and SVM). The best results were obtained by BG (setup I) and SVM (setup II). Globally, low performances were achieved, in particular, if compared with the machine learning results of Table 3.12. The machine learning difficulties in classifying both the anglophone misclassified users (setup I) and the GTN uncorrected responses (setup II) reinforce the competitiveness of the GTN approach.

TABLE 3.11: Machine learning error analysis results (in %, best values in **bold**).

	Classification metric							
	Acc				**WF1**			
Setup	BG	RF	SVM	MLP	BG	RF	SVM	MLP
I	**41.7**$^{\diamond}$	21.3	38.8	23.7	**50.8**$^{\diamond}$	29.7	45.5	28.8
II	40.2	31.2	**43.1**$^{\diamond}$	34.3	42.2	32.4	**44.4**$^{\diamond}$	30.0

$\diamond$ - Statistically significant under a pairwise comparison when compared with other models (p-value < 0.05).

3.4.3 Machine learning classification results

While the proposed GTN approach provides competitive country geolocation results (Table 3.5), it requires a substantial computational effort in terms of GT requests. During the experiments performed in this work, a total of 24,269 GT queries were executed: one for each distinct noun, requiring an average of 1.4 s for each GT query. Because there are 3,298 users, the average user GTN response time is 10.3 s.

To reduce the GTN request effort, we tested whether the GTN classification responses could be directly modeled as targets by the machine learning methods (the GTN2 method). The 10-fold average test results for GTN2 are shown in Table 3.12. The best values were achieved by the deep learning method (MLP), which outperforms other machine learning models for both classification metrics, presenting a statistical significance when compared with BG,

RF, and SVM (for Acc), and BG and RF (for WF1). MLP obtained a high-quality predictive performance (Acc of 80% and WF1 of 77%). Using an Intel Xeon E5 2.30-GHz computational server, the whole MLP training (for one 10-fold iteration) required approximately 1,200 s and the MLP testing time is much faster, requiring approximately 3 ms per user. These results confirm that GTN2 is a valuable and computationally fast alternative to GTN. For future multiclass machine learning comparisons, the data used in this section has been made publicly available at https://github.com/paolazola/Twitter-country-geolocation.

TABLE 3.12: Country geolocation results for GTN2 (in %, best values in **bold**).

Metrics	BG	RF	SVM	MLP
Acc	61.3	69.6	73.8	**80.3**$^{\diamond}$
WF1	64.2	66.2	76.2	**77.4***

$\diamond$ – Statistically significant under a pairwise comparison when compared with RF, BG, and SVM (p-value $<$ 0.05).
$*$ – Statistically significant under a pairwise comparison when compared with RF and BG (p-value $<$ 0.05).

3.4.4 Demonstration application

To further demonstrate the applicability of GTN, we assume a decision scenario in which an analyst wants to distinguish the country of interest of Twitter users that tweet about commodity prices. New data was fetched during the first week of January 2019: this comprised the last 10 days of public tweets of users that typed at least one of the keywords {"copper commodity", "sugar commodity", "cotton commodity", and "silver commodity" }. The original user sample was composed of 100 unique accounts. The Twitter profiles of these users were manually inspected, analyzing both the metadata and historical tweets, to detect the country of interest. This resulted in a set of 71 users with a clear country label. Although the sample is small, we note that a larger sample (concerning 3,298 steel production-related users) and more robust validation (10-fold) was already tested in 3.4.1. Therefore, the goal of this demonstration is just to show, as a proof of concept, the potential applicability of GTN to other non-steel commodity domains (with other users and more recent Twitter data).

The GTN method was then applied (as detailed in Section 3.4.1) to estimate the country for the set of 71 users. Because the number of users is relatively small, the results are shown in terms of a three-class task that includes the two top countries of Table 3.2: "USA", "India", and "other". The prediction results are shown in Table 3.13, in terms of the confusion matrix and individual class measures (the last three rows show Acc_c, $prediction_c$, and $recall_c$). The obtained results show a very good classification performance for India (17 users, Acc_{India}=90.1%, $precision_{India}$=100.0%, $recall_{India}$=70.8%) and a reasonable classification for USA (39 users, Acc_{USA}=67.6%, $precision_{USA}$=53.8%, $recall_{USA}$=80.8%).

TABLE 3.13: Confusion matrix and classification measures for the GTN demonstration example.

		Target country			
		USA	India	other	Total
GTN predictions	USA	**21**	0	5	26
	India	6	**17**	1	24
	other	12	0	**9**	21
	Total	39	17	15	71
	Acc_c=	67.6%	90.1%	74.6%	
	$precision_c$=	53.8%	100.0%	60.0%	
	$recall_c$=	80.8%	70.8%	42.9%	

Chapter 4

Twitter financial disambiguation and user relevance

This chapter studies the topic of Twitter Financial Disambiguation (TFD), which is relevant to filter financial domain texts after performing a search query. TFD is a nontrivial and relevant task when no unique identifiers (e.g., cashtags) are used, which often occurs with alloy (e.g., steel) and commodity (e.g., coffee) tweets. To automatically perform TFD, we propose a transfer learning approach that uses freely labeled news titles as the data source to train diverse unary and binary learning TFD methods. These include different text handling transforms, adaptations of statistical measures and modern machine learning methods. As a case study, we analyzed the steel prices domain, collecting a recent Twitter dataset. Overall, the best results were achieved by a binary Support Vector Machine (SVM) fed with TFD statistical measures and topic model features, obtaining an 80% and 71% discrimination level when tested with 11,081 and 3,000 manually labeled tweets. The best unary performance (78% and 69% for the same test tweets) was obtained by a Term-Frequency Inverse Document Frequency based Classifier (TF-IDFC). These models were further used to generate a Financial User Relevance rank (FUR) score, aiming to filter relevant users. The SVM and TF-IDFC FUR models obtained a predictive user discrimination level of 80% and 75% when tested with a manually labeled test sample of 418 users. These results attest that the proposed jointly TFD-FUR approach is a valuable tool to automatically select Twitter texts and users for financial expert systems (e.g., sentiment analysis, detection of influential users).

4.1 Problem specification

More than 300 million people use Twitter every month, which results in 500 million tweets sent each day[1]. Thus, Twitter is a powerful big data source

[1] https://blog.hootsuite.com/twitter-statistics/

TABLE 4.1: Financial domain sentiment analysis studies.

Study	Target[a]	Markets[b]	Textual Data[c]	Period
Bollen, Mao, and Zeng, 2011	SI	DJIA	TW	2008-2008
Lechthaler and Leinert, 2012	CF	CO	TR	2003-2010
Feuerriegel and Neumann, 2013	CF	G, CO	TR	2003-2012
Rao and Srivastava, 2013	SI,CF,F,VIX	DJIA, NAS, CO, G, EUR/USD	TW,SVI	2010-2011
Pröllochs, Feuerriegel, and Neumann, 2015	SI	TRD	RA	2004-2011
Nguyen, Shirai, and Velcin, 2015	S	18S	YFMB	2012-2013
Pagolu et al., 2016	S	MS	TW	2015-2016
Li et al., 2017	CF	CO	TR	2008-2014
Oliveira, Cortez, and Areal, 2017	S,P	SP, RSL, RMRF, DJIA, NAS, HML, MOM, SMB, VIX, Pind, Psize	TW	2012-2015
Daniel, Neves, and Horta, 2017	S	DJIA	TW	2013-2015
Maslyuk-Escobedo, Rotaru, and Dokumentov, 2017	CF	CO, NG, PR, GAS, HO	TR	2003-2014
Huang et al., 2018	S,B,CF,F,H	SP, HPI, 3-YGB, USD, TRC	TR	1998-2016
Mudinas, Zhang, and Levene, 2019	SI,S,F	DJIA, APPL, GOOGL, HP, JPM, EUR/USD, GBP/USD	FT,Re,TW	2011-2014
Groß-Klußmann, König, and Ebner, 2019	SIF	Related to Europe, USA, Asia and Australia	TW	2010-2018

[a] B: Bond, CF: Commodity Futures, F: Forex, H: Housing prices, P: Portfolio, S: Stocks, SI: Stock Index, SIF: Stock Index Futures, VIX: Volatility Index.
[b] 18S: 18 different Stocks quoted on DIJA, 3-YGB: 3 Years Government Bond, AAPL: Apple, CO: Crude Oil, DJIA: Dow Jones Industrial Average, EUR/USD: forex euro US dollar, G: gold, GAS: gasoline, GOOGL: google, GBP/USD: forex Great British Pound and US dollar, HO: heating oil, HML: high minus low, HP: Hewlett-Packard, HPI: housing price index, MOM: momentum factor, MS: Microsoft, NG: natural gas, NAS: Nasdaq, PInd: 10 Industry Portfolio, PR: propane, PSize: Portfolio formed on size, SMB: small minus big, SP: S&P500, RMRF: excess return on the market, RSL: Russell 2000, TRC: Thomson Reuters commodity prices, USD: US dollar currency, VIX: volatility Index.
[c] FT: Financial Times, TR: Thomson Reuters, TW: Twitter, RA: Regulatory Announces, Re: Reddit, SVI: Search Volume Index from Google, YFMB: Yahoo Finance Message Board.

of freely opinionated texts for social media based expert systems, with a wide range of applications, including sports and political event detection (Adedoyin-Olowe et al., 2016) and inferring the user country of interest (Zola, Cortez, and Carpita, 2019).

In particular, there has been a recent research trend of using social media sentiment analysis for financial expert systems (Oliveira, Cortez, and Areal, 2017; Groß-Klußmann, König, and Ebner, 2019). Regarding Twitter, the most common approach to retrieve texts is based on a keywords match by using the Application Programming Interface (API). It is easy to extract tweets about financial stocks or indexes, since specific company cashtags are commonly used (e.g., the cashtag $AAPL univocally identifies the Apple technology stock prices) (Oliveira, Cortez, and Areal, 2017). As shown in Table 4.1, several research studies used these unique cashtag identifiers to analyze the sentiment of tweets related to company stocks or indexes (e.g., (Pagolu et al., 2016; Oliveira, Cortez, and Areal, 2017)). However, there is scarce research addressing alloy or commodity prices social media texts. The few studies in this domain used mostly texts from authoritative sources, such as Thomson Reuters (Lechthaler and Leinert, 2012; Li et al., 2017). In fact, Twitter sentiment analysis in this domain is not as simple as for financial stocks, since alloy and commodity texts do not typically have a unique ticker. Thus, a generic keywords search needs to be used (e.g., *silver prices*). Yet, this often results in misleading tweets. This problem was recently pointed out by Groß-Klußmann, König, and Ebner (2019), which detected a large amount of noisy tweets when using generic keywords for filtering stock index futures and thus needed to use *a priori* list of known financial experts to filter the data.

As a demonstration example, we random extracted 100 tweets using the keywords *cocoa*, *silver price* and *steel price*. After a manual inspection of the tweets, we found that only 13%, 43% and 47% of tweets were related respectively to cocoa, silver and steel in sense of financial materials. When using the keyword *steel price*, four of the extracted tweets were:

1. “us stainless steel sheet prices moved up to start april as mills lowered base price discounts and demand increased";
2. “galvanized steel sheet roofing corrugated iron prices";
3. “sale stainless steel commercial kitchen list price";
4. “low prices on our top selling cylinder blanks in brass steel follow link below".

All four tweets are related with steel products but only the first two refer to steel industrial production. In effect, the last two are relevant for retail consumers and thus should be discarded when executing alloy steel price analytics. And Word Sense Disambiguation (WSD) methods, which disambiguate words based on lexicons (e.g., *commercial bank* versus *river bank*), do not distinguish well these tweets. For instance, when we apply the known Lesk WSD (Banerjee and Pedersen, 2002), the resulting synsets classify all four tweets as not related to alloy steel. Following this WSD limitation, in this chapter we introduce the concept of Twitter Financial Disambiguation (TFD), which can be seen as a form of text classification specifically built for filtering financial tweets when the search keyword has an unique meaning but that can be related with different contexts (e.g., *steel sheet* versus *steel kitchen*).

Within our knowledge, no studies have performed Twitter sentiment analysis of alloy or commodity prices (Table 4.1), which could be due to the difficulty of retrieving the relevant texts. In this chapter, we approach the TFD concept aiming to solve this filtering task. As a case study, we consider alloy steel prices, which is a financially relevant domain. Steel is the fourth most commonly used metal in the world and it is highly important to the global economy, since trends in production are an indicator of the health of a country's economy[2]. In the United States, the steel industries employee around 142,000 people and about 6.5 million Americans are employed by steel-consuming companies[3].

To address the TFD task, we propose an automatic transfer learning approach (Pan and Yang, 2010), in which freely available labeled news titles are used to compute the statistical and learning models. Two main transfer learning strategies are explored, based on having access to a training set of news titles with only positive financial texts (unary classification) or with positive and negative examples (binary case). For the former strategy, we adapt different distance measures (cosine and dynamic time warping), autoencoders (simple and deep learning), a Term-Frequency Inverse Document Frequency

[2] https://www.focus-economics.com/blog/steel-facts-commodity-explainer

[3] https://money.cnn.com/2018/03/07/news/companies/trump-tariffs-steel-jobs/index.html

Classification (TF-IFC) measure and a One-Class Support Vector Machine (OC-SVM). For the latter strategy, we adapt several distance and statistical measures (e.g., cosine, information gain, TF-IDFC) and also explore three supervised machine learning algorithms: Random Forest (RF), Support Vector Machine (SVM) and deep Multilayer Perceptron (MLP). All TFD methods generate a relevance score for each tweet. We aggregate these scores, aiming to create a Financial User Relevance rank (FUR) score, which indicates the degree of relevance of a user, thus being useful for filtering users (e.g., Twitter users that are interesting to follow). As explained in Table 4.3, most research studies measure user influence or expertise by adopting specific user data (e.g., metadata, historical tweets) or social network graph analysis. The novelty of the FUR score is that it only considers the texts retrieved by the keywords query, thus it does not require an access, storage and analysis of user metadata, historical tweets or social network interaction data. The main contributions of our jointly TFD-FUR approach are:

1. we address the TFD task, focusing on the case study of alloy steel prices;
2. we use freely and easily available news titles to compute the TFD models, thus making use of a transfer learning approach that avoids a costly human labeling;
3. we compare several TFD unary and binary learning approaches that are based on novel adaptations of statistical measures and modern machine learning algorithms;
4. we propose a new FUR score that only considers the texts returned by a keywords Twitter query;
5. we collect and analyze a recent alloy steel Twitter dataset that is publicly made available for further TFD researches.

The chapter is structured as follows. Section 4.2 details the related work about text classification and user relevance. Then, Section 4.3 describes the proposed approach, which includes TFD and FUR methods. Next, Section 4.4 reports the data used (Section 4.4.1), experiments performed and the obtained results (Section 4.4.2 and Section 4.4.3).

4.2 Related work

4.2.1 Twitter financial disambiguation

The proposed TFD concept is related with the research topics of Text Similarity (TXS), WSD and Topic Modeling (TM), all related to text classification. Table

4.2 summarizes the most relevant studies covering these topics, assuming a chronological order and a particular focus on short texts, as provided by microblogs. The table contains the following columns: **Aim** – the main research topic (TXS, WSD, TM or TFD); **Learning** – use of unsupervised or supervised learning (with labeled data); **Text size** – use of long or short (microblog) texts; **Training source** – data used to tune or train the method (if any and when different from target source); **Token handing** – preprocessing method used to handle the texts; **Model** – model adopted for the research topic; **Target source** – data where the model was validated; **Metrics** – model performance metrics; and **Validation** – type of validation method (e.g., $k-$fold cross validation, rolling window).

TABLE 4.2: Summary of the related work for Financial Twitter Disambiguation (TFD).

Study	Aim[a]	Learning[b]	Text size[c]	Training source[d]	Token handling[e]	Model[f]	Target source[g]	Metrics[h]	Validation[i]
Banerjee and Pedersen, 2002	WSD	U	-	WN	STR	Lesk	SensEval-2	ACC	-
Liu, Zhou, and Zheng, 2007	TXS	U	S	WN	STR	DTW	-	COR	-
Yan et al., 2013	TM	U	S	-	STR	BTM	Tweets2011, Q&A, 20NewsGroup	ACC,Purity, NMI,ARI	5-CV
Iosif and Potamianos, 2015	TXS	U	L	YS,G	STR, BOW, TFIDF	PCTXS	Charls Miller, MeSH	COR	-
Kenter and De Rijke, 2015	TXS	S^2	S	AWE	W2V, WE	SVM	MSC	ACC	-
Song and Roth, 2015	TXS	U	S	-	W2V	DESA	Lee, Pincombe, and Welsh, 2005, ACE2005, Chang et al., 2008	ACC,F1,COR	5-CV
Zhang, Yang, and Jacob, 2015	WSD	U	L,S	Wiki	STR	LMSK	AQUAINT, Blog06	MAP	-
Amiri et al., 2016	TXS	U,$S^{>2}$	-	-	WE	CS AE	SCWS, Q&A	MAP,MRR	HO
Neculoiu, Versteegh, and Rotaru, 2016	TXS	$S^{>2}$	S	-	WE	SiRNN	Job titles	ACC	-
Lim, Karunasekera, and Harwood, 2017	TM	U	S	-	STR	ClusTop	Twitter	TC,PMI, P,R,F1	4-CV
Chaplot and Salakhutdinov, 2018	WSD	U	-	WN	STR	WSDTM	SemEval(-2,-3,-2013,-2015)	F1	-
Li et al., 2018	TM	U	L,S	-	STR	EW	8 datasets (e.g., Reuters)	ACC, NMI	5-CV
This chapter	**TFD**	**$S^{1,2}$**	**S**	**NT**	**W2V, STR, TF-IDF**	**TF-IDFC, CD,DTW, SiAE, IG,PMI, RF,SVM, MLP**	**Twitter**	**AUC**	**RW**

[a] TFD: Financial Twitter Disambiguation, TM: Topic Modeling, TXS: Text Similarity, WSD: Word Sense Disambiguation.
[b] S: Supervised ([1] – unary texts; [2] – binary texts; $^{>2}$ – more than 2 classes), U: Unsupervised.
[c] L: Long text, S: Short text.
[d] AWE: Augmented Word Embedding, NT: News Titles, YS: Yahoo search, G: Google, Wiki: Wikipedia, WN: WordNet.
[e] BOW: Bag of Words, STR: String, TF-IDF: Term-Frequency Inverse-Document-Frequency matrix, WE: Word Embedding, W2V: Word2Vec.
[f] BTM: Biterm Topic Model, CD: Cosine Distance, CS AE: Context Sensitive Autoencoder, DESA: Dense Explicit Semantic Analysis, DTW: Dynamic Time Warping, EW: Entropy Weighting, LMSK: Language Model and Structural Knowledge, MLP: Multilayer Perceptron, PCTXS: Page Count and Text Based Similarity, SiAE: Siamese Autoencoder, SiRNN: Siamese RNN, SVM: Support Vector Machine, TF-IDFC: TF-IDF Classifier, WSDTM: Word Sense Disambiguation Topic Modelling.
[g] MeSH: Medical Subject Headings, MSC: Microsoft Paraphrase Corpus, Q&A: Question and Answering corpus, SCWS: Word similarity dataset.
[h] ACC: Accuracy, ARI: Adjusted Rand Index, AUC: Area Under the receiver operating characteristic Curve, COR: correlation, F1: F1-score, MAP: Mean Average Precision, MRR: Mean Reciprocal Rank, PMI: Pointwise Mutual Information, NMI: Normalized Mutual Information, P: Precision, R: Recall, TC: Topic Coherence,
[i] HO: Holdout train and test split, k-CV: k-fold Cross Validation, RW: Rolling Window.

Measuring the similarity between two texts (TXS) is a nontrivial task, especially if the texts have different sizes and include slang or abbreviations, often used in short microblog messages. TXS is often achieved by computing a text similarity measure. The most common measures are (Lin, Jiang, and Lee, 2014): Euclidian distance, Jaccard similarity and Cosine Distance. Yet, these traditional measures require vectors with the same length. To solve this issue, Liu, Zhou, and Zheng, 2007 used Dynamic time warping (DTW) for TXS. Other approaches used augmented Web documents (Iosif and Potamianos, 2015). The use of augmented texts is also often adopted for WSD tasks (e.g., WordNet lexical database) (Liu, Zhou, and Zheng, 2007; Chaplot and Salakhutdinov, 2018). Moreover, the WSD works from Table 4.2 combine features extracted using a TM algorithm. The Latent Dirichlet Allocation (LDA)

(Blei, Ng, and Jordan, 2003) is a popular algorithm for TM. More recently, the Biterm Topic model (BTM) method was proposed, aiming to achieve a better TM for short texts (Yan et al., 2013). In Table 4.2, the initial studies were mainly based on string comparisons, with the original words. Recent TXS works use a word embedding (e.g., Word2Vec) to get a numerical representation of the texts (Song and Roth, 2015; Neculoiu, Versteegh, and Rotaru, 2016). Only the most recent studies employ deep learning models, such as recurrent neural networks (Sanborn and Skryzalin, 2015) and autoencoders (Amiri et al., 2016).

The approach proposed in this chapter appears at the last row of Table 4.2. Our approach differs from the ones in the Table 4.2 since it is specifically built for financial tweets already filtered by specific keywords. Only one other study adopted Twitter (Lim, Karunasekera, and Harwood, 2017), performing a topic clustering based on networks of words that automatically define the number of topics, using a series of tweet features (e.g., hastags, mentions and nouns). Moreover, most supervised learning studies used binary labels, while we approach two training setups: unary, in which only positive financial texts are available; and binary, which assumes an access to both positive (financial) and negative (non financial) messages. Since Twitter texts are unlabeled, and in order to avoid a laborious manual effort, we use public and freely available news titles to set the positive and negative messages, thus making use of a transfer learning (Pan and Yang, 2010; Zola et al., 2019). As for the TFD models, we adjust and compare several data preprocessing, statistical measures and machine learning algorithms, including recent Word2Vec encoding and deep learning methods (e.g., siamese autoencoder, deep multilayer perceptron). The models are evaluated using a robust and realistic rolling window procedure (Tashman, 2000; Oliveira, Cortez, and Areal, 2017).

4.2.2 Social media user relevance

In general, there are two main ways to measure what is an influential or relevant social media user: based on user social network features or user data (e.g., metadata, historical texts). Table 4.3 surveys these influential user research approaches, with a particular focus on studies that analyze one specific user relevance topic, as our case study. Table 4.3 includes the columns: **Model** – proposed model to measure user relevance; **User network** – based on the usage of social network attributes (e.g., followers); **User history** – based on the usage of user metadata or historical messages; **Target source**, **Metrics** and **Validation** – similar meaning of Table 4.2.

Most studies of Table 4.3 focus on Twitter. Also, the state-of-the-art works assume two major sources of data: social networks (e.g., graphs of user interactions) and/or user history (e.g., metadata, user past tweets). The former source

TABLE 4.3: Summary of the related work for Financial User Relevance (FUR).

Study	Model[a]	User network	User history	Target source	Metrics[b]	Validation[c]
Yamaguchi et al., 2010	TuRank	X	-	Twitter	AA	-
Castillo, Mendoza, and Poblete, 2011	Fea,SVM, DT,BN	-	X	Twitter	MAE,P,R, ACC,F1	3-CV
Pal and Counts, 2011	Fea,GMM	X	-	Twitter	P,R,COR	-
Gayo-Avello, 2013	PD	X	-	Twitter	Min,Med Mean	-
Ito et al., 2015	LDA,Fea, RF	-	X	Twitter	AUC	10-CV
Cortez, Oliveira, and Ferreira, 2016	Fea	X	-	StockTwits	COR, PQU	RW
Eliacik and Erdogan, 2018	Fea,PgR	X	-	Twitter	COR	10-CV
This chapter	**TFD**	**-**	**-**	**Twitter**	**AUC**	**RW**

[a] BN: Bayesian Network, DT: Decision Trees, Fea: Feature Analysis, GMM: Gaussian Mixture Model, LDA: Latent Dirichlet Allocation, PD: Paradoxical Discounted, RF: Random Forest, SVM: Support Vector Machine, TFD: Financial Disambiguation based.

[b] AA: Average Adequacy, ACC: Accuracy, AUC: Area Under the receiver operating characteristic Curve, COR: Correlation, F1: F1-score, Min: Minimum, Med: Median, P: Precision, PQU: Percentage of Quality Users, R: Recall.

[c] *k*-CV: *k* fold Cross Validation, RW: Rolling Window.

is often modeled by using graph network analysis, computing measures such as indegree or Page Rank (Pal and Counts, 2011; Cortez, Oliveira, and Ferreira, 2016). The latter involves specific user metadata attributes, such as age (Castillo, Mendoza, and Poblete, 2011), or access to user past tweets (Ito et al., 2015). The novelty of our FUR approach (shown in the last row of Table 4.3) is that it works directly over the messages retrieved from a keywords query, with no need to access social network or user history data.

4.3 Proposed Approach

The proposed approach for TFD and FUR is depicted in Figure 4.1 and it includes five main steps: data source, data handling, TFD modeling, evaluation and FUR.

First, a Twitter keywords search is executed, resulting in a set of tweets that should be related with a financial topic but that also include other irrelevant texts. As a case study, this chapter addresses alloy steel prices. For TFD, this chapter adopts a supervised learning, under two main approaches: unary and binary classification. In order to get labeled data for train the models, we use easy to collect and freely available news titles (as detailed in Section 4.4.1). Positive financial texts (P) consist of steel domain news titles (Daudert, Buitelaar, and Negi, 2018), allowing to adjust unary classifiers. To represent Negative texts (N), not related to the financial domain, we use generic news titles. Thus,

binary classifiers are trained using $P \cup N$. The TFD models use a transfer learning (Pan and Yang, 2010; Zola et al., 2019), where the models are adjusted to one training source (news titles) and tested on a different source (Twitter).

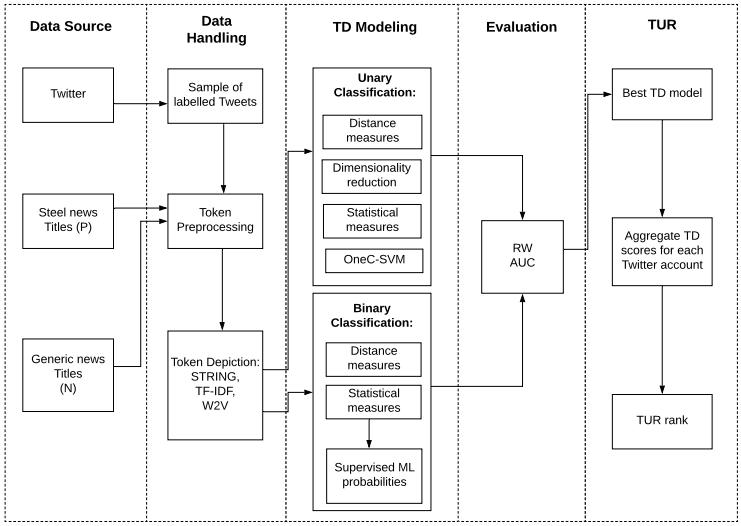

FIGURE 4.1: Schematic of the research approach for TFD and FUR.

In the second step, the collected tweets and news titles are preprocessed. Since the retrieved tweets are not labeled, we manually classify a sample of tweets and users, as explained in Section 4.4.1, in order to build a ground truth dataset that is used for testing the TFD methods and FUR rank. All texts (news titles and tweets) are transformed into a lowercase representation removing punctuation and stop words (e.g., "the", "and"). The resulting tokens might be used directly (as string) or further processed into a numeric representation, via a Term-Frequency Inverse Document Frequency matrix (TF-IDF) or Word2Vec (W2V) transform.

TF-IDF is a common transform for texts (Oliveira, Cortez, and Areal, 2016) that is computed as:

$$\begin{aligned} tf_{i,j} &= \frac{n_{i,j}}{n_{d_j}} \\ idf_i &= \log \frac{n_D}{n_{d:i \in d}} \\ tf\text{-}idf_{i,j} &= tf_{i,j} \times idf_i \end{aligned} \tag{4.1}$$

where $n_{i,j}$ is the number of occurrences of token i in document d_j, n_{d_j} is the number of tokens in document d_j, n_D is the number of documents in the collection and n_d is the number of documents in the collection that contain token

i. W2V is a modern word encoding method that was proposed by Mikolov et al., 2013a. W2V is based on a multilayer perceptron neural network with an input, projection and output layer. This work uses the unsupervised W2V algorithm with Continuous Bag-of-Words Model (CBOW) that is implemented at the `gensim` module in Python. The algorithm includes only one hyperparameter, the embedding size E (vector size for each token). To fix the hyperparameter, the embedding size is ranged within the values $E \in \{1, 8, 16, 32\}$.

All tested Supervised Machine Learning (ML) methods (random forest, multilayer perceptron and support vector machine) and autoencoders require a fixed input size but the analyzed texts include a variable number of tokens. To handle this issue, when using direct text token inputs (TF-IDF or W2V), the truncation technique employed in Wood-Doughty, Andrews, and Dredze, 2018; Zola et al., 2019 is adopted, which considers only the first M tokens, as they appear in the texts. If the texts have less than M tokens then we use padding, which consists in adding null values (e.g., 0) (Senin, 2008; Zola et al., 2019). Thus, supervised ML algorithms and autoencoders assume M inputs when using the TF-IDF transform and $E \times M$ inputs when the W2V encoding is adopted.

The third step performs the TFD, under a unary or binary classification. Unary methods include: Cosine Distance (CD) and Dynamic Time Warping (DTW) distance measures; dimensionality reduction via autoencoders; a TF-IDF based statistical measure; and One-class Support Vector Machine (OC-SVM), which is a popular Machine Learning (ML) algorithm for unary classification (Manevitz and Yousef, 2001). As for the binary methods, they include: CD and DTW distance measures; a higher range of adapted statistical measures, namely TF-IDF based, Information Gain (IG) and Pointwise Mutual Information (PMI); and binary classifier ML algorithms, namely Random Forest (RF), Support Vector Machine (SVM) and deep Multilayer Perceptron (MLP).

The fourth step is detailed in Section 4.3.4. It involves the usage of a realistic Rolling Window (RW) evaluation, which includes several train and test model updates through time. The TFD method predictions are contrasted with a tweet labeled sample ground truth, allowing the computation of the Area Under the receiver operating characteristic Curve (AUC).

Finally, in the fifth step, the best TFD model is selected and used to score all tweets. For each distinct Twitter user account, the scores are aggregated, resulting in the FUR rank (Section 4.3.3).

4.3.1 Unary training methods for Twitter Financial Disambiguation (TFD)

The unary methods assume a training data composed by only positive texts (P). In this chapter, these texts are represented by steel domain news titles (Section 4.4.1). The unary models output a TFD relevance score (S_t), which is computed as presented in Table 4.4. The S_t can be interpreted as the degree of proximity of the tweet t to the training data. Thus, the higher is the S_t, the higher is the probability that the tweet t is related to the positive concept. For a binary classification, it is possible to label a text (or tweet) t as a positive class (value of 1) if the respective relevance score is $S_t > T_{\text{TFD}}$, where T_{TFD} is a decision threshold that can range through any value of the S_t function domain; otherwise t is considered as belonging to the negative class (value of 0).

TABLE 4.4: TFD relevance scores when using unary (P) or binary ($P \cup N$) texts.

TFD model	Token handling	Training	TFD relevance scores $(S_t)^a$
CD	TF-IDF,W2V	unary	$\sum_{u \in P} \frac{t \cdot u}{t \cdot u}$
		binary	$\frac{1}{n_P} \sum_{u \in P} \frac{t \cdot u}{t \cdot u} - \frac{1}{n_N} \sum_{v \in N} \frac{t \cdot v}{t \cdot v}$
DTW	TF-IDF,W2V	unary	$-\sum_{u \in P} DTW(t,u)$
		binary	$\frac{1}{n_N} \sum_{v \in N} DTW(t,v) - \frac{1}{n_P} \sum_{u \in P} DTW(t,u)$
SiAE	TF-IDF,W2V	unary	$-\sum_{u \in P} \lVert \boldsymbol{h}_t - \boldsymbol{h}_u \rVert$
TF-IDFC	TF-IDF	unary	$\sum_{i \in t} tf\text{-}idf_{i,t}$
		binary	$\sum_{i \in t} [(tf\text{-}idf1_{i,t}) - (tf\text{-}idf0_{i,t})]$
IG	string	binary	$\sum_{i \in t} IG(i)$
PMI	string	binary	$\sum_{i \in t} PMI(i,1) - PMI(i,0)$

[a] $tf\text{-}idf$ – TF-IDF computed using the positive ($tf\text{-}idf1$) or negative ($tf\text{-}idf0$) texts; n_P – number of positive financial texts; n_N – number of negative texts; DTW – DTW distance function; $\boldsymbol{h}_t$ autoencoder function for text t; $IG(i)$ – IG function for token i; PMI – PMI function computed for token i and positive (1) or negative (0) classes.

In this chapter, we adapt two distance measures: the classical CD and DTW. DTW is popular for time series analysis and it can handle texts with different sizes, without the need of padding, as required by the CD measure (Senin, 2008). The relevance scores proposed in Table 4.4 allow to directly use CD and DTW as TFD unary classification methods.

Since the analyzed texts have different dimension sizes, we also adopt a dimensionality reduction algorithm. Autoencoders (AE) are a type of generative neural network in which the output is the same as the input. In particular, we

use the Siamese Autoencoder (SiAE) (Utkin et al., 2017). The SiAE is trained using positive texts (P), using as inputs the TF-IDF or W2V encoded numerical values. It also includes a squeezed hidden layer, which allows to reduce the texts. After the SiAE structure is trained, it can be used to compress any new texts, including tweets. Two SiAE structures are explored (Table 4.5): a simpler one, with just one encoder and decoder layer with hidden size equal to 1 (Model number 0), and a Deep SiAE, with several hidden layers (10 distinct structures are tested, from Model number 1 to 10). The SiAE networks can directly perform a TFD unary classification by using the relevance score proposed in Table 4.4, where $\boldsymbol{h}_i$ denotes the autoencoder squeezed hidden layer function for text i.

TABLE 4.5: Different SiAE models compared.

SiAE number	Hidden layer size (h1)	Hidden layer size (h2)	Hidden layer size (h3)	Hidden layer size (h4)	Hidden layer size (h5)	Hidden layer size (h6)
0	-	-	-	-	-	1
1	6	5	4	3	2	1
2	10	7	5	4	3	1
3	25	20	15	10	5	1
4	50	40	30	20	10	1
5	150	50	25	10	5	1
6	5	2	-	-	-	1
7	10	5	-	-	-	1
8	20	10	-	-	-	1
9	50	25	-	-	-	1
10	100	50	-	-	-	1

Another unary method is provided by the TF-IDF Classifier (TF-IDFC), which is based on the TF-IDF function of Function 4.1. The idea behind TF-IDFC is that TF-IDF assigns higher values to the most relevant tokens of a text, thus tweets with higher accumulated TF-IDF scores are more likely to be related with the positive concept defined by the training domain. The proposed unary TF-IDFC relevance score is presented in Table 4.4.

The last explored unary method is OC-SVM, which has been used for the classification of texts (Manevitz and Yousef, 2001). In this chapter, we test two OC-SVM kernels: linear and Gaussian. Both models contain the $\nu \in [0, 1]$ hyperparameter, a lower bound for the number of samples that are support vectors and an upper bound for the number of samples that are on the wrong side of the hyperplane. The Gaussian kernel as the γ hyperparameter that controls the bias-variance trade-off. In this chapter, the hyperparameteres were ranged using $\nu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ and $\gamma \in \{0.001, 0.01, 0.05, 0.1, 0.5, 1\}$.

4.3.2 Binary training methods for Twitter Financial Disambiguation (TFD)

The binary methods assume a training data with both positive and negative texts ($P \cup N$). Similarly to the unary case, all binary training methods produce a TFD relevance score (S_t) and a text t is considered positive (value of 1) if $S_t > T_{\text{TFD}}$. The extra negative (generic news) texts allow an adaptation of the TF-IDFC method, as defined in Table 4.4. Moreover, binary texts enable the computation of other information measures, namely IG and PMI, which are popular in text mining tasks (Xu et al., 2007; Oliveira, Cortez, and Areal, 2016). Following the formulation reported in Oliveira, Cortez, and Areal, 2016, for each token i of a text t, IG is computed as:

$$\begin{aligned} IG(i) = p(i,1)\log\frac{p(i,1)}{p(i)p(1)} + p(\bar{i},0)\log\frac{p(\bar{i},0)}{p(\bar{i})p(0)} \\ - p(\bar{i},1)\log\frac{p(\bar{i},1)}{p(\bar{i})p(1))} - p(i,0)\log\frac{p(i,0)}{p(i)p(0)} \end{aligned} \tag{4.2}$$

where the probabilities $p(i), p(\bar{i}), p(1), p(0), p(i,1), p(\bar{i},0), p(\bar{i},1)$ and $p(i,0)$ are derived from the training set ($P \cup N$) and $\bar{i}$ refers to the absence of i. The PMI measures the probability of word co-occurrence in a corpus as:

$$PMI(i,y) = \log\frac{p(i,y)}{p(i)p(y)} \tag{4.3}$$

where $p(y)$ is the probability of occurrence of class $y \in \{0,1\}$ in the set of training documents (corpus). The adapted IG and PMI TFD relevance scores are shown in Table 4.4.

Having access to binary labeled texts also enables the training of supervised ML algorithms. In this chapter, we compare three modern classifiers (Hastie, Tibshirani, and Friedman, 2008; Goodfellow et al., 2016): RF, SVM and a MLP. RF is an ensemble method that combines N_T decision trees based on bagging and random selection of input features. SVM are widely used in text classification (Joachims, 1998), computing the best separating hyperplane in a feature space, which is defined by a kernel transformation. The model includes the C hyperparameter, which controls the trade-off between fitting the errors and obtaining a smooth decision boundary. The adopted MLP, also known as Deep Feedforward Neural Network (DFFN), includes (Goodfellow et al., 2016): the ReLU activation function on all hidden units (with the sizes h_1, h_2 and h_3), the logistic function on the output layer, a dropout regularization of 0.3 and early stopping (to reduce overfitting). Since the TFD task is unbalanced, a undersampling procedure was applied to the ML training data, which reduces the computational cost when compared with oversampling (Batista, Prati, and Monard, 2004). Although the training sets are

balanced, the test data (from Twitter) is kept with the original unbalanced distribution. The ML algorithms were implemented by using the `keras` and `sklearn` Python modules. The tested hyperparameters include: RF – $N_T \in \{10, 50, 100, 250, 500, 1000, 1500, 3000, 5000, 10000\}$; SVM – $C \in \{0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10,$ linear and Gaussian kernel ($\gamma \in 0.001, 0.01, 0.05, 0.1, 0.5, 1\}$); MLP – ten different MLP structures related with different combinations of number of hidden nodes, as detailed in Table 4.6.

The adopted ML binary classifiers (RF, SVM and MLP) output a relevance class probability that can be interpreted as the relevance score $S_t = p(t)$, where $p(t) \in [0, 1]$ denotes the class probability for text t. In terms of input variables, we tested three different types of setups: TF-IDF, W2V or TFD features. TF-IDF and W2V are described in Section 4.3. The last setup is based on TFD binary statistical measures (TF-IDFC, IG and PMI scores, as computed in Table 4.4) and, as proposed in Cai, Lee, and Teh, 2007, k topic relevance features, as obtained using both LDA and BTM text clustering algorithms. Thus, the number of inputs for the TFD features setup is $3 + 2k$ (α_k values for LDA and θ_k values for BTM). To set k, we apply the Griffiths test (Griffiths and Steyvers, 2004) on the sample of binary texts when searching for $k \in \{2, ..., 100\}$.

TABLE 4.6: Different MLP structures compared.

Network number	Hidden layer size (h1)	Hidden layer size (h2)	Hidden layer size (h3)
1	50	25	10
2	100	50	25
3	100	25	5
4	150	100	20
5	150	50	10
6	200	100	50
7	250	200	20
8	300	150	10
9	500	250	50
10	500	100	5

4.3.3 Financial users relevance rank (FUR)

By using a TFD model, the keywords query resulting texts (Q) can be assigned with a financial relevance score $S_t, \forall t \in Q$. Let Q_u denote the subset of Q texts written by user $u \in U$, where U represents the full set of users that have written the retrieved Q texts. The aggregated FUR score is obtained by summing or averaging all user u texts, where $FUR_u = \sum_{t \in Q_u} S_t$ (sum) or $FUR_u = \frac{\sum_{t \in Q_u} S_t}{|Q_u|}$ (mean).

Similarly to the TFD classification case (Section 5.4), a FUR user binary classification can be achieved by adopting a T_{FUR} a decision threshold, which can

range through any $FUR_{u \in U}$ domain value. If $FUR_u > T_{\text{FUR}}$ then user u is classified as relevant (value of 1) for the specific financial application, else it is considered as irrelevant (value of 0).

4.3.4 Evaluation

The TFD models are validated by adopting the realistic rolling window procedure (Figure 4.2) (Tashman, 2000; Oliveira, Cortez, and Areal, 2017). This procedure simulates several training and test model iterations through time (total of I iterations), thus preserving the time order of the news titles and tweets. A fixed time period is used to dimension the training (t_{train}) and test window (t_{test}) texts. In the first iteration, the oldest news titles data are used to train the classifiers. Then, TFD predictions are performed over a Twitter test set, with more recent data. In the second iteration, both the training (news titles) and test (tweets) sets are updated by discarding the oldest texts and adding more recent ones, allowing to train new classifiers and obtain new TFD tweet predictions, an so on. Using the same procedure of Oliveira, Cortez, and Areal, 2017, to get an overall classification performance we average all I iteration predictive performance metrics. Then, we apply the non-parametric Wilcoxon test for measuring statistical significance (Hollander and Wolfe, 1999).

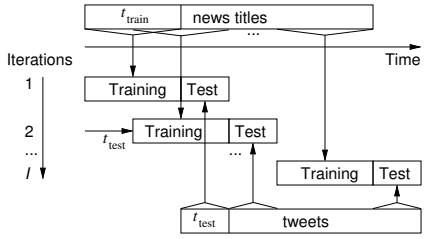

FIGURE 4.2: Schematic of the rolling window procedure.

To compare the different classifiers, we use the popular Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve (Ito et al., 2015), computed on the rolling window test data. The ROC curve shows the performance a binary classifier across all decision threshold values (T), plotting the False Positive Rate (FPR), in x-axis, versus the True Positive Rate (TPR), in the y-axis. The $AUC = \int ROC dT$ measures the global discriminatory performance of a classifier. Often, the AUC values are interpreted as: 0.5 – equal to a random classifier; 0.6 – reasonable, 0.7 – good; 0.8 – very good; 0.9 – excellent; and 1 – perfect. The ROC curve analysis contains two main advantages to evaluate binary classifiers (Fawcett, 2006). First, it is not dependent on a specific decision threshold value, which corresponds to a particular TPR (sensitivity) versus FPR (one minus the sensitivity) trade-off. Second, it

is not dependent on the class frequency, thus it can be applied to unbalanced tasks that often occur in text classification, such as the alloy steel TFD. The evaluation metrics were computed using the Python `sklearn` module.

4.4 Experimental evaluation

4.4.1 Data

The Twitter data were collected from March 2017 to October 2018, using the API service and the `Rtwitter` R tool package. The tweets are written in English and related to the following keywords: *steel price*, *steel industry* and *steel production*. A total of 533,759 tweets were retrieved, related with 270,613 unique users.

Since the collected unlabeled Twitter dataset is quite large, we executed a manual labeling of randomly sampled tweets and users to set the ground truth to validate the TFD and FUR models. We created two sets of binary labeled tweets, with 11,081 and 3,000 texts each. The first set is used to tune the TFD model hyperparameters, thus it can be also viewed as a validation set, and to compare the diverse TFD models. The second set is used as an external test set, to estimate the generalization capabilities of the best TFD models on a different unseen dataset. We note that these tweets are unbalanced, presenting an average around 36% of positive texts. Regarding the Twitter user ground truth, we first filtered users that have at least one non-retweet message. Recently, the steel sector received an increased news coverage due to tariffs imposed by the US Government. As a consequence, many users retweeted steel news just for political reasons, thus the filter allowed to discard a large portion of such users, resulting in 52,653 user accounts. From this set, we randomly selected 418 users that were manually labeled as relevant (1) or irrelevant (0) for the alloy steel domain. The user ground truth set is smaller than the labeled tweets since the manual inspection of a user (e.g., historical tweets, user profile metadata, user web pages) requires much more effort when compared with a single tweet analysis.

To build the training labeled data, we adopted news titles for two main reasons. First, the titles are freely available and easy to collect, while the full news content requires the payment of a fee, specially for steel news media. Second, the length of a title is shorter than the news, thus being closer to the tweet size. The P positive texts were collected from authoritative steel news media: *Kallanish Commodities*[4] and *SteelOrbis*[5]. The news titles are related to

[4] https://kallanish.com/en/

[5] https://www.steelorbis.com/

the same period of tweets, thus from March 2017 to October 2018. The total number of news titles are 20,366 from *Kallanish Commodities* and 9,418 from *StellOrbis*. Regarding the N negative texts, we used three different generic news sources: 2,554 titles from *The New York Times*[6], 2,990 titles from *Reuters*[7] and 44,182 from the dataset built in Kulkarni, 2017[8]. The generic news texts are related to the same time period of the collected tweets and steel news. The news titles and the 3,000 labeled tweets are publicly made available[9].

4.4.2 TFD results

For the TFD model experiments, we adopted a rolling window with a fixed training window size of $t_{\text{train}} = 2$ months and test window of $t_{\text{test}} = 1$ month, which results in a total of $I = 18$ iterations (Twitter test data from May 2017 to October 2018). In the first set of experiments, the overall rolling window test data is composed of the 11,081 labeled tweets. Diverse unary and binary TFD models were compared, using different token handling (as detailed in Table 4.4) and input setups (for the binary ML methods described in Section 4.3.2).

Several of the TFD models include parameters (e.g., E for the W2V embedding size, C value of SVM, M maximum number of tokens). Both tweets and news titles were first preprocessed (e.g., punctuation and stopword removal), resulting in an average size of 7 words for news titles and 14 tokens for tweets. The token truncation value (M), used by the TF-IDF or W2V input ML models, was set to the average text length since preliminary experiments have shown a better performance of average truncation when compared with the max length value. To set the other parameters, a grid search was executed with the ranges described in Section 4.3. Similarly to the work of Zola, Cortez, and Carpita (2019), to facilitate the comparison and select a single model throughout all rolling window iterations, the best average AUC configuration model was selected, as presented in Table 4.7. For comparison purposes, the last rows of Table 4.7 show the results of three methods: the Lesk WSD algorithm (Banerjee and Pedersen, 2002), implemented using the `nltk` Python module; the LDA when the number of topics is set equal to two (aiming to distinguish steel alloy texts); and a supervised binary SVM that is trained using labeled Twitter data and a Bag of Words (BOW) approach (the SVM uses all input words and it is set using the same modeling procedure, namely rolling window with two months of undersample training data and grid search for hyperparameter selection).

[6] https://www.nytimes.com/
[7] https://www.reuters.com/
[8] https://www.kaggle.com/therohk/million-headlines/home
[9] https://github.com/paolazola/Twitter-Financial-Disambiguation-Financial-Users-Relevance

TABLE 4.7: TFD classification performance using the 11,081 labeled tweets (average AUC values, best results when using the same type of training data are in **bold**).

Training	**Model**	**Token handling/ Input setup**	**AUC**
Unary Steel news titles	CD	W2V ($E = 1$)	0.49
	DTW	W2V ($E = 1$)	0.44
	SiAE (network 0)	W2V ($E = 16$)	0.60
	Deep SiAE (network 9)	W2V ($E = 8$)	0.62
	TF-IDFC	TF-IDF	**0.78***
	OC-SVM (linear kernel, $\nu = 0.1$)	TF-IDF	0.76
Binary news titles	CD	TF-IDF	0.64
	DTW	W2V ($E = 16$)	0.72
	TF-IDFC	TF-IDF	0.78
	IG	string	0.60
	PMI	string	0.76
	RF ($N_T = 10000$)	W2V ($E = 8$)	0.75
	SVM (linear kernel, $C = 100$)	W2V ($E = 32$)	0.77
	MLP (network 10)	W2V ($E = 16$)	0.78
	RF ($N_T = 50$)	TFD features ($k = 17$)	0.76
	SVM (linear kernel, $C = 0.001$)	TFD features ($k = 17$)	**0.80**$^{\diamond}$
	MLP (network 6)	TFD features ($k = 17$)	0.79
–	Lesk WSD	string	0.50
News titles	LDA	string	0.52
Twitter	SVM (linear kernel, $C = 0.5$)	BOW	0.91

$\star$ – Statistically significant (p-value < 0.05) under a pairwise comparison when compared with the unary models: CD, DTW, SiAE and Deep SiAE.

$\diamond$ – Statistically significant (p-value < 0.05) under a pairwise comparison when compared with the binary models: CD, RF ($N_T = 10000$), SVM (linear kernel, $C = 100$) and MLP (network 10)).

When analyzing the results, it is relevant to note that the unsupervised Lesk WSD method and the unsupervised LDA provide a poor performance (AUC of 0.50 for Lesk and 0.52 for LDA, equivalent to a random classifier) and that is clearly outperformed by most TFD models. Overall, the best results are achieved by the Twitter trained SVM model (AUC of 0.91). Yet, this model requires a substantial human effort for labeling data, which is prone to errors and it is often unfeasible in practice (e.g., when analysing big data). Regarding the transfer learning models, the best unary performance of AUC=0.78 is provided by the TF-IDFC statistical method, which is fast to compute and does not contain hyperparameters. The TF-IDFC model AUC differences are statistically significant when compared with all unary methods except OC-SVM. The second best unary method is OC-SVM (AUC of 0.76), which uses the same set of TF-IDF input features, followed by the autoencoders (AUC of 0.62 and 0.60). The distance based measures (CD and DTW) achieve the worst unary performances (lower than random classifier). Turning to the binary methods based on string, TF-IDF or W2V tokens, the best results are obtained by TF-IDFC and MLP with W2V, with an AUC of 0.78, which is equal to the unary TF-IDFC performance. Several of the other direct token input binary methods achieve an AUC higher than 0.7 (SVM, RF, PMI and DTW). The binary distance measures (CD with AUC of 0.64 and DTW with AUC of 0.72) obtain a substantial performance improvement when compared with their unary versions (e.g., there is a 28 percentage point increase for DTW). Overall, the best binary performance is achieved by the SVM that uses the TFD features as inputs, obtaining a very good discrimination level (AUC of 0.80), which is statistically significant when compared with 5 other binary models, as shown in Table 4.7. This binary SVM presents an improvement of 2 percentage points when compared with the best unary model (TF-IDFC), although such difference is not statistically significant.

For further TFD model experiments, we selected three best models: the Twitter trained SVM model (for comparion purposes); and the proposed TF-IDFC and the SVM (linear kernel, C =0.001, fed with TFD features, $k = 17$) classifiers, which were the best unary and binary representatives of Table 4.7. A second rolling window procedure was executed, using the same fixed train and test time periods ($t_{\text{train}} = 2$ months and $t_{\text{test}} = 1$ month, 18 iterations). During this execution, we reused the previously trained TF-IDFC and SVM TFD models and performed predictions for all 533,759 collected tweets (labeled and unlabeled). All these predictions were stored, allowing a later filtering of the relevant Twitter predictions, needed to compute the additional TFD (shown next) and FUR (Section 4.4.3) results. Figure 4.3 plots the global ROC curves for the selected TFD models when considering the second (and extra) TFD labeled test set with 3,000 tweets. The global ROC curves were obtained by

merging all the predictions from the 18 rolling window iterations into a single test set (Fawcett, 2006). When executing this additional predictive test, the proposed news titles trained SVM obtains a global AUC value (0.71), which corresponds to a good discrimination level. This model presents the same 2 percentage point difference (as in Table 4.7) when compared with the unary TF-IDFC method (AUC of 0.69). In particular, the ROC curve comparison of Figure 4.3 shows that the news titles SVM provides better TPR values when FPR is low (higher specificity trade-off region) and a very similar TPR results when FPR is high (higher sensitivity area). While the Twitter trained SVM achieves the best results, this model is less useful in practice, since it requires human labeled costly data (as previously discussed). Nevertheless, the comparison results attest the quality of the proposed transfer learning TFD models (e.g., difference of just 9 percentage points).

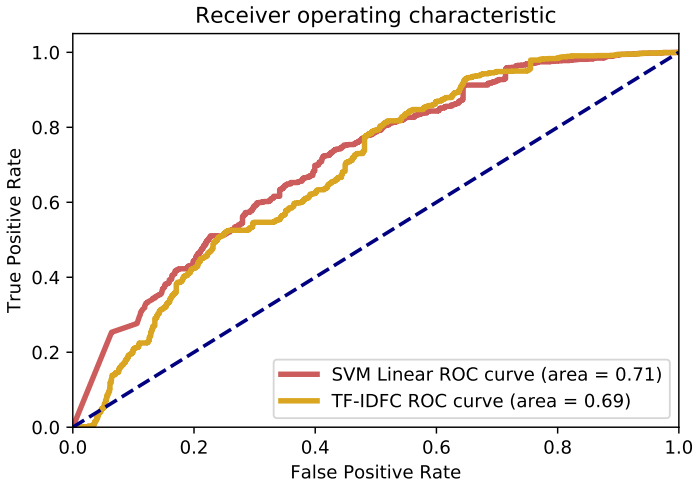

FIGURE 4.3: Global TFD ROC curves and AUC values when using the test sample of 3,000 labeled tweets (dashed line denotes a random classifier).

4.4.3 FUR results

The FUR experiments used the best TFD models (unary TF-IDFC and binary SVM) and their predictions when executing the second rolling window procedure (described in Section 4.4.2). In particular, we filtered the rolling window predictions to include all tweets related with the ground truth set of 418 users, which resulted in TFD S_t scores for 2,893 unlabeled tweets. These predictions were aggregated by each user u, allowing to compute the global FUR_u and respective ROC curves (Figure 4.4).

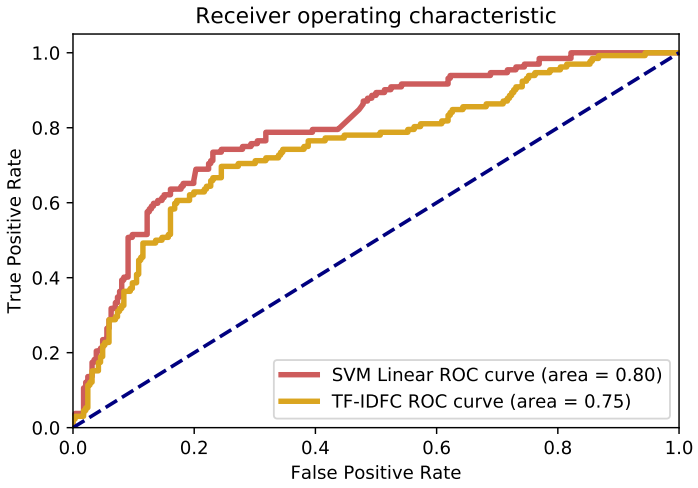

FIGURE 4.4: Global FUR ROC curves and AUC values when using the test sample of 418 labeled users (dashed line denotes a random classifier).

For both TDF models (SVM and TF-IDFC), the best FUR aggregation function is sum, resulting in higher AUC values (13 percentage point difference for SVM and 12 percentage point difference for TF-IDFC). When using the sum aggregation, the best FUR ROC curve is obtained by the binary SVM model, showing improved TPR values when compared with the unary TF-IDFC for most of the FPR axis range. Overall, the SVM model produced a very good discrimination, presenting an AUC of 0.80 and that is 5 percentage points better than the AUC value of TF-IDFC. It should be noted that the SVM user relevance predictive performance is similar to the one achieved by Ito et al., 2015, whose best model provided an AUC of 0.81. However, the authors considered a different Twitter dataset, a different notion of user relevance (not related with alloy steel), and more importantly, used all user history tweets (which requires more memory and computation). In contrast, our FUR approach only considers the tweets that resulted from the financial keywords query (Q).

For demonstration purposes, Table 4.8 reports the top 20 ranked user accounts when considering the binary SVM and unary TF-IDFC FUR sum scores. The **User name** column presents the Twitter account name and Web page for public company profiles. Due to privacy issues, the private accounts were anonymized. As for the **Ground truth** column, it presents the manual label result, where 1 denotes an alloy steel price relevant user and 0 an irrelevant one. The SVM and TF-IDFC rankings only differ after the ninth row. Globally, SVM correctly identifies 15 relevant users and TF-IDFC accurately classifies 14 ones.

TABLE 4.8: Top 20 steel price relevant users generated by the FUR scores.

User name	SVM rank	TF-IDFC rank	Ground truth	User name	SVM rank	TF-IDFC rank	Ground Truth
scrapindustry https://www.scrapmonster.com/	1	1	1	private user #4	13	18	1
aonesteelgroup http://aonesteelgroup.com/	2	2	1	private user #5	14	15	1
marketrnest http://marketresearchnest.com/	3	3	1	private user #6	15	16	1
trendy_girl_toy	4	4	0	yicaichina https://yicaiglobal.com/	16	-	1
sxcoal http://www.sxcoal.com/	5	5	1	private user #7	17	20	1
Cakestreamgo*	6	6	0	ywcdeals https://yeswecoupon.com/	18	-	0
foodrecipesgo*	7	7	0	private user #8	19	-	1
breakfastchild*	8	8	0	SPGlobalPlatts https://www.spglobal.com/platts/en	20	-	1
private user #1	9	9	1	private user #9	-	10	0
private user #2	10	11	1	private user #10	-	12	0
private user #3	11	14	1	private user #11	-	13	1
Northernweldarc http://northern-weldarc.com/	12	17	1	DTradingAcademy https://daytradingacademy.com/	-	19	1

∗ – These three Twitter profiles (probably bots) have the same contents and aim to sell or advertise products.

Chapter 5

Sentiment analysis for social media texts

Due to the expansion of Internet and Web 2.0 phenomenon, there is a growing interest in sentiment analysis of freely opinionated text. In this chapter, we propose a novel cross-source cross-domain sentiment classification, in which cross-domain labeled Web sources (Amazon and Tripadvisor) are used to train supervised learning models (including two deep learning algorithms) that are tested on typically non labeled social media reviews (Facebook and Twitter). We explored a three step methodology, in which distinct balanced training, text preprocessing and machine learning methods were tested, using two languages: English and Italian. The best results were achieved using undersampling training and a Convolutional Neural Network. Interesting cross-source classification performances were achieved, in particular when using Amazon and Tripadvisor reviews to train a model that is tested on Facebook data for both English and Italian.

5.1 Problem specification

Technological advances, such as the Internet expansion, Web 2.0 phenomenon and massive mobile device adoption, have increased the availability of freely opinionated text (e.g., blog reviews, social network comments). This big data source of unstructured texts enriches the value of sentiment analysis, also termed opinion mining, which uses computational methods to automatically analyze human opinions, sentiments and evaluations towards entities (e.g., products, services, organizations) (Liu, 2012). Indeed, several studies have analyzed opinion dynamics in social networks and their potential impact in decision making (Dong et al., 2018b; Dong et al., 2018a; Ureña et al., 2019). Thus, sentiment analysis is a key tool of modern decision support systems, helping to support decisions in several real-world applications, such as involving hotels,(Shi

and Li, 2011) stock markets,(Oliveira, Cortez, and Areal, 2016; Wang et al., 2018) and traffic accidents (Fu et al., 2018).

Given the importance of social media platforms (e.g., Facebook, Twitter), several works have proposed supervised machine learning algorithms for the sentiment analysis of social media texts (e.g., Naive Bayes, Support Vector Machines) (Liu, 2015). Yet, designing an accurate machine learning classifier for a particular sentiment domain and data source requires a substantial effort in terms of the data analyst time and execution of computational experiments. Moreover, some specific domains have less labeled data when compared with others (e.g., most Amazon reviews are about electronics). These two issues can be handled by using a cross-domain sentiment analysis,(Blitzer, Dredze, and Pereira, 2007; Wallin, 2014) which is a recent transfer learning research trend that aims to reuse sentiment models, previously fitted to some domains (e.g., electronics), to predict the sentiment of texts from other domains (e.g., books).

Some modern Internet platforms commonly ask for user labeled inputs. For instance, Amazon and Tripadvisor promote the writing of reviews under a 5-star rating system. However, sentiment labeled data is much scarce in other social media platforms. For example, Facebook is a popular social network with around 2 billion monthly active users[1] but only a small fraction of Facebook pages allow labeled reviews. Moreover, Twitter is a another relevant social network, with 330 million monthly active users[2], and that is commonly used to spread opinions about a wide range of domains, such as products (Go, Bhayani, and Huang, 2009) or stock markets (Oliveira, Cortez, and Areal, 2017). Yet, Twitter labeled data is much difficult to get, often requiring a laborious manual effort. In addition, there may be differences in the types of texts written in different Web platforms. For example, Twitter restricts the maximum size of text characters, while Facebook does not. As explained in Section 5.2, the majority of cross-domain studies consider a single Web data source (e.g., Amazon reviews). As shown in (Dalla Valle and Kenett, 2015; Dalla Valle and Kenett, 2018), the combination of multiple data sources if often valuable, allowing to augment information quality and reduce bias. Therefore, there is a potential gain and research interest in studying what we term here as "cross-source cross-domain" sentiment classification, in which cross-domain data, from one or more labeled sources, is used to create sentiment analysis models that are later applied to classify non labeled cross-domain texts from other sources.

In this work, we propose such approach, under the following main contributions:

[1] https://sproutsocial.com/insights/facebook-stats-for-marketers

[2] https://blog.hootsuite.com/twitter-statistics/

1. We approach a cross-source cross-domain sentiment classification, using distinct data sources and domain for training and testing the models. We adopt cross-domain big data labeled sources from different Web platforms (Amazon and Tripadvisor) to train the sentiment classification models. Then, the learned models are used to predict the sentiment of cross-domain texts from two unlabeled social media sources (Facebook and Twitter). Moreover, we consider datasets written in two distinct languages (English and Italian). The analyzed datasets are made publicly available[3] and thus can be used in future cross-source or cross-domain research studies.
2. We compare distinct data-driven approaches, in terms of: number of sentiment classes (2 or 3); feature engineering (stemming or part-of-speech tagging for the removal of nouns, pronouns and conjunctions); and balanced training methods (oversampling or undersampling). As for the learning algorithm, we propose a word embedded Convolutional Neural Network, which is compared with another deep learning model (Deep Feedforward Network) and two other classifiers (Support Vector Machines and Naive Bayes).
3. The proposed cross-source cross-domain approach is compared with a recent sentiment lexicon (Mohammad and Turney, 2013) and a state of the art cross-domain method that is based on a autoencoder structural correspondence learning (AE-SCL) method (Ziser and Reichart, 2017).

The chapter is structured as follows. The next Section 5.2 reports a summary of previous work for sentiment analysis and domain adaptation. In Section 5.3, we describe the data, modeling approaches and evaluation procedure. The Section 5.4 reports a brief description of the models used and evaluation metrics. Then, we describe the conducted experiments and obtained results (Section 5.5).

5.2 Related works

The related works are summarized in Table 5.1, which assumes a chronological order. Each study is characterized in terms of the language used, if it is a cross-domain or cross-source approach, data source used and size, type of text preprocessing (**L** – lemmatization, **S** – stemming, **P** – part-of-speech tagging), sentiment analysis method and number of sentiment classes adopted.

Sentiment analysis studies typically focuses on one specific domain at a time, such as hotels,(Shi and Li, 2011) movies (Mesnil et al., 2014) or stock

[3] https://github.com/paolazola/Cross-source-cross-domain-sentiment-analysis

markets (Oliveira, Cortez, and Areal, 2016; Li et al., 2017). Cross-domain sentiment analysis, (Blitzer, Dredze, and Pereira, 2007; Wallin, 2014; Ganin et al., 2016; Dragoni and Petrucci, 2017) also known as domain adaptation sentiment analysis, is a recent form of transfer learning (Pan and Yang, 2010). The goal is to learn a classification model from some domains (e.g., electronics, books) and then reuse the models to classify other domain texts (e.g., music reviews). This alleviates the need to collect and curate data for each new domain, and it is particularly relevant for accessing the sentiment of new product opinions for which scarce data are available (Fang and Zhan, 2015). The **Cross-domain** column of Table 5.1 signals the relevant works in this field.

The rationale for adopting a cross-domain sentiment analysis also translates into cross-source sentiment analysis. Developing an accurate model for one source is costly and several social media sources, such as Facebook or Twitter, contain a huge amount of unlabeled reviews. However, most cross-domain sentiment analysis works assume a single data source, as shown by the column **Cross-source** of Table 5.1. Often, this source consists in the popular Amazon platform, (Pan and Yang, 2010) with the analysis of distinct reviews of sold products (Blitzer, Dredze, and Pereira, 2007; Pan et al., 2010; Bollegala, Weir, and Carroll, 2011; Wallin, 2014; Zhang et al., 2019). Within our knowledge, there are only two cross-domain works that use distinct sources. Aue and Gamon (2005) considered only traditional Web sites. More recently, Ziser and Reichart, 2017 used a single source, the Blitzer's Amazon dataset with reviews of products (e.g., books, electronics) to train binary sentiment classification models that were then tested on blog texts (from 16 nondisclosed domains).

There are two main sentiment classification methods: lexicon and machine learning based. A lexicon is a special dictionary in which words are assigned to sentiment scores (Ghosh and Kar, 2013; Kumar, Desai, and Majumdar, 2016). The main advantage is that, once a lexicon is built, a fast unsupervised sentiment classification is achieved, by summing the overall word scores. Thus, there is no need for labeled data. However, lexicons tend to produce lower performances when compared with supervised machine learning approaches (Li et al., 2011b). Thus, machine learning is widely used for sentiment analysis (Pang, Lee, and Vaithyanathan, 2002; Salvetti, Lewis, and Reichenbach, 2004; Pouransari and Ghili, 2014).

Sentiment classification studies initially explored simpler feature engineering (e.g., N-grams or Bag-of-Words) and machine learning algorithms (e.g., Naive Bayes, Support Vector Machines). After 2014, recent text classification advances, such as word embedding and deep learning, (Goodfellow et al., 2016) were naturally incorporated into sentiment analysis works (Ortigosa, Martín, and Carro, 2014; Lai et al., 2015; Conneau et al., 2017). Focusing on transfer learning problems, Ganin et al. (2016) proposed a domain adversarial

TABLE 5.1: Summary of related work.

Study	Lang[a]	Cross-domain	Cross-source	Source[b]	Data Size[c]	L	S	P	Method[d]	Sentiment Classes
Pang, Lee, and Vaithyanathan (2002)	ENG			WS	2K			X	N-gram+NB, N-gram+SVM, N-gram+ME	2
Dave, Lawrence, and Pennock (2003)	ENG	X		WS			X	X	N-gram+NB, N-gram+SVM	2
Salvetti, Lewis, and Reichenbach (2004)	ENG			WS	27K		X	X	L+NB, L+MM	2
Aue and Gamon (2005)	ENG	X	X	WS	2K,5K,12K				N-grams+NB, N-grams +SVM	2
Cui, Mittal, and Datar (2006)				WS	200K				N-grams+PA, N-grams+LM, N-grams+Winnow	2
Ng, Dasgupta, and Arifin (2006)	ENG			WS	4K			X	L+SVM	2
Blitzer, Dredze, and Pereira (2007)	ENG	X		WS	8K				SCL, SCL-MI	2
Go, Bhayani, and Huang (2009)				SM	1.6M			X	N-grams+NB, N-grams+ME, N-grams+SVM	2
Ohana and Tierney (2009)	ENG			WS	2K				SW+SVM	2
Dang, Zhang, and Chen (2010)	ENG			WS	2K,8K			X	N-grams +SVM	2
Pan et al. (2010)	ENG	X		WS	20K				SFA	2
Glorot, Bordes, and Bengio (2011)	ENG	X		WS	340K				SDA	2
Shi and Li (2011)	CHI			WS	4K				Fr+SVM, Tf-Idf+SVM	2
Jo and Oh (2011)	ENG			WS	24K, 27K				S-LDA, ASUM	2
Yoshida et al. (2011)	ENG	X		WS	10K			X	GmWdDinD	3
Gräbner et al. (2012)	ENG			WS	80K			X	LDB	3 5
Neri et al. (2012)	ITA			SM	1K				SKMs	-
Bollegala, Weir, and Carroll (2011)	ENG	X		WS	8K,68K	X		X	FE+L1LR	2
Ghosh and Kar (2013)				WS	300			X	SLX	2
Ortigosa, Martín, and Carro (2014)	SPA			SM	3K			X	L,NB,J48,SVM	2
Santos and Gatti (2014)	ENG			WS,SM	12K,80K				We+CNN, Ce+CNN	2
Mesnil et al. (2014)	ENG			WS	50K				N-GM,NB, SVM,RNN	2
Pouransari and Ghili (2014)	ENG			WS	60K				BOW/W2V+LR, BOW/W2V+RF, BOW/W2V+SVM RNN	2 5
Tang et al. (2015)				WS	335K,5K				UWCVMC	4,5
Wallin (2014)	ENG	X		WS	636K			X	LR+BOW	2,5
Fang and Zhan (2015)	ENG	X		WS	5.1M			X	SVM,NB,RF	2
Lai et al. (2015)	ENG,CHI			D	230K,20K				We+RNN, We +RCNN	4 to 20
Ganin et al. (2016)	ENG	X		WS	8K				DANN	2
Kumar, Desai, and Majumdar (2016)				WS					NB,LR,SW	2
Tripathy, Agrawal, and Rath (2016)	ENG			WS	50K				N-grams+NB, N-grams+ ME, N-gram+SVM, N-gram+SGD	2
Conneau et al. (2017)	ENG,CHI			WS	11M				Ce+VDCNN	2 to 14
Dragoni and Petrucci (2017)	ENG	X		WS	1M				We+NN	2
Radford, Jozefowicz, and Sutskever (2017)				WS	82M				LSTM	2
Ziser and Reichart (2017)	ENG	X	X	WS,B	78K,40K				AE-SCL	2
Dragoni and Petrucci (2018)	ENG	X		WS	1M				FM+L	2
Zhang et al. (2019)	ENG	X		WS	56K				IATN	2
This work	ENG,ITA	X	X	WS,SM	1.3M		X	X	We+NB, We+SVM, We+MLP, We+CNN	2,3

[a] **Language** – ENG (English), CHI (Chinese), SPA (Spanish), ITA (Italian).

[b] **Data source type** – B: blogs, D: documents (e.g., Stanford sentiment treebank, News database), SM: social media (Facebook and Twitter), WS: Web sites (Amazon, Citysearch, Electronics reviews, MyMovies and other movies reviews, Tripadvisor, Yelp).

[c] **Number of instances** – K: thousand, M: million.

[d] **Sentiment Analysis method** – AE-SCL: autoencoder structural correspondence learning, ASUM: aspect and sentiment unification model, BOW: bag of words, Ce: character embedding, CNN: convolutional neural network, DANN: domain-adversarial neural network, FE: feature extraction, FM: fuzzy model, Fr: frequency, GmWdDinD: generative Bayesian model of word with domain dependence or domain independence, IATN: interactive attention transfer network, J48: decision tree, L: lexicon information, L1LR: L1 regularized logistic regression, LDB: lexicon database, LM: language modeling, LR: logistic regression, LSTM: long-short term memory neural network, ME: maximum-entropy, MM: Markov model, MI: mutual information, NB: naive Bayes, PA: passive-aggressive algorithm, S-LDA: sentence latent Dirichlet allocation, SDA: stacked denoising auto-encoders, SFA: spectral feature alignment, SGD: stochastic gradient descent, SKMs: sentiment knowledge mining system, SLX: sentiment lexicon database, SVM: support vector machine, SCL: structural correspondence learning, SW: SentiWordNet Baccianella, Esuli, and Sebastiani (2010), RCNN: recurrent convolutional neural network, RF: random forest, RNN: recursive neural network, UWCVMC: user word composition vector model, VDCNN: very deep convolutional neural network, W2V: word to vec, We: word embedding.

neural network where the hyperparameter are determined by a reverse cross-validation approach. Recently, Zhang et al. (2019) analyzed the jointly impact of sentence network attention and aspect network attention in the interactive attention transfer network (IATN). The novelty of our work is highlighted in the last row of Table 5.1. We address a novel cross-source cross-domain sentiment analysis, in which Web sources that contain easy labeled reviews (Amazon and Tripadvisor) are used to fit a sentiment analysis model, which is then reused to predict the sentiment of two typically unlabeled social media platforms (Facebook and Twitter). Moreover, we propose a recent deep learning method, which is based on a word embedded Convolutional Neural Network and that is compared with three machine learning methods (a modern Deep Feedforward Network, a Support Vector Machine and Naive Bayes), a recent sentiment lexicon and state of the art cross-domain method. We also explore stemming or part-of-speech tagging, to reduce the word sparsity, and oversampling or undersampling methods, to deal with the unbalanced sentiment datasets. Finally, to enrich the experimental comparison analysis, we consider two languages (English and Italian) and two sentiment classification tasks ("negative","positive" and "negative", "neutral", "positive").

5.3 Materials and methods

5.3.1 Sentiment analysis data

In this work, we consider texts written in two languages, English and Italian. We also consider two sentiment output label sets, with 2 ("negative", "positive") and 3 ("negative", "neutral", "positive") classes. The datasets analyzed are made freely available at https://github.com/paolazola/Cross-source%-cross-domain-sentiment-analysis. The texts come from four major sources of data:

1. **Amazon**: we gathered the data directly from the *Amazon.com* Web site. The reviews regard different products, such as electronic devices, kitchen objects, clothes and house accessories. For the polarity classification, we consider two 5-star rating value transformations: $\{1,2,3\} \rightarrow$ "negative" and $\{4,5\} \rightarrow$ "positive"; and $\{1,2\} \rightarrow$ "negative", $3 \rightarrow$ "neutral" and $\{4,5\} \rightarrow$ "positive". The data was collected from January to February 2018 and it includes 282,781 English and 161,443 Italian reviews.

2. **Tripadvisor**: we collected reviews directly from the *Tripadvisor.com* Web site. The 5-star reviews are related with restaurants, hotels, monuments

and interest points, cities and activities. The same Amazon label transform was adopted to create the 2 and 3 class outputs. The data was collected from January to February 2018 and the dataset is composed by 519,735 randomly sample reviews for the English language and 324,376 for the Italian one.

3. **Facebook**: the data was retrieved directly from the *Facebook.com* social network. We considered only comments from specific public pages having a 5-start rating system, such that we could compute the same 2 and 3 class sentiment labels. The sampled reviews performed from January to February 2018 are about several topics, namely universities, events, famous people, locals, parties, shops and cities (total of 5,792 English and 1,077 Italian texts).

4. **Twitter**: to reduce the manual labeling effort, we selected preferentially publicly labeled data. For English, we used the Sentiment140 labeled test set developed by Stanford University, (Go, Bhayani, and Huang, 2009) which has 497 reviews about companies, events, locations, movies, persons, etc. The data was collected in 2009. The data are structured in three label classification: "negative", "neutral" and "positive". As for Italian, we adopted the SENTIPOLC (SENTIment POLarity Classification) labeled dataset that was organized within Evalita 2014 (SENTIPOLC, 2014). It includes a set of 4,513 twitter status IDs, with annotations concerning polarity classification and irony detection about politics, news and famous people. Since the dataset only includes two classes ("positive" and "negative") and two authors are Italians, we performed an extra manually 3 level classification ("negative", "neutral" and "positive") of 937 tweets, collected at April 2018 and regarding Italian television shows and other more general topics. To get binary versions of Sentiment140 and our Italian manually labeled data, we merge the original negative and neutral classes into the "negative" label.

Figure 5.1 plots the data source percentage rating/sentiment class distributions. In all cases, the sentiment classes are unbalanced. Some sources (Amazon, Tripadvisor) present the common J-shaped distribution, with a much lesser number of negative reviews (Wallin, 2014).

As reported in Li et al. (2011a), this might be due to the following reasons: people tend to publish opinions about popular products, which are more likely positive; and there may exist many flaunt positive reviews from the product companies and dealers.

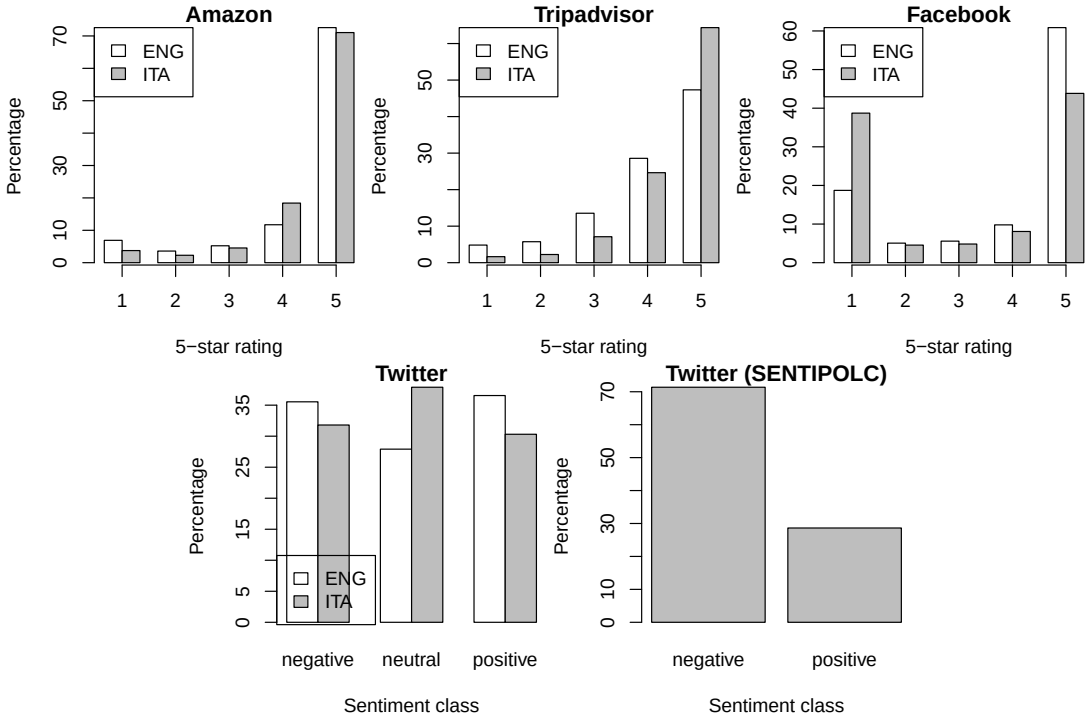

FIGURE 5.1: Sentiment distribution values for the distinct data sources.

5.3.2 Cross-source methodology

We adopt four learning algorithms, as detailed in Section 5.4: Naive Bayes (NB), Support Vector Machine (SVM), Deep Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN). Also, the text reviews are firstly preprocessed in order to remove numbers, capitalized letters, whitespaces, punctuation, stopwords and urls. After this preprocessing, we further apply stemming (Stem) or part-of-speech (POS) tagging (Section 5.3.3).

In this work, we assume a research methodology that contains three main steps (Figure 5.2). Let $A \rightarrow B$ denote a sentiment classification model that was trained on A and tested on B, where A and B denote cross-domain corpus. In step 1, we execute single source experiments ($A = B$, $A \in \{$Amazon,Tripadvisor$\}$). In the step 1 of Figure 5.2, the dashed boxes and arrows (e.g.,) denote the path followed by the Amazon dataset, while the dotted box and arrows (e.g.,) represent the Tripadvisor analysis path. The goal is to perform initial experiments to gather insights about balanced training (oversampling or undersampling) and hyperparameter (e.g., number of neural network hidden nodes) selection. This selection is based on a grid search, which uses a range of grid values for several hyperparameters (Section 5.4). The best hyperparameter values, in terms of classification performance on the test data, are then fixed for steps 2 and 3. We note that step 1 test data is from the same training data source, while in step 3, we perform the target cross-source tests using external source data (Facebook and Twitter) that was never used in the modeling decisions defined in steps 1 and 2. Moreover, step 1 also provides the estimation of single source test classification performances, which can be used to evaluate the quality of the proposed transfer learning sentiment approach. In effect, any cross-source test classification measure (of steps 2 or 3) close to the single source performance (of step 1) would indicate a high quality sentiment analysis.

Next, in step 2 we conduct Amazon→Tripadvisor and Tripadvisor→Amazon cross-source experiments, aiming to select the best text processing and machine learning method. The solid arrows (⟶) in step 2 of Figure 5.2 represent the same paths that were followed by the Amazon→Tripadvisor and Tripadvisor→Amazon experiments. The learning models use fixed balanced training and hyperparameter values, as set in step 1. There are two main text processing options (Stem or POS) and four learning algorithms (NB, SVM, MLP and CNN).

Finally, in step 3 we use the labeled sentiment sources for training (input domain) and perform the testing on both non labeled sources (target domain). In step 3 of Figure 5.2, the dashed arrows () represent the path when the target test domain is Twitter, while the dotted arrows () refer to the Facebook target domain. A fixed text processing and machine learning model (set in step 2) is used. Only one training model is obtained for each language, allowing to

obtain the final cross-source results: Amazon ∪ Tripadvisor→Facebook and Amazon ∪ Tripadvisor→Twitter.

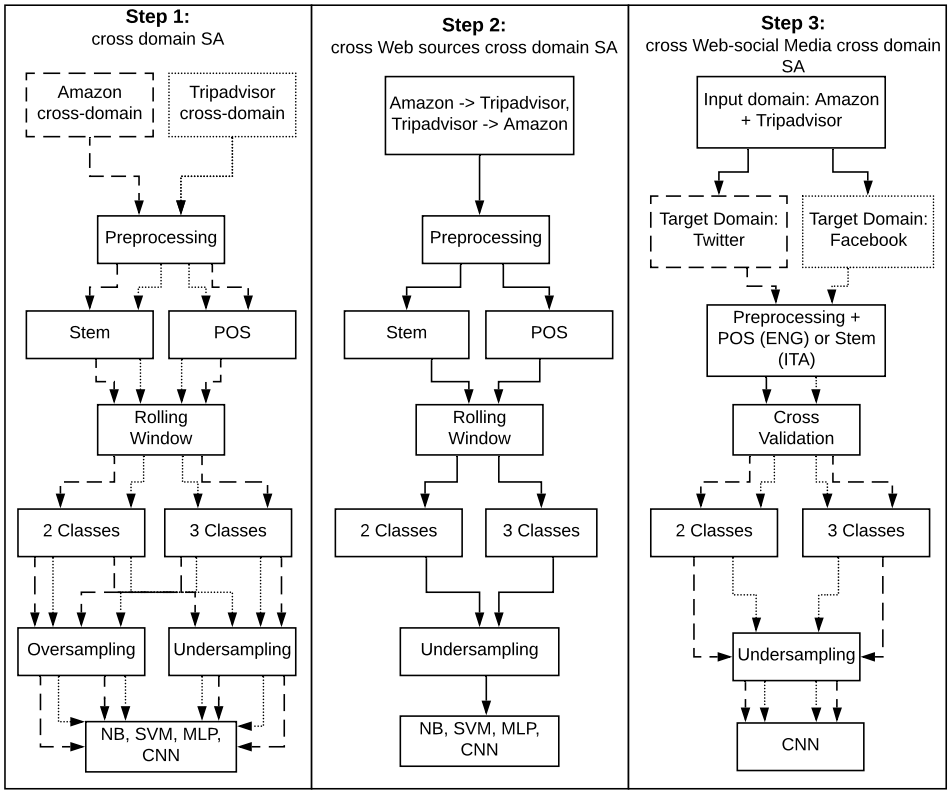

FIGURE 5.2: Adopted three step methodology for the cross-source cross-domain sentiment analysis (SA).

In steps 1 and 2, we use the three main features: *date*, *review text* and *sentiment class*. The *date* is used to chronologically order the messages, such that a rolling window evaluation scheme can be applied (Oliveira, Cortez, and Areal, 2017). The rolling window is a realistic and robust evaluation method that considers several training and test iterations through time. First, the texts are ordered by the *date* field and split into k distinct partitions of equal size. For a particular i iteration, the i-th partition is selected and further split into training (oldest data) and test sets (newest data). The training data are then balanced using undersampling or oversampling (Batista, Prati, and Monard, 2004). The former method decreases the dataset size by randomly sampling the majority examples in order to equal number of minority ones. The latter expands the data by sampling with repetition the minority examples in order to equal the number of majority ones. Next, the machine learning model is fit and evaluated using the test set, which keeps the original sentiment class distribution.

In the step 3, since the *date* feature is not available (at both Sentiment140 and SENTIPOLC), we execute a k-fold cross validation (Dragoni and Petrucci, 2017), which works as follows. First, the full training data is set by selecting all Amazon texts and a sample of Tripadvisor reviews, such that each source is similarly represented. We note that Tripadvisor is in general twice the size of Amazon data, thus a 50% sampling is often adopted. Then, the merged training data is randomly divided into k-folds. For a particular i iteration, all data samples except the ones belonging to the i-th fold are used to train the sentiment model. The balancing method is applied only to the training data. After fitting the model, it is tested two times, using the whole Facebook and Twitter messages as the test sets and leading to two sets of classification performance measures.

5.3.3 Stem and Part-of-Speech Text preprocessing

To reduce the word embedding size and computational effort, we test two alternative Natural Language Processing (NLP) techniques to compress text: stemming (Stem) and Part-of-Speech (POS) tag removal. As an example, **??** presents the text sentence reduction that is achieved when using Stem or POS preprocessing with the two largest data sources (Amazon and Tripadvisor), showing that in certain cases a high compression rate is achieved (e.g., around 70% for English Tripadvisor data when using POS).

TABLE 5.2: Comparison of text sentence size before and after preprocessing for Amazon and Tripadvisor sources (values denote the average number of words per sentence).

Language	**Source**	Original	Stem	POS
English	Amazon	35.60	17.60	10.70
	Tripadvisor	125.06	61.03	34.97
Italian	Amazon	42.20	26.08	13.50
	Tripadvisor	70.94	41.03	21.14

In the literature the *stemming* procedure refers to the process of stripping off affixes (both suffix and prefix) from the word and maintaining only the root of the word (Litvak and Vanetik, 2019). Similar to the stemming is the *lemmatization*, where each word is re conducted to its lemma or lexeme. The benefit of stemming and lemmatization is in data sparseness reduction even if for some languages, such as English, the dictionary is characterized by a diminish morphology and therefore the stemming procedure might not show a considerable improvement in the performance. However for other languages, such as Latin ones (Italian in our research), the vocabulary is very rich of morphology and

the stemming might help in reducing the number of features in the text, increasing the classification performance. To implement the data stem we used the *Snowball Stemmer* (Porter, 2001) available on *NLTK* module in Python. Two stem illustrative examples are: "affordable" → "afford" (English) and "bellissima" → "bell" (Italian).

Part-of-Speech (POS) tagging is a technique used to assign the appropriate parts of speech tag to each word in a text (it is also known as word classes, morphological classes or lexical tags) (Manning and Schütze, 1999).

We use the POS tagging in order to exclude all *nouns*, *pronouns* and *conjunctions* from the text in order to remove potential "domain" terms from the reviews and thus maintain more useful words for the cross-domain sentiment extraction, such as *adjectives* and *adverbs*. It has been demonstrated that *adjectives* are good indicator for opinion classification (Wiebe, 2000). Also, some literature works in sentiment classification did not consider *nouns* (Dang, Zhang, and Chen, 2010; Ortigosa, Martín, and Carro, 2014). The POS tagging was performed by using the *RDRPOSTagger* (Nguyen et al., 2014) library developed in the R software. The *RDRPOSTagger* supports both English and Italian languages and it is more fast in tagging when compared with other POS taggers, such as *Treetagger* (Schmid, 2013) available in Python. Two POS tag removal demonstrative examples are: "really worthy the money" → "really worthy" (English) and "Città meravigliosa in tutto" → "meravigliosa" (Italian).

5.3.4 Word embedding

Word Embedding is a distributed representation in which each word is represented as a vector in a continuous space and similar words are mapped to nearby points. The Vector Space Models (VSM) has been applied to text data since the 1960s and they assumed a greater interest in recent years. Among VSM it is possible to distinguish two main approaches:Baroni, Dinu, and Kruszewski (2014)

1. count based method: it is based on word co-occurence in order to build dense vectors. An example of this approach is the Latent Semantic Allocation (LSA);

2. predictive method: predict a word based on its neighbours. N-grams, Neural Probabilistic Language Models (NNLM) (Bengio et al., 2003) and the Word2Vec (Mikolov et al., 2013b) model are some examples of this approach.

The state of the art in VSM is associated to Mikolov et al. (2013b), which proposed a Feedforward Neural Network with an input, projection and output

layer under two versions: Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram Model (Skip-gram). In the CBOW model, the word w_t is predicted by considering the nearby words (context), while in the Skip-gram it tries to maximize the classification of a word based on another word in the same sentence. In this last case, the word w_t is used to classify the context.

In this chapter, we performed a word level embedding by using the *Keras* library tool based on a Feedforward Neural Network. The input is a integer matrix called **I**, where each word is mapped to its absolute frequency given the dataset words' distribution. The matrix has n rows which denotes the different reviews in the dataset and c columns. Each review has a variable number of words and, in order to reduce the sparseness in the matrix **I** and ensure the same dimension to each review, we defined the column number c as follow:

$$c = \frac{\sum_{i=1}^{n} length(r_i)}{n} \tag{5.1}$$

where represents the round function and $length(r_i)$ denotes the number of words in the i-th review. The matrix **I** is then passed to the embedding layer. The embedding layer maps a two-dimensional matrix in a sequence of e matrices. In this chapter the number of matrices are e =128. The embedded matrix **O** is then composed by n rows (for each review) and c columns. Each element $\mathbf{O}(i, j \times 128)$ represents the numerical depiction (real number) of the $n-^{th}$ sentence.

A small demonstration example is provided, which considers three messages:

1. "sicly beaches were fantastic and food amazing. What a super happy holiday";
2. "The hotel is good, receptionist helpful in giving advises and the swimmingpool was wonderful"; and
3. "A new car has been promoted by the company. It is fantastic, the best on the market with many new accessories.".

After preprocessing (e.g., with removal of punctuation, stop words and POS nouns), the sentences become:

1. "fantastic amazing super happy";
2. "good helpful wonderful"; and
3. "new promoted fantastic best many new".

The demonstration assumes text data with the following term frequency values: {*good*=3245, *helpful*=1700, *new*=1200, *many*=2400, *great*=3000, *fantastic*=2500, *free*=1400, *amazing*=1000, *super*=600, *happy*=1100, *wonderfull*=300, *best*=734, *promoted*=5}. Thus, the initial **I** integer matrix becomes:

$$\begin{bmatrix} 2500 & 1000 & 600 & 1100 & & \\ 3245 & 1700 & 300 & & & \\ 1200 & 5 & 2500 & 734 & 2400 & 1200 \end{bmatrix}$$

In this example, the average size is $c = 4$. Sentences with a length greater than 4 are truncated and sentences with less than 4 elements are padded with zeros (Wood-Doughty, Andrews, and Dredze, 2018), resulting in the final **I** matrix:

$$\begin{bmatrix} 2500 & 1000 & 600 & 1100 \\ 3245 & 1700 & 300 & 0 \\ 1200 & 5 & 2500 & 734 \end{bmatrix}$$

Since now the matrix **I** is composed by sentences with the same number of columns (tokens), it is possible to compute the word embedding via a Feed-forward Neural Network, obtaining for each token a real numbers representation. In this example, it is denoted with a sequence of 128 real values. Thus, for each sentence we concatenate the single word embedding (1×128) obtaining a sentence embedding equal to $(1 \times (4 \times 128))$. Considering a flatted representation, the matrix **O** is then composed by 3 rows and 512 columns denoting the concatenated word embedding (Santos and Gatti, 2014).

5.4 Models

The models described here were used for both binary and multiclass classification. As reported in Section 5.3.4, the input of all the machine learning algorithms is the word embedding matrix **O**, with n rows corresponding to the n reviews in the data set, while the output is related with the rating vector V (with 2 or 3 classes). Three of the learning models (SVM, MLP and CNN) have hyperparameters that were tuned using a grid search. Using only single source data (step 1 of Section 5.3.2), a rolling window validation was executed, providing several training and test iterations thought time. For each learning model (SVM, MLP or CNN), we select the hyperparameter value that resulted in the best average classification performance (Area Under the Curve metric, see Section 5.4.5) on the rolling window single source test data. The details of the selected hyperparameters, fixed in step 1 and used in steps 2 and 3 of Section 5.3.2, are presented in Section 5.5.

5.4.1 Naive Bayes

The label l^* can be assigned to a review r using the formulation: $l^* = \arg\max_l P(l|r)$. The Naive Bayes (NB) method is based on the Bayes' rule and on the strong hypothesis that there is independence between every pair of input features.Ng and Jordan (2002) The probability of label l based on r is computed as:

$$P(l|r) = \frac{P(r|l) * P(l)}{P(r)} \tag{5.2}$$

And, in case of binary classification (0,1), the label for the r-th review is based by on:

$$\frac{P(l_0|r)}{P(l_1|r)} = \frac{P(r|l_0) * P(l_0)}{P(r|l_1) * P(l_1)} \tag{5.3}$$

5.4.2 Support Vector Machine

Support Vector Machines (SVM) are widely used in text classification, (Liu, Bi, and Fan, 2017) often outperforming the NB algorithm (Joachims, 1998). It can be used for both classification and regression tasks and the model is based on a maximized margin criterion (Wang and Xue, 2014). For the binary classification, the SVM algorithm can compute the best separating hyperplane in a feature space (after the kernel transformation). Given $l_j \in 1, -1$, corresponding to negative (-1) or positive (1) classes, the solution of the SVM model for the review r_j is given by:

$$\overrightarrow{w} = \sum_j \alpha_j l_j \overrightarrow{r_j}, \alpha_j \geq 0; \tag{5.4}$$

where the α_j are obtained by solving a dual optimization problem. The support vectors are the $\overrightarrow{r_j}$ values such that $\alpha_j > 0$ (Cristianini and Shawe-Taylor, 2000). In this work, we selected the popular Gaussian kernel, also termed Radial Basis Function (RBF), which presents less hyperparameters when compared with other polynomial kernels. The model contains just two hyperparameters: the γ Gaussian kernel parameter and C, a penality parameter that indicates the sensibility of the model to misclassification. To set these hyperparameters, a grid-search was adopted in step 1 using the values $\gamma \in \{0.01, 0.1, 1.0\}, C \in \{0.1, 1.0, 10.0\}$. The best values were selected using test data (from step 1, using the same data source) and are reported in Section 5.3.1. The SVM model was implemented using the *sklearn* module in Python, which is based on the popular *libsvm* library.

5.4.3 Multilayer Perceptron

The adopted Multilayer Perceptron (MLP) model corresponds to a modern deep learning variant of the original feedforward neural network, (Mahendhiran and Kannimuthu, 2018) which includes three hidden layers (with H_1, H_2 and H_3 hidden nodes), usage of Dropout regularization, Adagrad gradient training and ReLU activation functions on the hidden nodes: (Goodfellow et al., 2016)

$$\begin{aligned} \text{ReLU} \quad & f(z_i) = max(0, z_i) \\ \text{Softmax} \quad & P(l|r) = f(z_l) = \frac{\exp(z_l)}{\sum_{k \in K} \exp(z_k)} \end{aligned} \tag{5.5}$$

where z_i is the weighted sum of the i-th neural unit a, f is the activation function and K is the set of output nodes. ReLU is a popular activation function often used in deep learning experiments due to its good convergence property and faster training of deep layers (LeCun, Bengio, and Hinton, 2015). The Softmax function allows the outputs to be interpreted as class probabilities (where $\sum_{k \in K} f(z_k) = 1$). The weights of the MLP are typically estimated by using a gradient descent algorithm (Ruder, 2016). To fit the weights, we used the Adagrad gradient descent variant, which automatically adapts the learning rate η, performing smaller updates for more frequently used weights and larger updates for infrequent weights. This algorithm is particularly suitable for sparse data tasks, such as text classification, which often contains very frequent and infrequent words (Pennington, Socher, and Manning, 2014; Pouransari and Ghili, 2014). To prevent overfitting, we use a Dropout value of 20% as the regularization method. Dropout randomly ignores neural connections during training and this significantly reduces overfitting, often obtaining major improvements when compared with other regularization methods (Srivastava et al., 2014). The grid search ranges for the MLP hyperparmeters were set to: $H_1 \in \{50, 60, ..., 90, 100, 150, 200, 250\}$, $H_2 \in \{10, 20, 30, ..., 50, 100, 125\}$ and $H_3 \in \{5, 10, 15, ..., 25\}$. The grid search was restricted to present a decreasing order in the number of hidden units per layer, such that $H_1 > H_2 > H_3$. Similarly to Prusa and Khoshgoftaar (2017) and Mahendhiran and Kannimuthu (2018) that used a fixed number of epochs (e.g., 100) for each experiment, this hyperparameter value was set to 100.

5.4.4 Convolutional Neural Network

Convolutional Neural Network (CNN) is a class of deep feedforward neural networks that exploits local connectivity patterns designed to process data that comes in the form of multiple arrays (Dragoni and Petrucci, 2017). CNNs have obtained competitive state-of-the-art results in several classification tasks, including image classification and text classification Kim (2014). The design of a

CNN is composed by an input layer, M convolutional layers, H MLP hidden layers and an output layer. When compared with MLP, the main difference is the presence of the initial convolutional layers, each composed by a convolutional layer and a pooling layer.

The contribution of the convolutional layer in the CNN regards the convolution operation itself, which is a kind of sliding window function that performs a matrix product between the input and a filter matrix or vector, called also kernel or feature detector, and that is smaller than the input matrix size. This convolution operation leads to a sparser interaction in CNN, thus fewer parameters are estimated, improving the computational efficiency. Another feature that distinguishes CNN from the other neural networks is the parameter sharing, which refers to the use of the same parameter for more than one function in a model, since each member of the kernel is used at every position of the input. By adopting this parameter sharing, the layers also assume a equivariance in translation property (Goodfellow et al., 2016). Another important element in a CNN is the pooling layer which further modifies the convolutional layer output, replacing the values in some location by the summary statistics of the nearby outputs. Two famous pooling functions are the max polling and the average pooling. For example, if the convolutional vector output c is divided into v rectangular areas, each composed by $e = \frac{c}{v}$ elements, then the pooling output is a vector of length v such that each element corresponds to the maximum or average of the e-th rectangular. In this chapter, we adopt a CNN with a convolution layer with its max pooling layer followed by another convolutional layer and an average pooling layer. Then, as described in Section 5.4.3, the same MLP procedure is added. The CNN hyperparameters were searched using the ranges: first filter $\in \{1, 3, 6, 12, 24, 32, 64, 96, 128\}$, first kernel $\in \{1, 5, 9, 14, 20\}$ max pooling $\in \{1, 2\}$, second filter $\in \{2, 6, 12, 24, 36, 48, 64, 128, 184, 256\}$ and second kernel $\in \{2, 3, ..., 6\}$.

5.4.5 Evaluation

The classification performance is based on three metrics: Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve, the macro-averaging F1-score and Accuracy.

Each classifier outputs a probability for a particular class label (l) and review (r): $P(l|r)$. For the decision parameter $D \in [0, 1]$, it can be assumed that class l is positive if $P(l|r) > D$. The ROC curve plots all the trade-offs (distinct D values) between correctly predicting the positive or negative l class values, showing one minus the specificity (x-axis) versus the sensitivity (y-axis). ROC curves can be applied to unbalanced tasks and without knowing *a priori* the

false positive and false negative costs (Fawcett, 2006). To obtain a single metric, the $AUC = \int_0^1 ROCdD$ is often used. A random classifier presents an AUC of 0.5, while the ideal classifier should present an AUC of 1.0. For the multiclass models, we compute the global AUC, which weights each class AUC according to the most frequent classes.

In classification, it is often assumed that the predicted class label l is the one with the highest probability. The confusion matrix maps the predicted versus the desired labels, allowing to compute several metrics, such as Accuracy, Precision, Recall and F1-score (Witten et al., 2017):

$$\begin{aligned} \text{Accuracy} &= \frac{TP+TN}{TP+FP+TN+FN} \\ \text{Precision}_l &= \frac{TP_l}{TP_l+FP_l} \\ \text{Recall}_l &= \frac{TP_l}{TP_l+FN_l} \\ \text{F1-score}_l &= 2 \times \frac{\text{Precision}_l \times \text{Recall}_l}{\text{Precision}_l + \text{Recall}_l} \end{aligned} \tag{5.6}$$

where TP_l, FP_l, FN_l denote the number of true positives, false positives and false negatives for class l. To combine the F1-score multiclass results into a single measure, we use the macro-averaging F1-score, which first computes the F1-score$_l$ for all l labels and then averages the overall result. The classification metrics were implemented using the *rminer* R package (Cortez, 2010).

To evaluate the overall performance of the sentiment models, we use the same procedure adopted by Oliveira, Cortez, and Areal (2016): first, we compute the metric (AUC, macro-averaging F1-score or accuracy) for each iteration of the rolling window (steps 1 and 2) or k-fold cross validation (step 3); then, we average the k distinct results. Statistical significance is obtained by applying the non-parametric Wilcoxon signed rank test (Hollander and Wolfe, 1999). The model selection decision (e.g., best hyperparameter value, best balancing method) is mainly based on AUC values as the single metric. In fact, on one hand the macro-averaging F1-score corresponds to just one specificity versus sensitivity trade-off, while the AUC is computed over all possible D trade-offs. On the other hand, accuracy is sensitive to unbalance data as our test sets and it might be misleading to performance evaluation. However, accuracy is a common metric often used in sentiment classification, thus, we deemed appropriate to include it in the results.

5.5 Experimental evaluation

We conducted the computational experiments using code written in the Python language and executed using two different multi-core servers (e.g., Intel Xeon E5 at 2.30 GHz). In both steps 1 and 2, we used k=20 iterations of the rolling

window evaluation scheme. In step 3, we used the same $k = 5$-fold cross validation employed in the recent work of Dragoni and Petrucci (2018).

5.5.1 Step 1 results

In each rolling window iteration of step1, the reviews were sorted, such that 60% of the oldest data was used for training and 40% for testing. Also due to computational requirements, we conducted the step 1 hyperparameter grid selection only for the undersampling and binary classification case. Hyperparameters are then fixed with the best searched values used for the oversampling and multiclass models. The selected values are shown in Table 5.3.

TABLE 5.3: List of selected hyperparameters.

	Stem				POS			
	English		Italian		English		Italian	
	Tripadvisor	Amazon	Tripadvisor	Amazon	Tripadvisor	Amazon	Tripadvisor	Amazon
SVM:								
C	0.1	1	0.1	1	0.1	1	0.1	0.1
γ	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
MLP:								
H_1	200	100	200	90	100	100	200	200
H_2	125	30	125	50	50	30	125	125
H_3	25	10	25	20	15	10	25	25
CNN:								
first filter	12	32	12	6	12	32	12	24
first kernel	9	5	9	5	9	5	9	9
max pooling	2	1	2	2	2	1	2	1
second filter	24	64	24	12	24	64	24	48
second kernel	6	2	6	3	3	2	4	2
H_1	200	100	200	90	100	100	200	200
H_2	125	30	125	50	50	30	125	125
H_3	25	10	25	20	15	10	25	25

The sentiment classification results for step 1 are presented in Table 5.4, which shows interesting AUC results for Tripadvisor and Amazon data sources, in both languages. The best AUC values were obtained for the English Tripadvisor data: 81% AUC for binary task and 78% for the three sentiment classification.

5.5.2 Step 2 results

Step 2 aims to select the best text processing (Stem or POS) and machine learning methods (NB, SVM, MLP, CNN). The respective results are presented in Table 5.5 and in terms of two cross-source types of results: Amazon→Tripadvisor and Tripadvisor→Amazon. The table highlights in **bold** the best AUC result per test target source (Tripadvisor or Amazon), language (English or Italian) and number of classes (2 or 3).

TABLE 5.4: AUC (macro-average F1-score, accuracy) results for sentiment classification in step 1 (best AUC values per dataset and same number of classes are in **bold**).

Balance	Class	Model	English				Italian			
			Stem		POS		Stem		POS	
			Amazon	Tripadvisor	Amazon	Tripadvisor	Amazon	Tripadvisor	Amazon	Tripadvisor
Under	2	NB	0.52 (0.48, 0.59)	0.53 (0.49, 0.54)	0.65 (0.54, 0.64)	0.61 (0.56, 0.62)	0.57 (0.48, 0.61)	0.59 (0.46, 0.58)	0.59 (0.47, 0.59)	0.59 (0.46, 0.58)
		SVM	**0.54** (0.45, 0.58)	0.52 (0.46, 0.67)	0.67 (0.54, 0.63)	0.64 (0.55, 0.61)	0.55 (0.46, 0.62)	0.53 (0.47, 0.76)	0.42 (0.39, 0.71)	0.50 (0.47, 0.87)
		MLP	0.50 (0.40, 0.53)	0.51 (0.47, 0.56)	0.64 (0.59, 0.75)	0.66 (0.61, 0.68)	0.62 (0.49, 0.62)	0.55 (0.49, 0.65)	0.63 (0.50, 0.64)	0.50 (0.25, 0.41)
		CNN	0.51 (0.44, 0.54)	0.56 (0.50, 0.55)	**0.74*** (0.64, 0.74)	**0.81** (0.72, 0.76)	0.70 (0.53, 0.65)	**0.75*** (0.57, 0.70)	**0.73** (0.56, 0.69)	0.61 (0.44, 0.61)
	3	NB	0.52 (0.28, 0.47)	0.52 (0.28, 0.44)	0.63 (0.35, 0.61)	0.61 (0.34, 0.54)	0.57 (0.30, 0.56)	0.61 (0.32, 0.49)	0.59 (0.30, 0.55)	0.56 (0.27, 0.51)
		SVM	0.51 (0.20, 0.35)	0.50 (0.18, 0.29)	0.64 (0.32, 0.57)	0.62 (0.31, 0.51)	0.46 (0.16, 0.28)	0.46 (0.23, 0.44)	0.41 (0.12, 0.31)	0.43 (0.14, 0.37)
		MLP	0.52 (0.20, 0.32)	0.52 (0.30, 0.44)	0.55 (0.31, 0.51)	0.65 (0.40, 0.53)	0.60 (0.31, 0.48)	0.60 (0.33, 0.46)	0.61 (0.30, 0.48)	0.50 (0.10, 0.26)
		CNN	0.52 (0.26, 0.46)	0.55 (0.29, 0.36)	**0.69** (0.38, 0.58)	**0.78*** (0.50, 0.62)	0.66 (0.32, 0.47)	**0.74*** (0.40, 0.52)	**0.70** (0.33, 0.53)	0.55 (0.16, 0.33)
Over	2	NB	0.53 (0.49, 0.62)	0.53 (0.50, 0.55)	0.65 (0.54, 0.64)	0.61 (0.54, 0.60)	0.57 (0.47, 0.61)	0.60 (0.47, 0.59)	0.59 (0.47, 0.60)	0.59 (0.46, 0.58)
		SVM	**0.54** (0.49, 0.62)	0.51 (0.47, 0.68)	0.70 (0.55, 0.64)	0.65 (0.54, 0.60)	0.61 (0.49, 0.64)	0.61 (0.47, 0.59)	0.55 (0.47, 0.74)	0.60 (0.46, 0.57)
		MLP	0.49 (0.49, 0.79)	0.51 (0.50, 0.59)	0.65 (0.59, 0.75)	0.58 (0.50, 0.64)	0.60 (0.53, 0.85)	0.51(0.53, 0.84)	0.61 (0.55, 0.84)	0.50 (0.26, 0.43)
		CNN	0.50 (0.51, 0.70)	0.54 (0.50, 0.54)	0.70 (0.69, 0.83)	0.81 (0.75, 0.82)	0.67 (0.59, 0.86)	0.68(0.62, 0.87)	0.72 (0.58, 0.87)	0.65 (0.57, 0.81)
	3	NB	0.53 (0.29, 0.54)	0.53 (0.29, 0.43)	0.63 (0.34, 0.60)	0.50 (0.26, 0.38)	0.56 (0.29, 0.58)	0.59 (0.28, 0.54)	0.59 (0.31, 0.57)	0.50 (0.22, 0.37)
		SVM	0.53 (0.29, 0.46)	0.53 (0.23, 0.35)	**0.70** (0.37, 0.57)	0.51 (0.11, 0.15)	0.62 (0.33, 0.64)	0.61(0.27, 0.49)	0.58 (0.24, 0.42)	0.50 (0.09, 0.16)
		MLP	0.48 (0.29, 0.61)	0.49 (0.31, 0.65)	0.59 (0.37, 0.70)	0.50 (0.21, 0.39)	0.58 (0.36, 0.86)	0.51(0.35, 0.83)	0.58 (0.36, 0.84)	0.50 (0.31, 0.75)
		CNN	0.50 (0.32, 0.61)	0.53 (0.24, 0.32)	0.68 (0.46, 0.77)	0.50 (0.21, 0.32)	0.65 (0.40, 0.86)	0.65 (0.42, 0.86)	**0.70** (0.40, 0.86)	0.50 (0.33, 0.77)

* Statistically significant under a pairwise comparison when compared with the respective oversampling approach (p-value < 0.05).

For the Italian language, quality results were achieved, with all AUC values higher or equal to 0.70, thus similar to the single source experiments (Table 5.4). However, the English results are much lower than the ones obtained in step 1, being closer to the random classification (AUC of 0.50). To better understand this behavior, we analyzed the sentiment data source distributions (Table 5.6). Table 5.6 shows that the Amazon English (ENG) reviews are related to a reduced number of products (45). Moreover, this dataset presents a much higher standard deviation when compared with other data sources. To check if this difference is affecting the English results, we created a new dataset, termed Amazon ENG2, by removing the most reviewed product from Amazon ENG. This new dataset has a standard deviation that is more similar to the other sources (Table 5.6). We tested the new dataset in step 2 (Table 5.7). The obtained results show a substantial improvement in the classification performances (with statistical significance), with the best models obtaining AUC values that range from 0.74 to 0.81.

Analyzing the best step 2 results (Table 5.5 and Table 5.7), we conclude that the deep CNN model is the best machine learning algorithm, presenting the best overall AUC performances. Moreover, the POS tag processing method is the best option for the English language (using Amazon ENG2). For the Italian language, stemming leads to better results when Tripadvisor is used as the

TABLE 5.5: AUC (macro-average F1-score, accuracy) results for cross-source sentiment classification in step 2 (best AUC values per test source, language and same number of classes are in **bold**).

Classes	Algorithm	Amazon→Tripadvisor				Tripadvisor→Amazon			
		Stem		POS		Stem		POS	
		English	Italian	English	Italian	English	Italian	English	Italian
2	NB	**0.54** (0.52, 0.72)	0.59 (0.51, 0.70)	0.49 (0.45, 0.49)	0.60 (0.52, 0.72)	0.53 (0.47, 0.50)	0.57 (0.44, 0.53)	0.50 (0.45, 0.48)	0.58 (0.43, 0.52)
	SVM	**0.54** (0.49, 0.65)	0.59 (0.49, 0.67)	0.51 (0.41, 0.49)	0.44 (0.40, 0.64)	0.51 (0.46, 0.71)	0.60 (0.44, 0.53)	0.51 (0.44, 0.72)	0.49 (0.47, 0.87)
	MLP	0.52 (0.50, 0.77)	0.58 (0.47, 0.62)	0.51 (0.48, 0.51)	0.61 (0.51, 0.68)	0.52 (0.49, 0.55)	0.59 (0.49, 0.62)	0.50 (0.45, 0.49)	0.58 (0.47, 0.59)
	CNN	0.53 (0.44, 0.50)	0.72 (0.53, 0.67)	0.51 (0.48, 0.51)	**0.75** (0.56, 0.69)	**0.55*** (0.50, 0.54)	**0.76*** (0.58, 0.71)	0.49 (0.46, 0.50)	0.70 (0.54, 0.65)
3	NB	**0.52** (0.31, 0.57)	0.58 (0.34, 0.68)	0.50 (0.27, 0.34)	0.55 (0.32, 0.65)	0.52 (0.28, 0.39)	0.58 (0.26, 0.44)	0.51 (0.26, 0.32)	0.56 (0.27, 0.47)
	SVM	0.52 (0.21, 0.32)	0.48 (0.22, 0.37)	0.50 (0.18, 0.28)	0.44 (0.11, 0.19)	0.48 (0.15, 0.23)	0.45 (0.14, 0.25)	0.45 (0.15, 0.23)	0.46 (0.19, 0.35)
	MLP	0.51 (0.25, 0.48)	0.56 (0.29, 0.52)	0.50 (0.28, 0.37)	0.59 (0.28, 0.44)	0.51 (0.30, 0.40)	0.58 (0.29, 0.46)	0.50 (0.27, 0.33)	0.57 (0.27, 0.42)
	CNN	0.52 (0.26, 0.36)	0.69 (0.30, 0.46)	0.50 (0.27, 0.33)	**0.70** (0.35, 0.52)	**0.54*** (0.28, 0.35)	**0.72*** (0.36, 0.52)	0.50 (0.28, 0.35)	0.69 (0.31, 0.46)

* Statistically significant under a pairwise comparison when compared with other approaches for the same test source and language (p-value < 0.05).

TABLE 5.6: Statistics of the data source reviews.

	Amazon ENG	Amazon ENG2	Tripadvisor ENG	Amazon ITA	Tripadvisor ITA
Number of items	45	44	96	123	116
Number reviews	282,781	207,898	519,735	161,443	324,376
Mean (reviews/items)	6,289	4,730	5,413	1,312	2,816
Median	1,803	1,716	2,377	1,123	1,162
Standard Deviation	12,042	6,039	6,707	1,337	7,031
Minimum	10	10	71	20	219
Maximum	74,883	28,888	27,141	7,475	57,864

training source, while POS tag outperforms stemming when Amazon training data is used. Since the performance differences are slight (ranging from 1 to 6 percentage points), we opted to select stemming for the Italian language, since it provides the highest AUC values (0.76 for 2 classes and 0.72 for 3 classes).

5.5.3 Step 3 results

Using the sentiment models selected in step 2 (undersampling, usage of CNN, POS tag for the English language, stemming for the Italian language), we executed the final step 3 (Section 5.3.2), aiming to measure the value of using easy labeled sources (Amazon and Tripadvisor) to train sentiment models that are evaluated on typically non labeled sources (Facebook and Twitter). Table 5.8 shows the obtained performances for the proposed cross-source cross-domain CNN (CS-CD CNN). This approach is compared with two methods: a sentiment lexicon and a cross-domain sentiment classification method. We selected

TABLE 5.7: AUC (macro-average F1-score, accuracy) results for cross-source sentiment classification in step 2 and using Amazon ENG2 (best AUC values per number of classes are in **bold**).

Classes	Algorithm	Amazon ENG2 → Trip. ENG		Trip. ENG→ Amazon ENG2	
		Stem	POS	Stem	POS
2	NB	0.50 (0.45, 0.59)	0.63 (0.59, 0.77)	0.50 (0.46, 0.50)	0.62 (0.55, 0.59)
	SVM	0.50 (0.44, 0.59)	0.67 (0.55, 0.66)	0.50 (0.45, 0.72)	0.62 (0.54, 0.59)
	MLP	0.50 (0.42, 0.61)	0.64 (0.59, 0.76)	0.50 (0.44, 0.52)	0.66 (0.61, 0.68)
	CNN	0.49 (0.39, 0.53)	**0.78*** (0.66, 0.76)	0.50 (0.44, 0.49)	**0.81*** (0.70, 0.75)
3	NB	0.49 (0.23, 0.37)	0.61 (0.39, 0.74)	0.50 (0.25, 0.32)	0.62 (0.33, 0.51)
	SVM	0.52 (0.22, 0.37)	0.66 (0.36, 0.65)	0.51 (0.16, 0.22)	0.61 (0.30, 0.48)
	MLP	0.49 (0.19, 0.37)	0.62 (0.35, 0.58)	0.50 (0.23, 0.37)	0.63 (0.41, 0.58)
	CNN	0.51 (0.19, 0.36)	**0.74*** (0.41, 0.60)	0.51 (0.26, 0.34)	**0.76*** (0.48, 0.60)

* Statistically significant under a pairwise comparison when compared with other approaches using the same number of classes (p-value < 0.05).

the crowdsourcing lexicon proposed by Mohammad and Turney, 2013, since it supports both English and Italian languages. As for the cross-domain method, we used the AE-SCL version whose code is freely available in GitHub[4]. The AE-SCL was trained using Blitzer's Amazon product reviews and tested on Twitter and Facebook data. We note that the AE-SCL code only supports the English language and a binary sentiment classification, thus the Italian and three class results are omitted for this method in Table 5.8.

The best results are achieved by the CS-CD CNN method for Facebook (English and Italian). When compared with the lexicon (Mohammad and Turney, 2013) and AE-SCL (Ziser and Reichart, 2017), the proposed CS-CD CNN is competitive for the Facebook data, producing the best AUC values (with statistical significance). For Twitter, CS-CD CNN compares favourably in terms of AUC values for the Italian binary classification and English three class, obtaining the same AUC values as the crowdsourcing lexicon for the English binary classification. The AE-SCL produces the second best Facebook English AUC values. The generic crowdsourcing lexicon achieves the worst Facebook English AUC results but obtains the best AUC value for the Twitter Italian three

[4] https://github.com/yftah89/structural-correspondence-learning-SCL

TABLE 5.8: AUC (macro-average F1-score, accuracy) results for cross-source sentiment classification in step 3 (includes a comparison with two other methods; best AUC values in **bold**).

Classes	Algorithm	Target: Facebook		Target: Twitter	
		English	Italian	English	Italian
2	CS-CD CNN	**0.81*** (0.72, 0.81)	**0.78*** (0.73, 0.60)	**0.68** (0.60, 0.61)	**0.60** (0.56, 0.56)
	Lexicon	0.67 (0.64, 0.70)	0.56 (0.58, 0.58)	**0.68** (0.68, 0.70)	0.56 (0.56, 0.62)
	AE-SCL	0.74 (0.25, 0.28)	-	0.50 (0.50, 0.56)	-
3	CS-CD CNN	**0.76*** (0.49, 0.60)	**0.80*** (0.55, 0.51)	**0.65** (0.37, 0.46)	0.50 (0.35, 0.35)
	Lexicon	0.59 (0.46, 0.63)	0.54 (0.37, 0.49)	0.62 (0.51, 0.51)	**0.55** (0.36, 0.47)

* Statistically significant under a pairwise comparison when compared with other approaches using the same number of classes (p-value < 0.05).

class case, although the 0.55 value is close to the random AUC discrimination of 0.50.

5.5.4 Discussion

Table 5.9 summarizes the main AUC results achieved by the proposed CNN method in all three steps. It is interesting to notice that step 2 (cross Web sources and cross domain SA) improves the test classification performance for Amazon when Tripadvisor is used as training domain. Specifically, Amazon English AUC in step 2 raises by 7 percentage points (p.p.) for both classification tasks (with 2 and 3 classes) when compared with the step 1 results. Similarly, the Amazon Italian AUC increases by 3 p.p. for the binary classification and 2 p.p. for the three-class task. In contrast, there is slight decrease in the AUC performance (from 2 to 4 p.p.) for Tripadvisor when using Amazon training data. The exception is the binary Italian case, which results in the same AUC (75%).

More important are the CS-CD CNN step 3 results for Facebook, which correspond to high quality AUC values: 0.81 for the binary and 0.76 for the three class English classification. Similar quality results were reached for the Italian language, with 0.78 and 0.80 for the binary and three class classifications. As shown in Table 5.9, these AUC results compare well with the single source (step 1) and cross Web sources (step2) test performances. In effect, the AUC values range from: step1 – 0.69 to 0.81; and step2 – 0.70 to 0.81. This comparison confirms that the proposed CS-CD CNN method is valuable when

TABLE 5.9: Summary of the main CNN sentiment classification results (AUC values).

Classes	Target domain	English		Italian		Target domain	English	Italian
		Step 1	Step 2	Step 1	Step 2		Step 3	Step 3
2	Amazon	0.74	0.81	0.73	0.76	Facebook	0.81	0.78
3		0.69	0.76	0.70	0.72		0.76	0.80
2	Tripadvisor	0.81	0.78	0.75	0.75	Twitter	0.68	0.60
3		0.78	0.74	0.74	0.70		0.65	0.55

using Facebook as a target test source. For demonstration purposes, Figure 5.3 presents the word clouds, after the POS tag removal, for the most frequent 100 words when using the Amazon, Tripadvisor and Facebook English data. The word clouds denote some similarity among the text sources (e.g., high frequency of *great*, *good*, *best*, *nice* and *bad* terms), helping to explain why the CS-CD approach provides good results for Facebook.

FIGURE 5.3: Example of word clouds (first 100 words) for pre-processed English data.

For Twitter, a reasonable discrimination was achieved in three cases (AUC>0.60). Better results were obtained for the English language, while a poor performance (similar to a random classifier) was achieved for the Italian three class classification. The best performance on Facebook target source was expected, since Facebook comments are not restricted to the character size limit of Twitter, thus the used sentiment words should be more similar to the Amazon and Tripadvisor source reviews. Also, the language differences might be explained by higher complexity of the Italian Latin language in terms of the type of tenses and adjectives used. Indeed, we note that in step 3 the average number of words is: English – 10.2 for Facebook and 4.0 for Twitter; and Italian – 27.0 for Facebook and 13.0 for Twitter. These values denote differences between the text sources, especially for Twitter. For example, the POS tag average sentence size is 10.7 for English Amazon (Table 5.2), which is much closer to the 10.2 value of Facebook than the 4.0 of Twitter. Moreover, users tend to write tweets

with slang and abbreviations, which typically are sparse and thus are not easily visible when analyzing word clouds. Two real examples of such tweets are:

- "i luv the book'da vinci code"; and
- "omgg i ohhdee want mcdonalds damn i wonder if its open lol".

Since slang and abbreviations (e.g., *luv*, *omgg*) are not often used in Amazon or Tripadvisor reviews, the CS-CD model would produce poor results when tested with these type of tweets. For demonstration purposes, Table 5.10 shows 10 examples of the binary CS-CD CNN probability for the positive class. In this example, the model correctly identifies the sentiment of 4 Facebook posts and 3 tweets. In particular, the last two rows of Table 5.10 exemplify that the CS-CD CNN does not correctly detect the sentiment polarity for tweets with the *argghhhh* slang and *omgg* abbreviation.

TABLE 5.10: Examples of binary CS-CD CNN positive sentiment classification (correct values using a 0.5 classification threshold are in **bold**).

Text[a]	**Source**[b]	**Target**[c] **class**	**CS-CD CNN probability**
Just back from a superb few days in Liverpool, much of which was spent in this wonderful club. The staff and musicians were excellent[...]	FB	1	**1.00**
First time in with hen party and must say barmaid was sooo rude n sharp wen asked for some merchendise even tho[...]	FB	0	**0.00**
Absolutely fabulous want to go again went with my three girls. Next time I would stay alot longer and want to write my name on the wall[...]	FB	1	**0.93**
Needed at least 3 full days... going back, absolute!! Fantastic 'premium' exclusive collections.	FB	1	**0.74**
A incredible journey back in time. You can feel the history surrounding you[...]	FB	1	0.27
Jquery is my new best friend.	TW	1	**0.74**
I'm itchy and miserable!	TW	0	**0.02**
Obama's speech was pretty awesome last night!	TW	1	**0.74**
argghhhh why won't my jquery appear in safari bad safari!!!	TW	0	0.78
omgg i ohhdee want mcdonalds damn i wonder if its open lol	TW	1	0.39

[a] **Text** – [...]: truncated text. The complete data are available at https://github.com/paolazola/Cross-source-cross-domain-sentiment-analysis.

[b] **Source** – FB: Facebook, TW: Twitter.

[c] **Target class** – 0: negative sentiment, 1: positive sentiment.

The obtained results for the proposed CNN model confirm that the combination of freely available labeled Web sources, such as Amazon and Tripadvisor, can help to train generic sentiment analysis models that provide valuable predictions when applied to unlabeled social media texts, particularly for Facebook. The proposed CS-CD approach alleviates the need for arduous human labeling of these social media texts and thus it can be a key element of modern decision support systems. For instance, to perform social media analytics in the areas of Marketing and Finance (e.g., brand monitoring, customer support, analysis of commodity price opinions).

Chapter 6

Conclusion and future works

This thesis researched about the recent trend of evaluating financial prices dynamics given opinionated texts. In particular, the sentiment is extracted by social media or modern web 2.0 platforms where people create free textual contents. Specifically, the focus of this research is related to the steel prices dynamics given the importance of steel in the world and the scarce amount of studies about it. However, evaluating the predictability of steel prices given messages from social media, like Twitter, involves further challenges that are not common to the well know sentiment analysis for stocks or financial indexes, as reported in the Chapter 2. This thesis presented a coherent framework and set of social media analytics tools that is capable of extracting from the web 2.0 community the useful data to study the dynamics of tweets and steel prices. However, all the proposed the proposed framework and tools can be easily extended and applied to other commodities/alloys prices dynamics. In particular, the following tools were developed:

- GTN/GTN2 algorithms: aiming to locate the country of interest of Twitter users, in order to infer the steel market to which the tweet is referred;
- TFD-FUR algorithms: targeted to disambiguate financial tweets that shared the same keyword but are related to the alloy steel and create a rank of the most expert users in the financial application domain;
- CS-CD CNN SA: the model focuses on sentiment classification for short texts by applying a transfer learning approach.

The following Sections 6.1–6.2–6.3 report the main conclusions and provide a brief discussion of the results achieved in the Chapters 3–4–5.

6.1 Geolocation of financial social media texts

With the expansion of the Internet, web and social media analytics are a key tool of diverse decision support systems. Several of these social media analytic

systems require user geographic location data. In the work proposed in Chapter 3, we propose a novel GTN approach to detect the most probable Twitter user country of interest when such context is not explicitly known. GTN is a purely word distribution method that does not require training data. It is based on the frequency of users' tweet nouns and GT country word distribution data. The main advantage of the GTN method, with respect to existing geographic dictionary models, is its ability to obtain information from generic and adaptable nouns, dynamically provided by GT, such as "Brexit", "Trump", or "cricket". Moreover, using GT as source, the GTN method is able to benefit from country term frequency or language differences. Conversely, the GTN has some limitations. For example, as shown in Table 3.7, there are popular generic nouns (e.g., "time" and "year") that show a similar frequency of use in different countries. In addition, GTN assumes just one implicit country of interest, whereas some users might travel or tweet implicitly about more than one country.

Following a design science research methodology Arnott and Pervan (2014), we validated GTN empirically. Using a conservative procedure, we created a recent dataset with 3,298 Twitter users from 54 countries with 744,830 tweets written in 48 languages. The obtained GTN results are of high quality (83% accuracy and weighted F1-score) and competitive when compared with a state-of-the-art word distribution method (Lee et al., 2015). An error analysis was also performed on the GTN misclassifications, revealing different types of errors, such as mismatches between different anglophone countries (32% of the errors) and between countries that are similar or share a language or location (3%). Several experiments were conducted, using four machine learning classifiers: bagging (BG), random forest (RF), support vector machines (SVM), and a deep learning inspired multilayer perceptron (MLP). The experiments have shown that the GTN errors are difficult to outperform, confirming the value of the GTN responses. One limitation of GTN is its dependency on GT and the required GT request time. As an alternative, we tested the GTN2 approach, in which a machine learning method models the GTN responses. The best results were achieved by the GTN2 MLP model (80% accuracy and 78% weighted F1-score when modeling GTN), which is a much faster method than GTN. Finally, we have demonstrated the applicability of GTN to non-steel commodities (such as cotton), using more recent Twitter data and a different but smaller sample of users.

Because the percentage of geotagged tweets is small and Twitter user profile location data is frequently unreliable (Cheng, Caverlee, and Lee, 2010; Hecht et al., 2011), as also shown in this study, the proposed GTN and GTN2 approaches can be valuable to support Web and social media analytic systems. In future work, we intend to apply GTN in real-world applications, such as

for filtering country tweets related to a particular commodity price (for example, gold or wheat prices from Germany). In addition, we wish to complement GTN with extra geolocation features, such as friendship networks or user profile metadata, and investigate more fine-grained location levels. Finally, we plan to research whether feature selection filtering methods, such as pointwise mutual information (Oliveira, Cortez, and Areal, 2016), can be used to discard the GTN generic nouns that are used equally by different countries, thereby potentially improving the GTN performance. However, we note that such a filtering approach would require a GTN adaptation that involves a training set.

6.2 Twitter financial disambiguation and user relevance

Twitter is becoming a valuable big data source for social media analytics. Focusing on financial stocks or indexes, Twitter messages are easily retrieved by using search queries with specific casthags (e.g., $AAPL for Apple stocks). However, the Twitter extraction of other financial opinion tweets, such as related with alloys (e.g., steel, bronze) or commodities (e.g., gold, coffee), is a non-trivial task, as it requires a keywords search that often results in irrelevant texts.

In the Chapter 4, we propose an automatic filter approach, termed Twitter Financial Disambiguation (TFD), aiming to extract financial related tweets and without the need of an human labeling. We achieve this by using a transfer learning, in which freely news titles are used to train diverse TFD models, under two main learning approaches: unary, with only positive texts; and binary, with positive and negative texts. The TFD models include: adaptations of distance and statistical measures (CD and DTW), Term-Frequency Inverse Document Frequency Classification (TF-IDFC), Information Gain (IG) and Pointwise Mutual Information (PMI); and recent machine learning methods, namely simple and deep Siamese Autoencoder (SiAE), Support Vector Machine (SVM), Random Forest (RF) and deep Multilayer Perceptron (MLP). Also, we test distinct text handling methods, namely the raw string, a TF-IDF transform and a Word2Vec (W2V) encoding. Moreover, we propose a Financial User Relevance rank (FUR) score. The advantage of FUR is that is allows to filter relevant users by using only the keywords query texts and not additional social media or user data features that are required by the state of the art studies.

As a case study, we considered the alloy steel prices domain. We performed several steel prices Twitter queries that resulted in 533,759 unlabeled tweets collected from March 2017 to October 2018. Then, we executed a realistic

rolling window validation procedure, with several train and test model updates, aiming to tune and compare the diverse unary and binary TFD models. The first rolling window experiments, using 11,081 manually labeled tweets as the test set, revealed that the best unary discrimination performance is obtained by TF-IDFC, while the best binary training method was obtained by a SVM fed with TFD binary statistical measures (TF-IDFC, IG and PMI) and topic relevance features obtained using the Latent Dirichlet Allocation (LDA) and Biterm Topic Model (BTM) text clustering algorithms. Overall, the binary trained SVM model obtained an Area Under the receiver operating characteristic Curve (AUC) of 80%, while the unary TF-IDFC achieved a slight lower value (AUC of 78%). These two models were selected for further experiments that used a second rolling window procedure. The experiments confirmed that SVM produces a better discrimination for TFD prediction when using an extra (unseen) set of 3,000 labeled tweets (the AUC was 71% for SVM and 0.69% for TF-IDFC). Moreover, the same rolling window experiment was used to test the SVM and TF-IDFC TFD models predictive performance to discriminate relevant users when using the FUR score and a manually labeled set of 418 users. The best predictive performance was also obtained by SVM, which presented an AUC of 80%, while TF-IDFC obtained an AUC of 75%. Given these results, we recommend the usage of the binary SVM model for TFD-FUR, since it consistently provided the best results. As an alternative, in particular if labeled negative tests are not easy to collect, we suggest the simpler unary TF-IDFC.

The proposed approach, based on freely labeled news titles, allows an automatic TFD-FUR for Twitter, alleviating the need for a laborious human labeling of tweets or curated lists of relevant user accounts (e.g., web companies) regarding a specific financial domain. Thus, it is valuable as filtering step to be used by financial social media expert systems (e.g., sentiment analysis, recommendation users to follow). In future work, we intend to address other case studies, such as commodity (e.g., gold, coffee) or other alloy (e.g., bronze, copper) prices. Moreover, the proposed FUR models only rely on the tweets retrieved using the keywords query, thus the models could be further complemented by using other features, such as user account profile data (e.g., web site).

6.3 Sentiment analysis for social media texts

In the work proposed in the Chapter 5 a novel cross-source cross-domain sentiment analysis has been explored. The goal is to easily classify the sentiment of distinct items (e.g., restaurant, hotel, book, music) by first fitting a sentiment classifier to easy-to-collect labeled Web sources (from Amazon and Tripadvisor) and then reusing such model to predict the sentiment of typically

unlabeled social media reviews (from Facebook and Twitter). Thus, the cross-source transfer learning approach alleviates the need to construct sentiment models for each single data source and does not require any human effort to classify unlabeled texts.

We adopted a three step experimental methodology, in which distinct modeling methods were tested: balancing training methods – undersampling and oversampling; text preprocessing – stemming and Part-of-Speech (POS) tagging; and learning algorithms – Naive Bayes (NB), Support Vector Machine (SVM), deep Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN). We also considered two different languages (English and Italian) and two types of sentiment classification ({"negative", "positive"} and {"negative", "neutral", "positive"}). The first two steps confirmed the undersampling and CNN learning algorithm as the best modeling approach. Also, the selection of adverbs and adjectives via POS tagging resulted in the best English results, while stemming led to slight better Italian classification performances. In the last step, we applied the selected models under the proposed cross-source cross-domain approach. When using both Amazon and Tripadvisor training sources, the most important results are the high quality classification performances that were obtained using the Facebook source as the target domain. Indeed, the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve was 81% for the polarity classification and 76% for three class classification on the English language. Similar results were reached for the Italian language: 78% AUC for polarity and 80% for three classes. As for Twitter, a reasonable discrimination was achieved for the English language (AUC from 65% to 68%).

To the best of our knowledge, we believe that this is the first work that considered a social media cross-source cross-domain sentiment classification, which is valuable to reduce the laborious human labeling of texts in modern social network platforms, in particular when using Facebook as test target source. In the future, we expect to extend the proposed methodology to other languages (e.g., German, Portuguese), aiming to discover patterns among the language families (e.g., Germanic, Latin). Moreover, other improvements could be achieved by adopting a deep contextualized word representation based on attention networks Peters et al. (2018). We also intend to experiment with other Web opinion platforms, such as Foursquare (https://foursquare.com/) or StockTwits (https://stocktwits.com/).

Bibliography

Adedoyin-Olowe, Mariam et al. (2016). "A rule dynamics approach to event detection in twitter with its application to sports and politics". In: *Expert Systems with Applications* 55, pp. 351–360.

Amiri, Hadi et al. (2016). "Learning text pair similarity with context-sensitive autoencoders". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 1882–1892.

Arnott, David and Graham Pervan (2014). "A critical analysis of decision support systems research revisited: the rise of design science". In: *JIT* 29.4, pp. 269–293. DOI: 10.1057/jit.2014.16. URL: https://doi.org/10.1057/jit.2014.16.

Aue, Anthony and Michael Gamon (2005). "Customizing sentiment classifiers to new domains: A case study". In: *Proceedings of recent advances in natural language processing (RANLP)*. Vol. 1. 3.1. Citeseer, pp. 2–1.

Aulov, Oleg and Milton Halem (2012). "Human sensor networks for improved modeling of natural disasters". In: *Proceedings of the IEEE* 100.10, pp. 2812–2823.

Avvenuti, Marco et al. (2018). "GSP (Geo-Semantic-Parsing): Geoparsing and Geotagging with Machine Learning on top of Linked Data". In: *European Semantic Web Conference*. Springer, pp. 17–32.

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani (2010). "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." In: *LREC*. Vol. 10. 2010, pp. 2200–2204.

Backstrom, Lars, Eric Sun, and Cameron Marlow (2010). "Find me if you can: improving geographical prediction with social and spatial proximity". In: *Proceedings of the 19th international conference on World wide web*. ACM, pp. 61–70.

Baker, Malcolm and Jeffrey Wurgler (2007). "Investor sentiment in the stock market". In: *Journal of economic perspectives* 21.2, pp. 129–152.

Baldridge, Jason (2005). "The opennlp project". In: *URL: http://opennlp. apache. org/index. html, (accessed 2 February 2012)*.

Banerjee, Satanjeev and Ted Pedersen (2002). "An adapted Lesk algorithm for word sense disambiguation using WordNet". In: *International conference on intelligent text processing and computational linguistics*. Springer, pp. 136–145.

Bao, Te, Cars Hommes, and Tomasz Makarewicz (2017). *Bubble Formation and (In) Efficient Markets in Learning-to-forecast and optimise Experiments*.

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski (2014). "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 238–247.

Batista, Gustavo EAPA, Ronaldo C Prati, and Maria Carolina Monard (2004). "A study of the behavior of several methods for balancing machine learning training data". In: *ACM SIGKDD explorations newsletter* 6.1, pp. 20–29.

Bengio, Yoshua et al. (2003). "A neural probabilistic language model". In: *Journal of machine learning research* 3.Feb, pp. 1137–1155.

Bishop, Christopher M. (2007). *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer. ISBN: 9780387310732. URL: http://www.worldcat.org/oclc/71008143.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Blitzer, John, Mark Dredze, and Fernando Pereira (2007). "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification". In: *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440–447.

Bollegala, Danushka, David Weir, and John Carroll (2011). "Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pp. 132–141.

Bollen, Johan, Huina Mao, and Xiaojun Zeng (2011). "Twitter mood predicts the stock market". In: *Journal of computational science* 2.1, pp. 1–8.

Borthwick, Andrew and Ralph Grishman (1999). "A maximum entropy approach to named entity recognition". PhD thesis. Citeseer.

Breiman, L. (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Buyukkokten, Orkut et al. (1999). "Exploiting Geographical Location Information of Web Pages". In: *ACM SIGMOD Workshop on The Web and Databases, WebDB 1999, Philadelphia, Pennsylvania, USA, June 3-4, 1999. Informal Proceedings*. Ed. by Sophie Cluet and Tova Milo. INRIA, pp. 91–96. URL: http://www-rocq.inria.fr/\%7Ecluet/WEBDB/gravano.ps.

Cai, Junfu, Wee Sun Lee, and Yee Whye Teh (2007). "Improving word sense disambiguation using topic features". In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

Castillo, Carlos, Marcelo Mendoza, and Barbara Poblete (2011). "Information credibility on twitter". In: *Proceedings of the 20th international conference on World wide web*. ACM, pp. 675–684.

Chang, Hau-wen et al. (2012). "@ Phillies tweeting from Philly? Predicting Twitter user locations with spatial word usage". In: *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. IEEE Computer Society, pp. 111–118.

Chang, Ming-Wei et al. (2008). "Importance of Semantic Representation: Dataless Classification." In: *AAAI*. Vol. 2, pp. 830–835.

Chaplot, Devendra Singh and Ruslan Salakhutdinov (2018). "Knowledge-based word sense disambiguation using topic models". In: *Thirty-Second AAAI Conference on Artificial Intelligence*.

Chen, Danqi and Christopher Manning (2014). "A fast and accurate dependency parser using neural networks". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750.

Cheng, Zhiyuan, James Caverlee, and Kyumin Lee (2010). "You are where you tweet: a content-based approach to geo-locating twitter users". In: *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, pp. 759–768.

Choi, Hyunyoung and Hal Varian (2009). "Predicting initial claims for unemployment benefits". In: *Google Inc*, pp. 1–5.

— (2012). "Predicting the present with Google Trends". In: *Economic Record* 88.s1, pp. 2–9.

Compton, Ryan, David Jurgens, and David Allen (2014). "Geotagging one hundred million twitter accounts with total variation minimization". In: *Big Data (Big Data), 2014 IEEE International Conference on*. IEEE, pp. 393–401.

Conneau, Alexis et al. (2017). "Very deep convolutional networks for text classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Vol. 1, pp. 1107–1116.

Cortez, Paulo (2010). "Data mining with neural networks and support vector machines using the R/rminer tool". In: *Industrial Conference on Data Mining*. Springer, pp. 572–583.

Cortez, Paulo, Nuno Oliveira, and João Peixoto Ferreira (2016). "Measuring user influence in financial microblogs: experiments using stocktwits data". In: *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*. ACM, p. 23.

Crandall, David J et al. (2009). "Mapping the world's photos". In: *Proceedings of the 18th international conference on World wide web*. ACM, pp. 761–770.

Cristianini, Nello and John Shawe-Taylor (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Cui, Hang, Vibhu Mittal, and Mayur Datar (2006). "Comparative experiments on sentiment classification for online product reviews". In: *AAAI*. Vol. 6, pp. 1265–1270.

Dalla Valle, L. and R. Kenett (2018). "Social media big data integration: A new approach based on calibration". In: *Expert Systems with Applications* 111, pp. 76–90.

Dalla Valle, L. and R. S. Kenett (2015). "Official statistics data integration for enhanced information quality". In: *Quality and Reliability Engineering International* 31.7, pp. 1281–1300.

Dalvi, Nilesh, Ravi Kumar, and Bo Pang (2012). "Object matching in tweets with spatial models". In: *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, pp. 43–52.

Dang, Yan, Yulei Zhang, and Hsinchun Chen (2010). "A lexicon-enhanced method for sentiment classification: An experiment on online product reviews". In: *IEEE Intelligent Systems* 25.4, pp. 46–53.

Daniel, Mariana, Rui Ferreira Neves, and Nuno Horta (2017). "Company event popularity for financial markets using Twitter and sentiment analysis". In: *Expert Systems with Applications* 71, pp. 111–124.

Daudert, Tobias, Paul Buitelaar, and Sapna Negi (2018). "Leveraging News Sentiment to Improve Microblog Sentiment Classification in the Financial Domain". In: *Proceedings of the First Workshop on Economics and Natural Language Processing*, pp. 49–54.

Dave, Kushal, Steve Lawrence, and David M Pennock (2003). "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews". In: *Proceedings of the 12th international conference on World Wide Web*. ACM, pp. 519–528.

Davis Jr, Clodoveu A et al. (2011). "Inferring the location of twitter messages based on user relationships". In: *Transactions in GIS* 15.6, pp. 735–751.

Dong, Y. et al. (2018a). "A survey on the fusion process in opinion dynamics". In: *Information Fusion* 43, pp. 57–65. DOI: 10.1016/j.inffus.2017.11.009. URL: https://doi.org/10.1016/j.inffus.2017.11.009.

Dong, Y. et al. (2018b). "Consensus reaching in social network group decision making: Research paradigms and challenges". In: *Knowledge-Based Systems* 162, pp. 3–13. DOI: 10.1016/j.knosys.2018.06.036. URL: https://doi.org/10.1016/j.knosys.2018.06.036.

Dragoni, Mauro and Giulio Petrucci (2017). "A neural word embeddings approach for multi-domain sentiment analysis". In: *IEEE Transactions on Affective Computing* 8.4, pp. 457–470.

— (2018). "A fuzzy-based strategy for multi-domain sentiment analysis". In: *International Journal of Approximate Reasoning* 93, pp. 59–73.

Eliacik, Alpaslan Burak and Nadia Erdogan (2018). "Influential user weighted sentiment analysis on topic based microblogging community". In: *Expert Systems with Applications* 92, pp. 403–418.

Fang, Xing and Justin Zhan (2015). "Sentiment analysis using product review data". In: *Journal of Big Data* 2.1, p. 5.

Fang, Ze-Han and Chien Chin Chen (2016). "A novel trend surveillance system using the information from web search engines". In: *Decision Support Systems* 88, pp. 85–97. DOI: 10.1016/j.dss.2016.06.001. URL: https://doi.org/10.1016/j.dss.2016.06.001.

Fawcett, Tom (2006). "An introduction to ROC analysis". In: *Pattern recognition letters* 27.8, pp. 861–874.

Feinerer, Ingo et al. (2013). "The textcat package for n-gram based text categorization in R". In: *Journal of statistical software* 52.6, pp. 1–17.

Fernández-Delgado, Manuel et al. (2014). "Do we need hundreds of classifiers to solve real world classification problems?" In: *The Journal of Machine Learning Research* 15.1, pp. 3133–3181.

Feuerriegel, Stefan and Dirk Neumann (2013). "News or noise? How news drives commodity prices". In:

Fu, X. et al. (2018). "Mining Newsworthy Events in the Traffic Accident Domain From Chinese Microblog". In: *International Journal of Information Technology & Decision Making*.

Ganin, Y. et al. (2016). "Domain-adversarial training of neural networks". In: *The Journal of Machine Learning Research* 17.1, pp. 2096–2030.

Gao, Lin and Stephan Süss (2015). "Market sentiment in commodity futures returns". In: *Journal of Empirical Finance* 33, pp. 84–103.

Gayo-Avello, Daniel (2013). "Nepotistic relationships in twitter and their impact on rank prestige algorithms". In: *Information Processing & Management* 49.6, pp. 1250–1280.

Geană, Corina Mihaela et al. (2018). ""CYBERSLANG" OR THE LANGUAGE USED BY THE INTERNET USERS (A–L)". In: *Journal of Romanian Literary Studies* 15, pp. 472–480.

Ghosh, Monalisa and Animesh Kar (2013). "Unsupervised linguistic approach for sentiment classification from online reviews using SentiWordNet 3.0". In: *Int J Eng Res Technol* 2.9.

Ginsberg, Jeremy et al. (2009). "Detecting influenza epidemics using search engine query data". In: *Nature* 457.7232, p. 1012.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). "Domain adaptation for large-scale sentiment classification: A deep learning approach". In:

Proceedings of the 28th international conference on machine learning (ICML-11), pp. 513–520.

Go, Alec, Richa Bhayani, and Lei Huang (2009). "Twitter sentiment classification using distant supervision". In: *CS224N Project Report, Stanford* 1.12.

Goldberg, Yoav (2017). "Neural network methods for natural language processing". In: *Synthesis Lectures on Human Language Technologies* 10.1, pp. 1–309.

Goodfellow, Ian et al. (2016). *Deep learning*. Vol. 1. MIT press Cambridge.

Gräbner, Dietmar et al. (2012). "Classification of Customer Reviews based on Sentiment Analysis". In: *Information and Communication Technologies in Tourism 2012*. Springer, pp. 460–470.

Greene, BB and GM Rubin (1971). *Automatic grammatical tagging of English. Department of Linguistics, Brown University, Providence*. Tech. rep. Rhode Island, Technical report.

Griffiths, Thomas L and Mark Steyvers (2004). "Finding scientific topics". In: *Proceedings of the National academy of Sciences* 101.suppl 1, pp. 5228–5235.

Groß-Klußmann, Axel, Stephan König, and Markus Ebner (2019). "Buzzwords build Momentum: Global Financial Twitter Sentiment and the Aggregate Stock Market". In: *Expert Systems with Applications* 136.1, pp. 171–186.

Guo, Kun, Yi Sun, and Xin Qian (2017). "Can investor sentiment be used to predict the stock price? Dynamic analysis based on China stock market". In: *Physica A: Statistical Mechanics and its Applications* 469, pp. 390–396.

Han, Bo, Paul Cook, and Timothy Baldwin (2014). "Text-based twitter user geolocation prediction". In: *Journal of Artificial Intelligence Research* 49, pp. 451–500.

Hand, D. (2006). "Classifier Technology and the Illusion of Progress". In: *Statistical Science* 21.1, pp. 1–15.

Hastie, T., R. Tibshirani, and J. Friedman (2008). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd. NY, USA: Springer-Verlag.

Hecht, Brent et al. (2011). "Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, pp. 237–246.

Hollander, Myles and Douglas A Wolfe (1999). *Nonparametric statistical methods*. Wiley-Interscience.

Huang, Dashan et al. (2018). "Sentiment across asset markets". In:

Iosif, Elias and Alexandros Potamianos (2015). "Similarity computation using semantic networks created from web-harvested data". In: *Natural Language Engineering* 21.1, pp. 49–79.

Ito, Jun et al. (2015). "Assessment of tweet credibility with LDA features". In: *Proceedings of the 24th International Conference on World Wide Web*. ACM, pp. 953–958.

Jo, Yohan and Alice H Oh (2011). "Aspect and sentiment unification model for online review analysis". In: *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, pp. 815–824.

Joachims, Thorsten (1998). "Text categorization with support vector machines: Learning with many relevant features". In: *European conference on machine learning*. Springer, pp. 137–142.

Kahle, David and Hadley Wickham (2013). "ggmap: Spatial Visualization with ggplot2." In: *R Journal* 5.1.

Kenter, Tom and Maarten De Rijke (2015). "Short text similarity with word embeddings". In: *Proceedings of the 24th ACM international on conference on information and knowledge management*. ACM, pp. 1411–1420.

Kim, Yoon (2014). "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751.

Kinsella, Sheila, Vanessa Murdock, and Neil O'Hare (2011). "I'm eating a sandwich in Glasgow: modeling locations with tweets". In: *Proceedings of the 3rd international workshop on Search and mining user-generated contents*. ACM, pp. 61–68.

Kotzias, Dimitrios, Theodoros Lappas, and Dimitrios Gunopulos (2016). "Home is where your friends are: Utilizing the social graph to locate twitter users in a city". In: *Information Systems* 57, pp. 77–87.

Kulkarni, Rohit (2017). *A Million News Headlines*. Tech. rep. doi:10.7910/DVN/SYBGZL.

Kumar, KL Santhosh, Jayanti Desai, and Jharna Majumdar (2016). "Opinion mining and sentiment analysis on online customer review". In: *Computational Intelligence and Computing Research (ICCIC), 2016 IEEE International Conference on*. IEEE, pp. 1–4.

Lai, Siwei et al. (2015). "Recurrent Convolutional Neural Networks for Text Classification." In: *AAAI*. Vol. 333, pp. 2267–2273.

Laylavi, Farhad, Abbas Rajabifard, and Mohsen Kalantari (2016). "A multi-element approach to location inference of twitter: A case for emergency response". In: *ISPRS International Journal of Geo-Information* 5.5, p. 56.

Lechthaler, Filippo and Lisa Leinert (2012). "Moody oil: What is driving the crude oil price?" In: *Empirical Economics*, pp. 1–32.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *Nature* 521.7553, p. 436.

Lee, Michael D, Brandon Pincombe, and Matthew Welsh (2005). "An empirical evaluation of models of text document similarity". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 27. 27.

Lee, Sunshin et al. (2015). "Read between the lines: A Machine Learning Approach for Disambiguating the Geo-location of Tweets". In: *Proceedings of*

the 15th ACM/IEEE-CS Joint Conference on Digital Libraries. ACM, pp. 273–274.

Li, J. et al. (2017). "Forecasting Oil Price Trends with Sentiment of Online News Articles". In: *Asia-Pacific Journal of Operational Research* 34.02, p. 1740019.

Li, Rui et al. (2012). "Towards social user profiling: unified and discriminative influence model for inferring home locations". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 1023–1031.

Li, Shoushan et al. (2011a). "Semi-supervised learning for imbalanced sentiment classification". In: *IJCAI proceedings-international joint conference on artificial intelligence*. Vol. 22. 3, p. 1826.

Li, Ximing et al. (2018). "Exploring coherent topics by topic modeling with term weighting". In: *Information Processing & Management* 54.6, pp. 1345–1358.

Li, Y. et al. (2011b). "Snippet-based unsupervised approach for sentiment classification of Chinese online reviews". In: *International Journal of Information Technology & Decision Making* 10.06, pp. 1097–1110.

Lim, Kwan Hui, Shanika Karunasekera, and Aaron Harwood (2017). "ClusTop: A clustering-based topic modelling algorithm for twitter using word networks". In: *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, pp. 2009–2018.

Lin, Yung-Shen, Jung-Yi Jiang, and Shie-Jue Lee (2014). "A similarity measure for text classification and clustering". In: *IEEE transactions on knowledge and data engineering* 26.7, pp. 1575–1590.

Litvak, M. and N. Vanetik (2019). *Multilingual Text Analysis Challenges, Models, and Approaches*. World Scientific.

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

Liu, Bing (2012). "Sentiment analysis and opinion mining". In: *Synthesis lectures on human language technologies* 5.1, pp. 1–167.

Liu, Xiaoying, Yiming Zhou, and Ruoshi Zheng (2007). "Sentence similarity based on dynamic time warping". In: *null*. IEEE, pp. 250–256.

Liu, Y., J.W. Bi, and Z.P. Fan (2017). "A method for ranking products through online reviews based on sentiment classification and interval-valued intuitionistic fuzzy TOPSIS". In: *International Journal of Information Technology & Decision Making* 16.06, pp. 1497–1522.

Lo, Andrew W (2004). "The adaptive markets hypothesis: Market efficiency from an evolutionary perspective". In:

Loria, Steven et al. (2014). "Textblob: simplified text processing". In: *Secondary TextBlob: Simplified Text Processing*.

Mahendhiran, P.D. and S. Kannimuthu (2018). "Deep Learning Techniques for Polarity Classification in Multimodal Sentiment Analysis". In: *International Journal of Information Technology & Decision Making* 17.03, pp. 883–910.

Mahmud, Jalal, Jeffrey Nichols, and Clemens Drews (2014). "Home location identification of twitter users". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 5.3, p. 47.

Malanichev, AG and PV Vorobyev (2011). "Forecast of global steel prices". In: *Studies on Russian Economic Development* 22.3, p. 304.

Malkiel, Burton G and Eugene F Fama (1970). "Efficient capital markets: A review of theory and empirical work". In: *The journal of Finance* 25.2, pp. 383–417.

Manevitz, Larry M and Malik Yousef (2001). "One-class SVMs for document classification". In: *Journal of machine Learning research* 2.Dec, pp. 139–154.

Manning, Christopher D and Hinrich Schütze (1999). *Foundations of statistical natural language processing*. MIT press.

Maslyuk-Escobedo, Svetlana, Kristian Rotaru, and Alexander Dokumentov (2017). "News sentiment and jumps in energy spot and futures markets". In: *Pacific-Basin Finance Journal* 45, pp. 186–210.

Mesnil, Grégoire et al. (2014). "Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews". In: *arXiv preprint arXiv:1412.5335*.

Middleton, Stuart E, Lee Middleton, and Stefano Modafferi (2014). "Real-time crisis mapping of natural disasters using social media". In: *IEEE Intelligent Systems* 29.2, pp. 9–17.

Mikolov, Tomas et al. (2013a). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119.

Mikolov, Tomas et al. (2013b). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Minot, Ariana S et al. (2015). "Searching for twitter posts by location". In: *Proceedings of the 2015 international conference on the theory of information retrieval*. ACM, pp. 357–360.

Mitchell, Ryan (2018). *Web Scraping with Python: Collecting More Data from the Modern Web*. " O'Reilly Media, Inc."

Mohammad, Saif M and Peter D Turney (2013). "Crowdsourcing a word–emotion association lexicon". In: *Computational Intelligence* 29.3, pp. 436–465.

Moor, James (2003). *The Turing test: the elusive standard of artificial intelligence*. Vol. 30. Springer Science & Business Media.

Morstatter, Fred et al. (2013). "Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose". In: *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013.* Ed. by Emre Kiciman et al.

The AAAI Press. ISBN: 978-1-57735-610-3. URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6071.

Mudinas, Andrius, Dell Zhang, and Mark Levene (2019). "Market trend prediction using sentiment analysis: lessons learned and paths forward". In: *arXiv preprint arXiv:1903.05440*.

Munzert, Simon et al. (2014). *Automated data collection with R: A practical guide to web scraping and text mining*. John Wiley & Sons.

Neculoiu, Paul, Maarten Versteegh, and Mihai Rotaru (2016). "Learning text similarity with siamese recurrent networks". In: *Proceedings of the 1st Workshop on Representation Learning for NLP*, pp. 148–157.

Neri, F. et al. (2012). "Sentiment analysis on social media". In: *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, pp. 919–926.

Ng, Andrew (2018). *Machine Learning Yearning*. deeplearning.ai.

Ng, A.Y and M.I Jordan (2002). "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". In: *Advances in neural information processing systems*, pp. 841–848.

Ng, Vincent, Sajib Dasgupta, and SM Arifin (2006). "Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews". In: *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, pp. 611–618.

Nguyen, Dat Quoc et al. (2014). "RDRPOSTagger: A ripple down rules-based part-of-speech tagger". In: *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 17–20.

Nguyen, Thien Hai, Kiyoaki Shirai, and Julien Velcin (2015). "Sentiment analysis on social media for stock movement prediction". In: *Expert Systems with Applications* 42.24, pp. 9603–9611.

Ohana, Bruno and Brendan Tierney (2009). "Sentiment classification of reviews using SentiWordNet". In: *9th. IT & T Conference*, p. 13.

Oliveira, Nuno, Paulo Cortez, and Nelson Areal (2013). "Some experiments on modeling stock market behavior using investor sentiment analysis and posting volume from Twitter". In: *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*. ACM, p. 31.

— (2016). "Stock market sentiment lexicon acquisition using microblogging data and statistical measures". In: *Decision Support Systems* 85, pp. 62–73. DOI: 10.1016/j.dss.2016.02.013. URL: https://doi.org/10.1016/j.dss.2016.02.013.

— (2017). "The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices". In: *Expert Syst. Appl.* 73, pp. 125–144. DOI: 10.1016/j.eswa.2016.12.036. URL: https://doi.org/10.1016/j.eswa.2016.12.036.

Ortigosa, Alvaro, José M Martín, and Rosa M Carro (2014). "Sentiment analysis in Facebook and its application to e-learning". In: *Computers in Human Behavior* 31, pp. 527–541.

Pagolu, Venkata Sasank et al. (2016). "Sentiment analysis of Twitter data for predicting stock market movements". In: *2016 international conference on signal processing, communication, power and embedded system (SCOPES)*. IEEE, pp. 1345–1350.

Pal, Aditya and Scott Counts (2011). "Identifying topical authorities in microblogs". In: *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, pp. 45–54.

Pan, Sinno Jialin et al. (2010). "Cross-domain sentiment classification via spectral feature alignment". In: *Proceedings of the 19th international conference on World wide web*. ACM, pp. 751–760.

Pan, S.J. and Q. Yang (2010). "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (2002). "Thumbs up?: sentiment classification using machine learning techniques". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pp. 79–86.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Peters, Matthew E. et al. (2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. Ed. by Marilyn A. Walker, Heng Ji, and Amanda Stent. Association for Computational Linguistics, pp. 2227–2237. ISBN: 978-1-948087-27-8. URL: https://aclanthology.info/papers/N18-1202/n18-1202.

Plank, Barbara, Anders Søgaard, and Yoav Goldberg (2016). "Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss". In: *arXiv preprint arXiv:1604.05529*.

Porter, Martin F (2001). *Snowball: A language for stemming algorithms*.

Pouransari, Hadi and Saman Ghili (2014). *Deep learning for sentiment analysis of movie reviews*. Tech. rep. Stanford University.

Preis, Tobias, Helen Susannah Moat, and H Eugene Stanley (2013). "Quantifying trading behavior in financial markets using Google Trends". In: *Scientific reports* 3, p. 1684.

Pröllochs, Nicolas, Stefan Feuerriegel, and Dirk Neumann (2015). "Enhancing sentiment analysis of financial news by detecting negation scopes". In: *2015 48th Hawaii International Conference on System Sciences*. IEEE, pp. 959–968.

Prusa, J.D. and T.M. Khoshgoftaar (2017). "Improving deep neural network design with new text data representations". In: *Journal of Big Data* 4.1, p. 7.

Qian, Yujie et al. (2017). "A Probabilistic Framework for Location Inference from Social Media". In: *arXiv preprint arXiv:1702.07281*.

Radford, Alec, Rafal Jozefowicz, and Ilya Sutskever (2017). "Learning to generate reviews and discovering sentiment". In: *arXiv preprint arXiv:1704.01444*.

Rahimi, Afshin, Trevor Cohn, and Tim Baldwin (2018). "Semi-supervised User Geolocation via Graph Convolutional Networks". In: *arXiv preprint arXiv:1804.08049*.

Rahimi, Afshin, Trevor Cohn, and Timothy Baldwin (2015). "Twitter user geolocation using a unified text and network prediction model". In: *arXiv preprint arXiv:1506.08259*.

Rahimi, Afshin et al. (2015). "Exploiting text and network context for geolocation of social media users". In: *arXiv preprint arXiv:1506.04803*.

Rao, Tushar and Saket Srivastava (2013). "Modeling movements in oil, gold, forex and market indices using search volume index and twitter sentiments". In: *Proceedings of the 5th Annual ACM Web Science Conference*. ACM, pp. 336–345.

Rodrigues, Erica et al. (2016). "Exploring multiple evidence to infer users' location in Twitter". In: *Neurocomputing* 171, pp. 30–38.

Ruder, Sebastian (2016). "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747*.

Rui, Huaxia, Yizao Liu, and Andrew Whinston (2013). "Whose and what chatter matters? The effect of tweets on movie sales". In: *Decision Support Systems* 55.4, pp. 863–870.

Ryoo, KyoungMin and Sue Moon (2014). "Inferring twitter user locations with 10 km accuracy". In: *Proceedings of the 23rd International Conference on World Wide Web*. ACM, pp. 643–648.

Salvetti, Franco, Stephen Lewis, and Christoph Reichenbach (2004). "Automatic opinion polarity classification of movie". In: *Colorado research in linguistics* 17.1, p. 2.

Sanborn, Adrian and Jacek Skryzalin (2015). "Deep learning for semantic similarity". In: *CS224d: Deep Learning for Natural Language Processing. Stanford, CA, USA: Stanford University*.

Santos, Cicero dos and Maira Gatti (2014). "Deep convolutional neural networks for sentiment analysis of short texts". In: *Proceedings of COLING 2014,*

the 25th International Conference on Computational Linguistics: Technical Papers, pp. 69–78.

Schmid, Helmut (2013). "Probabilistic part-ofispeech tagging using decision trees". In: *New methods in language processing*, p. 154.

Schumaker, Robert P., A. Tomasz Jarmoszko, and Chester S. Labedz (2016). "Predicting wins and spread in the Premier League using a sentiment analysis of twitter". In: *Decision Support Systems* 88, pp. 76–84. DOI: 10.1016/j.dss.2016.05.010. URL: https://doi.org/10.1016/j.dss.2016.05.010.

Schütze, Hinrich and Yoram Singer (1994). "Part-of-speech tagging using a variable memory Markov model". In: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 181–187.

Senin, Pavel (2008). "Dynamic time warping algorithm review". In: *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855, pp. 1–23.

SENTIPOLC (2014). "http://www.di.unito.it/~tutreeb/sentipolc-evalita14/index.html". In: *Evalita*.

Shi, Han-Xiao and Xiao-Jun Li (2011). "A sentiment analysis model for hotel reviews based on supervised learning". In: *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*. Vol. 3. IEEE, pp. 950–954.

Singh, Jyoti Prakash et al. (2017). "Event classification and location prediction from tweets during disasters". In: *Annals of Operations Research*, pp. 1–21.

Song, Yangqiu and Dan Roth (2015). "Unsupervised sparse vector densification for short text similarity". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1275–1280.

Srivastava, Nitish et al. (2014). "Dropout: A simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958.

Tang, Duyu et al. (2015). "User Modeling with Neural Network for Review Rating Prediction." In: *IJCAI*, pp. 1340–1346.

Tashman, L.J. (2000). "Out-of-sample tests of forecasting accuracy: an analysis and review". In: *International Forecasting Journal* 16.4, pp. 437–450.

Trian, Hendro Asmoro (2013). "Exploring Gold Equivalency for forecasting steel prices on Pipeline projects." In: 2.5, pp. 1–22.

Tripathy, Abinash, Ankit Agrawal, and Santanu Kumar Rath (2016). "Classification of sentiment reviews using n-gram machine learning approach". In: *Expert Systems with Applications* 57, pp. 117–126.

Tumasjan, Andranik et al. (2010). "Predicting elections with twitter: What 140 characters reveal about political sentiment." In: *Icwsm* 10.1, pp. 178–185.

Ureña, R. et al. (2019). "A review on trust propagation and opinion dynamics in social networks and group decision making frameworks". In: *Information Sciences* 478, pp. 461–475. DOI: 10.1016/j.ins.2018.11.037. URL: https://doi.org/10.1016/j.ins.2018.11.037.

Utkin, Lev V et al. (2017). "A Siamese Autoencoder Preserving Distances for Anomaly Detection in Multi-robot Systems". In: *2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*. IEEE, pp. 39–44.

Vomfell, Lara, Wolfgang K Härdle, and Stefan Lessmann (2018). "Improving Crime Count Forecasts Using Twitter and Taxi Data". In: *Decision Support Systems* 113, pp. 73–85.

Walczak, Steven and Narciso Cerpa (1999). "Heuristic principles for the design of artificial neural networks". In: *Information & Software Technology* 41.2, pp. 107–117. DOI: 10.1016/S0950-5849(98)00116-5. URL: https://doi.org/10.1016/S0950-5849(98)00116-5.

Wallin, Alexander (2014). "Sentiment analysis of Amazon reviews and perception of product features". PhD thesis. Master's thesis, Lund University.

Wang, N. et al. (2018). "Textual Sentiment of Chinese microblog toward the Stock Market". In: *International Journal of Information Technology & Decision Making*, pp. 1–23.

Wang, Zhe and Xiangyang Xue (2014). "Multi-class support vector machine". In: *Support Vector Machines Applications*. Springer, pp. 23–48.

Wiebe, Janyce (2000). "Learning subjective adjectives from corpora". In: *Aaai/iaai* 20.

Williams, Elizabeth, Jeff Gray, and Brandon Dixon (2017). "Improving geolocation of social media posts". In: *Pervasive and Mobile Computing* 36, pp. 68–79.

Witten, I. et al. (2017). *Data Mining: Practical Machine Learning Tools and Techniques*. 4th. San Francisco, CA: Morgan Kaufmann, San Franscico, USA.

Wood-Doughty, Zach, Nicholas Andrews, and Mark Dredze (2018). "Convolutions Are All You Need (For Classifying Character Sequences)". In: *Proceedings of the 4th Workshop on Noisy User-generated Text, NUT@EMNLP 2018, Brussels, Belgium, November 1, 2018*. Ed. by Wei Xu et al. Association for Computational Linguistics, pp. 208–213. ISBN: 978-1-948087-79-7. URL: https://aclanthology.info/papers/W18-6127/w18-6127.

Wu, Desheng and Yiwen Cui (2018). "Disaster early warning and damage assessment analysis using social media data and geo-location information". In: *Decision Support Systems* 111, pp. 48–59. DOI: 10.1016/j.dss.2018.04.005. URL: https://doi.org/10.1016/j.dss.2018.04.005.

Xing, Frank Z, Erik Cambria, and Roy E Welsch (2018). "Natural language based financial forecasting: a survey". In: *Artificial Intelligence Review* 50.1, pp. 49–73.

Xu, Yang et al. (2007). "A study on mutual information-based feature selection for text categorization". In: *Journal of Computational Information Systems* 3.3, pp. 1007–1012.

Yamaguchi, Yuto et al. (2010). "Turank: Twitter user ranking based on user-tweet graph analysis". In: *International Conference on Web Information Systems Engineering*. Springer, pp. 240–253.

Yan, Xiaohui et al. (2013). "A biterm topic model for short texts". In: *Proceedings of the 22nd international conference on World Wide Web*. ACM, pp. 1445–1456.

Yoshida, Y. et al. (2011). "Transfer learning for multiple-domain sentiment analysis—identifying domain dependent/independent word polarity". In: *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Zhang, Hui, Kiduk Yang, and Elin Jacob (2015). "Topic Level Disambiguation for Weak Queries". In: *arXiv preprint arXiv:1502.04823*.

Zhang, K. et al. (2019). "Interactive Attention Transfer Network for Cross-domain Sentiment Classification". In: *The 33rd AAAI Conference on Artificial Intelligence (AAAI-2019)*. Honolulu, Hawaii, USA.

Ziser, Yftah and Roi Reichart (2017). "Neural Structural Correspondence Learning for Domain Adaptation". In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 400–410.

Zola, Paola and Maurizio Carpita (2016). "FORECASTING THE STEEL PRODUCT PRICES WITH THE ARIMA MODEL." In: *Statistica & Applicazioni* 14.1.

Zola, Paola, Paulo Cortez, and Maurizio Carpita (2019). "Twitter user geolocation using web country noun searches". In: *Decision Support Systems* 120, pp. 50–59.

Zola, Paola and Silvia Golia (2019). "PART OF SPEECH TAGGING FOR BLOG AND MICROBLOGS DATA". In: *Data Science & Social Research 2019 Book of Abstracts*, p. 128.

Zola, Paola et al. (2019). "Social Media Cross-Source and Cross-Domain Sentiment Classification". In: *International Journal of Information Technology & Decision Making*.