Adaptive Sentiment Analysis

Andrius Mudinas

Supervisors: Prof. Mark Levene
Dr. Dell Zhang

The Department of Computer Science and Information Systems
Birkbeck, University of London

This dissertation is submitted for the degree of
Doctor of Philosophy

Birkbeck, University of London December 2018

I would like to dedicate this thesis to my loving parents, wife Egidija and wonderful daughter Olivia . . .

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 100 000 words, including appendices, bibliography, footnotes, tables and equations, and has fewer than 150 figures.

Andrius Mudinas
December 2018

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisors, Prof. Mark Levene and Dr. Dell Zhang, for their continuous support of my Ph.D. study and related research, and for their patience, motivation and immense knowledge. Their guidance has helped me throughout the research for and writing of this thesis. I could not have imagined having better advisors and mentors for my Ph.D. study.

Last but not least, I would like to thank my family, my parents, my wife and wonderful daughter for supporting me throughout the writing of this thesis.

Abstract

Domain dependency is one of the most challenging problems in the field of sentiment analysis. Although most sentiment analysis methods have decent performance if they are targeted at a specific domain and writing style, they do not usually work well with texts that are originated outside of their domain boundaries. Often there is a need to perform sentiment analysis in a domain where no labelled document is available. To address this scenario, researchers have proposed many domain adaptation or unsupervised sentiment analysis methods. However, there is still much room for improvement, as those methods typically cannot match conventional supervised sentiment analysis methods.

In this thesis, we propose a novel *aspect-level sentiment analysis* method that seamlessly integrates lexicon- and learning-based methods. While its performance is comparable to existing approaches, it is less sensitive to domain boundaries and can be applied to cross-domain sentiment analysis when the target domain is similar to the source domain. It also offers more structured and readable results by detecting individual topic aspects and determining their sentiment strengths. Furthermore, we investigate a novel approach to automatically constructing *domain-specific sentiment lexicons* based on distributed word representations (aka word embeddings). The induced lexicon has quality on a par with a handcrafted one and could be used directly in a lexicon-based algorithm for sentiment analysis, but we find that a two-stage bootstrapping strategy could further boost the sentiment classification performance. Compared to existing methods, such an end-to-end *nearly-unsupervised approach to domain-specific sentiment analysis* works out of the box for any target domain, requires no handcrafted lexicon or labelled corpus, and achieves sentiment classification accuracy comparable to that of fully supervised approaches.

Overall, the contribution of this Ph.D. work to the research field of sentiment analysis is twofold. First, we develop a new sentiment analysis system which can — in a nearly-unsupervised manner — adapt to the domain at hand and perform sentiment analysis with minimal loss of performance. Second, we showcase this system in several areas (including finance, politics, and e-business), and investigate particularly the temporal dynamics of sentiment in such contexts.

Table of contents

List of figures

List of tables

Listings

Chapter 1

Introduction

During its natural evolution, our species has developed various traits that have helped to improve our ability to survive in our surrounding environment. The ability to express and understand emotions is one such trait. It has a direct impact on our cognition and behaviour, and plays an important role in our everyday lives. Although there is no clear agreement on how to define emotions, some researchers define them using a set of basic states, such as `anger`, `fear`, `sadness`, `disgust`, `surprise`, `anticipation`, `trust`, and `joy` [61]. Others have expanded this set by including additional emotions such as moral [84, 225, 170] or even sensory perception [85]. In this thesis, we use the term "sentiment" to describe a variety of affective states [186, 244], and we draw a distinction between sentiment *attitudes* and sentiment *emotions*, following the typology proposed by Scherer [216]. By attitude, we mean the narrow sense of sentiment (as in most research papers on sentiment analysis) — whether people are `positive` or `negative` about something. By emotion, we mean the eight "basic emotions" in four opposing pairs — `joy-sadness`, `anger-fear`, `trust-disgust`, and `anticipation-surprise`, as identified by Plutchik [189].

Emotions and sentiment are present in almost all information sources. Every day, a large number of opinionated documents are published on the Internet: people post product reviews, express political views and share their feelings on social networks. Thus, naturally, this sentiment information not only affects our behaviour but also has a significant impact on our decision-making process. Prior to making a purchase or visiting a local restaurant, most individuals check online reviews and read customer feedback. On social media platforms such as Twitter and Facebook, people also share attitudes on personally important topics, day traders provide their trading stances and politicians pitch their messages to voters. More recently, Twitter attracted a lot of negative attention because of attempts by malicious actors to manipulate public opinion and political voting [12]. Therefore, the ability to extract sentiments from various information sources can not only provide invaluable information about people's views

on various topics, but also help to predict customer behaviour, stock market movements or even election results.

Sentiment analysis is a domain-specific problem (i.e. all approaches perform well only if targeted at a specific domain, and they suffer significant performance loss once domain boundaries are crossed [200]). Bag-of-words learning approaches [182, 179] are among the most susceptible to the domain dependency problem. Conversely, numerous studies have suggested that a typical lexicon-based system [58, 228, 227] has lower domain sensitivity, may be easier to maintain by a human user, and has an output that is self-explanatory, yet it cannot match the accuracy of bag-of-words supervised learning.

Thus, considerable effort has been invested in finding an automated way of domain adaptation by designing unsupervised sentiment-detection systems [60], various knowledge transfer methods [65, 27, 25] and systems less prone to cross domain boundaries [26]. Unfortunately, most of these approaches suffer from various limitations. Supervised domain adaptation requires labelled domain-specific training data, and the collection of such data is an expensive and time-consuming task. Unsupervised approaches typically have inferior performance and cannot match conventional supervised sentiment analysis methods.

Moreover, most domain-adaptation methods are designed with specific domain boundaries and constraints in mind, such as targeting different topics in a social media [126] or various categories of product reviews [76], thus performing domain adaptation between similar domains (also known as *near* domains). Attempts to adapt *distant* domains [158, 184] suffer from a significant drop in efficiency. Another important aspect is that researchers typically work with clearly defined domain boundaries [126, 76], having their datasets split into distinctive and separable categories. However, many sentiment sources, such as social media, are noisy, having a mix of *cross-style* or *near-domain* documents. In addition to the mentioned limitations, in this thesis we will also highlight that domain adaptation does not eliminate the domain dependency problem. Hence, the thesis attempts to address some of these issues.

1.1 Motivation and Objectives

In order to address the problems described in the section above, this thesis introduces a new approach to domain adaptation and presents our exploration towards answering several related research questions.

First, is it possible to overcome the above lexicon and bag-of-words learning limitations and reduce sensitivity to crossing domain boundaries? Second, can lexicon-based systems improve their performance by learning a domain-specific lexicon? Third, can

an unsupervised domain-adaptation and sentiment-analysis method close the gap to the performance of supervised methods?

In order to answer the given research questions, the thesis will explore the possibility of combining lexicon- and machine-learning techniques for opinion analysis. In addition, it will investigate methods that are less sensitive to crossing domain boundaries, with lower topic and style dependency compared to a pure bag-of-words machine-learning implementation. It will also evaluate the advantage of inducing domain-specific sentiment lexicons, as well as evidence that different domains have different sentiment vector spaces. Moreover, it will explore the possibility of building domain-specific sentiment classifiers with unlabelled documents only, which can achieve sentiment classification accuracy comparable to that of fully supervised approaches.

Finally, to validate the proposed models, we will explore practical applications and evaluate domain adaptation in a series of case studies using a diverse range of domains.

We evaluate the realisation of our objectives and the contributions made by this thesis to the subject of sentiment analysis and domain adaptation in Chapter 7.

1.2 Contributions

As part of the research presented in this thesis, we have made several contributions to adaptive sentiment analysis and explored various domain-adaptation and cross-domain sentiment analysis scenarios. Exploration of these questions contributes to our understanding of adaptive sentiment analysis, which we define as a novel set of sentiment analysis methods that can adapt to any domain at hand, are less sensitive to crossing domain boundaries, and can be applied in both supervised and semi-supervised modes. These features make our approach suitable for a wide range of practical sentiment analysis applications. Specifically, our main four contributions are listed below.

First, to overcome the limitations of lexicon- and bag-of-words learning, we have developed a **novel sentiment analysis method, *pSenti***, which is less sensitive to crossing domain boundaries and has similar performance to the pure learning-based methods. Here, we have shown that the sentiment analysis results produced by our *hybrid* approach are favourable compared to the lexicon-only and learning-only baselines.

Second, we created a **novel lexicon-induction method** and integrated it into the previously built *pSenti* sentiment-detection system. Using our novel approach, we have demonstrated that a high-quality domain-specific sentiment lexicon can be induced from word embeddings [160, 185] of that domain with just a few seed words. We have also confirmed the advantage of generating domain-specific sentiment lexicons and provided evidence that different domains have different sentiment vector spaces. The

induced lexicon could be applied directly in a lexicon-based algorithm for sentiment analysis and operate as an nearly-unsupervised sentiment method. It can effortlessly adapt to new domains, give high accuracy in a sentiment analysis task, and offer similar rich features to other lexicon-based sentiment analysis approaches.

Third, we have **proposed a system which can, in an nearly-unsupervised manner, adapt to the domain at hand** and perform near-cross-domain sentiment analysis without additional adaptation and with minimum loss of performance. Our results confirm our deep-learning-based [94] method's superiority over more traditional SVM-based approaches [182, 179] in the domain-adaptation task.

Fourth, we have **evaluated our models** in several areas: (i) we have investigated the potential of using sentiment *attitudes* (positive vs. negative) and also sentiment *emotions* extracted from financial news or tweets to help predict stock price movements, (ii) we have considered political-sentiment analysis and *stance detection*, (iii) we have analysed product reviews rating seasonality and trend analysis, (iv) we have considered multidimensional and temporal model integration with a hybrid sentiment analysis method.

The source code for our implemented systems and the datasets have been made available to the research community [1] [2] [3] [4].

1.3 Publications

The following publications by the author are related to this thesis:

[1] A. Mudinas, D. Zhang, and M. Levene. "Combining Lexicon and Learning Based Approaches for Concept-level Sentiment Analysis". In: *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*. WISDOM '12. Beijing, China: ACM Press, 2012. ISBN: 9781450315432.

[2] A. Mudinas, D. Zhang, and M. Levene. "Bootstrap Domain-Specific Sentiment Classifiers from Unlabeled Corpora". In: *Transactions of the Association for Computational Linguistics* 6 (2018), pp. 269–285.

[3] A. Mudinas, D. Zhang, and M. Levene. "Market Trend Prediction using Sentiment Analysis: Lessons Learned and Paths Forward". In: *Proceedings of the 7th KDD Workshop on Issues of Sentiment Discovery and Opinion Mining*. WISDOM '18. London, UK: ACM Press, 2018.

[1] http://www.dcs.bbk.ac.uk/~andrius/psenti/
[2] https://github.com/AndMu/Wikiled.Sentiment
[3] https://github.com/AndMu/Unsupervised-Domain-Specific-Sentiment-Analysis
[4] https://github.com/AndMu/Market-Wisdom

Chapter 2

Background

Sentiment analysis, also known as opinion mining, is the computational study of people's underlying feelings, opinions, attitudes, appraisals, and emotions towards entities, events, topics, individuals, issues, and their attributes [155]. It is a mature research field, whose origins can be traced back to the 1960s. Stone et al. published a pioneering resource, the *General Inquirer* lexicon dictionary, which, fifty years later, is still relevant and maintained [221]. However, significant progress in sentiment analysis was not achieved until the 1990s after the technological revolution of the Internet made available enormous volumes of opinionated text. In 1990 Wiebe [245] published work which established many of the ground rules in the sentiment analysis field. Several years later, Hatzivassiloglou and McKeown [89] presented a method of predicting the semantic orientation of adjective words and phrases with a high accuracy of 82%. More research followed [244, 90] and helped to establish opinion mining and sentiment analysis as a research area in its own right. In the 2000s, new sentiment analysis algorithms started to emerge. Turney [232] introduced one of the first algorithms for document-level sentiment analysis, which achieved an average accuracy of 74% for product reviews; but on movie reviews, the performance was much worse, only 66%. In his design, rather than focusing on isolated adjectives, Turney proposed to detect sentiment based on selected phrases chosen via several part-of-speech (POS) patterns [232]. The emergence of social media, the much-increased availability of subjective and opinionated text on the Web, and advances in machine learning and natural language processing (NLP) techniques started another tide of sentiment analysis publications. As Mäntylä et al. [150] in their survey found, 99% of sentiment research papers have been published after 2004.

Nowadays, opinion mining is still attracting interest from many researchers and covers a broad range of research topics and techniques. Over the past few years, deep-learning algorithms have made impressive advances, with the introduction of new NLP techniques such as word embedding [160, 185], creating an opportunity to develop

novel sentiment analysis methods and investigate sentiment analysis from a different perspective. The topic has become so popular that various yearly competitions are being organised, one of the most famous of which is SemEval [222, 252, 171, 206, 205, 172, 204], as well as competitions organised by Kaggle[1], ESWC [209] and others [15, 162]. Some approaches are so well tweaked, that in one of Kaggle's events, *"Bag of Words Meets Bags of Popcorn"*, one of the participants managed to achieve the impressive score of 0.99259 for the area under a receiver operating characteristic (ROC) curve (AUC). The area under the ROC curve (AUC) can be used to describe the quality of a classification algorithm, which we will also briefly use in later chapters. The ROC curve is a two-dimensional plot in which the false-positive rate is plotted on the X-axis, and the true-positive rate is plotted on the Y-axis. Calculating the area under the curve is one way to present this in a single value, which lies between 0.5 and 1. If classifier A has a higher AUC than classifier B, then it is considered the better.

The rest of this chapter is organised as follows. In Section 2.1 we describe the main components required for opinion mining and sentiment analysis, describe data collection methods and sentiment sources, and introduce text processing and transformation in the context of the adaptive sentiment analysis story. In Section 2.2 we cover the main sentiment-detection approaches, starting with the lexicon-based approach and ending with deep learning. We also discuss word embedding, its importance for domain adaptation and its integration into existing sentiment analysis approaches, and review both supervised and unsupervised sentiment analysis methods. In Section 2.3 we discuss different domain-adaptation methods, the challenges they face and discuss their importance. Sections 2.4 and 2.5 briefly cover sentiment dimensionality and temporal sentiment analysis. In Section 2.6 we examine areas of applied sentiment analysis and discuss how they are related to the thesis. This chapter also discusses aspect extraction, opinion targets and many other related topics.

2.1 The Components of Sentiment Analysis

Sentiment analysis can be partitioned into components in several ways, depending on scope, approach and learning algorithm selection. We can group components into several major categories: *data retrieval* and *pre-processing*, *feature extraction*, *learning method* selection and *domain adaptation*. We note that sentiment information is most frequently stored in a text format. Therefore, it must be retrieved, normalised and converted into a machine format, typically numerical feature vectors. Many of the components are shared with other Information Retrieval (IR) methods. They are well defined and were established long before the topic of sentiment analysis became popular.

[1] http://www.kaggle.com

There is also a broad set of various NLP techniques and approaches involved, and their choice depends on the strategy taken and varies from one method to another.

Methods based on machine learning use varied methods to perform **feature extraction**, as well as additional pre-processing. To extract features, classic machine learning approaches typically use a bag-of-words model [182, 179], in which every single word is a feature, and it also is not uncommon to enrich them with additional steps. We will cover the bag-of-words model in more depth later. In this thesis, we use a variety of strategies to generate features. For example, in Chapter 3, we will limit features to well-known and potential sentiment words. In Chapter 4, to induce a sentiment lexicon, we will use multidimensional word vectors, and, finally, in Chapter 5 we will employ a bag-of-words with multidimensional vectors strategy. A similar feature selection process can be applied to other tasks such as sentiment summarisation and aggregation, aspect extraction, identification of an opinion holder, sentiment target and discussed topics [14, 39].

Component choice also depends on the selection of sentiment analysis **learning method** as well as other factors, such as sentiment granularity level selection or the ability to detect neutral-objective information. **Domain adaptation**, cross-domain and multi-language sentiment analysis may require additional components and techniques (e.g. extracting sentiment analysis from long, professionally edited text is a very different task compared to extracting sentiment from microblogs). Short text extracted from Twitter typically requires a unique approach with additional components to normalise and process irregular language [1]. As we mentioned previously, each of these topics can be a separate research topic, and we will cover them in more depth in later sections.

2.1.1 Data collection and sources

The rise in popularity of research into sentiment analysis can be directly correlated with the information technology revolution and the increased availability of subjective text data sets. Prior to the widespread use of the World Wide Web and social media, there was little need for methods for sentiment mining and collection. With the extensive availability of third-party review sites such as CNET[2], IMDB[3], and Rotten Tomatoes[4] (see Figure 2.1), or, more importantly, the increased use of social networks such as Twitter, Facebook and Reddit, the popularity of sentiment analysis gained significant traction. In a sense, opinion mining opened the possibility of attaining a sneak peek into human thoughts on a broad range of topics.

[2]https://download.cnet.com
[3]https://www.imdb.com
[4]https://www.rottentomatoes.com

Fig. 2.1: Rotten Tomatoes review

Most early sentiment datasets were acquired by scraping Web pages and normalising retrieved data [208], a procedure that is well covered in the literature [122] and one for which there is a wide selection of open-source tools available. A basic Web scraper can be implemented in a couple lines of Python code. However, in many cases, implementations may be more complicated, as many content providers are continuously trying to prevent scrapers from working [51] and even taking legal action against them [203]. This can also be a challenging problem due to the complexity of page structure. In some of the experiments, we will make use of a Web scraping; for instance, in Chapter 3 we will use product reviews scraped from CNET, and in Chapter 6, datasets collected from the FT and Reuters websites.

Most social networks, including Twitter and Facebook, provide an API to access their data. In this thesis, we will use the Twitter API to download tweets and user information. In Chapter 5 we will use messages collected using the Twitter API to build embeddings and improve semi-supervised domain adaptation. In Chapter 6, we will collect and use two different Twitter datasets. More specifically, to obtain relevant sentiment signals in the financial domain, we will assemble an extensive collection of financial tweets. We will also collect a vast dataset containing 142 million messages from 7.6 million unique users, in which they express their political stance towards Donald Trump. Not all functionality is available via the API, thus to collect additional information from Twitter, such as user profile photos, we will also employ Web scraping.

To annotate datasets and gain quick access to training and testing data, researchers in a variety of disciplines use crowdsourcing platforms such as Amazon Mechanical Turk (MTurk)[5], CrowdFlower[6] and SurveyMonkey[7] [237]. In Chapter 5 we will use a Twitter dataset annotated using CrowdFlower. In Chapter 6 we will use MTurk to annotate a 2016 US presidential election Twitter dataset. That will allow us to evaluate our proposed sentiment analysis method efficiency in the political-sentiment domain and answer a research question on the demographics of Trump's supporters.

[5]https://www.mturk.com
[6]https://www.crowdflower.com
[7]https://www.surveymonkey.co.uk

Another common strategy for evaluating the proposed sentiment analysis methods is to use standard benchmark datasets. In papers analysing Amazon customer reviews, the dataset collected by McAuley and Leskovec [154] is the most commonly employed. In papers looking into movie reviews, the datasets collected by Maas et al. [144] and Pang and Lee [179] are often used. The Twitter domain is unique, can cover many topics, and thus has many datasets, some of the most important of which are from the SemEval competition [204]. Hence in this thesis, we also use these datasets (Amazon, IMDB, Twitter), to validate our proposed methods and make a direct comparison with other researchers.

There are also plenty of commercial datasets. Big players in the financial news domain and trading, such as Thomson Reuters and Bloomberg, have their own commercial solutions. Many companies, such as Amazon, Twitter and Facebook, also provide commercial access to their extensive datasets, including to historical data.

2.1.2 Text preprocessing and transformation

Retrieved information must be processed and transformed into a format suitable for computer processing. The procedure was established as part of the evolution of IR methods and is similar across many of the sentiment analysis approaches [132]. It typically starts with text pre-processing. Pre-processing steps such as tokenisation, stopword removal and morphological normalisation were introduced and developed in the late 1960s [211]. Each of these steps has a number alternative implementations. For example, in its simplest form, **text tokenisation** can be done using a whitespace tokeniser, which performs down-casing and splitting of the text into any sequence of whitespace, tab or newline characters. The sample sentence, *"I have visited many countries"* would be split into the tokens: *"i", "have", "visited", "many" and "countries"*.

For more sophisticated tokenisation, a wide selection of open-source NLP software packages is available. Products such as NLTK [19], OpenNlp [7] and CoreNlp [149] provide not only tokenisation options but also additional functionality, such as sentence splitting, part-of-speech (POS) annotations, morphological analysis, Named Entity Recognition (NER), syntactic parsing and co-reference resolution, and can even detect temporal definitions [40]. In this thesis, we make use of various third-party NLP software packages and libraries. The core of *pSenti* from Chapter 3 is a lexicon-based system, so it shares many common components with NLP processing techniques. It supports two different NLP frameworks, CoreNlp and OpenNlp, and uses them in tokenisation, POS and entity tagging. In later chapters, we use NLTK and our custom regular-expression-based tokeniser for the Twitter domain.

In Chapters 3 and 6, we also extract **temporal orientations** from a text. A temporal orientation is calculated using two different methods: using the *SuTime* temporal tagger

[40], and using the tense of a sentence. *SuTime* is a rule-based tagger built on regular-expression patterns to recognise and normalise temporal expressions in English text in the form of TIMEX3 tags. TIMEX3 is part of the TimeML annotation language [193] for marking up events, times and their temporal relationships in documents (see Table 2.1 for some examples).

Type	Text	Tag
Date	October of 1963	<TIMEX3 tid="t1" value="1963-10" type="DATE"> October of 1963</TIMEX3>
Duration	fifty-six years	<TIMEX3 tid="t1" type="DURATION" value="P56Y"> fifty-six years</TIMEX3>
Set	Every third Sunday	<TIMEX3 tid="t1" value="XXXX-WXX-7" type="SET" quant="every third" periodicity="P3W"> Every third Sunday</TIMEX3>
Time	5:05 in the afternoon	<TIMEX3 tid="t1" value="2011-08-01T17:05:00" type="TIME">5:05 in the afternoon</TIMEX3>
Date - written out year	winter of nineteen ninety-four	<TIMEX3 tid="t1" value="1994-WI" type="DATE">winter of nineteen ninety-four</TIMEX3>
Duration Range	two to three months	<TIMEX3 tid="t1" altvalue="P2M/P3M" type="DURATION">two to three months</TIMEX3>
Holiday	last Christmas	<TIMEX3 tid="t1" type="DATE" altvalue="20101225">last Christmas</TIMEX3>
Ambiguous words	The spring water was cool and refreshing	The <TIMEX3 tid="t1" value="2011-SP" type="DATE">spring</TIMEX3> water was cool and refreshing

Table 2.1. SuTime tagging examples

Not all information extracted from a text is useful. In many IR approaches it is considered that stopwords such as "an", "and", "by", "for" and "the" do not carry any valuable information; they merely represent noise which requires additional unproductive processing and decreases IR efficiency [215]. Thus, it is quite common in sentiment analysis to remove stopwords. That can be done using a simple stopword list or by exploiting POS information. However, as Manning et al. [148] noted, in recent years, there has been a trend in IR to either keep stopwords in place or to reduce the stop list to a minimum, as the overhead is not considerable. Eliminating stopwords can also reduce system performance, as it frequently fails to recognise acronyms such as *"IT engineer"* and is thus prone to false positives. In most of our experiments, we remove stopwords, but not in all. We found that our deep-learning models performed better with stopwords in texts.

Many sentiment analysis methods also use morphological normalisation [132], converting words to the singular or applying various stemming procedures. Similar to the situation with stopwords, we found that stemming improves *pSenti* performance. However, it does reduce the performance of deep-learning-based models.

Twitter and other microblogs typically require a unique approach. They use irregular language, emoticons, Internet slang words and abbreviations, and are full of misspelled words. As Hogenboom et al. [95] identified, graphical emoticon recognition in these

domains significantly improves sentiment classification accuracy. Microblog authors also use misspellings and word lengthening to express their feelings [34]. Short text snippets are also difficult to process using NLP toolkits and frequently require additional pre-processing [169]. To address the issue and domain specifics mentioned above, we incorporate hashtag and emoticon recognition, use a custom-built tokeniser and include word-lengthening recognition.

NLP methods can be useful not only for the initial text pre-processing but also in other applications, such as generating additional machine-learning features or performing many sentiment analysis subtasks. One of the most widely used is so-called POS information. The Penn Treebank, originally introduced by Marcus et al. [151], opened up new research capabilities in syntactic sentence analysis. As we will demonstrate in later chapters, POS information may be employed in stopword removal, aspect detection and candidate sentiment lexicon generation, and can help with the word sense disambiguation problem and provide the ability to understand the surrounding context better. POS information can also be included as a machine-learning feature [165], which can improve sentiment-detection performance, although some researchers have found that is not that useful [182, 77]. Among other NLP methods, it is essential to mention the lexical category and NER information resolution. This information can help with an aspect, author, attribute or sentiment target resolution, which we will cover in more depth in later chapters.

Machine-learning-based approaches typically require an additional vectorisation step, in which selected and processed word tokens are transformed into numerical vectors. Using the bag-of-words model, a document is represented as an unordered list of unigrams also known as terms. Vectors are typically generated using a vocabulary scheme with a high-dimensional feature space, having a tens-of-thousands-element-long vector for each document. Each dimension can be represented by a binary value (with 1 indicating term presence) or using other weight calculation approaches. One of the most common is *term frequency* or *term frequency–inverse document frequency* (*tf-idf*), which was originally introduced by Salton and Buckley [212]. *Tf-idf* consists of two parts: the term frequency multiplied by the inverse document frequency. Given a document collection D, a word w, and an individual document $d \in D$, we calculate a word weight w_d using Equation (2.1), where $tf_{w,D}$ is the frequency of w appearance in d, $|D|$ is the size of the corpus, and $f_{w,D}$ equals the number of documents in which w appears in D [212]. One of the main advantages of *tf-idf* weighting is that it reduces the impact of words that occur very frequently and emphasises features that occur in a small fraction of in a given corpus. Another, similar approach is one-hot encoding (OHE), which is commonly used in deep-learning models. Using OHE, each word is represented by a vector of zeros except for the element at the index representing the

corresponding word from a vocabulary scheme. Vectorised data also typically requires additional data normalisation and standardisation.

$$w_d = tf_{w,d} * \log \frac{|D|}{f_{w,D}} \tag{2.1}$$

2.1.3 Sentiment granularity

In its simplest form, sentiment detection can be defined as a procedure of **binary** document classification into positive and negative classes [232, 182]. Nowadays, using off-the-shelf machine-learning libraries and a few lines of Python code, it is possible to create a simple linear bag-of-words support-vector machine (SVM) sentiment classifier which can achieve as high as 90% accuracy in the binary sentiment classification task. However, such a **document-level** approach has many limitations. Most documents are not monolithic items with a single opinion: they can contain areas with positive, negative and factual content, which is also known as sentiment-neutral information. A sentiment author also often expresses multiple opinions and can have different opinions about various aspects of the discussed topic. Therefore, by just detecting that a given document is positive or negative, we would lose a lot of information about which aspects (e.g., product features) the author liked or disliked, and to what degree. To address that, Hatzivassiloglou and Wiebe [90] proposed **sentence-level** sentiment analysis, Hu and Liu introduced the two-step **aspect-level** method [98], which was later improved by Popescu and Etzioni [191]. Using the two-step method, the sentiment analysis task can be divided into two separate subtasks: aspect identification and sentiment-strength measurement.

According to the output, sentiment analysis can be divided into three families:

- **Binary** classification into positive and negative classes. As we have already mentioned, many early machine-learning-based sentiment analysis approaches treated sentiment analysis as a *document-level* binary classification problem [232, 182]. Another flavour of binary classification in sentiment analysis would be classification into subjective and objective classes [246]. This type of classification is frequently applied to the *sentence* or *paragraph* level and can be employed in multi-stage sentiment analysis to eliminate subjective text [260, 202]. Binary classification can also be utilised to answer sentiment-related questions. For example, market sentiment analysis can generate **BUY** and **SELL** signals [236]. In Chapter 6 we will investigate the financial domain and sentiment analysis in financial news. We will use mood and sentiment to generate binary **BUY** and **SELL** signals and will demonstrate that, in some cases, the model using sentiment information outperforms the baseline method. There are many other examples of binary sentiment classification applications (e.g. Tumasjan et al. [231]

demonstrated a model to detect political-sentiment application). In later chapters, we also discuss the *binary* sentiment model to find Trump presidential election supporters.

- **Three-class** classification into positive, negative and neutral classes. One common way to handle neutral sentiment is to treat the set of neutral documents as a separate "neutral" class, which is the method advocated by Koppel and Schler [116] and investigated by many others [98, 113, 66]. This approach is more representative of how sentiment can be expressed, as it considers the fact that sentiment is not a binary value. In a typical opinionated document, most common areas are not positive or negative, but in fact neutral or factual blocks. Thus, a simple binary classifier is not able to differentiate between factual and opinionated information.

- **Multi-class** classification, regression and ordinal regression. All methods in this class attempt to measure the sentiment intensity level on a more granular scale which addresses the fact that many sentiment information sources have a broad sentiment scale. Amazon product reviews (see Figure 2.2) have an ordinal five-star rating scale, and IMDB and Rotten Tomatoes (see Figure 2.1) use a ten-star scale. Thus, to reflect this, some researchers have tried to consider the problem as a multi-class classification problem. Koppel and Schler [116] investigated the use of various stacks, and others, such as Almeida et al. [3], designed a multi-class classifier. Pang and Lee [180] investigated various methods, including regression with an ordinal rating score with a possibility to measure sentiment strength, where a value could be measured on a continuous scale from -1 to $+1$, or from 1 to 5, or from 0 to 100. Goldberg and Zhu [79] also investigated sentiment rating prediction as both an ordinal regression problem and as a metric regression problem.

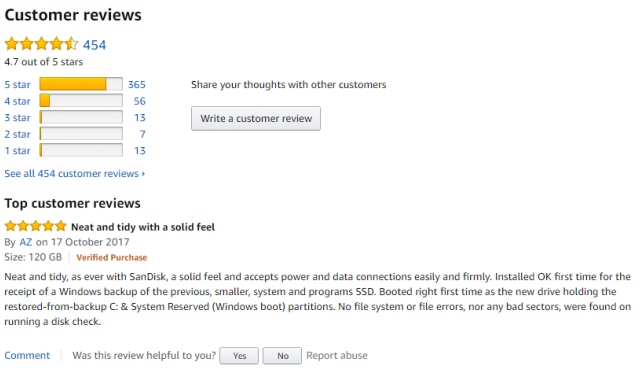


Fig. 2.2: Amazon review

In general, the one-versus-one (OVO) strategy is regarded as one of the most effective SVM strategies available [70] for multi-class sentiment analysis. In later chapters, with the pseudo-labelled training examples of three classes (-1: negative, 0: neutral, and $+1$: positive), we tried both standard multi-class classification [97] and ordinal classification [68]. However, neither of them could deliver a decent performance. After carefully inspecting the classification results, we realised that it is very difficult to obtain a set of representative training examples with good coverage for the neutral class. This is because the neutral class is not homogeneous: a document could be neutral because it is equally positive and negative, or because it does not hold any sentiment. In practice, the latter case is more often seen than the former, which implies that the absence of sentiment word features more often defines the neutral class rather than their presence, which would be problematic to most supervised learning algorithms. What we have discovered is that the simple method of identifying neutral documents from the binary sentiment classifier's decision boundary works surprisingly well if appropriate thresholds are found. Specifically, we take the probabilistic outputs of a binary sentiment classifier, and then put all the documents whose probability of being positive is close to neither 0 nor 1 but in the middle range into the neutral class.

In Chapter 5, we will cover the *probability calibration* application in more depth.

2.1.4 Opinion holder and sentiment target

Sentiment analysis typically requires identification of both the opinion holder and the target. This information can be used in sentiment summarisation and aggregation to provide a better explanation of sentiment flow and justification. In their study, Choi et al. [45] also found that the incorporation of semantic roles in sentiment improves the performance of the opinion recognition task. In product or service reviews, an opinion holder is typically a review author, and in other sources they can be explicitly mentioned [18]. The presence of an opinion holder in a sentence or text area is also a good indicator of an expressed sentiment, and this may be employed for the identification of subjective areas [32, 249].

In his book, Liu [134] identified two main sentiment types according to their target, *direct sentiment*, which is targeted at an object or its feature, and *comparative sentiment*, where two or more objects/features are compared to each other. Frequently, both are present in the opinionated text, as illustrated by the following product review snippet.

These phones are exceptionally portable due to the collapsible head bridge. They stand up to everyday use and keep sounding good, even after I sweat on the ear piece. I have not found a better pair after a long search. My only wish would be a lower price and better availability

In this example, the author expressed a direct opinion about an item and, at the same time, compared it to other products. In the case of direct opinion, Liu [134] defined it as a quintuple $(o_j, f_{jk}, s_{ijkl}, h_i, t_l)$, where o_j is an object, f_{jk} is an aspect of the object o_j, s_{ijkl} is the sentiment polarity or orientation, h_i is the opinion holder, and t_l is the time when the opinion is expressed by h_i. This definition can be further extended with t_o being a temporal sentiment dimension, which defines a temporal opinion target (i.e. expressed with the past, current or future in mind). For example, *"I hope the next version will be perfect"*, has a positive sentiment pointing to the future, and *"Last year, I lost my phone"* has negative sentiment targeting the past. Each of these dimensions is an important sentiment analysis component and can be the target of a separate research topic.

A sentiment target, which is also referred to as an aspect in the literature, also plays a key role in our thesis and is covered in more depth below and in Chapter 3. Both opinion-holder and sentiment target identification are also tightly related to the problem of *stance detection* (SD) [218]. As defined by Mohammad et al. [167], a typical sentiment-detection system classifies a text into positive, negative or neutral categories, while in SD the task is to detect a text that is favourable or unfavourable to a specific given target. Most of the existing research on SD is focused on the area of politics [119, 120, 226]. In Chapter 6, in a political-sentiment analysis use case, we briefly investigate SD, opinion holders and their demographics. More specifically, we collected and annotated 6200 user profiles, identified their stances towards Donald Trump, and examined whether Trump supporters prefer whiter areas to move to than Trump opponents.

2.1.5 Aspect detection and aspect-level sentiment classification

Aspect and view extraction can play multiple roles in sentiment analysis. The most common application of aspect extraction is in aspect-level sentiment classification, where the aim is to identify the sentiment polarity of discussed targets [181]. They can also enrich a sentiment-detection process [115], help in domain adaptation [115], or be employed to create domain-specific signatures. It is one of the best ways to present sentiment information [14, 39], as it can provide a quick overview of discussed product features, as well as visualise what was good and what was not so good in each of them. Aggregated by aspects, a sentiment analysis result can be consumed by other automatic processes such as Customer Relation Model (CRM) systems [258], and by doing so, allow companies to take advantage of collected reviews and customer feedback, and to improve their sales [43]. It can also help in a product comparison task [98], which can be presented to both consumers and producers. It has thus attracted the attention of many researchers.

Similar to sentiment classification, the aspect-level sentiment-detection task can be divided into lexicon- [98, 58] and learning-based [105, 264] approaches. A lexicon-based approach typically uses a so-called Separate Aspect Sentiment (SAS) model, which consists of two separate tasks: the aspect detection, and its sentiment pair prediction [152]. In this approach, aspects are extracted independently and later have their sentiment value calculated, aggregating contextual sentiment polarity within a predefined token window, sentence or paragraph. A learning-based approach is frequently based on a Joint Multi-Aspect Sentiment (JMAS) model. In a JMAS model, aspect sentiment is predicted in pairs, where the aspect is associated with the sentiment, thus jointly predicting which pairs can be found in the document [152].

Extracting aspects from a "high-quality" text is usually a relatively straightforward procedure and, in many cases, can be solved using an unsupervised task [35]. As Hu and Liu [98] have found in their research, the aspect extraction task can be implemented by selecting frequent nouns and noun phrases. However, customer reviews and microblog messages are usually short, informal, and sometimes even ungrammatical (e.g., consisting of incomplete sentences), which makes this task more challenging. To overcome this problem, Hu and Liu [99] proposed using Labelled Sequential Rules (LSR), where rules are a special kind of sequential pattern. As in a sentiment lexicon generation task, an aspect's lexicon can also be generated by exploring synonyms [38]. Similar aspects can also be discovered using word-embedding solutions.

Machine-learning-based approaches traditionally employ topic modelling and clustering [137, 143, 156, 35]. As an example, Kohail [115], in their work, demonstrated an aspect extraction method based on Latent Dirichlet Allocation (LDA) topic modelling. They created an unsupervised framework for extracting dominant topics from multi-domain document collections and demonstrated that aspects play an influential role in the domain detection task. The rise in popularity of deep-learning-based methods opened a way for new, neural-network-based methods [152, 121, 250] and various solutions based on word embedding [257]. Jebbara and Cimiano [104] and Marx and Yellin-Flaherty [152] demonstrated that neural networks outperform traditional models by a significant margin. Such a design also allows seamless integration of extracted aspect information into neural networks based on sentiment-analysis models.

In the next chapter, we also discuss another motivation to emphasise aspect/view extraction. We will demonstrate that exclusion of the domain-specific aspect words from the machine-learning step will reduce the dependence on the domain topic, writing style or time period. We will use an aspect extraction method similar to those of Hu and Liu [98], such as generating a list of candidate aspects by including frequent nouns and noun phrases. The main distinctive difference in our approach is that we use additional steps to enhance aspect detection, which we will cover in more depth later.

2.1.6 Subjectivity and objectivity

Many researchers frequently focus only on binary positive/negative sentiment classification and ignore neutral sentiment. However, as Koppel and Schler [116] identified in their research, neutral sentiment detection is a critical part of the sentiment-detection process. Neutral sentiment can typically be defined as objective or factual information which does not express any subjective opinion about a discussed topic. It can be a simple statement of facts, a physical description of an item or any other objective information. In some information sources, objective information can be dominant and even use well-known strong sentiment words to express factual information.

To demonstrate the importance of the ability to distinguish between subjective and objective information, we can take a compelling example from a movie review, in which the author briefly describes the plot: *'The film opens with a flashback, in which Derek brutally kills two men vandalising his car'*. This sentence does not carry any sentiment information; the author simply describes the film's plot. However, what makes this example especially compelling is that it contains so-called strong sentiment words, *kills* and *vandalising*, and most lexicon-based systems would flag this sentence as a strong negative opinion. Therefore, the ability to distinguish between objective and subjective-text blocks is essential for most sentiment-detection systems. This example also illustrates that objective information can be domain dependent. Any human reader can understand that it is impossible to kill or hurt somebody in a movie review, and we tend to ignore such sentiment words depending on the information source and context.

Another excellent example of why the ability to distinguish between subjective and objective information is essential can be found in Chapter 6. More specifically, in the political-sentiment analysis use case, all our sentiment-analysis models performed poorly with the non-related messages class. However, in this case, *non-related messages class* has a broader definition and can include sentiment messages targeted at a different (irrelevant) topic.

One of the approaches of how to handle subjective text is to use the so-called two-step approach [260, 202]. In the first step, we classify sentences into subjective/objective classes. Then, in the second step, we classify subjective sentences into positive/negative. For the implementation of the first step, there are many different options, from supervised to many different variations of unsupervised methods. As an example, Yu and Hatzivassiloglou [260] proposed a supervised method for subjectivity detection; Riloff and Wiebe [202] demonstrated a semi-supervised expressions learning-based technique; and, more recently, Ortega et al. [174] proposed an unsupervised approach.

Another conventional approach is to incorporate subjectivity as a third class and consider the problem as a three-class (negative/positive/neutral) classification problem. In their study, Koppel and Schler [116] compared various three-class sentiment classi-

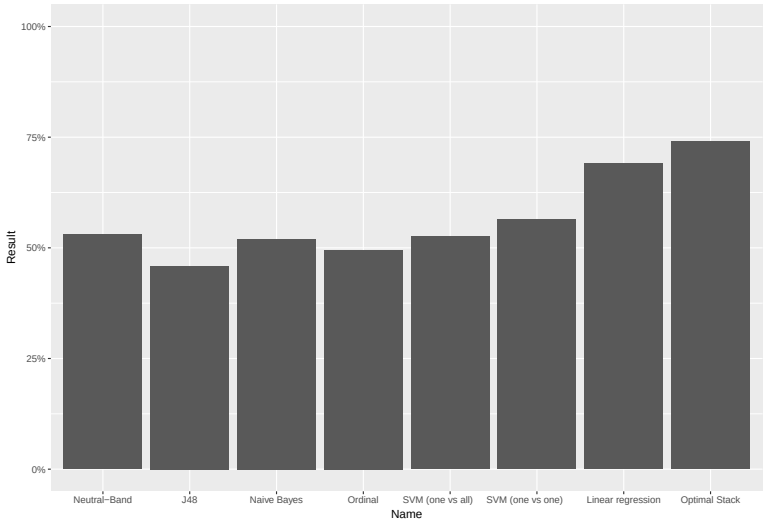


Fig. 2.3: Three-class classification performance using a variety of learning methods [116]

fication methods and found that this problem might be best handled using a pairwise coupling, where a custom stack-based classifier outperformed all other classification methods. In Figure 2.3 we present the results of various three-class classification methods. The figure shows the *optimal stack*, as proposed by Koppel and Schler [116], is superior to other approaches by a significant margin.

2.2 Sentiment-Detection Methods

Sentiment analysis includes a diverse family of approaches to how to find and measure sentiment. These can be divided into three main branches (see Figure 2.4): *lexicon-based* [58, 228, 227], *learning-based* [179, 232, 105, 8, 130] and *hybrid*, between the two [169, 262]. Most of the early sentiment-analysis methods were *lexicon-based*. As the name implies, they are typically designed around a lexicon dictionary. *Learning-based* approaches gained popularity slightly later [182, 179], and instantly established themselves as the default and preferable solutions. Supervised classification is far more accurate than *lexicon-based* classification [135, 60]. However, lexicons have not lost their importance: they are usually easier to understand and to maintain by non-experts, and they can also be integrated into *learning-based* approaches [169].

In this thesis, we make use of almost all sentiment-detection methods. We start Chapter 3 with the *lexicon-based* method and expand it using the *supervised learning* method (linear SVM). In Chapter 4, to induce a high-quality domain-specific sentiment lexicon, we will use shallow *neural networks* and compare many *supervised* and *semi-supervised/transductive learning* algorithms. In Chapter 5, we will develop the *unsupervised* approach to *domain-specific* sentiment classification using *distributed*

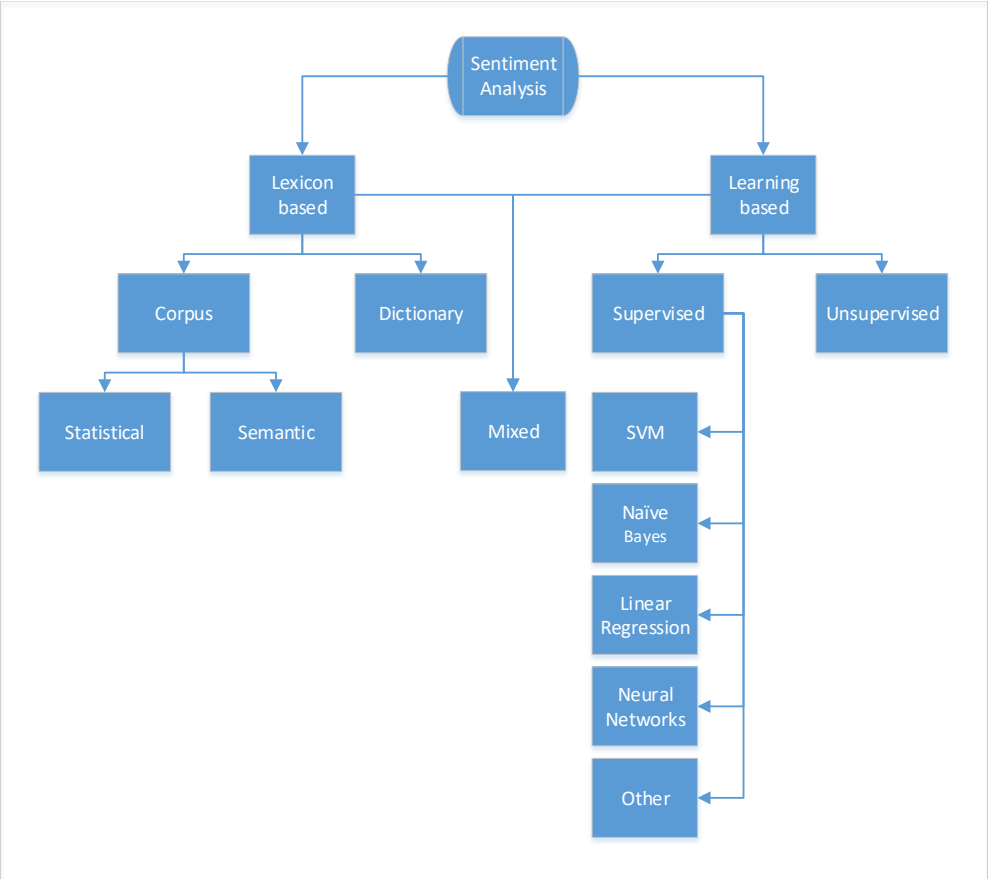


Fig. 2.4: Sentiment-detection approaches

word representations and *deep-learning* models. We will also discuss and compare its performance with almost all possible sentiment-detection methods, such as *lexicon-based*, *unsupervised*, *semi* and *fully supervised learning*. Finally, in Chapter 6, we will present several case studies. As in Chapter 5, almost all methods of sentiment detection and their adaptation to various domains will be demonstrated.

2.2.1 Lexicon-based sentiment detection

As mentioned before, lexicon-based systems require a pre-compiled sentiment lexicon corpus, where each word has an assigned sentiment value. Such lexicons can be either manually crafted [221, 133] or automatically generated using seed words [98, 58]. Many research papers use already published and well-known sentiment lexicons, with one of the first being the *General Inquirer* sentiment lexicon published by Stone et al. [221]. More recently, Liu [133] compiled the *Opinion Lexicon*, which consists of 2,006 positive words and 4,683 negative words, plus misspelled words, slang words and some morphological variants. Another prominent annotated corpus is the multi-perspective question-answering (*MPQA*) compiled by Wiebe et al. [247]. Among many

others, it is worth mentioning *SentiStrength* [227, 228], *SentiWordnet* by Baccianella et al. [11] and the lexicon compiled by Warriner et al. [241]. Twitter messages have their lexicons published as part of SemEval tasks [205]. Moreover, in the case of microblogs, the use of emoticons as a universal sentiment indicator is widely accepted. Lexicons are typically domain independent and single lexicons, such as that published by Go et al. [77], can be shared across multiple domains.

The sentiment value of a text snippet in a lexicon-based system can be calculated by aggregating positive and negative words. In Equation (2.2), we present the most straightforward aggregation implementation, in which the total sentiment f_S is calculated by counting all the positive words w_p and subtracting this number from the count of all negative words w_n.

$$f_S = \sum w_p - \sum w_n \tag{2.2}$$

However, using such a simple approach, a sentiment score would be unscaled and typically skewed by document size. There are many methods for performing a sentiment calculation and scaling, and Lowe et al. [141] highlighted the three most popular:

Absolute proportional difference with bounds $[0,1]$. Using this calculation method (see Equation (2.3)), from the sum of all positive words w_p we subtract the sum of all negative words w_n and divide then by the total number of word occurrences w_a. The main disadvantage of this method is that a sentiment score can be affected by non-sentiment words, as the denominator is a count of all words in a document. Besides that, it is well suited to short text snippets.

$$f_S = \frac{\sum w_p - \sum w_n}{\sum w_a} \tag{2.3}$$

Relative proportional difference with bounds: $[-1,1]$. In contrast to Equation (2.3), in Equation (2.4) the denominator includes only the count of sentiment words. The main disadvantage of this method is that sentiment values cluster around positive and negative poles and are not evenly distributed.

$$f_S = \frac{\sum w_p - \sum w_n}{\sum w_p + \sum w_n} \tag{2.4}$$

Logit scale with bounds: $[-infinity, +infinity]$. The logit (also known as log-odds) [93] is given in Equation (2.5).

$$f_S = log(\frac{p}{p-1}) = log(p) - log(1-p) \tag{2.5}$$

In sentiment-analysis result calculation, we are primarily interested in the relative balance of positive and negative sentiment, or $\frac{P}{N}$ and as Lowe et al. [141] identified,

for such a task the *logit* scale is superior compared to other available methods. In the original logit equation, p is between 0 and 1, but in the sentiment-analysis case we use a slightly modified version, where p is the sum of all positive words w_p, and $1-p$ is the sum of all negative w_n (see Equation (2.6)). It has the smoothest sentiment distribution and is symmetric around zero. β is a fixed coefficient to prevent $log(0)$ from occurring.

$$f_S = \log\left(\sum w_p + \beta\right) - \log\left(\sum w_n + \beta\right) \tag{2.6}$$

In other common variations, equations can be expanded to include sentiment strength as a coefficient or weight and re-scale the final value to fit it into the desirable sentiment scale.

As we have already highlighted, the basic lexicon-based system can be implemented by simply counting positive and negative words. On the contrary, most state-of-the-art systems [63, 169] employ a more sophisticated design using various NLP techniques:

- **Negations**. The most critical part, which can invert sentiment strength (e.g. *'not good'* has a negative sentiment).
- **Intensification**. They can decrease or increase sentiment value strength (e.g. *'much better'*).
- **Text repair**. The use of various heuristic rules to replace idioms, slang and irregular language.
- **Spelling correction**. This is especially important when processing user messages from various microblogging platforms. In the Twitter domain, repeated letters added to the word (e.g. Miiiiike), or the use of multiple exclamations or question marks can also be a reliable indicator of expressed emotion or sentiment [227].

In later chapters, we will employ similar sentiment calculation techniques and design. Unless noted otherwise, we will use Equation (2.6) to calculate the final sentiment value.

2.2.2 Supervised learning

Widely available training and testing data made the supervised learning-based approach the most common sentiment-analysis method. Extracting reviews from Amazon or IMDB and training a linear classifier would give access to a high-accuracy domain-specific sentiment-analysis system [182].

In their paper, Pang et al. [182] evaluated and compared several different supervised machine-learning algorithms for classifying a sentiment extracted from movie reviews. They included the comparison of Naïve Bayes (NB), Maximum Entropy (ME), and

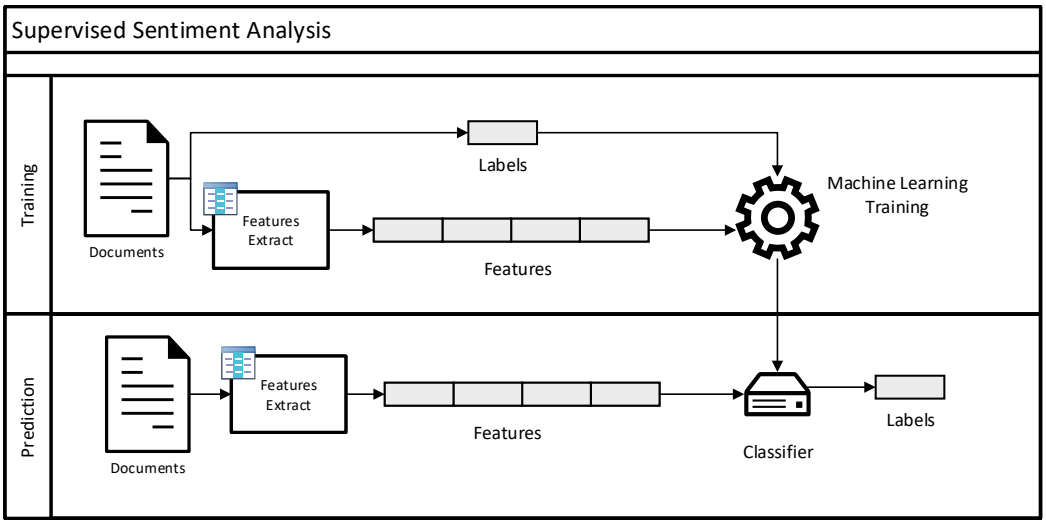


Fig. 2.5: Supervised sentiment-analysis architecture

Support Vector Machines (SVM), with SVM slightly outperforming other learning algorithms. Using a basic design, SVM trained on unigram features (bag-of-words), they managed to achieve a high accuracy of 82.9%. This design was further improved in their later work [179] and achieved an even higher, 87.2%, accuracy. In Figure 2.5 we present the architecture of a typical supervised sentiment-analysis system consisting of feature extraction, training and classifications steps.

As we already mentioned earlier, the design of a bag-of-words supervised sentiment system can be implemented in just a few lines of Python code, and such an implementation can surpass the method proposed by Pang and Lee [179] and achieve 90% accuracy on the IMDB dataset [144]. In the bag-of-words approach, each word is represented as a separate and independent feature. Using a *linear SVM classifier*, the hyperplane decision boundary can be calculated using Equation (2.7), and items are separated into positive and negative classes by the straight line (see Figure 2.6).

$$w^t x_i + \beta = 0 \tag{2.7}$$

Still, as other researchers have found [200], such a straightforward design solely based on supervised machine learning typically suffers from *style*, *domain*, or even *time* dependencies. In comparison, lexicon-based sentiment-detection systems have lower overall accuracy but suffer less from domain dependency. It is important to mention that lexicon-based systems also have a domain dependency problem but with lower sensitivity to crossing domain boundaries. In later chapters, we will demonstrate that, in some domains, lexicon-based systems can suffer a significant performance loss.

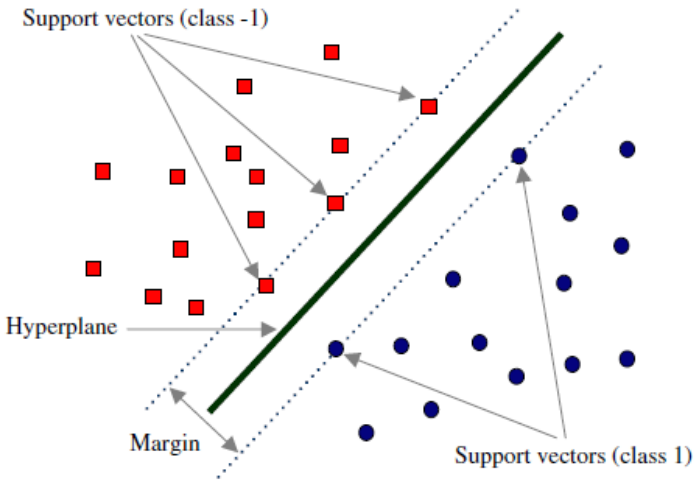


Fig. 2.6: SVM separating hyperplane

In contrast to lexicon systems, a naive document or sentence-level machine-learning-based implementation is a black-box system, is difficult to understand and maintain by a human user, and solely relies on training dataset quality. It performs a sentiment classification and provides only an overall sentiment score, without any further explanation or justification. It is possible to extrapolate some explanation using feature weights; however, that will not explain why a model decided to assign such weights in the first place. Thus, many attempts have been made to incorporate lexicon knowledge into machine-learning classifiers [6, 91, 60]. Other researchers have designed so-called multi-stage systems, in which lexicon and machine learning are employed only for certain subtasks, such as extending a sentiment lexicon [260] or identification of subjective-text blocks [179]. In the case of mixed approaches, the first wave followed the increasing popularity of various generative probabilistic models based on Latent Dirichlet Allocation (LDA) [91, 105, 130]. However, their sentiment-analysis performance was often worse than the simple bag-of-words SVM approach [182]. The second, more successful wave followed advances in deep learning and word-embedding techniques [60].

In Chapter 3, we will explore the possibility of combining lexicon and machine-learning techniques for opinion analysis. As Mladenić et al. [164] identified, weights extracted from a linear SVM may be a good indicator of feature importance. This observation is based on the fact that a feature weight obtained from the linear model represents a vector coordinate which is orthogonal to the hyperplane, and its direction indicates the predicted class. The magnitude of a vector or distance from the hyperplane also tells us how informative the feature is for the classification and its importance in the data-separation task. By selecting sentiment words and adjectives as SVM features, we will use machine learning to discover a domain-specific sentiment lexicon, measure the importance of sentiment words and adjust other lexicon-inspired components. In

a sense, the training phase will work as an adaptation link between a generic lexicon approach and a target domain.

2.2.3 Unsupervised learning

Until recently, unsupervised methods in sentiment analysis were less prevalent. Most frequently they are employed only for one of the sentiment-detection subtasks, such as dealing with lexicon induction, aspect extraction or a domain-adaptation task. Unsupervised sentiment-analysis methods are also closely related to lexicon induction and domain adaptation, which we will cover in more depth in later sections. In one of the first unsupervised lexicon-induction papers, Turney [232] utilised POS tags to find candidate sentiment words by exploiting, now a widely accepted fact, that most sentiment words are adjectives and adverbs. Many other works followed proposing various methods [10, 78, 110]. In the early days, some researchers tried to exploit semantic representation using WordNet [98, 111], but typically they suffered from lower accuracy, and WordNet-based approaches also struggled with domain adaptation. Later research shifted to LDA-based semantic discovery, and now it is more common to use word embedding [86] for similar tasks.

As an alternative, other researchers tried various clustering methods [266, 44], although such an approach typically performed worse than weakly-supervised or multi-stage methods. As Eisenstein [60] has recently discovered, lexicon-based systems can improve unsupervised binary sentiment classification, and such a mixed system can go a considerable way towards closing the gap to supervised methods. As we already mentioned before, in Chapter 5, we will present our *nearly-unsupervised* approach to *domain-specific* sentiment classification of documents for a new domain based on distributed word representations (vectors) with performance on a par with conventional supervised sentiment analysis.

2.2.4 Deep learning

Recent advances in deep-learning techniques have opened up the possibility of using neural networks to perform sentiment analysis [49, 114, 96]. It has been demonstrated that deep-learning models are more effective in tackling sentiment-detection problems [96] and have been extensively applied in the field of NLP and sentiment analysis. They are better at capturing the semantic relationship between words, and models such as Long Short-Term Memory (LSTM) recurrent neural network (RNN) [94] allow the memorisation of long-term contextual information. One of the many appeals of LSTM is that it can connect previous information to the current context and allow seamless integration of word embeddings as the projection layer of the neural network; thus, we

can use domain-aware word vectors. As was reported by Dai and Le [49], the LSTM RNN can reach or even surpass the performance levels of all previous baselines and produce better results than other RNNs. Researchers have also reported good results using other neural networks such as LSTM autoencoders [219] and Convolutional Neural Networks (CNN) [114, 47].

More recently, Radford et al. [197], using multiplicative LSTM with 4096 units, discovered the so-called sentiment unit, which learned an accurate representation of the sentiment, a promising step towards developing a system with unsupervised sentiment empathy. Despite being trained only to predict the next character in the text of Amazon reviews, a single "sentiment neuron" was highly predictive of sentiment, and by tweaking the "sentiment neuron", this network was able to generate positive or negative reviews.

Technical innovations using attention models have introduced a set of new approaches that have obtained new state-of-the-art results in a number of NLP tasks, including sentiment analysis [55]. Such approaches are typically based on bidirectional training of Transformers [233] to learn a language model (LM), with the most important research in this area having been undertaken by Radford et al. [198], Devlin et al. [55], and more recently by Radford et al. [199].

We use various deep-learning networks in Chapters 5 and 6. Our results will show that for most sentiment-analysis tasks, LSTM neural networks will demonstrate superior performance, surpassing all other methods. Only in the market-prediction task, in Chapter 6, SVM with an RBF kernel will outperform the LSTM model, which can be explained by the size of the training dataset.

2.2.5 Word embedding

Maas et al. [144] were among the first to identify the importance of learning word vectors and the possibility of using them in the construction of an unsupervised sentiment-detection system. More recently in sentiment analysis researchers have started using the two-layer neural networks *word2vec* [160] and *GloVe* [185]. The introduction of word-embedding methods has had a substantial impact on various NLP-related areas and opened up the possibility of new sentiment-detection methods. Both approaches learn word vectors from their co-occurrence information that helps understand semantic relationships between words and allows the grouping of words based on their linguistic similarity. Representations are typically constructed from a sizeable unlabelled corpus, and produce a vector space of several hundred dimensions, with each word being assigned a corresponding vector. Words with similar meaning are usually located nearby within the vector space, and that vital feature has been exploited in various

sentiment-analysis methods [60]. Many of the latest sentiment-analysis methods use pre-trained word embeddings as the first (projection) layer of the neural network.

Rothe et al. [207] proposed a *DENSIFIER* method to reduce the dimensionality of word embedding without losing semantic information and explored applications in various domains. *DENSIFIER* performed slightly worse in the SemEval-2015 task [205] compared to *word2vec*, although its training time was shorter by a factor of 21.

We discuss word embedding in more depth in Chapter 4, where we will present our lexicon-induction approach and confirm that words with different sentiment polarities form distinct clusters in a word vector space.

2.3 Domain Adaptation

All sentiment-analysis approaches perform well if targeted at a specific domain. However, they suffer significant performance loss once domain boundaries are crossed [200]. Read [200] identified three types of boundaries: **topic**, **domain** and **temporal**. Crossing one of these boundaries typically has a detrimental impact on the performance of a sentiment system. Kaur and R. Saini [112] also found that **writing style** has a significant impact on sentiment-analysis performance and can be classified as another domain boundary. As Xiao and Guo [254] noted, **cross-language** sentiment classification can be identified as a special domain-adaptation case. We would also mention other instances of sentiment anomalies, which require adaptation, such as sarcasm and irony. E.g. sentence *'Nice perfume. Must you marinate in it?'* contains positive sentiment words. However, it should be classified as a negative sentiment text block.

The simplest way to adapt a sentiment-detection system to an underlying domain is by collecting labelled domain-specific training data. However, that is an expensive and time-consuming task. Thus, considerable effort has been invested in finding an automated method of domain adaptation by designing unsupervised sentiment-detection systems [60], various knowledge transfer methods [65, 27, 25] and systems less sensitive to crossing domain boundaries [169]. Research into adaptive sentiment analysis can be categorised into **domain-to-domain** and **general-to-domain** adaptation. Another approach would be **lexicon induction** and expansion with domain-specific sentiment words.

It is also important to mention that most domain-adaptation solutions, even in the case of unsupervised methods, retain domain dependency and would still have difficulties processing documents from new and unknown domains. To process documents from multiple domains, we would also need a topic-detection component. Each domain typically has a distinctive aspect signature, and their extraction can help in domain identification and classification [115].

Through this thesis, we explore various domain-adaptation and cross-domain sentiment-analysis scenarios. In Chapter 3 we will explore the *near-cross-domain* environment, or, in other words, *cross-style*, as both datasets are from the same *topic domain*, despite using different writing styles. The experimental results in that chapter will illustrate one of the principal advantages of our *pSenti* algorithm (i.e. lower topic and style dependency compared to a pure bag-of-words machine-learning implementation). To further improve performance, in Chapter 4 we will explore lexicon induction, adaptation and expansion with domain-specific sentiment words. Finally, in Chapter 5 we will take this one step further and propose a novel nearly-unsupervised domain-adaptation method, which almost matches the performance of the supervised method. In a sense, semi-unsupervised sentiment analysis is one of domain-adaptation methods. Such a system has the ability to discover domain-specific sentiment without supervision and adapt to an underlying domain.

The political-sentiment analysis in Chapter 6 will demonstrate the importance of domain adaptation and the advantages of our proposed method. The lexicon-based approach with the general-purpose sentiment lexicon will show inferior performance with 0.584 AUC, just fractionally better than a random selection. However, domain adaptation will significantly improve its performance. The method based on the lexicon induction from Chapter 4 will produce reasonable results with 0.723, and the semi-supervised approach based on the model from Chapter 5 will improve results further and achieve a high 0.803 AUC.

2.3.1 Knowledge transfer

The purpose of transfer learning is to use the knowledge of a source domain to perform sentiment analysis in a target domain. In their survey, Pan and Yang [178], categorised knowledge transfer methods into three main approaches: feature-based, instance-based and model-parameter-based. Faralli and Navigli [65] proposed a domain-driven Word Sense Disambiguation (WSD) method, where they iteratively created glossaries for several domains using a bootstrapping technique. It does not cover sentiment detection, yet it demonstrated the importance of identifying word sense.

Another approach is based on various techniques for uncovering the correlation between the source and target domains. For example, Bollegala et al. [27] developed an unsupervised cross-domain sentiment classifier using an automatically extracted sentiment-sensitive thesaurus and computing the correlation between the source and target domains. Similarly, Bollegala et al. [25] created an unsupervised method for learning cross-domain word representations using a given set of pivots and non-pivots [25] selected from a source and a target domain. Pan et al. [177] proposed a spectral feature alignment (SFA) method, based on labelled data from a source domain and a

set of domain-independent features. Using these classification settings, they managed to reduce the gap between domain-specific words of the two domains and to improve sentiment classification performance on a new domain.

It is also popular to use various types of ensembles. For example, Samdani and Yih [214] and Xia and Zong [253] proposed feature-set-based ensembles, while Li et al. [127] trained multiple classifiers on different domains for the final decision-making. However, all these transfer learning approaches are supervised and hence require labelled data for training.

2.3.2 Lexicon induction

Domain adaptation is also tightly related to the problem of constructing a sentiment lexicon. Owsley et al. [175] found that to achieve "good" results using a lexicon-based system you must build a domain-specific lexicon that is related to both the entities and their sentiment identification. In different domains the same word could have an opposite meaning or a very different sentiment strength. Moreover, as Lu et al. [142] found in their research, even a different context may have an impact on a sentiment orientation. One of the most reliable methods to build a domain-specific lexicon is to use professional human annotators. Many of the publicly available lexicons were manually crafted by human annotators (e.g. Mohammad et al., Stone et al., Liu [165, 221, 133]).

A lexicon induction can also be performed using both supervised and semi-supervised learning methods. Overall, supervised lexicon-induction methods are less common, yet in Chapter 3 we will present a novel sentiment-analysis method which may also be employed as a supervised lexicon-induction method. Another common approach to generating a domain-specific lexicon is to expand it gradually using an initial small set of seed of words. This family of semi-supervised lexicon-induction algorithms is far more dominant among other alternatives. Early methods have successfully utilised WordNet [98, 58]. By exploring synonym and antonym sets in WordNet, they predicted the semantic orientations of adjectives and expanded sentiment lexicons. However, it is worth mentioning that such methods do not adjust the sentiment value for each sentiment word in the lexicon; they merely expand the lexicon with previously unknown sentiment words. To generate a domain-specific lexicon, it is also necessary to acquire an adapted version of WordNet, and only a few domains have specific WordNet versions available.

More recently, Hamilton et al. [86] demonstrated in their work that using label propagation with high-quality word vector embeddings could induce a domain-specific sentiment lexicon which can achieve performance competitive with methods that rely on hand-curated dictionaries.

Authors	Count	Emotions
Ekman [62]	6	anger, disgust, fear, joy, sadness, surprise
Parrott [183]	6	anger, fear, joy, love, sadness, surprise
Frijda [69]	6	desire, happiness, interest, surprise, wonder, sorrow
Plutchik [189] (see Figure 2.7)	8	acceptance, anger, anticipation, disgust, joy, fear, sadness, surprise
Tomkins [230]	9	desire, happiness, interest, surprise, wonder, sorrow
Matsumoto [153]	22	joy, anticipation, anger, disgust, sadness, surprise, fear, acceptance, shy, pride, appreciate, calmness, admire, contempt, love, happiness, exciting, reg

Table 2.2. Some existing definitions of basic emotions

As Eisenstein [60] found, lexicon-based systems can also improve an unsupervised binary sentiment classification, and such a mixed system may be a considerable step towards closing the gap to supervised methods. However, even a superior-quality lexicon does not guarantee good performance in a real-life sentiment-analysis task, and they have not tested the induced lexicon in a sentiment-analysis task. Our results in Chapter 4 show that lexicon-based systems can be susceptible to borderline noise in a generated lexicon. We will also demonstrate that better lexicon-induction results can be achieved using a more straightforward approach. The simple SVM-based model, trained on only a couple of seed words, can outperform all other models and can be a better alternative to more complicated label-propagation methods. We will also demonstrate the advantage of generating domain-specific sentiment lexicons and provide evidence that different domains have different sentiment vector spaces.

2.4 Sentiment Dimensionality

In real life, sentiment is not a binary value and can have quite a complicated structure, with more than ninety different emotions which can have a different meaning to different people [163]. To understand human emotions, researchers and engineers started employing various psychological models. One of the most popular emotion-definition models was introduced by Ekman [61], which defined six primary human emotions and twenty-four secondary ones. Cambria et al. [36] analysed existing models and proposed a new biological and psychological emotion-categorisation model. In Table 2.2 we list some of the most popular approaches, together with a list of primary emotions.

Typically in opinion mining we are trying to answer a two-dimensional question on how much an audience likes or dislikes something. Thus, not all sentiment dimensions

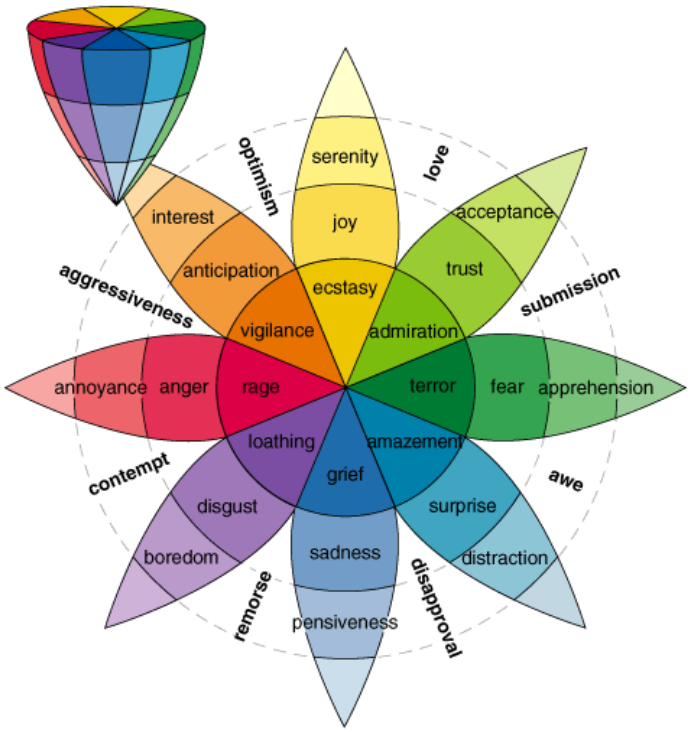


Fig. 2.7: Plutchik's sentiment wheel [190]

are required, and most of the time the distance from the two extreme poles is sufficient. In other cases, specific dimensions can be instrumental. For example, Zhang et al. [263] found that sentiment dimensions such as *hope* and *fear* displayed a significant positive correlation with the *Market Volatility Index*, also known as *VIX*. Moreover, those two dimensions in Twitter messages were shown to help stock market prediction. The multidimensional sentiment is also sensitive to domain boundaries and requires adaptation; however, not much research has been done in that area yet.

We will first briefly use multidimensional sentiment in Chapter 3. In Chapter 6 we will investigate them in more depth and apply them to multiple sentiment-analysis scenarios (e.g. we will investigate the causal relationship between sentiment attitude/emotion signals and stock price movements using various sentiment signal sources and different time periods).

2.5 Temporal Analysis and Sentiment Time series

A compelling language characteristic is that lexical word meaning can change over time, and can have a direct impact on sentiment analysis. Over time, new words and terms can also be introduced. There are two primary methods to study word sense change . In the **semasiology** method [73], we track how a word changes its sense,

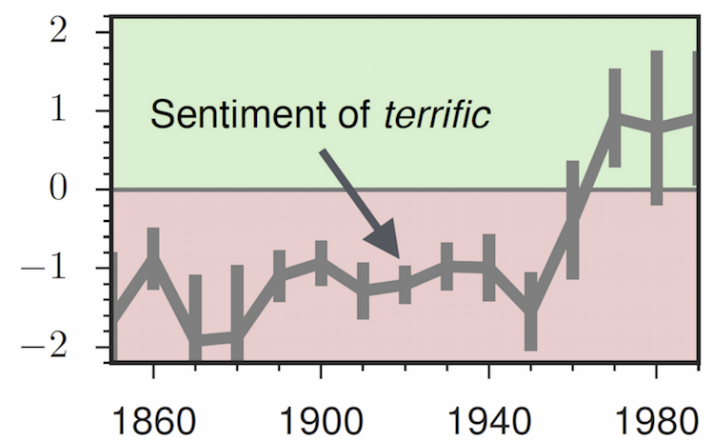


Fig. 2.8: The sentiment of "terrific" changed from negative to positive over the last 150 years [86]

and in the **onomasiology** [72], we want to find out how a concept expression (in our case sentiment) changes and what new expression forms arise. A typical change in word meaning requires one of four triggers [21]: it can be triggered by linguistic, psychological, sociocultural or cultural forces. Language evolution and change detection have been investigated by many researchers [117, 159]; however, significantly less so from the sentiment-analysis perspective. In their work, Hamilton et al. [86] investigated how sentiment meaning and their polarities can drift over time (see Figure 2.8). They also demonstrated that a sentiment word could become obsolete due to temporal changes, yet it frequently requires many decades to appear. All those findings suggested the importance of a sentiment change point detection and constant domain adaptation.

In Chapter 6, we will analyse temporal sentiment drift in Amazon reviews, and our results will confirm that temporal dependencies can indeed be observed in Amazon reviews. This finding will confirm the importance of temporal sentiment monitoring and continuous sentiment system adaptation to the underlying domain.

Another related topic is sentiment time series and the investigation of its characteristics. Mei et al. [157, 156] investigated mining subtopics and analysing their dynamics over time. Others have tried to exploit sentiment data in forecasting market movements [9], election results [17] and future sales [261]. There have also been attempts to predict sentiment using its past time-series values [74]. However, we could argue that such an approach is typical for any time-series data and does not explain why sentiment changes.

We will investigate sentiment time series in Chapter 6. Our results indicate that aggregating past sentiment can significantly boost performance and can be employed to improve the sentiment analysis of customer reviews.

2.6 Application Areas

As we have already highlighted, sentiment analysis has many practical applications. One of the most popular applications is product review sentiment analysis, and there are many commercial review-management platforms to choose from, such as Bazaarvoice[8], PowerReviews[9], Yotpo[10] among others. They offer numerous services, from promises to help boost brand recognition, to the possibility of increasing sales by collecting and analysing product reviews. One of the typical sentiment-analysis examples would be the Aspectiva[11] product, which performs aspect-level review aggregation and analysis of what people discuss about various product aspects and their feeling towards them. Others offer more niche services, such as PowerReviews, which promises to detect fake reviews and attempts to damage a brand. We explore similar sentiment application scenarios in Chapters 3, 5 and 6, where we detect customers' feelings towards various products, analyse movie reviews and provide a detailed explanation of their opinion on individual aspects.

Twitter is another important source of sentiment information with an even more extensive range of applications, from brand reputation monitoring[12], political campaign analysis [217, 119, 120, 226] to financial market [28, 9] and movies box office revenue prediction [13]. Similar trends are delivered on other social platforms as well, where start-ups such as SentiOne[13] offer sentiment monitoring across a set of various sources from Facebook and Twitter, to public forums and other portals. We will introduce our novel Twitter analysis method in Chapter 5 and demonstrate various practical applications in Chapter 6.

The successful campaign and use of Twitter by Barack Obama in 2008 and, more notably, by Trump in the 2016 US presidential election, confirmed the importance of social media and its impact on politics. It also demonstrated the importance of understanding how it affects surrounding society and has been a favourite subject for research. Numerous studies have been performed to tackle this problem [102, 46]. Borondo et al. [30] even developed a conceptual model to predict the election winner. More recently, researchers have investigated Trump's election campaign [234, 136] to study users who follow the presidential candidates. In Chapter 6, we also investigate Donald Trump supporters and opponents, and their stances and political-sentiment in the 2016 US presidential election.

[8] http://www.bazaarvoice.com
[9] http://www.powerreviews.com
[10] http://www.yotpo.com
[11] http://www.aspectiva.com
[12] http://www.tweetreports.com
[13] http://sentione.com

In recent years, a whole new industry has been formed around financial market sentiment detection [256, 255]. Traditional financial news/data providers, notably Thomson Reuters [67] and Bloomberg [23], have started providing commercial sentiment-analysis services. As a result, new financial platforms, such as StockTwits[14] (see Figure 2.9), which offers sentiment-analysis tools, have also emerged. Nowadays, many investment banks and hedge funds are trying to exploit the sentiments of investors to help make better predictions about the financial market. Some of the most prominent financial institutions (including DE Shaw, Two Sigma and Renaissance Technologies) have been reported to utilise sentiment signals [103], in addition to structured transactional data (such as past prices, historical earnings, and dividends) in their sophisticated machine-learning models for algorithmic trading.

According to the *efficient market hypothesis* (EMH) [147], it is impossible to "beat the market", since stock market efficiency always causes existing share prices to incorporate and reflect all relevant market information. However, many people have challenged this claim and declared that it is possible to predict price movements with more than 50% accuracy [100, 194].

A variety of technical approaches to market trend prediction have been proposed in the research literature, ranging from AutoRegressive Integrated Moving Average (ARIMA) [239, 176] to ensemble methods [194]. In their work, Huang et al. [100] demonstrated the superiority of SVM in forecasting weekly movement directions of the NIKKEI 225 index, and Lin et al. [131] managed to achieve 70% accuracy by combining decision trees and neural networks. Recent advances in deep learning have brought a new wave of methods [42, 71] to this field. In particular, the Long Short-Term Memory (LSTM) (RNN) has been shown to be very effective.

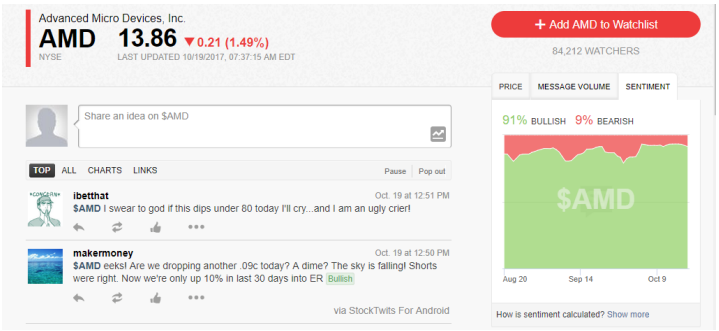


Fig. 2.9: Market sentiment

Numerous studies have been carried out in an attempt to understand the intricate relationship between sentiment and price on the financial market. Wang et al. [236] investigated the correlation between stock performance and user sentiment extracted

[14]http://stocktwits.com

from StockTwits and SeekingAlpha[15]. Ding et al. [57] proposed a deep-learning method for event-driven stock market prediction and achieved nearly 6% improvements on S&P 500 index prediction. Arias et al. [9] investigated whether information extracted from Twitter can improve time-series prediction and found that indeed it could help predict the trend of volatility indices (e.g., VXO, VIX) and historical volatilities of stocks. In their research, Bollen et al. [28] identified that some emotion dimensions extracted from Twitter messages can be good market trend predictors. A more recent study by Tabari et al. [224] drew a similar conclusion. Similar to our approach in Chapter 6, Deng et al. [54] combined technical analysis with sentiment analysis. However, they used only a limited set of technical indicators, together with a generic lexicon-based sentiment-analysis model, and attempted to predict future prices using simple regression models.

In Chapter 6, we perform various experiments in the financial domain. First, we use mood and sentiment extracted from *Financial Times* articles, news headlines and tweets. Second, *Granger causality* analysis is carried out to assess whether they have any potential predictability power on the stock price change. Later, we use mood and sentiment to generate **BUY** and **SELL** signals and demonstrate that, in some cases, the model using sentiment information outperforms the baseline method. Our experimental results on stock market prediction show that for some selected stocks both general sentiment and mood data integration could enhance the baseline and improve prediction results.

2.7 Conclusion

In this chapter, we provided a broad overview of the sentiment-analysis area and reviewed previous research related to our work on adaptive sentiment analysis.

In Section 2.1 we briefly described the main components required for the sentiment-analysis task, data collection methods and sources, text processing and transformation, as well as how it is crucial to adaptive sentiment analysis. We also reviewed sentiment granularity levels and their selection, with an overview of why neutral, also known as subjective, classes are essential in sentiment analysis. In this chapter, we also considered aspect detection and opinion-holder identification, and described why this is important to our research topic and domain adaptation.

In Section 2.2 we discussed main sentiment-detection approaches starting from a lexicon-based approach and ending with a learning-based approach. The chapter also covered the evolution of sentiment-analysis methods and included design samples of the most popular sentiment-analysis methods; reviewed research in both supervised and

[15]https://seekingalpha.com/

unsupervised methods; and covered deep-learning-based sentiment analysis, including word embedding.

In Section 2.3 we covered domain-adaptation methods proposed by other researchers and their evolution over time. We also presented the challenges they face and discuss their importance. More specifically, we covered knowledge transfer, domain-specific lexicon selection, and induction and adaptation, as well as unsupervised sentiment analysis. We reviewed the main components required for domain adaptation, highlighted the importance of aspect detection and presented the value of word embedding. In Sections 2.4 and 2.5, we briefly touched on sentiment dimensionality and temporal sentiment analysis. In Section 2.6 we reviewed the most popular applications and the challenges they face.

We will also discuss some related work in more detail in later chapters when it is relevant to specific problems that we consider in our research.

Chapter 3

Concept-Level Domain Sentiment Discovery

3.1 Introduction

Most of the early sentiment-analysis systems took a *lexicon-based* approach to a document sentiment classification task. This approach is based on the so-called *lexicon design*, having domain-specific sentiment lexicons as the main sentiment information source [58, 228, 227]. Later, the focus of research shifted more to learning-based approaches [182, 179]. Sentiment-analysis systems based on supervised machine-learning techniques usually achieve the best performance in sentiment detection. However in many cases, they are black boxes in the sense that no explanation or justification can be provided to users.

Another concern in sentiment analysis is the domain dependency problem. With a large enough training corpus, a supervised learning-based method can perfectly fit a target domain and achieve a high sentiment classification accuracy. Unfortunately, this comes at a cost, such as the domain overfitting or dependency issue. Domain dependency is not unique to learning-based methods. Other approaches also have difficulties dealing with documents outside of their domain boundaries. However, learning-based methods are more susceptible to this problem and have a higher sensitivity to crossing domain boundaries. On the one hand, machine-learning solutions have superior performance, but they suffer a significant loss of accuracy if domain boundaries are crossed.

To address domain adaptation, researchers have proposed various methods. Almost all of the domain-adaptation experiments have been done on synthetic datasets, which have clearly defined domain boundaries. Yet real-world information sources typically contain a mixture of cross-domain documents and have different characteristics from static experimental datasets. Moreover, the domain-adaptation process does not make the underlying sentiment-analysis model less agnostic to domain boundaries. In other

Customer Review => { (Aspect$_1$: View$_1$), (Aspect$_2$: View$_2$), ..., (Aspect$_k$: View$_k$) },
e.g.,
a user comment on Google Chrome => { (Appearance: +0.8), (Plugins: +0.6), ..., (Speed: +0.9) }.

Fig. 3.1: An example of *pSenti*'s aspect-oriented output.

words, even after domain adaptation, learning-based sentiment-analysis methods retain their sensitivity to crossing domain boundaries and typically have difficulties dealing with noisy sentiment sources. As we will demonstrate later, lexicon-based systems are less sensitive near domain boundaries. Thus, there is a need for a concept-level sentiment-analysis system that could seamlessly integrate lexicon-based and learning-based approaches to get the best of both.

3.2 Contribution

To overcome the above lexicon and machine-learning limitations, we have developed a novel sentiment-analysis method which is less sensitive to crossing domain boundaries and has similar performance to pure learning-based methods. In this chapter, we present the anatomy of *pSenti*[1][2] — **a concept-level sentiment-analysis system** that seamlessly integrates lexicon-based and learning-based approaches to acquire **adaptive sentiment analysis**.

The main advantage of our *hybrid* approach using a lexicon/learning symbiosis is to get the best of both worlds — the stability and readability of a carefully hand-picked lexicon, and the high accuracy from a powerful supervised learning algorithm. Thanks to the built-in sentiment lexicon and numerous linguistic rules, *pSenti* can detect and measure sentiments at the concept level, providing structured and readable aspect-oriented outputs, as illustrated in Figure 3.1.

The main idea of *pSenti* is to generate feature vectors for supervised machine learning in the same fashion as lexicon-based sentiment-analysis systems see it. In a sense, *pSenti* is a lexicon-based sentiment-analysis system with an integrated learning-based domain-adaptation module. Our experimental results confirmed that such a two-step design is less prone to domain overfitting and less sensitive to a change of topic or writing style. Compared to pure lexicon-based systems, it achieves significantly higher accuracy in sentiment-polarity classification and sentiment-strength detection. Compared to pure learning-based systems, our method offers more structured and readable results with aspect-oriented explanation and justification, while being less sensitive to the writing style of a text. Moreover, contrary to a bag-of-words design, it can be modified and further adjusted after a learning phase (i.e. we can introduce

[1]https://github.com/AndMu/Wikiled.Sentiment
[2]http://www.dcs.bbk.ac.uk/~andrius/psenti/

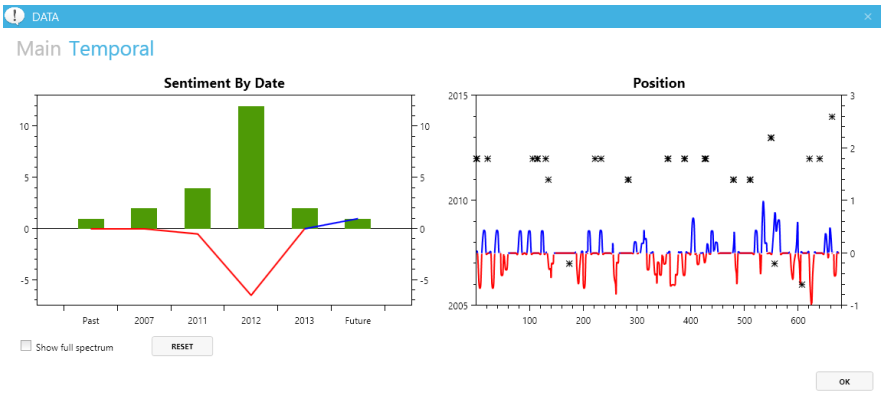


Fig. 3.2: *pSenti* article temporal sentiment-analysis output

new linguistic rules or expand a sentiment lexicon at any time to further improve the system's performance). Also, in the case of insufficient labelled training data, it can fall back to a lexicon-based component and perform sentiment analysis of unseen examples.

The ability to perform cross-style sentiment analysis is significant, as it implies that we can train the system using formal professional reviews as training examples and then apply the system for sentiment analysis on informal customer reviews. We cover the anatomy of our proposed approach in **Section 3.4**. The extensive experiments we have carried out on two real-world datasets are reported in **Section 3.5**. Both datasets, CNET software reviews and IMDB movie reviews, confirm the superiority of the proposed composite approach over state-of-the-art systems such as *SentiStrength* [227, 228].

In addition to a single-dimensional sentiment output, it is also able to calculate the eight Plutchik [189] mood dimensions — *anger, anticipation, disgust, fear, joy, sadness, surprise and trust*. Mood dimensions are extracted using the NRC sentiment lexicon [166] and are generated as an XML output for each document and a whole dataset (see Listing 3.1).

```
<MoodData>
    <Mood name="Anger" value="0.028"/>
    <Mood name="Anticipation" value="0.054"/>
    <Mood name="Disgust" value="0.011" />
    <Mood name="Fear" value="0.038" />
    <Mood name="Joy" value="0.028" />
    <Mood name="Sadness" value="0.028" />
    <Mood name="Surprise" value="0.015" />
    <Mood name="Trust" value="0.108" />
</MoodData>
```

Listing 3.1: Mood information XML output

pSenti can also resolve four sentiment temporal orientations: *past, present, future*, and *undefined* as well as present results with the greater granularity (see Figure 3.2). A temporal orientation is calculated using two different methods: using the *SuTime* temporal tagger [40] and by finding tense of a sentence.

In later chapters we will make more extensive use of mood and temporal sentiment information.

3.3 Datasets

To empirically evaluate our *pSenti* system, we conducted experiments on two real-world datasets.

- *The first dataset*: Software-Product-Reviews[3] consists of software product reviews collected by the thesis author from CNET's software download website. The dataset includes five software product categories: Browser, Antivirus, Video, Action Games and Utilities. Most software reviews are written by customers (average users), but there are some which are written by professionals (CNET editors).
- *The second dataset*: Movie Reviews[4] consists of movie reviews collected by Pang and Lee [179] from the IMDB website. It is a well-known standard benchmark dataset for sentiment analysis.

The first dataset was collected using a custom-built Web scraper by applying the following procedure:

- Extracted all available product categories from the review website[5].
- For each of the categories, crawled all the pages, containing customer and editor reviews.
- For each review, extracted the name of the user, the text, the time stamp of the review and the original rating.

A customer review is typically a short text snippet with an average length of a couple of hundred characters (see Table 3.1). Editor reviews are longer, however, with an average length of over a thousand characters (see Table 3.1). The following example illustrates a typical customer review:

[3]http://www.dcs.bbk.ac.uk/~andrius/psenti/
[4]http://www.cs.cornell.edu/People/pabo/movie-review-data/
[5]https://download.cnet.com

Dataset		Labels of Reviews	Number of Reviews	Avg Length of Reviews
Software Reviews	Miscellaneous (Editor)	Pos/Neg	1660	1056.82
	Browser (Editor)	Pos/Neg	360	1091.61
	Browser (Customer)	Pos/Neg	2000	158.07
	Antivirus (Customer)	Pos/Neg	2000	165.06
	Video (Customer)	Pos/Neg	2000	152.43
	Action Games (Customer)	Pos/Neg	2000	136.21
	Utilities 1 (Customer)	Pos/Neg	2000	155.80
	Utilities 2 (Customer)	1-5 Stars	1850	295.19
Movie Reviews	Movies 1	Pos/Neg	2000	3892.96
	Movies 2	1-5 Stars	5000	2257.44

Table 3.1. The experimental datasets.

"It comes with great features, no worries of updates as it does it all with automatic updates and keeps your computer running smooth."

The datasets have been pre-processed to remove duplicates, spam and inconsistencies. A number of researchers [192, 128, 31] have highlighted that a potential concern when performing sentiment classification is that the training data may contain class imbalance that can negatively affect classification performance. Thus, to address this issue and avoid sampling bias, we are using random undersampling to produce a balanced dataset (i.e. each class has a similar number of reviews). The detailed characteristics of these datasets are shown in Table 3.1.

3.4 Model

Our concept-level sentiment-analysis system, *pSenti*, is developed by combining lexicon-based and learning-based approaches. As shown in Figure 3.3, the supervised machine-learning component is responsible for multiple tasks, such as adjusting sentiment values and new sentiment word discovery. To derive the final output, it performs adjustment

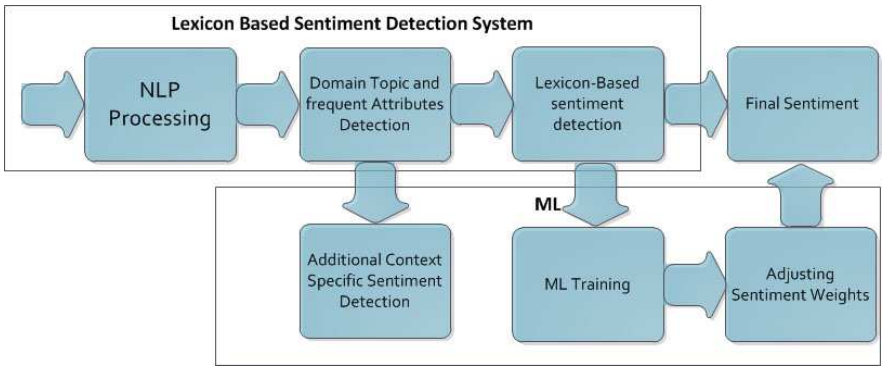


Fig. 3.3: The system architecture of *pSenti*.

Fig. 3.4: Sentence-level *pSenti's* analysis interface

of all lexicon components, including semantic rules. The *pSenti* system measures and reports the overall sentiment of a given opinionated text, such as a customer review, as a real-valued score between -1 and $+1$, which can then be easily transformed into a positive/negative classification or a range of 1-5 stars (see Equation (3.3)). It can also output sentiment as an eight-dimensional mood vector.

The system has a rich UI, making it easy to use to analyse sentiment dynamics and understand how sentiment changes over time. Using its interface, we can inspect sentiment changes on both sentence (see Figure 3.4) and word (see Figure 3.5) levels.

3.4.1 Preprocessing

The core of *pSenti* is its lexicon-based system, so it shares many common NLP processing techniques with other similar approaches. It supports two different NLP frameworks: Stanford CoreNlp [149] and OpenNlp [7]. During the first step of text processing, we carry out tokenisation, POS and entity tagging. Before feeding a piece of a document into the parser, we perform some text clean-up, simplification and transformations.

As part of the transformation, we replace known idioms and emoticons with text masks. *pSenti* can read both text emoticons and those encoded as Unicode images. For example, the emoticon ":-)" or its Unicode representation, will be replaced by the token `EMOTICON_SMILE`. `EMOTICON_SMILE` is listed in the default lexicon as a sentiment word with $+2$ sentiment value. Similarly, ":|", which has a negative sentiment strength -1,

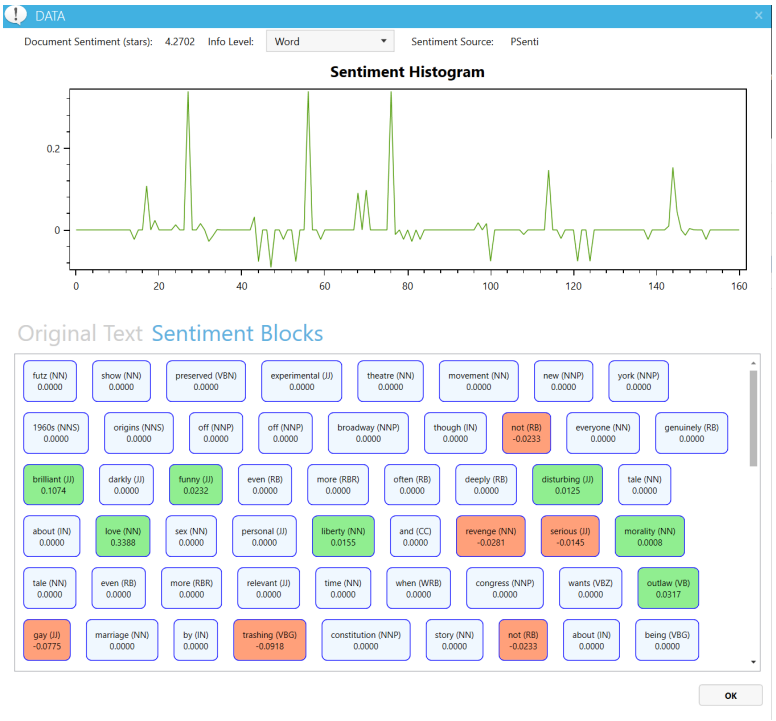


Fig. 3.5: Word-level *pSenti's* analysis interface

will be replaced by the token `EMOTICON_CONFUSED`. The assumption here is that various emoticons express different sentiment strength, which has already been measured, differentiated and added into the standard lexicon. Emoticon tokenisation simplifies their processing and understanding by machine-learning algorithms, and allows them to be further adjusted, depending on the underlying domain.

Emoticons are commonly used as a universal method to express sentiment and have similar sentiment strength across many domains [238]. As they are ubiquitous in social media domains, *pSenti* has an option to use the emoticon-only sentiment lexicon. Such an approach is useful in the procedure of bootstrapping the training sample, which we extensively employ in the following chapters.

Idioms follow slightly different heuristic rules, and they are replaced using system-defined tokens. Thus "crocodile tears", known to have sentiment strength -3, should be replaced by `_Bad_Three_` token. The range of sentiment values for emoticons and idioms is from -3 to $+3$. Currently, *pSenti* knows about 116 emoticons and forty idioms.

3.4.2 Aspect and view extraction

Aspect and view extraction play multiple roles in sentiment analysis. People very often express multiple views in a single review (sometimes even of opposite polarity) about distinct aspects of the same item as a software product or a movie. Therefore, it is

essential for a practical sentiment-analysis system to extract the discussed aspects and the corresponding views from each document with a sentiment. A view is subjective information expressed in relation to an aspect; thus, we include them in the domain-specific sentiment lexicon-induction step. Aspect detection can also help to create a domain-specific signature and so contribute to recognising to which domain messages belong.

The current implementation of *pSenti* uses a simple aspect and view extraction algorithm as follows:

- **Find candidate aspects**. We generate a list of candidate aspects by including frequent nouns and noun phrases identified by the POS tagger. In this step, we also use Named Entity Recognition (NER) information. If a word has a named entity category assigned, we include only *location*, *organisation* and *person* categories. We excluded all words which have an initial sentiment value, as well as stopwords.

- **Find expressed views**. We generate a list of candidates by including adjectives and known sentiment words which occur near an aspect (in the same sentence) but excluding all stopwords and all types of named-entities.

- **Clean-up**. We further remove all candidate aspects or views that occur less than five times and ensure that the same word can be either an aspect or view.

- **Group similar aspects**. If multiple aspects share the same stem, they are assigned to the same aspect group; we also include the phrases in which they occurred.

- **Generate final aspects and views**. The final list includes only the top 100 of the aspect group, the top 100 views, plus the top ten views for each selected aspect.

Another motivation for *pSenti* to emphasise aspect/view extraction is that the domain-specific aspect words will be excluded from the machine-learning step to reduce the dependence on the current topic domain, writing style or time period. For example, in many of the browser category customer reviews, we can observe very negative sentiments towards "Internet Explorer" and "Microsoft", so if we include these words in the machine-learning step, they would be given high negative values. In the bag-of-word learning-based approach (e.g. using SVM as the learning algorithm), "Microsoft" would be in the top list with a strong negative weight of -1.36, and "Firefox" would have a positive weight of $+1.07$. However, these words do not carry any stable or robust sentiment value, and it is purely a coincidence that, at the time of sentiment analysis, Microsoft IE6 had such negative publicity.

After a couple of years, we might find that the sentiment polarity and strength for these aspect words have become entirely different from their current values. That helps

pSenti not only to be less sensitive to topic-domain boundaries but also less sensitive to crossing a time-domain boundary.

Besides, aspect/view extraction allows us to find frequently occurring adjectives (views), which can be used to expand the sentiment lexicon and enables us to perform context-aware sentiment-value estimation for such adjectives within the given aspect. For example, the same word, "large", could have very different sentiment implications in different contexts: the sentiment for a "large monitor" is usually positive, while the sentiment for a "large phone" is probably negative.

3.4.3 Lexicon-based sentiment-detection evaluation

For the first pass of sentiment detection, our system uses the sentiment lexicon constructed using public resources. It is a mixture of various publicly available lexicons, including the Opinion Lexicon compiled by Liu [133], the General Inquirer compiled by Stone et al. [221] and *SentiStrength* by Thelwall et al. [227]. Currently, the sentiment lexicon consists of 7048 sentiment words including words with wildcards. The wildcard character "*" in such words represents a number of any characters or an empty string (e.g. "graceful*" will match words "graceful", "gracefully" and "gracefulness").

Their sentiment values are marked in the range from -3 to $+3$. Based on this sentiment lexicon, we apply the following heuristic linguistic rules to detect sentiments from a text:

- **Negation**. We included both traditional negation words such as "not" and "don't", as well as pattern-based negations such as "stop" + "*vb*-ing", "quit" + "*vb*-ing". Our system also employs an algorithm in which negation could be applied to more distant sentiments. If a negation word could not be attached to sentiment or another known adjective, it is treated as a negative sentiment word with a weight -1.5 and will generate the feature $w_{not-word}$ for the machine-learning algorithm. As part of the processing, we perform various sentence repairs using heuristic rules for more reliable negation detection. For example, the system detects negation words in phrases such as "not just ..." and "not only ... but also", and excludes them as sentiment negations. Besides, it splits words with the "non-" prefix (e.g. the word "non-violent" will be separated into two words, "not violent", in advance).

- **Modifier**. Since words such as "more" and "less" can boost or reduce the sentiment value of their associated sentiment word, they are considered by our sentiment-detection algorithm. Intensifiers increase the sentiment value by several times, whereas diminishers decrease it several times. Currently, we have forty such handcrafted modifiers with their impact value in the range from 0.4x to 2.5x.

The lexicon-based algorithm is presented below:

Algorithm 1 *pSenti* lexicon-based algorithm

Input: Text Document τ, The Sentiment lexicon $\mathscr{L}$
Output: Sentiment strength F_{senti}

```
procedure LEXICONSENTIMENT
    for all w_word ∈ τ do
        s_w ← WORDSENTIMENT(w_word)
        if s_w > 0 then
            w_p ← w_p + |s_w|
        else if s_w < 0 then
            w_n ← w_n + |s_w|
    F_senti = 1/2 (log_2 (Σ w_p + β) − log_2 (Σ w_n + β))
procedure WORDSENTIMENT(w_word)
    if w_word ∈ 𝓛 then
        s_w ← GETSTRENGTH(w_word)        ▷ Get a sentiment value from lexicon
        if ISINVERTED(w_word) then       ▷ Check for a presence of negation
            s_w ← −s_w
        s_w ← MODIFIERS(w_word) × s_w    ▷ Apply sentiment strength modifiers
    else
        if ISINVERTED(w_word) then
            s_w ← C_n
```

3.4.4 Learning-feature extraction

As was already mentioned above, in our proposed model, the learning phase is responsible for the lexicon part of domain adaptation by adjusting sentiment word values and participating in a domain-specific lexicon expansion. The supervised machine-learning algorithm used in our system is the linear SVM implementation from LibSVM[6], with an L2 objective function for optimisation and grid search for parameter tuning. We chose linear SVM since in previous studies [182] it has been shown that it outperforms other popular learning algorithms for sentiment analysis. Another reason for this selection was the observation identified by Mladenić et al. [164] that weights extracted from a linear SVM can be a good indicator of feature importance. A feature weight obtained from a linear model represents a vector coordinate which is orthogonal to the hyperplane, and its direction indicates the predicted class. The magnitude also tells us how informative a feature is for classification and its importance in a data-separation task.

A classic bag-of-words supervised learning approach takes all words as features; however, not all words carry sentiment information. Thus, if we limit features only to

[6]http://www.csie.ntu.edu.tw/cjlin/libsvm/

well-known and potential sentiment words, we would be able to use weights extracted from a linear SVM to learn their importance and impact on a sentiment classification task. In other words, we would discover their domain-specific sentiment values. Due to normalisation and standardisation, feature weights discovered in the training phase are on the same scale, and their interpretation can be mapped back into the lexicon-based part and represent their domain-specific sentiment strength. Other lexicon features, such as mood dimensions and overall lexicon-based sentiment strength, are represented by so-called *lexicon bias*. In Section 3.4.7, we will further validate our model and confirm that linear SVM weights can indeed generate a high-quality domain-specific sentiment lexicon.

To perform domain adaptation the following elements are included:

- **Sentiment words**. The weight of this feature is its frequency in a given document multiplied by its *absolute* sentiment strength. For example, if we have a document with the word "good" (sentiment value $+2$), appearing twice, we would generate the feature w_{good} with a weight of $2 \times 2 = 4$. A similar calculation would be performed for the word "bad" (sentiment value -2). It would generate the feature w_{bad} with a weight of $2 \times 2 = 4$. That makes the learning part responsible for the polarity identification. Such a model is agnostic to an initial lexicon polarity, and thus transferable across other lexicons and easier to interpret by human users.

  In the case of sentiment-value modification, the feature and value generation are slightly more complicated. If the sentiment source has been inverted, we generate a new feature to reflect inversion. In the case of the word "good", the feature $w_{not-good}$ would have a sentiment value of -2 with a feature value of $+2$ (absolute value). A similar situation arises with intensifiers or diminishers (e.g. for the bigram "extremely good", where we have the sentiment word "good" appearing next to one of the strongest intensifiers with $x2.5$ impact value, we would generate the $w_{more-good}$ feature with $+2 \times 2.5 = +5$ as its value). In the case of "sometimes good", we would generate $w_{less-good}$ with a value of $+2 \div 1.5 = +1.33$

- **Other adjectives**. Turney [232] was one of the first to identify that adjectives are among the most likely sentiment candidates. Thus, by including adjectives as learning-phase features, we are performing domain-specific sentiment word discovery. For adjective-based features, we use the term frequency weighting. For example, if the word "large" appears twice, we would have the feature w_{large} with a value of 2.0. In this case, a negation, intensifier or diminisher does not modify a feature's weight but only triggers the generation of a new feature. An instance such as the "not clean" bigram, would generate the feature $w_{not-clean}$.

- **Inverted words**. Inverters are typically good indicators of negative sentiment (e.g. the phrase "not working" does not have any well-known sentiment words in it, yet it expresses a strong negative sentiment). Negative sentiments in our lexicon have their strength in the range from -1 to -3, with most words having a value of -1 or -2. Based on empirical evidence, for inverted words in lexicon-based sentiment analysis, we use a value of -1.5, which is in the middle of this range. To mitigate the empirical bias, in the learning phase we use the term frequency weighting.

- **Lexicon-based sentiment score**. We call this feature *lexicon bias*, which is essential in processing documents which contain sentiment words unseen in the training examples. The fallback sentiment lexicon-based classifier plays an important role in a situation where we might have insufficient labelled training data. In such a case, the sentiment strength of these documents can still be derived using a standard fallback sentiment lexicon and lexicon-based heuristic rules. Using this feature, we are measuring how biased the lexicon-based classifier is in a particular domain.

- **Mood dimensions**. We have an option to include eight mood dimensions (see Listing 3.1), extracted using the NRC sentiment lexicon [166]. For each dimension, as its feature weight, we use the probability of occurrence in a given document.

3.4.5 Sentiment scoring

Most of our results are reported in terms of classification into positive and negative classes. However, the actual output is a real-valued sentiment score in the range of $[-1,+1]$. F_{senti} is calculated using the *log-odds*, also known as the *logit* equation (see Equation (3.1)). As discussed in Section 2.2.1, this equation has the smoothest symmetrical distribution in the range of $[-1,+1]$, is symmetric around zero and is most suitable for representing the proportion of two different sentiment poles. In Equation (3.1), w_p is the sum of positive, and w_n of negative (absolute) sentiment values, and β is a fixed coefficient to prevent ill-defined $log_2(0)$. F_{senti} has upper-bound $+1$ and lower-bound -1 (see Equation (3.2)). If the value is greater than +1.0, it will be reset to $+1.0$, and if it is lower than -1.0, it will be reset to -1.0.

$$F'_{senti} = \frac{1}{2}(\log_2(\sum w_p + \beta) - \log_2(\sum w_n + \beta)) \tag{3.1}$$

$$F_{senti} = \begin{cases} 1, & \text{if } F'_{senti} \geq 1 \\ -1, & \text{if } F'_{senti} \leq -1 \\ F'_{senti}, & \text{otherwise} \end{cases} \tag{3.2}$$

If neither positive nor negative sentiment is detected, our algorithm treats such text as a neutral text and assigns it the sentiment value 0. The sentiment value can be easily transformed into a five-star scale using the simple Equation (3.3).

$$F_{stars} = 2 * F_{senti} + 3 \tag{3.3}$$

It is important to note that the final sentiment calculation also includes fallback sentiment words adjusted by a penalised *lexicon bias* coefficient.

3.4.6 Sentiment measurement example

To illustrate the calculation process, consider the following review as an example:

"After reading very good reviews online, I bought this one for Evolution class. It is a horrible excuse for a new textbook. Do not buy this horrible book unless it is for a middle school student. If the authors think this book has been written for an advanced audience, then I would suggest that anyone interested in learning evolution not attend University of Washington."

To simplify all calculations, we omit normalisation, mood dimensions and fallback sentiment words. All calculation steps are presented below in Table 3.3 and the sentiment lexicon in Table 3.2. The learning-based logic is also presented in Algorithm 2.

Word	Sentiment Value
good	$+2$
horrible	-3
excuse	-1
advance	$+1$
interest	$+2$

Table 3.2. The sentiment lexicon snapshot

- **Lexicon-based sentiment-strength calculation**. The review contains five sentiment words: *"good"* (with boosted sentiment value $+2.0 \times 1.5 = +3$), two occurrences of *"horrible"* (with a sentiment value $-3.0 \times 2 = -6$), *"excuse"* (with a sentiment value -1.0), *"advance"* (with a sentiment value $+1.0$) and

Algorithm 2 *pSenti* learning-based algorithm

Input: Text Document τ, SVM weights ν, The Sentiment lexicon $\mathscr{L}$
Output: Sentiment strength F_{senti}
  procedure LEARNINGSENTIMENT
    for all $w_{feature}, w_{word} \in \tau$ **do**
      if $w_{feature} \in \nu$ **then**
        $s_w \leftarrow \text{GETWEIGTH}(w_{feature})$ ▷ Retrieve an SVM weight
      else if $w_{word} \in \mathscr{L}$ **then**
        $s_w \leftarrow \text{WORDSENTIMENT}(w_{word})$ ▷ Call the *pSenti* lexicon component
        $s_w \leftarrow \text{ADJUSTFALLBACK}(s_w)$ ▷ Apply fallback penalty
      if $s_w > 0$ **then**
        $w_p \leftarrow w_p + |s_w|$
      else if $s_w < 0$ **then**
        $w_n \leftarrow w_n + |s_w|$
    $F_{senti} = \frac{1}{2}(\log_2(\sum w_p + \beta) - \log_2(\sum w_n + \beta))$

"interest" (with a sentiment value +2.0). The review also contains two inverted verbs: *"do not buy"* and *"not attend"*, for which we generate features $w_{not-buy}$ and $w_{not-attend}$. As described in the previous section, all negated words are treated as negative sentiment words and given a weight of -1.5. The sum of all positive sentiment values is $w_p = 3+1+2 = 6$, and the sum of all negative is $w_n = 6+1+1.5+1.5 = 10$. The lexicon-based sentiment value, calculated using Equation (3.1), is -0.387.

- **Learning features extraction**. As it was outlined in the section above, for each sentiment word, we generate a separate feature and use its aggregated sentiment value. For each non-sentiment feature, we will use the term *frequency*. Following the outlined procedure, we would generate the document vector as: $[+3.0, +6.0, +1.0, +1.0, +1.0, +1.0, +2.0, +1.0, -0.387]$ (see Table 3.3). The last value in the vector (-0.387), is the lexicon-based document sentiment value.

- **Learning-based weight discovery**. All documents in a training dataset have to be labelled with a positive or negative label. In this phase, we train linear SVM using a training dataset and extract estimated SVM coefficients from a model. Sample weights are provided in Table 3.3.

- **Learning-based weight adjustment**. To demonstrate how we calculate domain-specific sentiment, we will use the same review. To determine a domain-specific sentiment value for each feature, we will multiply the previously calculated weights by their SVM coefficient (e.g. two occurrences of *"horrible"*, with an original sentiment value of -2 (absolute $+2$), has an overall -1.8 domain-specific sentiment value ($-0.3 \times 2 \times 3 = -1.8$). On the other hand, some sentiment words

such as *"advance"* lost their sentiment value ($+1 \times 0 = 0$). For the last feature, which we also call the "*pSenti bias*", we include the originally calculated lexicon sentiment value (-0.387) adjusted by its SVM weight ($+0.05$) plus the SVM hyperplane bias (-0.12), with a final value $-0.387 \times 0.05 + (-0.12) = -0.138$.

To calculate the domain-specific sentiment, we follow the same procedure as above. First, we calculate $w_p = 0.3 + 0.1 = 0.4$ and $w_n = 1.8 + 0.1 + 0.3 + 0.05 + 0.138 = 2.388$. Finally using *log-odds* (see Equation (3.1)) we calculate the review sentiment value. In Table 3.4 the results of the sentiment calculation are shown, with the final sentiment after adjustment (-1), significantly lower than in the original calculations (-0.368). The final result, transformed into five-star grades using Equation (3.3), is just one star out of a possible five.

	Features								
	good	horrible	new	excuse	not-buy	advance	interest	not-attend	F_{senti}
Lexicon step	$+3$	-6	0	-1	-1.5	$+1$	$+2$	-1.5	-0.387
Learning Vector	$+3$	$+6$	$+1$	$+1$	$+1$	$+1$	$+2$	$+1$	-0.387
SVM Weights	$+0.1$	-0.3	$+0.05$	-0.1	-0.3	0	$+0.025$	-0.05	$+0.05$
Learning adjustment	$+0.3$	-1.8	$+0.05$	-0.1	-0.3	0	$+0.05$	-0.05	-0.138

Table 3.3. Weight adjustment stages

Step	$\sum w_p$	$\sum w_n$	F_{senti}	F_{stars}
Lexicon-based	6	10	-0.368	2.263
Learning-based	0.4	2.388	-1	1

Table 3.4. Sentiment rating calculations

3.4.7 Sentiment lexicon information gain evaluation

In this section, to examine the effectiveness of learning-based lexicon adaptation, we attempt to recreate a sentiment lexicon using the Amazon domain dataset. As a benchmark, we took the sentiment lexicon compiled by Liu [133]. This lexicon is not domain specific, but its primary application was in the Amazon domain, and it is representative enough to allow evaluation of a generated lexicon quality.

To generate a domain-specific lexicon, as described before, we use feature weights extracted from a linear SVM model. Normalisation and standardisation make feature weights extracted from different domains comparable, with the values placed on the same scale. Different features have different information gain, with most of them having values just fractionally above zero. Figure 3.6 shows the top 40 features with the most

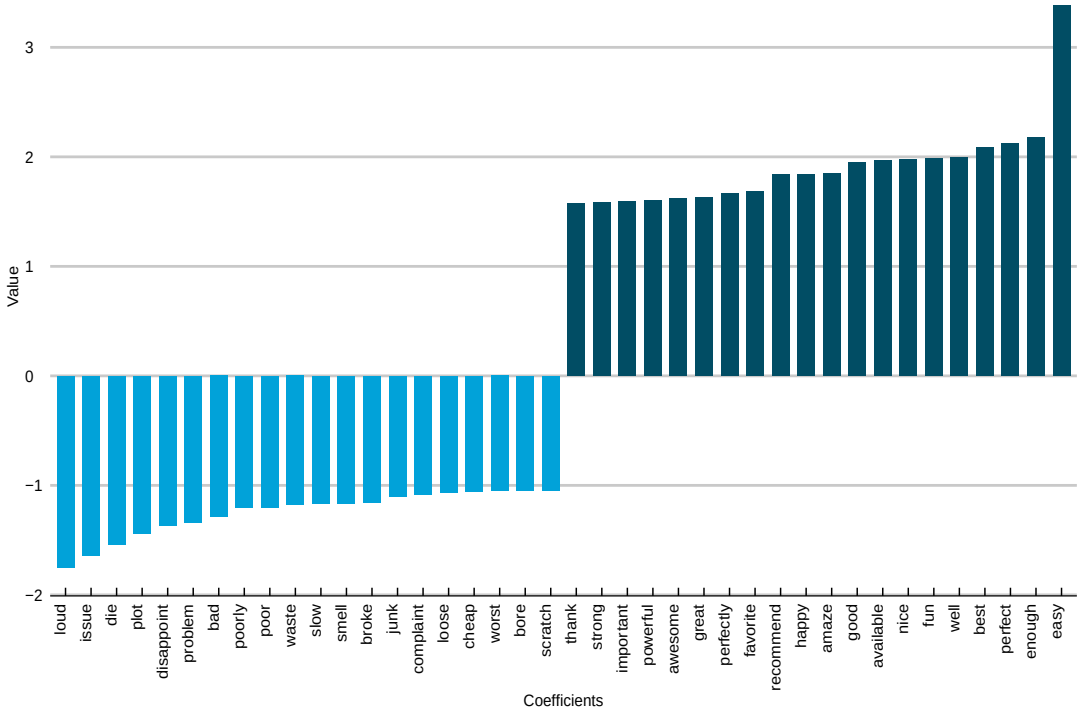


Fig. 3.6: Top 40 SVM feature weights

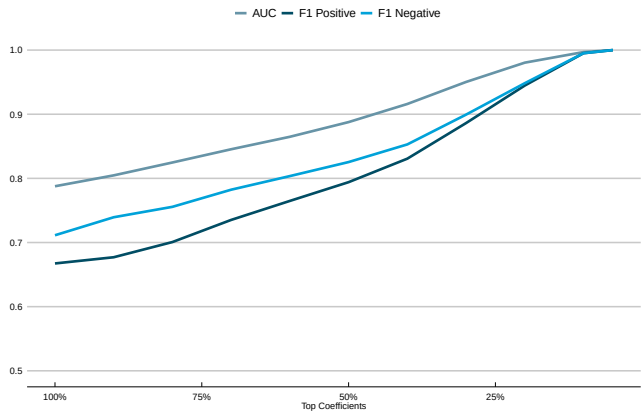


Fig. 3.7: How the performance of lexicon is influenced using feature information gain filtering

substantial information gain. Thus, to evaluate the adapted lexicon quality, we filter it using information gain and present the results in Figure 3.6 and Table 3.5.

The results indicate that even if we include all generated features, we will get a reasonable-quality lexicon with 0.7877 AUC. Imposing a higher cut-off threshold increases lexicon quality and demonstrates that features with the highest information gain are indeed the most accurate. Taking the top 20% of features generates an impressive AUC value of 0.9803 (see Table 3.5). Results also confirm that the generated lexicon

Top	*AUC*	F_1^{Positive}	F_1^{Negative}	Size
100%	0.7877	0.6673	0.7113	2064
90%	0.8046	0.6770	0.7393	1872
80%	0.8247	0.7007	0.7556	1654
70%	0.8453	0.7352	0.7823	1444
60%	0.8648	0.7648	0.8035	1238
50%	0.8877	0.7940	0.8254	1032
40%	0.9159	0.8307	0.8529	826
30%	0.9503	0.8862	0.8993	618
20%	0.9803	0.9447	0.9483	412
10%	0.9968	0.9951	0.9951	206
5%	1.0000	1.0000	1.0000	102

Table 3.5. How the performance of lexicon is influenced using feature information gain filtering

has not only good AUC but also well-balanced F1 scores. Moreover, the actual accuracy is likely to be much higher, as the benchmark lexicon is not domain specific. Manual inspection of the top features (see Figure 3.6) confirms that the strongest sentiment values are indeed identified in words typically associated with customer reviews, such as “loud”, “issue”, “recommend” and “easy”.

3.5 Experimental Results

The *pSenti* system, based on the proposed hybrid approach, is compared with the following baselines:

- *LexiconOnly*: The pure lexicon-based approach using the same sentiment lexicon as *pSenti*;
- *LearningOnly*: The pure learning-based approach using the same learning algorithm (linear SVM) as *pSenti*, with bag-of-words features; and
- *SentiStrength*[7]: a state-of-the-art sentiment-analysis system free for academic research [227, 228].

All experimental results are reported using 10-fold cross-validation.

3.5.1 Same-domain sentiment analysis

Sentiment-polarity classification

In this experiment, we used two datasets: (i) a set of browser reviews, and (ii) a set of professionally edited reviews of various software products, with both datasets containing

[7]http://sentistrength.wlv.ac.uk/

Dataset		*pSenti*	*LexiconOnly*	*LearningOnly*	*SentiStrength*
Software Reviews	Miscellaneous (Editor)	89.64%	79.40%	**90.78%**	64.93%
	Browser (Editor)	86.94%	76.94%	**91.39%**	62.77%
	Browser (Customer)	79.60%	74.50%	**80.54%**	52.25%
	Antivirus (Customer)	78.55%	70.60%	**82.91%**	47.85%
	Video (Customer)	83.55%	75.95%	**85.83%**	52.80%
	Action Games (Customer)	78.75%	71.55%	**82.92%**	58.25%
	Utilities 1 (Customer)	78.80%	73.70%	**82.03%**	50.50%
Movie Reviews	Movies 1	82.30%	66.00%	**86.85%**	60.70%

Table 3.6. The sentiment-polarity classification performance (accuracy) in the standard (single-style) setting.

balanced data. As explored by Straube and Krell [223], F-measure in Information Retrieval (IR) is a suggested metric for imbalanced classes. However, in the case of a balanced binary classification task, accuracy is the most straightforward measure and is sufficient to explain results. Thus, in this section, we report only accuracy, while the other measures are omitted.

As we can see from the results shown in Table 3.6, our algorithm achieved consistent results across all domains. Performance on customer reviews is lower in comparison to professionally prepared editor reviews, but that could be easily explained by the text quality in customer reviews, rating inconsistencies and different writing styles. In our experiment we tried to mimic real-life situations and made the assumption that an author who wrote a review and assigned a rating is objective in his or her valuation; for that reason, we used all the original review ratings extracted from http://www.donwload.com. However, authors are not always consistent in their ratings — they may write a positive review and assign just a 1-star rating. Also, it is not uncommon to find reviews in which customers express opposite sentiment towards competing products. For example, in our dataset, we have a 5-star review with the sentence, “Glad to dump Explorer forever!”. In this review, an author expresses negative sentiment towards “Explorer”, yet, the review has a 5-star rating because it refers to the Firefox browser. Nevertheless, as we can see from Table 3.6, the maximum accuracy achieved by our algorithm was 83.55% (for customer video products), and the lowest accuracy was 78.55% (for antivirus products). In this thesis, to ensure statistical significance in binary sentiment classification tasks, we use the two-tailed binomial test [259]. Tests are performed at 95% significance level. The binomial test is typically used when an experiment has two possible outcomes and as Salzberg [213] has argued, it is well suited for the comparison of two classifiers. Hence, this method is commonly used in text categorisation [259] and to compare different sentiment analysis methods [4]. The binomial test on the *same-domain* sentiment

classification confirmed that *pSenti* is significantly better ($p_{value} = 0.000$) than the *LexiconOnly* and *SentiStrength* baselines.

In all our experiments using *LexiconOnly* and *SentiStrength* we used default configurations without any training or sentiment-value adjustments, and in both cases final sentiment was calculated using Equation (3.1). *SentiStrength* achieved the lowest scores in all categories; such low accuracy can be explained by the fact that in many reviews *SentiStrength* was not able to detect any sentiment or assigned neutral sentiment value.

To compare our algorithm's performance with other well-known methods or to a pure machine-learning implementation, we made use of the Pang and Lee [179] movie reviews dataset. Another reason for using this dataset is that movie reviews are usually more difficult to process, as is clearly illustrated in Figure 3.8c, where the pure lexicon-based approach to sentiment analysis would struggle with customer reviews in the movie domain. One reason for such a poor performance is that many of the movie reviews in the given dataset make extensive use of quotes and plot description. For example, in the sentence "when you get out of jail, you can kill him" the author uses several negative words. However, he is not expressing an opinion but just quoting one of the character's utterances. Such blocks of objective information could be a significant source of sentiment-value distortion, which can be addressed only by processing subjective information blocks. As Pang and Lee [179] have demonstrated in their work, by using such processing it is possible to significantly improve the accuracy of sentiment detection. We have tried to apply a similar (minimum-cut) subjectivity detection algorithm to the datasets we have processed; however, so far, this has not had any positive effect on overall system performance. In this context, we note that subjectivity detection is domain specific and therefore requires a domain-specific training dataset. Nevertheless, as we can see from the results shown in Table 3.6, even without subjectivity detection, our algorithm achieved 82.3% accuracy, a significant improvement over the state-of-the-art lexicon-based method but lower than the SVM unigram implementation.

Sentiment-strength detection

All the results have so far been reported as classification into positive/negative classes, but, as previously highlighted, the actual output is sentiment strength. The last experiment results are shown in Table 3.7. This experiment was conducted to illustrate performance on 5-star classifications. As the results show, for utility product customer reviews, our algorithm achieved, on average, an RMSE of 1.56. The one-versus-one (OVO) strategy is regarded as one of the most effective SVM strategies available [70] for multi-class sentiment analysis. Thus, in the *LearningOnly* case, we used a five-class one-versus-one classification.

Dataset		*pSenti*	*LexiconOnly*	*LearningOnly*	*SentiStrength*
Software Reviews	Utilities 2 (Customer)	1.56	1.50	**1.45**	1.77
Movie Reviews	Movies 2	0.87	0.98	**0.60**	1.13

Table 3.7. The sentiment-strength detection performance (RMSE) in the standard (single-style) setting.

Training	Testing	*pSenti*	*LexiconOnly*	*LearningOnly*	*SentiStrength*
Browser (Customer)	Miscellaneous (Editor)	77.47%	**79.40%**	71.92%	64.93%
Browser (Customer)	Browser (Editor)	**77.78%**	76.94%	75.28%	62.77%
Miscellaneous (Editor)	Browser (Customer)	**77.10%**	74.50%	68.55%	52.25%
Browser (Editor)	Browser (Customer)	**75.90%**	74.50%	65.80%	52.25%

Table 3.8. The sentiment-polarity classification performance (accuracy) in the cross-style setting.

3.5.2 Cross-style sentiment analysis

In this experiment we created a *near-cross-domain* environment, or, in other words, *cross-style*, as both datasets are from the same *topic domain*, yet they use a different writing style. It is important to note that results in this section are reported using only a single domain and a limited set of topics.

There are many writing styles on the Web, with very distinct features and characteristics. In this section, we define two types of writing style: formal and informal expressions. In a formal journalistic style, writers use well-structured sentences and have a certain composition and length requirement. On the other hand, an informal text is typically short, has irregular grammar and spelling problems, and shows creativity in sentiment expressions. It is not uncommon for a sentiment-analysis system to perform well with one style and significantly worse with another [146]. The experimental results in this section illustrate one of the principal advantages of our algorithm (i.e. lower topic and style dependency compared to a pure SVM implementation). To test this, we made use of two datasets, the professional and informal browser reviews within the same domain, and trained with both SVM and *pSenti* on one type of review to evaluate their performance on another.

As expected, compared to *pSenti*, the SVM-based model excelled in its performance, in all cases, on the same dataset. However, when tested on reviews from another type its performance dropped significantly. In particular, an SVM trained on editor reviews achieved only 68.55% accuracy on customer reviews, as shown in Table 3.8. Such a drop in accuracy illustrates the weakness of a pure machine-learning method (i.e. overfitting on the training dataset). In contrast, *pSenti* produced consistent results. For customer reviews, it achieved 77.10%, which is just a small 2.5% drop in accuracy, as shown in

Training	Testing	*pSenti*	*LexiconOnly*	*LearningOnly*	*SentiStrength*
Movies 1	Browser (Customer)	**76.00%**	74.50%	66.95%	52.25%
Movies 1	Utilities 1 (Customer)	**75.30%**	73.70%	65.02%	50.50%
Browser (Customer)	Movies 1	67.70%	66.00%	**67.90%**	60.70%
Utilities 1 (Customer)	Movies 1	67.75%	66.00%	**68.50%**	60.70%

Table 3.9. The sentiment-polarity classification performance (accuracy) in the cross-domain setting.

Table 3.8. This suggests that our mixed algorithm can be trained on one type of reviews and detect sentiments in another type without incurring a significant performance penalty. From the practical point of view, such a system simplifies sentiment processing in less structured social media sources such as Twitter, which usually does not have reliable training data. Moreover, sentiment-strength labelling using professionally prepared text is more reliable, which has more content and is less likely to contain sentiment anomalies. In conjunction with the algorithm's ability to detect the discussed aspects, we can train *pSenti* on different domains and, based on the discussed topic, switch between domain-specific weighting models.

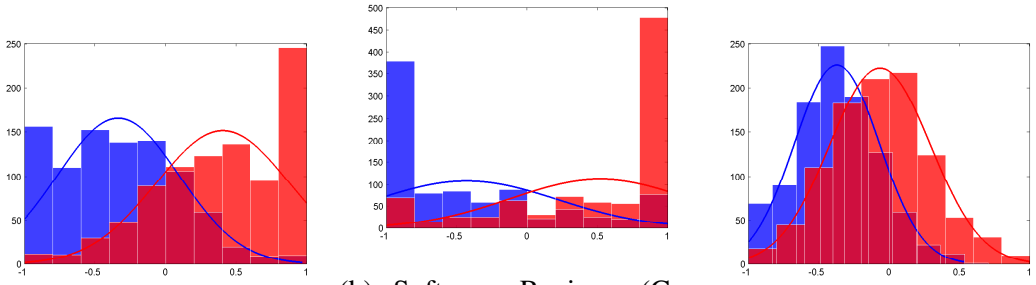
3.5.3 *Distant cross-domain* sentiment analysis

In the final part of our experiments, we analysed various aspects of the system's performance across *distant domain* boundaries. As expected, in all scenarios, *pSenti* was among the best-performing models. On short, informal text processing (see Table 3.9), it was the model with the best overall performance, with only the lexicon-based model producing comparable results.

In the movie reviews dataset, processing performance was significantly lower. We have already highlighted that this domain uses many sentiment words to describe objective information, and, without domain adaptation, all methods failed to take that into account; more specifically, the extensive use of quotes and plot description, as well as the domination of negative sentiment words. Looking at the lexicon-based sentiment-analysis results (see Figure 3.8c), we can see a shift towards the negative sentiment scale, which contributes to lower scores. It would be possible to address these issues by looking into the distribution graph and using output calibration.

3.6 Summary and Conclusions

According to the experimental results, the pure lexicon-based approach achieved its best performance on customer software reviews. As Figure 3.8b shows, in this case, the sentiment values have an evident polarisation into positive and negative clusters (i.e.



(a) Software Reviews (Editor)

(b) Software Reviews (Customer)

(c) Movie Reviews

Fig. 3.8: The lexicon-based sentiment-analysis results.

most reviews have a value of either -1 or $+1$). In editor software reviews, as shown in Figure 3.8a, the values are more scattered, but, overall, the classes are still clearly separable. On the other hand, in movie reviews, as shown in Figure 3.8c, the situation is more complicated. In this case, both classes tightly overlap, and that is reflected in the poor performance of the lexicon-based algorithm. Thus, the machine-learning contribution is especially noticeable, and SVM easily detects and offsets such domain anomalies.

Another important topic is how the inclusion of lexicon-based calculated sentiment into the machine-learning vector influenced the system's performance. As expected, its influence can be directly correlated to the original performance in the given domain; in this case, the greatest effect is in customer reviews, and the smallest is in movie reviews. By removing this feature from the movie classification dataset, accuracy drops by only 0.1%, which shows that this feature is entirely ignored by SVM (this can also be seen in the low SVM weight). In editor reviews, on the other hand, it has significantly higher influence. Removing the lexicon-based sentiment feature would result in a loss of 2% in algorithm accuracy.

We have shown that the sentiment-analysis results produced by our *hybrid* approach are favourable compared to the lexicon-only and learning-only baselines. For both sentiment-polarity classification and sentiment-strength detection, the *pSenti* system based on the proposed hybrid approach achieved high accuracy that is very close to the pure learning-based system and much higher than the pure lexicon-based system. Furthermore, *pSenti* can provide sentiment-analysis results in a structured and readable way by dividing the overall sentiment into aspects (e.g., product features) and their corresponding views. Moreover, it has much better tolerance to the writing style of text, as demonstrated by our cross-style experiments where the system is trained on editor reviews and then tested on customer reviews or vice versa. Compared with a representative state-of-the-art sentiment-analysis system *SentiStrength*, the *pSenti* system is consistently and significantly better. In summary, our proposed *hybrid*

approach combines the best of two worlds: it provides stability, as well as readability, through a carefully designed lexicon and the high accuracy of a powerful supervised learning algorithm. Results also demonstrated that a supervised linear SVM model could be employed to generate a high-quality domain-specific sentiment lexicon.

It would be promising to explore the potential of this approach further, and we will do so in later chapters.

Chapter 4

Domain Lexicon Induction using Word Embedding

4.1 Introduction

Machine-learning algorithms are the dominant approaches in sentiment analysis. Supervised learning algorithms typically deliver much higher accuracy in a sentiment classification than lexicon-based methods. However, lexicons have not entirely lost their attractiveness: they usually are easier to understand and to maintain by non-experts, and can be integrated into learning-based sentiment classifiers [169, 60] and used in other tasks such as aspect detection [169].

In Chapter 3 we introduced *pSenti*, a concept-level sentiment-analysis system that seamlessly integrates lexicon-based and learning-based approaches to acquire adaptive sentiment analysis. In addition to a customisable sentiment lexicon, it also uses shallow NLP techniques such as part-of-speech (POS) tagging, detection of sentiment inverters and modifiers (intensifying and diminishing adverbs). Many lexicon-based based approaches use a pre-compiled out-of-shelf sentiment lexicon [175]. However, to achieve the best results with a lexicon-based model, it is vital to perform either lexicon induction, adaptation or expansion with domain-specific sentiment words. The same word could drastically change its sentiment polarity (and/or strength) if it is used in a different domain. For example, being "small" is likely to be negative for a hotel room but positive for a digital camcorder; being "unexpected" may be a good thing for the ending of a movie but not for the engine of a car; and we will probably enjoy "interesting" books but not necessarily "interesting" food. The domain can be defined not only by the topic of the documents but also by the style of writing and can change over time. For example, the meanings of words such as "gay" and "terrific" would depend on whether the text was written in a historical era or modern times.

The experimental results in Chapter 3 confirmed that the lexicon component made *pSenti* less sensitive to crossing *near-domain* boundaries. Lower sensitivity is a desirable feature of *pSenti*, especially if an underlying sentiment source is noisy, or contains cross-style or near-domain documents. Without domain adaptation, in the lexicon-only mode it performed well in the product review domain; however, in the movie review domain it was behind the accuracy of the supervised learning-based method. In a sense, *pSenti*, in its hybrid mode, uses supervised machine learning to generate a high-quality domain-specific sentiment lexicon. The ability to adapt a general-purpose lexicon or bootstrap a specific lexicon from a target domain with minimal supervision would make the *pSenti* design even more attractive and close the gap in areas with a non-standard sentiment language.

The introduction of modern word-embedding techniques such as *word2vec* [160] and *GloVe* [185] have opened the possibility of new sentiment-analysis methods. Those techniques can learn word co-occurrence information from a large unlabelled text corpus and produce a vector space of several hundred dimensions, with each word being assigned a corresponding vector. The resulting vector space helps the understanding of the semantic relationship between words and allows the grouping of words based on their linguistic similarity. This feature makes word embedding an attractive option in a domain-specific lexicon-induction task. Researchers have proposed various methods to produce sentiment lexicons automatically [98, 58]. However, our experimental results indicate that word embedding is better at capturing sentiment relationships and context information. In their recent work, Hamilton et al. [86] demonstrated that, starting from a small set of seed words and conducting label propagation over the lexical graph derived from the pairwise proximities of word embeddings, they could induce a domain-specific sentiment lexicon comparable to a hand-curated one. Intuitively, the success of their method named SentProp requires a relatively clear separation between sentiment words of opposite polarity in the vector space, which, as we will show later, is not very realistic. Moreover, they have focused on the induction of sentiment lexicons alone, while we are trying to design an end-to-end pipeline sentiment analysis, with domain-specific sentiment lexicon induction as a critical component.

4.2 Contribution

In this chapter, we propose *domain-specific* lexicon induction based on distributed word representations (vectors). This should help to overcome issues with lexicon discovery and adaptation for new domains and improve the lexicon-based system use case. Our proposed contribution is twofold: first, we create a novel *lexicon-induction method*; and second, we integrate it into the previously built *pSenti* sentiment-detection system.

Our investigation indicates that in word embeddings learned from the unlabelled corpus of a given domain, the distributed word representations (vectors) for opposite sentiments form distinct clusters, although those clusters are not transferable across domains. By exploiting such a clustering structure, we would be able to utilise supervised or semi-supervised/transductive learning algorithms to induce a high-quality domain-specific sentiment lexicon from just a few typical sentiment words (as "seeds"). The induced lexicon could be applied directly in a lexicon-based algorithm for sentiment analysis. In other words, the primary motivation for this integration of lexicon-induction and lexicon-based *pSenti* is to create an semi-supervised sentiment method which can effortlessly adapt to new domains, give high accuracy in a sentiment-analysis task, and offer similar rich features to lexicon-based sentiment-analysis approaches.

The source code for our implemented system and the datasets for our experiments have been opened to the research community [1].

The rest of this chapter is organised as follows. In **Section 4.3**, we describe experimental datasets. In **Section 4.4**, we investigate and discuss domain-specific word embeddings, their structure, sentiment cluster visualisation and cross-domain characteristics. In **Section 4.5**, we represent the main stages of our approach and in **Section 4.6** perform two different experiments: lexicon induction from five domains and sentiment analysis in two domains. In **Section 4.7**, we conclude and discuss future work.

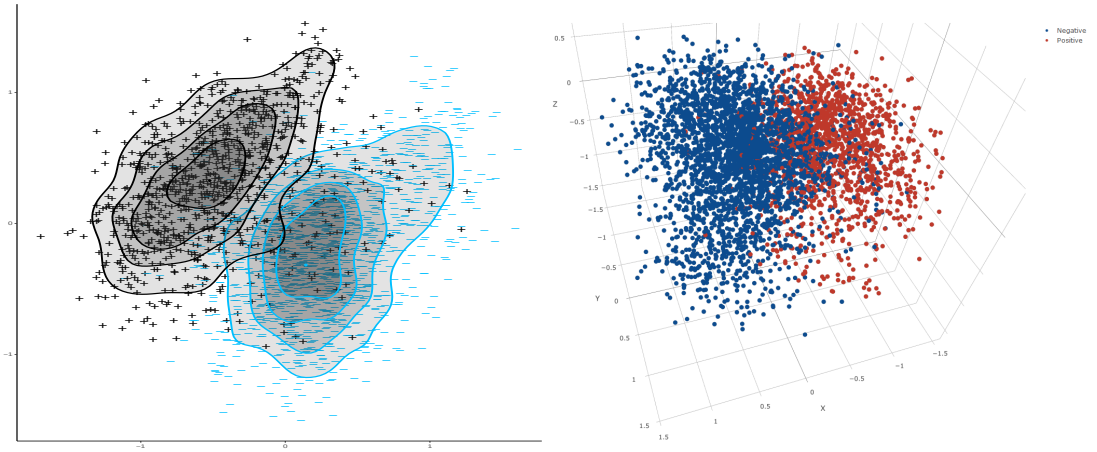
4.3 Datasets

To ensure a fair comparison with the state-of-the-art sentiment lexicon-induction technique SentProp[2] [86], we adopt precisely the same datasets for three domains, together with corresponding publicly available word embeddings, as theirs.

- *Standard English*. We use the 'General Inquirer' lexicon [221] with the-sentiment polarity scores collected by Warriner et al. [241] and Google News word embeddings[3].
- *Twitter*. We use the sentiment lexicon from the SemEval-2015 Task 10E competition [205] and word embeddings constructed by Rothe et al. [207].
- *Finance*. We use the finance-specific sentiment lexicon handcrafted by Hamilton et al. [86], and their word embeddings learned from the financial 8K corpus[4] [124] using an SVD-based method [148].

In Chapter 3 we identified, that lexicon-based systems with a general-purpose lexicon perform poorly in some domains. The difference in performance between lexicon-based and learning-based systems was most significant in processing the movie

[1] https://github.com/AndMu/Unsupervised-Domain-Specific-Sentiment-Analysis
[2] https://github.com/williamleif/socialsent
[3] ttps://code.google.com/archive/p/word2vec/
[4] https://nlp.stanford.edu/pubs/stock-event.html



(a) Sentiment words in 2D plane

(b) Sentiment words in 3D plane

Fig. 4.1: Visualisation of the sentiment words in the Standard-English domain

review dataset. To assess the effectiveness of our proposed method, in the second phase of the experiments we use the following review datasets:

- *IMDB*. We use 50k film reviews in English from IMDB [144] with 25k labelled training documents.
- *Amazon*. We use about 28k product reviews in English across four product categories from Amazon [22, 154] with 8k labelled training documents.

The sentiment lexicon created by Liu [135] is consistently one of the best for review analysis [201], so it is used for both domains.

4.4 Word Embedding

4.4.1 Domain-specific sentiment word embedding

Drawing an analogy to the well-known *cluster hypothesis* in Information Retrieval (IR) [148], here we put forward the cluster hypothesis for sentiment analysis: *words in the same cluster behave similarly with respect to sentiment polarity in a specific domain*. That is to say, we expect positive and negative sentiment words to form distinct clusters, given that they have been represented in an appropriate vector space. To verify this hypothesis, it would be useful to visualise the high-dimensional domain-specific sentiment word vectors in a 2D plane. We have tried various dimensionality reduction techniques, including the t-distributed Stochastic Neighbour Embedding (t-SNE) [145], but we found that simply using the classic Principle Component Analysis (PCA) [20] works very well for this purpose. In all our figures, "+" denotes positive polarity and "-", negative polarity.

Fig. 4.2: A local region of the vector space zoomed in the Standard-English domain

We have found in general that the above cluster hypothesis holds for word embedding within a specific domain. Figure 4.1a shows that in the Standard-English domain, the sentiment words with opposite polarities form two distinct clusters. However, it can also be seen that those two clusters would overlap each other. This is because each word carries not only a sentiment value but also its linguistic and semantic information.

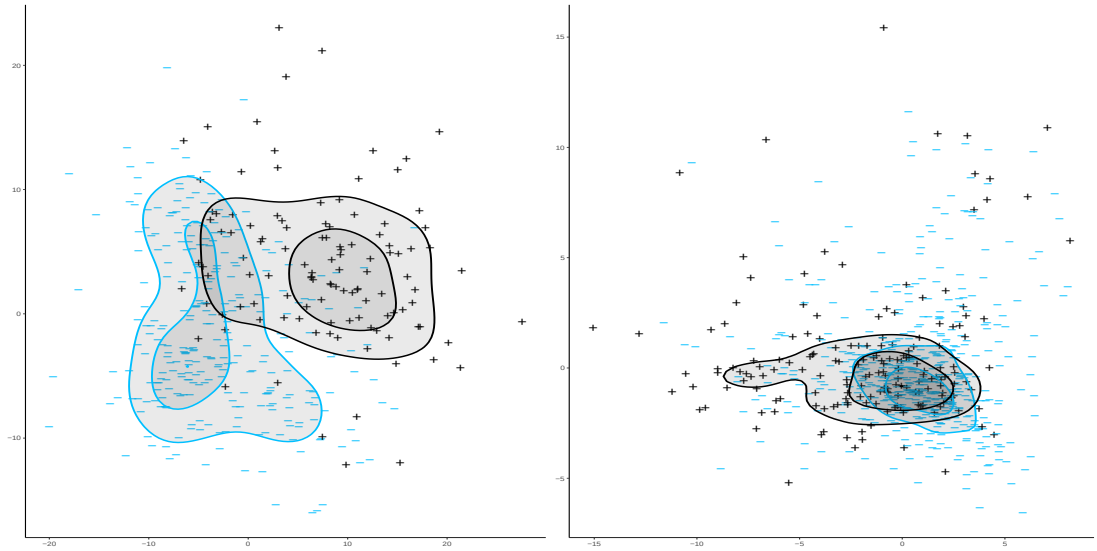
Zooming into one of the word vector space regions (see Figure 4.2) can help us to understand why sentiment words with different polarities could be grouped: 'hail', 'stormy' and 'sunny' are linguistically similar, as they all describe weather conditions, yet they convey very different sentiment values. Moreover, as explained by Plutchik [189], sentiment could be grouped into multiple dimensions such as joy–sadness, anger–fear, trust–disgust and anticipation–surprise. Putting that aside, some sentiment words can be classified as positive or negative depending on the context. These reasons lead to the phenomenon that many sentiment words are in the overlapping noisy region between two clusters in the vector space.

4.4.2 Cross-domain vector space characteristics

On visual inspection of the Finance (see Figure 4.3a) and IMDB (see Figure 4.6a) word-embedding spaces, we can see that their sentiment words with different polarities form distinct clusters that are mostly separable.

However, if we consider the Finance vector space and apply the IMDB sentiment weights (see Figure 4.3b), positive and negative words would be mixed and could not be easily separated.

Positive and negative words typically appear in the same context. For example, we could say "the room is good" and "the room is bad". Both are legitimate sentences; "good" and "bad" have the same context and thus they would result in similar word embedding vectors. However, in our results, we identified that positive and negative sentiment words form their respective clusters. The probable reason for the cluster hypothesis to be true is that, in reality, people tend to use positive sentiment words together much more often than mixing them with negative sentiment words, and vice versa. For example, we would much more often see sentences such as "the room is clean and tidy" than "the room is clean but messy". It is a long-established fact in computational linguistics that words with similar meanings tend to occur near each



(a) In the Finance (same-domain) vector space.(b) In the IMDB (different domain) vector space.

Fig. 4.3: Sentiment words of Finance in the same/different domain vector space.

other [161], and sentiment words are no exception [232]. Moreover, it has been widely observed that online customer reviews are affected by the so-called love-hate, self-selection bias: users tend to rate only products they either love or hate, leading to many more 1-star and 5-star ratings than other (moderate) ratings. If they consider the product just average or so-so, they probably will not bother to leave a review. The polarisation of online customer reviews would also encourage the clustering of sentiment words into opposite polarities.

4.5 Model

Our approach to domain-specific lexicon induction is built on the basis of word embedding — distributed word representations that could be learned from an unlabelled corpus to encode the semantic similarities between words [80].

As shown in Figure 4.4, our approach consists of four stages:

(1) collection of unlabelled domain-specific documents,
(2) extraction of word embeddings from a collected corpus,
(3) domain-specific sentiment lexicon induction, and
(4) sentiment detection using an induced lexicon.

Each stage in our proposed system generates an input to the following step. When we needed to construct word embeddings, we chose to use *word2vec*, the most widely used word-embedding technique, which employs a two-layer neural network [160]. Although any reasonable word-embedding technique could be used in our approach, the choice of embedding method may affect system performance. In fact, previous

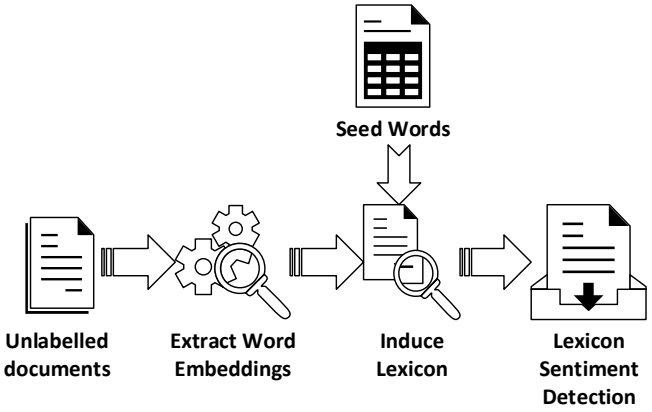


Fig. 4.4: Domain lexicon induction and integration into *pSenti*

studies [207, 47] suggest that *word2vec* usually provides the best word embedding for sentiment-analysis tasks. In our experiments, we use it with a skip-gram window of five words to construct word vectors of 500 dimensions, as recommended by previous studies[5]. Word embedding exploits statistical properties; thus, to create rich semantic relatedness, it is necessary to collect a reasonably sized dataset. As Altszyler et al. [5] found in their study, to achieve good results, *word2vec* requires datasets with around 10 million words. When the corpus size is reduced, the performance of *word2vec* severely decreases.

Many authors [210, 47, 59] have reported a successful application of other word embedding techniques, such as Glove and FastText [24]. FastText is a subword-based learning method, an extension to Word2Vec, which allows capturing more subtle semantic relationships among words. However, in our preliminary experiments, a simpler Word2vec approach delivered better results. That may be impacted by different vocabularies while using different word embedding methods and would require further validation.

The rise in popularity of new language modelling (LM) methods [55, 199] has introduced more advanced techniques, such as contextualised word-embeddings. They can produce embeddings for a word based on the context in which it appears, thus producing different embeddings for each of its occurrences. Although we have not performed any experiments with contextualised word-embeddings, such approaches may improve lexicon induction results or even lead to new insights.

Given word embeddings (2) for a specific domain, we can induce a sentiment lexicon (3) from a few typical sentiment words (as "seeds"). Table 4.1 shows the seed words for five different domains, which are *identical* to those used in Hamilton et al. [86]. The induction of a sentiment lexicon could then be formulated as a simple *word sentiment classification* problem with two classes (positive vs. negative): each word is represented

[5]https://www.kaggle.com/c/word2vec-nlp-tutorial

Corpus	Positive	Negative
Standard English	good, lovely, excellent, fortunate, pleasant, delightful, perfect, loved, love, happy	bad, horrible, poor, unfortunate, unpleasant, disgusting, evil, hated, hate, unhappy
Twitter	love, loved, loves, awesome, nice, amazing, best, fantastic, correct, happy	hate, hated, hates, terrible, nasty, awful, worst, horrible, wrong, sad
Finance	successful, excellent, profit, beneficial, improving, improved, success, gains, positive	negligent, loss, volatile, wrong, losses, damages, bad, litigation, failure, down, negative
IMDB	good, excellent, perfect, happy, interesting, amazing, unforgettable, genius, gifted, incredible	bad, bland, horrible, disgusting, poor, banal, shallow, disappointed, disappointing, lifeless, simplistic, bore
Amazon	IMDB domain seeds (as above) plus positive, fortunate, correct, nice	IMDB domain seeds (as above) plus negative, unfortunate, wrong, terrible, inferior

Table 4.1. The "seeds" for domain-specific sentiment lexicon induction.

as a vector by employing domain-specific word embeddings. The seed words are labelled with their corresponding classes, while all the other words (i.e. "candidates") are unlabelled. The task here is first to learn a classifier from the labelled examples, and then apply it to predict the sentiment polarity of each unlabelled candidate word. The probabilistic outputs of such a word sentiment classifier could be regarded as the measure of confidence in the predicted sentiment polarity. In the end, those candidate words with a high probability of being either positive or negative would be added to the sentiment lexicon. The final induced sentiment lexicon would include both the seed words and the selected candidate words.

As pointed out in Chapter 3, if we consider all words from the given corpus as candidate words, the word sentiment classifier described above tends to assign sentiment values not only to sentiment words but also to product features and *aspects* of the expressed view. An *aspect* is a sentiment target and should not be included in a sentiment lexicon. For example, if many customers do not like the weight of a product, the word classifier may assign a strong negative sentiment value to "weight", yet this is not stable, as the sentiment polarity of a word may be different when a new version of the product is released, or when the customer preference has changed, and, furthermore, it probably does not apply to other products. To avoid this potential issue, it would be necessary to consider only a high-quality list of candidate words, which are likely to be genuine sentiment words. Such a list of candidate words could be obtained directly from general-purpose sentiment lexicons. It is also possible to perform natural language processing on a target domain corpus and extract frequently occurring adjectives or other typical sentiment indicators such as emoticons as candidate words, which is beyond the scope of this chapter.

In a set of experiments, we evaluate and compare our lexicon discovery component with other state-of-the-art techniques. Our results show that the simple SVM-based

model, trained on only a couple of seed words, outperformed all the baseline models and can be a better alternative to more complicated label-propagation methods [86].

For the last step of sentiment-detection, step (4), we de-noise an induced lexicon by applying a cut-off probability threshold for candidate words to enter the induced lexicon and use it in the *pSenti* lexicon-based sentiment-detection mode. We evaluate our final model with Amazon and movie review domains and confirm that an induced lexicon not only improves the lexicon-based system's performance but also that the most improvements are in areas in which the general-purpose lexicon had its worst accuracy.

4.6 Experimental Results

4.6.1 Lexicon Induction

To examine the effectiveness of different machine-learning algorithms for building domain-specific word sentiment classifiers, we attempt to recreate known sentiment lexicons in three domains: *Standard English*, *Twitter*, and *Finance* (see Section 4.3), in the same way as Hamilton et al. [86] did. Put differently, for evaluation we would use a known sentiment lexicon in the corresponding domain as the list of candidate words and see how different machine-learning algorithms would classify those candidate words based on their domain-specific word embeddings. For those lexicons with ternary sentiment classification (positive vs. neutral vs. negative), the class-mass normalisation method [267] used in Hamilton et al. [86] has been applied here to identify the neutral category. The quality of each induced lexicon for a specific domain is evaluated by comparing it with its corresponding known lexicon as the ground-truth, according to the performance metrics, which are *precisely the same* as in Hamilton et al. [86]: Area Under the Receiver-Operating-Characteristic (ROC) Curve (*AUC*) for the binary classifications (ignoring the neutral class, as is common in previous work), macro-averaged F_1 for the ternary classification (positive vs. neutral vs. negative), and Kendall's τ rank correlation coefficient with continuous human-annotated polarity scores. Note that Kendall's τ is not suitable for the Finance domain, as its known sentiment lexicon is only binary. Therefore, our experimental setting and performance measures are all identical to those of Hamilton et al. [86], which ensures the validity of the empirical comparison between our approach and theirs.

In Table 4.2 and 4.3, we compare a number of typical supervised and semi-supervised/transductive learning algorithms for *word sentiment classification* in the context of domain-specific sentiment lexicon induction:

- *k*NN — *k* Nearest Neighbours [88],
- LR — Logistic Regression [88],

	Corpus	Supervised				Semi-Supervised/Transductive				
		*k*NN	LR	SVM_{lin}	SVM_{rbf}	TSVM	S3VM	CPLE	SGT	SentProp
AUC	Standard English	0.892	0.931	0.939	**0.941**	0.901	0.540	0.680	0.852	0.906
	Twitter	0.849	**0.900**	0.895	0.895	0.770	0.521	0.651	0.725	0.860
	Finance	0.711	**0.944**	0.942	0.932	0.665	0.561	0.836	0.725	0.916
τ	Standard English	0.469	0.495	**0.498**	0.495	0.487	0.038	0.162	0.409	0.440
	Twitter	0.490	**0.569**	0.548	0.547	0.522	0.001	0.211	0.437	0.500

Table 4.2. Comparing the induced lexicons with their corresponding known lexicons (ground-truth) according to the ranking of sentiment words measured by *AUC* and Kendall's τ.

- SVM_{lin} — Support Vector Machine with the linear kernel [106],
- SVM_{rbf} — Support Vector Machine with the nonlinear RBF kernel [106],
- TSVM — Transductive Support Vector Machine [107],
- S3VM — Semi-Supervised Support Vector Machine [75],
- CPLE — Contrastive Pessimistic Likelihood Estimation [140],
- SGT — Spectral Graph Transducer [108],
- SentProp — a label-propagation-based classification method proposed for the SocialSent system [86].

The suitable parameter values for SVM are found via a grid search with cross-validation, and the probabilistic outputs are given by Platt scaling [188] if the original learning algorithm does not provide them.

The experimental results shown in Tables 4.2 and 4.3 demonstrate that in almost every single domain, simple linear-model-based supervised learning algorithms (LR and SVM_{lin}) can achieve optimal or near-optimal accuracy for the sentiment lexicon-induction task, and they outperform the state-of-the-art sentiment lexicon-induction method SentProp [86] by a significant margin. We measured statistical significance using the two-tailed binomial test [259] with a confidence level of 95%. Results confirmed that the proposed method produced significantly better results. There does not seem to be any benefit to utilising semi-supervised/transductive learning algorithms (TSVM, S3VM, CPLE, SGT and SentProp). The qualitative analysis of the sentiment lexicons induced by different methods shows that they differ only on those borderline, ambiguous words (such as "soft") residing in the noisy overlapping regions between two clusters in the vector space (see Section 4.4.2). In particular, SentProp is based on label propagation over the lexical graph of words, so it could easily be misled by noisy borderline words when sentiment clusters have considerable overlap with each other, kind of "overfitting" [20]. Furthermore, according to our experiments on the same machine, those simple linear models are more than seventy times faster than SentProp. The speed difference is mainly due to the fact that supervised learning algorithms need to train only on a small number of labelled words ("seeds" in our context) while semi-

	Corpus	*k*NN	LR	SVM$_{lin}$	SVM$_{rbf}$	SentProp
F_1	Standard English	0.647	0.674	0.702	**0.723**	0.604
	Twitter	0.455	0.613	0.616	**0.623**	0.612
	Finance	0.406	0.497	0.549	**0.595**	0.508

Table 4.3. Comparing the induced lexicons with their corresponding known lexicons (ground-truth) according to the classification of sentiment words measured by macro-averaged F_1.

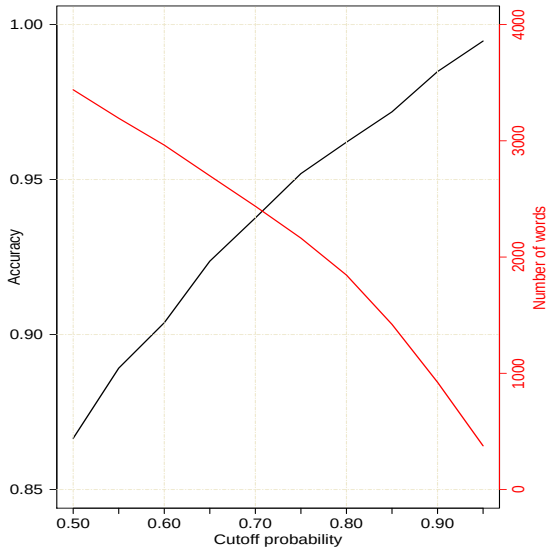
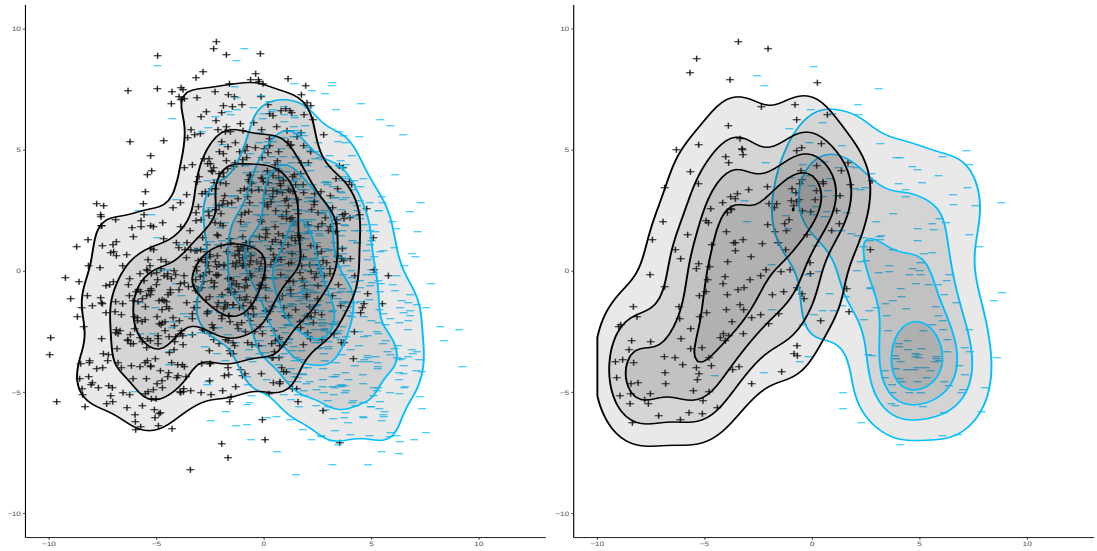


Fig. 4.5: How the accuracy and size of an induced lexicon are influenced by the cut-off probability threshold.

supervised/transductive learning algorithms need to train on not only a small number of labelled words but also a large number of unlabelled words.

It has also been observed in our experiments that there is a typical *precision/recall trade-off* [148] for the automatic induction of semantic lexicons. Assuming that classified candidate words would be added to the lexicon in the descending order of their probabilities (of being either positive or negative) when the lexicon becomes bigger and bigger, it becomes noisier and noisier. Figure 4.5 shows that imposing a higher cut-off probability threshold (for a candidate word to enter the induced lexicon) would decrease the size of the induced lexicon but increase its quality (accuracy). On the one hand, the induced lexicon needs to contain a sufficient number of sentiment words, especially when detecting sentiment from short texts, as a lexicon-based method cannot reasonably classify documents with no (or too few) sentiment words. On the other hand, the noise (misclassified sentiment words) in the induced lexicon would have a detrimental impact on the accuracy of the document sentiment classifier built on top of it. Contrary to most previous work, such as Qiu et al. [196], which tries to expand the lexicon as much as



(a) Original/Full. (b) Filtered.

Fig. 4.6: Sentiment words about movies in the IMDB vector space before/after filtering.

possible and thus maintain a high recall, we would put more emphasis on precision and keep tight control of the lexicon size.

4.6.2 Lexicon integration into the *pSenti* sentiment-analysis model

One of the best ways to evaluate induced lexicon quality is to use it in the actual sentiment classification task. For the experiments here, we use a list of 7866 sentiment candidate words constructed by merging two well-known general-purpose sentiment lexicons that are both publicly available: the 'General Inquirer' [221] and the sentiment lexicon from Liu [133]. This set of candidate words is itself a combined general-purpose sentiment lexicon, so we name it the GI+BL lexicon. Moreover, we set the cut-off probability threshold to the reasonable value of 0.7 in our sentiment lexicon-induction algorithm. Comparing the IMDB vector space, including all the candidate words (see Figure 4.6a), with that including only the high-probability candidate words (see Figure 4.6b), it is clear that the positive and negative sentiment clusters become more clearly separated in the latter.

First, we try the induced sentiment lexicons in the lexicon-based sentiment classifier *pSenti* and use it with Amazon product reviews. Given a sentiment lexicon, *pSenti* can perform not only binary sentiment classification but also ordinal sentiment classification on a five-point scale. To measure the binary classification performance, we use both micro-averaged F_1 (miF_1) and macro-averaged F_1 (maF_1), which are commonly used in text categorisation [259]. To measure the five-point scale classification performance, we use both Cohen's κ coefficient [148] and Root-Mean-Square Error (*RMSE*) [20].

	Lexicon	binary				5-point scale	
		miF_1	maF_1	F_1^{pos}	F_1^{neg}	Cohen's κ	*RMSE*
general-purpose	GI+BL	0.745	0.744	0.764	0.722	0.235	1.325
domain-specific	same domain (Kitchen)	**0.761**	**0.761**	**0.772**	**0.750**	**0.236**	**1.310**
	different domain (Electronics)	0.749	0.749	0.750	0.749	0.215	1.373
	different domain (Video)	0.736	0.735	0.752	0.717	0.206	1.372

Table 4.4. Lexicon-based sentiment classification of Amazon kitchen product reviews.

As the baseline, we use the previously mentioned combined general-purpose sentiment lexicon, GI+BL. As we can see from the results shown in Table 4.4, using the induced lexicon for the target domain would make the lexicon-based sentiment classifier *pSenti* perform better than simply employing an existing general-purpose sentiment lexicon, even though the former is noisier than the latter.

To measure statistical significance, three different tests were carried out. To assess the binary classification, we performed the two-tailed binomial test; to quantify the difference between Cohen's κ coefficients, we performed the two-sample z-test with a confidence interval for the difference [118, 87]; and to validate *RMSE*, we performed the two-sample t-test [48]. To obtain the standard error of κ (SE_κ) we used Equation (4.1), where P is the observed agreement, P_e is the chance agreement and n is the number of observations. All calculations were performed with a confidence level of 95%.

In the case of binary sentiment classification, the difference between the proposed method and the baseline performance was statistically significant ($p_{value} = 0.04$). However, in the case of the five-point scale classification, we observed a mixed picture. According to the z-test, the difference in Cohen's κ was not statistically significant ($SE_\kappa = 0.003$; $z = 0.224$; $p_{value} = 0.822$), yet according to the t-test, the difference in *RMSE* was statistically significant ($t = 14.981$; $p_{value} = 0.000$).

$$SE_\kappa = \sqrt{\frac{P(1-P)}{n(1-P_e)^2}} \tag{4.1}$$

Second, to evaluate the proposed lexicon-induction method, we make comparisons on the IMDB dataset (see Table 4.5). As highlighted previously in Chapter 3, movie reviews is a domain where lexicon-based sentiment classifier with a general-purpose lexicon typically struggles. To better illustrate our method, as an alternative we added a mix of Amazon reviews from all four categories. Movie reviews are also significantly longer than Amazon reviews. Thus, we can consider this experiment as *long* versus *short* text sentiment analysis. As in the previous experiment, *pSenti* with induced domain-specific lexicon outperformed the baseline with general-purpose lexicon and demonstrated the superiority of the proposed method in both short and long texts and was statistically significant in both experiments. Results also show that the most substantial

Method	IMDB		Amazon	
	AUC	F_1	*AUC*	F_1
with existing general-purpose lexicon	0.808	0.705	0.818	0.747
with induced domain-specific lexicon	**0.841**	**0.768**	**0.839**	**0.771**

Table 4.5. *pSenti* sentiment classification.

performance gains are in IMDB, the domain where, previously, lexicon-based *pSenti* had the worst performance. Domain-specific lexicon induction using word embeddings allowed us to close the gap to supervised sentiment-analysis methods further.

4.7 Summary and Conclusions

Can lexicon-based systems improve their performance by learning a domain-specific lexicon? This chapter presents our exploration towards answering the above research question. By capturing word co-occurrence information, word embedding efficiently discovers sentiment in new domains and simplifies the construction of new domain sentiment lexicons. We have also confirmed the advantage of generating domain-specific sentiment lexicons and provided evidence that different domains have different sentiment vector spaces. To the best of our knowledge, such an approach has not been tried previously, and our experimental results demonstrate its superiority over other state-of-art methods.

Specifically, the main contributions of this chapter are as follows.

- We have formulated the cluster hypothesis for sentiment analysis (i.e. words with different sentiment polarities form distinct clusters) and verified that, in general, it holds for word embeddings within a specific domain but not across domains.
- We have demonstrated that a high-quality domain-specific sentiment lexicon can be induced from the word embeddings of that domain together with just a few seed words. Surprisingly, a simple linear-model-based supervised learning algorithm such as Logistic Regression is good enough for this purpose; there is no benefit to utilising nonlinear models or semi-supervised/transductive learning algorithms, due to the noise at the borders of sentiment word clusters. Using those linear models, our system outperforms the state-of-the-art sentiment lexicon-induction method — SentProp [86].
- Experimental results show that the proposed methods in this chapter constitute the semi-supervised sentiment method, which can adapt to new domains and provide rich aspect-level analysis.

Our proposed lexicon-induction method is an opportunity for lexicon-based designs to close the gap to learning-based in domains, which requires specific adaptation. In contrast to the fully supervised hybrid *pSenti* mode form Chapter 3, the proposed

induction method requires minimal supervision, as it is nearly-unsupervised. There is a lot of potential for future work, and it provides a new way to create novel unsupervised methods for sentiment detection and domain adaptation.

Chapter 5

Semi-supervised Sentiment Analysis

5.1 Introduction

There is often the need to perform sentiment analysis in a domain where no labelled documents are available. In a previously unseen domain, there are usually neither a domain-specific lexicon available to employ lexicon-based sentiment classifiers nor a labelled corpus available to train learning-based sentiment classifiers. Although we could make use of a general-purpose off-the-shelf sentiment lexicon or a pre-built sentiment classifier for a different domain, the effectiveness would be inferior to a supervised domain-adaptation method, with an accuracy from mid-70% to as low as 50%. In Chapter 6, the political-sentiment-analysis task, the lexicon-based approach with the general-purpose sentiment lexicon shows inferior performance with 0.584 AUC, just fractionally better than a random selection.

In Chapter 3 we established that a lexicon-based system *pSenti* could be adapted to an underlying domain using supervised machine learning. In some sense, the *hybrid* model is a three-step sentiment-analysis method, in which the first step generates candidate sentiment lexicon, the second induces a domain-specific lexicon and the last uses a lexicon-based mode to calculate final sentiment. In Chapter 4 we demonstrated that a high-quality domain-specific sentiment lexicon could be induced from the word embeddings of that domain together with just a few seed words. Word embeddings are learned from an unlabelled corpus of a given domain, thus *pSenti* from Chapter 4 may be viewed as an nearly-unsupervised sentiment-detection system and the lexicon-induction process as an *semi-supervised* domain-adaptation step.

Recent advances in *deep learning* [123] have brought sentiment analysis to a new height [49, 114, 96]. As was reported in Dai and Le [49], the Long Short-Term Memory (LSTM) [94] Recurrent Neural Network (RNN) can reach or surpass the performance levels of all previous baselines for sentiment classification of documents. One of the many appeals of LSTM is that it can connect previous information to the current context

and allow seamless integration of pre-trained word embeddings as the first (projection) layer of the neural network. Moreover, more recently Radford et al. [197] discovered that LSTM could learn sentiment even though it was trained for an entirely different purpose — to predict the next character in the text of Amazon reviews. They discovered in a multiplicative LSTM with 4096 units, that one of them can behave as the "sentiment unit" and learn the perfect representation of text sentiment.

5.2 Contribution

In order to solve the problems described above, in this chapter we will integrate deep learning into our sentiment-analysis model and explore the possibility of building domain-specific sentiment classifiers with unlabelled documents only. We propose an end-to-end, pipelined *nearly-unsupervised* approach to *domain-specific* sentiment classification of documents for a new domain based on distributed word representations (vectors).

In comparison, in Chapter 3 we adapted to new domains using supervised training. In Chapter 4 we adapted using an semi-supervised lexicon-based model, which was created by exploiting domain-specific lexicon induction using word embedding. Finally, in this chapter, we take this one step further and propose a novel semi-supervised sentiment-analysis method, which almost matches the performance of the supervised method.

Our approach to domain-specific sentiment classification of documents is built on the basis of word embeddings — distributed word representations that could be learned from an unlabelled corpus to encode the semantic similarities between words [80]. Briefly speaking, given a large unlabelled corpus for a new domain, we would first set up the vector space for that domain using word embedding. We would then induce a sentiment lexicon in the discovered vector space and exploit the induced lexicon in a lexicon-based document sentiment classifier to bootstrap a more effective learning-based document sentiment classifier for that domain. Overall, the document sentiment classifier resulting from our nearly-unsupervised approach does not require any labelled document to be trained, and it outperforms the state-of-the-art semi-unsupervised method for document sentiment classification [60].

This was achieved using the *pSenti* system from Chapter 3 in lexicon-only mode, the lexicon-induction method from Chapter 4, and a deep-learning classification model. As in the previous chapter, the induced lexicon can be applied directly in a lexicon-based method for sentiment classification, but a higher performance could be achieved through a two-phase bootstrapping method which uses the induced lexicon to first assign positive/negative sentiment scores to unlabelled documents. Those documents found to

have clear sentiment signals as pseudo-labelled examples were used to train a document sentiment classifier via supervised learning algorithms (such as LSTM).

The source code for our implemented system and the datasets for our experiments are open to the research community [1].

The rest of this chapter is organised as follows. In **Section 5.3**, we describe the experimental datasets. In **Section 5.4**, we present the main stages of our approach, and in **Section 5.5** perform sentiment analysis in short and long texts. In **Section 5.6**, we briefly discuss *cross-domain* and *distant cross-domain* sentiment analysis using the semi-supervised lexicon-based model from Chapter 4 and the proposed deep-learning LSTM model from this chapter. In **Section 5.7**, we conclude and discuss future work.

5.3 Datasets

In this chapter, we performed two sets of experiments: sentiment classification of long and short texts. In the long-text experiment, to facilitate a fair comparison with the state-of-the-art semi-supervised document sentiment classification technique ProbLex-DCM[2] [60], we adopt the following two datasets, which are identical to what they used.

- *IMDB*. We use 50k film reviews in English from IMDB [144] with 25k labelled training documents.
- *Amazon*. We use about 28k product reviews in English across four product categories from Amazon [22, 154] with 25k labelled training documents, which we extracted from similar product categories.

The average length of each document in the IMDB dataset is 241 words and the maximum length of a document is 2,526 words. The Amazon dataset is similar, with an average 123 and maximum 2,801 length.

Source	Granularity	Count
Collected	Unlabelled	2 756 479
SemEval	2-point	26 696
	5-point	43 011

Table 5.1. Short-text sentiment classification dataset

In the second part, the short-text sentiment classification, we use the dataset from SemEval-2017 Task 4 [204]. The dataset consists of tweets annotated for sentiment on 2-point and 5-point scales. All annotations were performed on CrowdFlower. SemEval messages are also tagged with a topic; however, we do not include that information in our experiments.

[1] https://github.com/AndMu/Unsupervised-Domain-Specific-Sentiment-Analysis
[2] https://github.com/jacobeisenstein/probabilistic-lexicon-classification

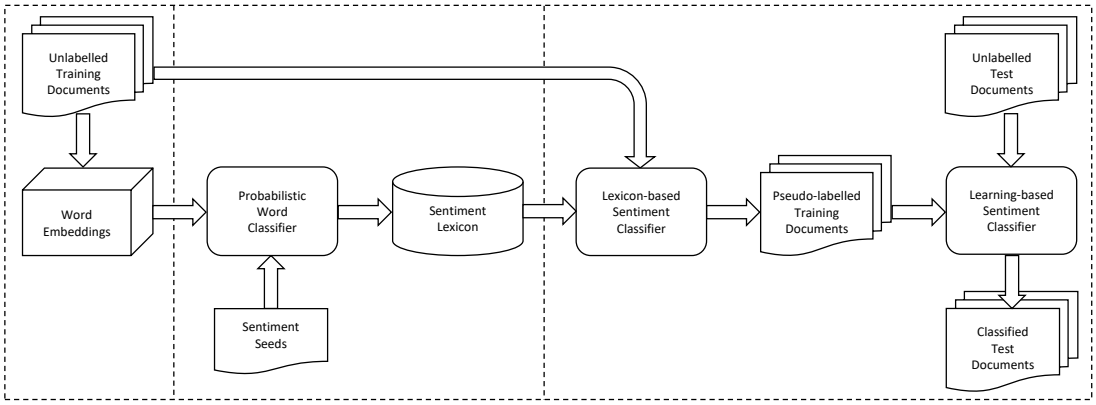


Fig. 5.1: Our *nearly-unsupervised* approach to *domain-specific* sentiment classification.

We used the Twitter API to download additional tweets, along with corresponding user information. All the tweets were automatically filtered for duplicates and also pre-processed by replacing emoticons with their corresponding text representations and encoding URLs as tokens.

5.4 Model

As shown in Figure 5.1, the proposed approach consists of three main components:
(1) domain-specific sentiment word embedding,
(2) domain-specific sentiment lexicon induction, and
(3) domain-specific sentiment classification of documents.

The first and second components in the proposed method are the same as in Chapter 4 and are considered as the first stage. Given word embeddings for a specific domain, we induce a sentiment lexicon from a few typical sentiment words (as "seeds"). We use the same experimental lexicon-induction settings, with emphasis on higher quality and a smaller-sized lexicon. For us, having a small, high-quality sentiment lexicon is affordable, because our proposed approach to document sentiment classification will be able to mitigate the low recall problem of lexicon-based methods by combining them with learning-based methods. Higher cut-off probability threshold (for candidate words to enter the induced lexicon) decreases the size of the induced lexicon but increase its quality (accuracy). In Section 4.5 we gave a more detailed description of the lexicon-induction approach.

Given the induced sentiment lexicon, we propose to use a lexicon-based sentiment classifier to classify unlabelled documents, and then use those classified documents containing at least three sentiment words as *pseudo-labelled* documents to be used later for the training of a learning-based sentiment classifier. The condition of "at least three

sentiment words" is to ensure that only reliably classified documents would be further utilised as training examples.

Why would the two-phase bootstrapping method described above be able to work better than running a lexicon-based sentiment classifier with the induced lexicon? The reason is actually quite similar to that of *pseudo-relevance feedback*, aka blind relevance feedback, in Information Retrieval (IR) [148]. The induced lexicon is analogous to the original keyword query from the user; the pseudo-labelled documents are analogous to the most highly ranked search results with respect to that query. It has been shown in IR research that pseudo-relevance feedback which assumes the most highly ranked search results to be truly relevant and then makes use of them to estimate the relevance of all other documents via supervised learning [101] can often bring performance improvements as real relevance feedback does. It is a similar story here.

Zhang et al. [263] tried to address the low recall problem of lexicon-based methods for Twitter sentiment classification by training a learning-based sentiment classifier using the noisy labels generated by a lexicon-based sentiment classifier [58]. Although the basic idea of their work is similar to what we do in our approach (see Section 5.5), there exist several notable differences. First, they adopted a single general-purpose sentiment lexicon provided by Ding et al. [58] and used it for all domains, while we would induce a different lexicon for each different domain. Consequently, their method could have a relatively large variance in the document sentiment classification performance because of the domain mismatch (e.g. $F_1 = 0.874$ for the "Tangled" tweets and $F_1 = 0.647$ for the "Obama" tweets), whereas our approach would perform quite consistently over different domains. Second, they would need to strip out all the previously known opinion words in their single general-purpose sentiment lexicon from the training documents in order to prevent *training bias* and force their document sentiment classifier to exploit domain-specific features. But doing this would obviously lose the very valuable sentiment signals carried by those opinion words. In contrast, we would be able to utilise all terms in the training documents, including those opinion words appearing in our automatically induced domain-specific lexicons as features when building our document sentiment classifiers. Third, they designed their method specifically for Twitter sentiment classification, while our approach would work not only for short texts such as tweets (see Section 5.5.2) but also for long texts such as customer reviews (see Section 5.5.1). Fourth, they had to use an intermediate step to identify additional opinionated tweets (according to the opinion indicators extracted through the χ^2 test on the results of their lexicon-based sentiment classifier) in order to handle the neutral class, but we would not require that time-consuming step, as we would use the calibrated probabilistic outputs of our document sentiment classifier to detect the neutral class (see Section 5.5.3).

5.5 Experimental Results

In this section, we present the results of two experiments. We tested the proposed approach in two very different domains: in a long-text domain with well-structured text, and with unstructured short text messages from Twitter. In the Twitter domain, we performed two experiments: binary and five-class classification with a neutral sentiment.

5.5.1 Sentiment classification of long texts

First, we try the induced sentiment lexicons in the lexicon-based sentiment classifier *pSenti* to see how good they are. Given a sentiment lexicon, *pSenti* is able to perform not only binary sentiment classification but also ordinal sentiment classification on a five-point scale. To measure binary classification performance, we use both micro-averaged F_1 (miF_1) and macro-averaged F_1 (maF_1), which are commonly used in text categorisation [259]. To measure five-point scale classification performance, we use both Cohen's κ coefficient [148] and also Root-Mean-Square Error (*RMSE*) [20].

In the lexicon baseline, we use the combined general-purpose sentiment lexicon, GI+BL, mentioned previously in Chapter 4, which was constructed by merging two well-known general-purpose sentiment lexicons that are both publicly available: the 'General Inquirer' [221] and the sentiment lexicon from Liu [133]. As we found in the previous chapter, using an adapted sentiment lexicon for a target domain results in better performance than by employing an existing general-purpose sentiment lexicon. Moreover, using the sentiment lexicons induced from the same domain would lead to a much better performance than using the sentiment lexicons induced from a different domain. Thus, to ensure a fair comparison, we also included *pSenti* with an induced domain-specific lexicon.

Second, to evaluate the proposed two-phase bootstrapping method, we make empirical comparisons between the IMDB and Amazon datasets using a number of representative methods for *document sentiment classification*:

- *pSenti* — a concept-level *hybrid* sentiment classifier,
- ProbLex-DCM — a probabilistic lexicon-based classification using the Dirichlet Compound Multinomial (DCM) likelihood to reduce effective counts for repeated words [60],
- SVM$_{lin}$ — Support Vector Machine with the linear kernel [106],
- CNN — Convolutional Neural Network [114], bootstrapped by *pSenti* training data.
- LSTM — Long Short-Term Memory, a Recurrent Neural Network (RNN) that can remember values over arbitrary time intervals [94, 49].

To increase the training speed of the deep-learning algorithms CNN and LSTM with a word-embedding projection layer, we use mini-batch training. To support batching,

Method			IMDB		Amazon	
			AUC	F_1	*AUC*	F_1
Unsupervised	Lexicon-based	*pSenti* with existing general-purpose lexicon	0.808	0.705	0.818	0.747
		pSenti with induced domain-specific lexicon	0.841	0.768	0.839	0.771
		ProbLex-DCM [60]	0.884	0.806	0.836	0.756
	Learning-based	SVM_{lin} trained on pseudo-labelled data	0.863	0.771	0.845	0.763
		CNN trained on pseudo-labelled data	0.879	0.781	0.849	0.773
		LSTM trained on pseudo-labelled data	**0.890**	**0.810**	**0.850**	**0.776**
Supervised	Learning-based	LSTM trained on real labelled data (full size)	0.971	0.912	0.933	0.860
		" (1/2 size)	0.934	0.862	0.913	0.835
		" (1/4 size)	0.892	0.821	0.875	0.795
		" (1/8 size)	0.850	0.746	0.833	0.756
		pSenti trained on real labelled data (full size)	0.928	0.852	0.877	0.803

Table 5.2. Sentiment classification of long texts.

we fix the review size to 500 words, truncating reviews longer than that and padding shorter reviews with null values. As pointed out by Greff et al. [82], the hidden layer size is an important hyperparameter of LSTM: usually, the larger the network, the better the performance but the longer the training time. In our experiments, we used an LSTM network with 400 units on the hidden layer, which is the capacity that a PC with one Nvidia GTX 1080 Ti GPU can afford and a dropout rate [235] of 0.5, which is the most common setting in research literature [220, 96, 47].

As shown in Table 5.2, the two-phase bootstrapping method described above has been demonstrated to be beneficial: the learning-based sentiment classifiers trained on pseudo-labelled data are superior to lexicon-based sentiment classifiers, including the state-of-the-art semi-supervised sentiment classifier ProbLex-DCM [60]. Furthermore, the two-phase bootstrapping method is a general framework which can utilise any lexicon-based sentiment classifier to produce pseudo-labelled data. Therefore, the more sophisticated ProbLex-DCM could also be used instead of *pSenti* in this framework, which is likely to bring us even higher performances. Among the three learning-based sentiment classifiers, LSTM achieved the best performance on both datasets, which is consistent with the observations in other studies such as Dai and Le [49].

Comparing the LSTM-based sentiment classifiers trained on pseudo-labelled and real labelled data, we can also see that using a large number of pseudo-labelled examples could achieve a similar effect as using $25/4 \approx 6$k and $25/8 \approx 3.12$k real labelled examples for IMDB and Amazon, respectively. This suggests that the semi-supervised approach is actually preferable to the supervised approach if there are only a few thousand (or fewer) labelled examples. As expected, in the hybrid mode and trained on real labelled data, *pSenti* performed worse than the supervised LSTM method and

System		*Acc*	F_1
Semi-supervised	Baseline $_{\text{all positive}}$	0.398	0.285
	Baseline $_{\text{all negative}}$	0.602	0.376
	Ours$_{\text{LSTM}}$	**0.804**	**0.795**
Supervised	Worst system	0.412	0.372
	Median system	0.802	0.801
	Best system	0.897	0.890

Table 5.3. Sentiment classification of short texts into two categories — SemEval-2017 Task 4B.

was not far ahead of the proposed semi-supervised method. As in previous chapters, we calculated the two-tailed binomial test [259] with a confidence level of 95%, and it confirmed the proposed method superiority over the method introduced in the previous chapter ($p_{value} = 0.000$).

5.5.2 Sentiment classification of short messages

To evaluate our proposed approach to sentiment classification of short texts, we carried out experiments on the Twitter sentiment classification benchmark dataset from SemEval-2017 Task 4B [204], which classifies 6185 tweets as either positive or negative. In addition to the Twitter-domain seed words listed in Table 4.1, we have also made use of common positive/negative emoticons, which are ubiquitous on Twitter, as additional seeds for the task of sentiment lexicon induction. Note that in all our experiments, we do not use the sentiment labels and the topic information provided in the training data.

Making use of the provided training data and our own unlabelled data collected from Twitter, we have constructed the domain-specific word embeddings, induced the sentiment lexicon, and bootstrapped the pseudo-labelled tweet data to train the binary tweet sentiment classifier. As the learning algorithm, we have chosen LSTM with a hidden layer of 150 units, which would be enough for tweets, as they are quite short (with an average length of only twenty words).

The official performance measures for this short-text sentiment classification task [204] include accuracy (*Acc*) and F_1. Although our approach is nearly-unsupervised (without any reliance on labelled documents), its performance on this benchmark dataset is comparable to that of supervised methods: it would be placed roughly the middle of all the participating systems in this competition (see Table 5.3).

5.5.3 Detecting neutral sentiment

Many real-world applications of sentiment classification (e.g., on social media) are not simply a binary classification task but also involve a neutral category. Although

many lexicon-based sentiment classifiers, including *pSenti*, can detect neutral sentiment, extending the above learning-based sentiment classifier (trained on pseudo-labelled data) to recognise neutral sentiment is challenging. To investigate this issue, we have done experiments on the Twitter sentiment classification benchmark dataset from SemEval-2017 Task 4C [204], which classifies 12 379 tweets into an ordinal five-point scale (-2, -1, 0, $+1$, $+2$), where 0 represents the neutral class.

One common way to handle neutral sentiment is to treat the set of neutral documents as a separate class for the classification algorithm, which is the method advocated by Koppel and Schler [116]. With the pseudo-labelled training examples of three classes (-1: negative, 0: neutral, and $+1$: positive), we tried both standard multi-class classification [97] and ordinal classification [68]. However, neither of them could deliver a decent performance. After carefully inspecting the classification results, we realised that it is very difficult to obtain a set of representative training examples with good coverage for the neutral class. This is because the neutral class is not homogeneous: a document could be neutral because it is equally positive and negative, or because it does not contain any sentiment. In practice, the latter case is more often seen than the former case, and it implies that the absence of sentiment words more often defines the neutral class features rather than their presence, which would be problematic to most supervised learning algorithms.

What we have discovered is that the simple method of identifying neutral documents from the binary sentiment classifier's decision boundary works surprisingly well, as long as the right thresholds are found. Specifically, we take the probabilistic outputs of a binary sentiment classifier, and then put all the documents whose probability is not close to 0 or 1 but in the middle range into the neutral class. It turns out that *probability calibration* [173] is crucially important for this simple method to work. Some supervised learning algorithms for classification can give poor estimates of the class probabilities, and some do not even support probability prediction. For instance, maximum-margin learning algorithms such as SVM focus on hard samples that are close to the decision boundary (the support vectors), which makes their probability prediction biased. The technique of probability calibration allows us to calibrate the probabilities of a given classifier better or to add support for probability prediction. If a classifier is well calibrated, its probabilistic output should be able to be directly interpreted as a confidence level of the prediction. For example, among the documents to which such a calibrated binary classifier gives a probabilistic output close to 0.8, approximately 80% of the documents would actually belong to the positive class.

Using the sigmoid model of Platt [188] with cross-validation on the pseudo-labelled training data, we carry out probability calibration for our LSTM-based binary sentiment classifier. Figure 5.2 shows that the calibrated probability prediction aligns with the true confidence of prediction much better than the raw probability prediction. In this case,

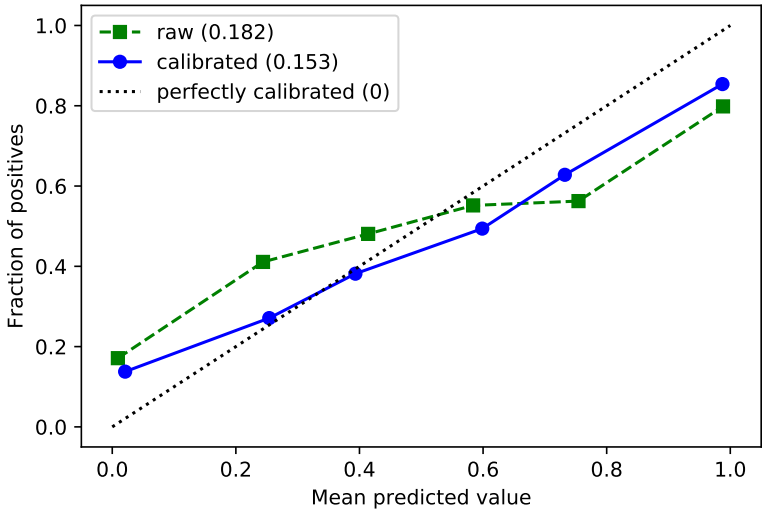
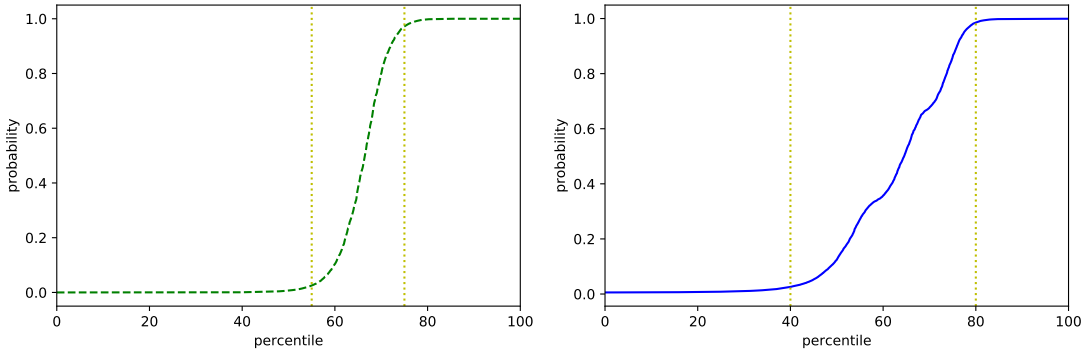


Fig. 5.2: The probability calibration plot of our LSTM-based sentiment classifier on the SemEval-2017 Task 4C dataset.



(a) raw (b) calibrated

Fig. 5.3: The probability curve with a region of intermediate probabilities representing the neutral class.

the Brier loss [33] that measures the mean squared difference between the predicted probability and the actual outcome could be reduced from 0.182 to 0.153 by probability calibration.

If we rank the estimated probabilities of being positive from low to high, the curve of probabilities would be in an "S"-shape, with a distinct middle range where the slope is steeper than the two ends, as shown in Figure 5.3. The documents with their probabilities of being positive in such a middle range should be neutral. Therefore, the two elbow points in the probability curve would make appropriate thresholds for the identification of neutral sentiment, and they could be found automatically by a simple algorithm using the central difference to approximate the second derivative. Let p_{L} and p_{U} denote the identified thresholds ($p_{\mathrm{L}} < p_{\mathrm{U}}$), then we assign class label "-1" to

System		MAE^{μ}	MAE^{M}	miF_1	maF_1
Semi-supervised	Baseline $_{\text{all -2}}$	1.895	2.000	0.006	0.014
	Baseline $_{\text{all -1}}$	0.923	1.400	0.089	0.286
	Baseline $_{\text{all 0}}$	0.525	1.200	0.133	0.500
	Baseline $_{\text{all +1}}$	1.127	1.400	0.063	0.188
	Baseline $_{\text{all +2}}$	2.105	2.000	0.004	0.011
	Lexicon-based	0.939	1.135	0.253	0.189
	Ours$_{\text{LSTM}}$	**0.536**	**0.815**	**0.537**	**0.326**
Supervised	Worst system	0.985	1.325	0.250	0.121
	Median system	0.509	0.823	0.545	0.299
	Best system	0.554	0.481	0.504	0.405

Table 5.4. Sentiment classification of short texts on a five-point scale — SemEval-2017 Task 4C.

all those documents with a probability below p_{L}, "+1" to all those documents with a probability above p_{U}, and "0" to all those documents with a probability within $[p_{\text{L}}, p_{\text{U}}]$.

The official performance measures for this sentiment classification task [204] are MAE^{μ} and MAE^{M}, which stand for micro-averaged and macro-averaged Mean Absolute Error (MAE), respectively. We would also like to report the micro-averaged and macro-averaged F_1 scores which are denoted as miF_1 and maF_1, respectively. As shown in Figure 5.3, the thresholds identified from the raw probability curve are roughly at the 55th and 75th percentiles, which would yield $MAE^{\mu} = 0.632$ and $MAE^{M} = 0.832$; the thresholds identified from the calibrated probability curve are roughly at the 40th and 80th percentiles, which would yield much better scores $MAE^{\mu} = 0.536$ and $MAE^{M} = 0.815$. So, with the aid of probability calibration, our proposed approach would be able to comfortably outperform all the baselines, including the lexicon-based method *pSenti* and compete with the average (median) participating systems (see Table 5.4). Please note that this is not a fair comparison: our approach is at a great disadvantage because (i) it is nearly-unsupervised, without any reliance on labelled documents, whereas all the other systems are supervised; and (ii) it performs only ternary classification, whereas all the other systems perform classification on the full five-point scale.

5.6 Cross-Domain Sentiment Analysis

In the last part of our experiments, we compared the semi-supervised lexicon-based model *pSenti* from Chapter 4 and the proposed deep-learning LSTM model from this chapter in *cross-domain* sentiment analysis.

In this section, we use the terms *near* and *distant cross-domain* to define the relative distance between the domains in question. In Chapter 3 we referred to Amazon and IMDB domains as *distant* domains. However, the distance between Twitter and other

Target	Source	LSTM		$pSenti_4$	
		AUC	F_1	*AUC*	F_1
Amazon	IMDB	**0.940**	**0.865**	0.900	0.812
	Twitter	0.761	0.702	**0.783**	**0.752**
IMDB	Amazon	**0.885**	**0.802**	0.859	0.786
	Twitter	0.582	0.596	**0.781**	**0.713**
Twitter	Amazon	0.736	0.634	**0.787**	**0.719**
	IMDB	0.734	0.658	**0.808**	**0.711**

Table 5.5. *Cross-domain* sentiment classification

domains is even greater. Hence, in this context, we define the distance between Amazon and IMDB as *near* and Twitter as *distant* to all other domains.

As we can see from the results shown in Table 5.5, the model from this chapter performed remarkably well in *cross-domain* sentiment analysis between Amazon and IMDB domains and outperformed *pSenti*. However, its performance dropped significantly once *distant domain* boundaries were crossed. The most significant drop was observed in the transition from IMDB to the Twitter domain. Such a drop in accuracy may illustrate the same weakness of a machine-learning-based approach, which we identified in previous chapters. Alternatively, it could be merely a case of inability to handle a transition from regular to irregular language. More research is needed to identify how to address this issue, as we believe that it may be possible to soften this effect by adding additional text pre-processing and using intermediate text representation to reduce language differences.

5.7 Summary and Conclusions

In this chapter, we explored the possibility of building domain-specific sentiment classifiers with only unlabelled data. Specifically, the main contributions of this chapter are as follows.

- Our proposed semi-supervised domain-adaptation method was significantly better compared to the method from the previous chapter. We have shown that similar to pseudo-relevance feedback, a lexicon-based sentiment classifier could be enhanced by using its outputs as pseudo-labels and employing supervised learning algorithms such as LSTM to train a learning-based sentiment classifier on pseudo-labelled documents. Our end-to-end pipelined approach, which is overall unsupervised (except for the very small set of seed words), works better than the state-of-the-art semi-supervised technique for document sentiment classification — ProbLex-DCM [60], and its performance is at least on a par with an average fully supervised sentiment classifier trained on real labelled data [204].

- We have revealed the crucial importance of probability calibration to the detection of neutral sentiment, which was overlooked in previous studies [116]. With the right thresholds found, neutral documents can be simply identified at the binary sentiment classifier's decision boundary.
- Our results confirmed a deep-learning method superiority over more traditional SVM-based approaches in the domain-adaptation task.
- We have shown that our proposed method performs remarkably well in *cross-domain* sentiment analysis; however, its performance drops significantly once *distant domain* boundaries are crossed.
- The proposed approach may be further improved by introducing the latest state-of-the-art. New LM methods [55, 199] demonstrated an improvement over existing sentiment analysis methods and introduced several new techniques. More specifically, contextualised word-embeddings may further improve both the lexicon induction and the bootstrap components. Besides, the LSTM classifier from the last sentiment classification component may be replaced with a more sophisticated BERT (Bidirectional Encoder Representations from Transformers) [55], using our model to fine-tune BERT for a sentiment classification task.

Chapter 6

Case Studies

6.1 Introduction

In previous chapters, we introduced our approach to adaptive sentiment analysis and its evolution from the *hybrid* approach using a lexicon/learning symbiosis to the semi-supervised sentiment-detection system. In this chapter, to demonstrate a practical application of our sentiment-analysis methods, we present four case studies in three different domains.

Section 6.2 focuses on Amazon product reviews and investigation into sentiment time-series dynamics, which also covers seasonality analysis and design of a *temporal-hybrid* sentiment-analysis system. The resulting system is an enhanced version of *pSenti* which considers past sentiment history to improve its performance. **Section 6.3** examines temporal dependency in Amazon reviews using static and dynamic learning approaches. Our results will confirm temporal dependency existence and the importance of continuous system adaptation to the underlying domain.

Financial market forecasting is one of the most attractive practical applications of sentiment analysis. In **Section 6.4**, we investigate the potential of using sentiment *attitudes* (positive vs. negative) and also sentiment *emotions* extracted from financial news or tweets to help predict stock price movements. We conduct the *Granger causality* test [81] to find out whether sentiment attitudes and sentiment emotions cause stock price changes, or if it is actually the other way around. We carry out extensive experiments to see if a strong baseline model that utilises fifteen technical indicators for market trend prediction can be further enhanced by adding sentiment attitude and/or sentiment emotion features.

In **Section 6.5** we will perform a sentiment analysis of the 2016 US presidential election and examine Donald Trump supporters and opponents. More specifically, we will use geotagged Twitter data to explore questions of *'white flight'* and political-sentiment demographics. Some of our findings contradict those of other researchers,

such as Trump support in highly educated areas and among younger age groups. The experimental results will confirm the importance of domain adaptation, as *pSenti* with the general-purpose lexicon performed just fractionally better than a random selection. We will also investigate two-stage lexicon induction and demonstrate that it could be used to improve domain adaptation further.

Finally, we complete the section with concluding remarks and discuss future directions.

6.2 Amazon Product Reviews Case Study

The Amazon product reviews domain is a favourite target for evaluating sentiment-analysis algorithms. Analysis of customer reviews has many practical applications, as millions of people daily base their purchases on product ratings generated by reviews. We have already made use of Amazon reviews in Chapters 3 to 5, but in this chapter we take a different perspective, looking at the temporal aspect. While considerable research has been done into opinion and sentiment extraction, little work has so far been done to investigate how sentiment changes over time, or on the importance of opinion shift detection. Notably, we study sentiment fluctuations, looking at whether they follow a trend, and whether the information can be utilised to forecast future sentiment.

6.2.1 Datasets

In this section, we analyse a set of product reviews from the Amazon dataset collected by McAuley and Leskovec [154]. It contains reviews from twenty-four different categories, but we use only a small subset of this dataset. Specifically, we use reviews from *electronics, video, kitchen* categories (see Table 6.1). To carry out product-oriented analysis, we selected their top two reviews from each category (see Table 6.2). This selection is justified by the need to have a reasonably sized sample to allow for the statistical analysis and model validation tests.

Type	Count
Electronics	1 194 638
Video	656 559
Kitchen	95 799

Table 6.1. Amazon dataset partitioned by categories

Type	Product	Product Code	Total
Electronics	Sennheiser RS120	B0001FTVEK	3185
Electronics	Creative ZEN 30 GB Player	B000E99YRM	1042
Video	Blade Runner	B001EC2J1G	1731
Video	True Blood: Season 1, Episode 1	B006GM8NXM	1079
Kitchen	Keurig B70 Platinum Brewing System	B000GTR2F6	1427
Kitchen	Crane Adorable 1 Gallon Cool Mist Humidifier	B000GWE2U6	1926

Table 6.2. Amazon dataset partitioned by products

6.2.2 Sentiment time-series analysis

A historical sentiment trend is a form of univariate discrete time series. Such time series can be described using a general model having two components (see Equation (6.1)): a signal or trend part $g(t)$; and a stochastic sequence δ_t, also called noise.

$$S_g = g(t) + \delta_t \tag{6.1}$$

Many researchers [138, 83] employ temporal sentiment analysis to find opinion change points and anomalies. In contrast, our goal is an investigation into *temporal sentiment* dynamics, understanding the mechanisms involved in generating the series, and exploiting them in a future sentiment prediction based on a past value trend. The research will also attempt to uncover *temporal domain dependency* and possible methods to mitigate it.

Traditionally, time-series investigation is focused on decomposition into a trend, seasonal effect and irregular fluctuations [41]. We started with the hypothesis that for short time spans, a sentiment trend for any product should follow a random walk as a stochastic stationary signal. However, over the longer term, it should form an observable and predictable trend that should help to improve future sentiment prediction.

For our experiments, we selected the electronics (see Figure 6.1), video and kitchen Amazon product review categories with an expectation that products in these categories would exhibit different temporal characteristics and popularity patterns. Each group is represented by two products (see Table 6.2), with at least a thousand reviews. Our results identified that products from each of the groups indeed have distinctive characteristics. However, one thing was common to many of the reviews: at the end of each year, there was a spike in the number of published reviews. One explanation for such an activity burst could be the fact that retail has strong seasonal factors falling around the Christmas holiday period. Heavy discounting and sales typically rise sharply in November, with sales peaking around the so-called *Black Friday* and *Cyber Monday*. Products from the kitchen category (see Figure 6.2) are somewhat typical examples of such seasonality.

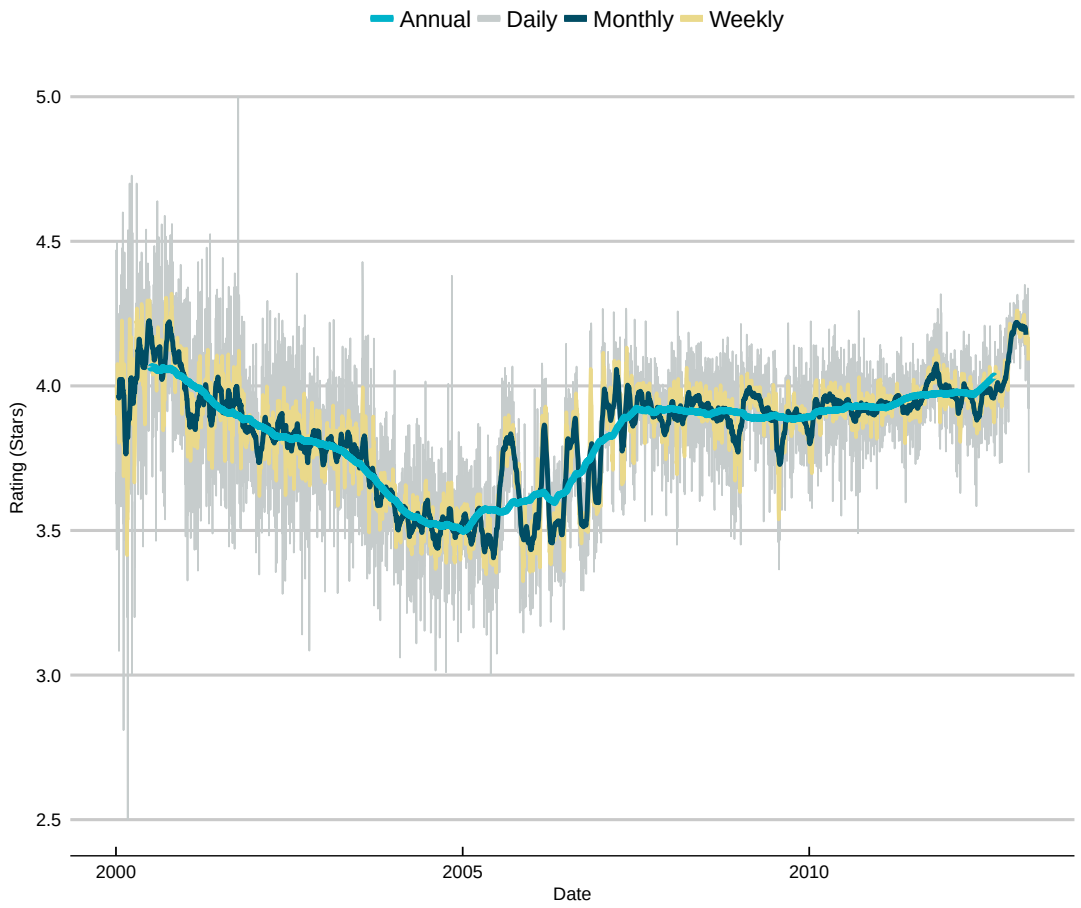
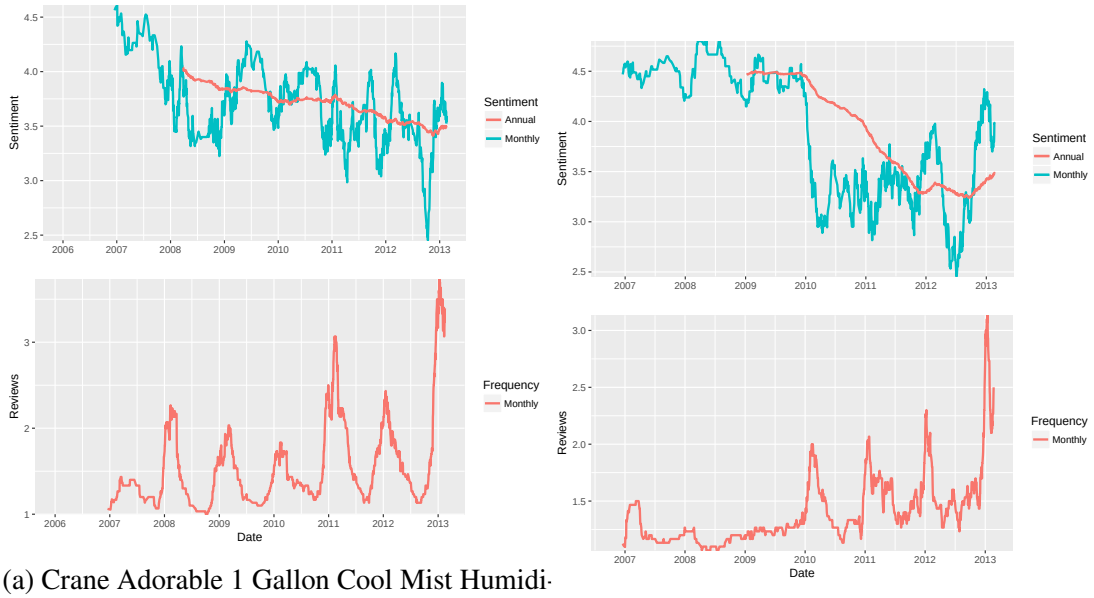


Fig. 6.1: Electronic product sentiment trends

Notably, the rise in the number of reviews can frequently be correlated to a significant change in the average rating. For example, robust seasonal activity can be observed in the *Crane Adorable Humidifier* (see Figure 6.2a), and similar patterns can also be found in the *Keurig B70 Brewing System* (see Figure 6.2b). Both product reviews suffer from average sentiment degradation, with the *Keurig B70* suffering a significant sentiment drop at the end of 2009.

Analysis of video product reviews (see Figure 6.3) uncovers slightly different patterns. In the *Blade Runner* movie review results (see Figure 6.3a) there is a massive spike in published reviews around the end of 2007, with a considerable jump in customer rating. That can be explained by the fact that in 2007 this movie was remastered and re-released as *The Final Cut* version, which caused this anomaly. In the case of the *True Blood* TV series (see Figure 6.3b), popularity peaked on release and slowly diminished over time, yet its sentiment's yearly moving average remained constant.

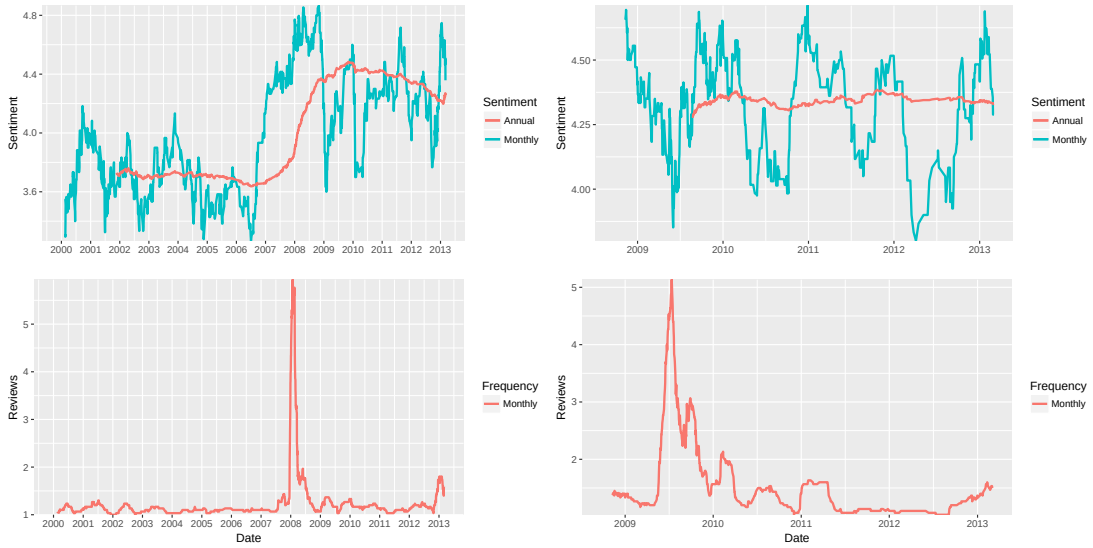
The *Sennheiser RS120 headphones* (see Figure 6.4a) from electronic products, has similar yearly fluctuations to those earlier examined kitchen products, except for the end of 2009, which had very low customer activity. Its yearly moving average



(a) Crane Adorable 1 Gallon Cool Mist Humidifier

(b) Keurig B70 Platinum Brewing System

Fig. 6.2: Kitchen products

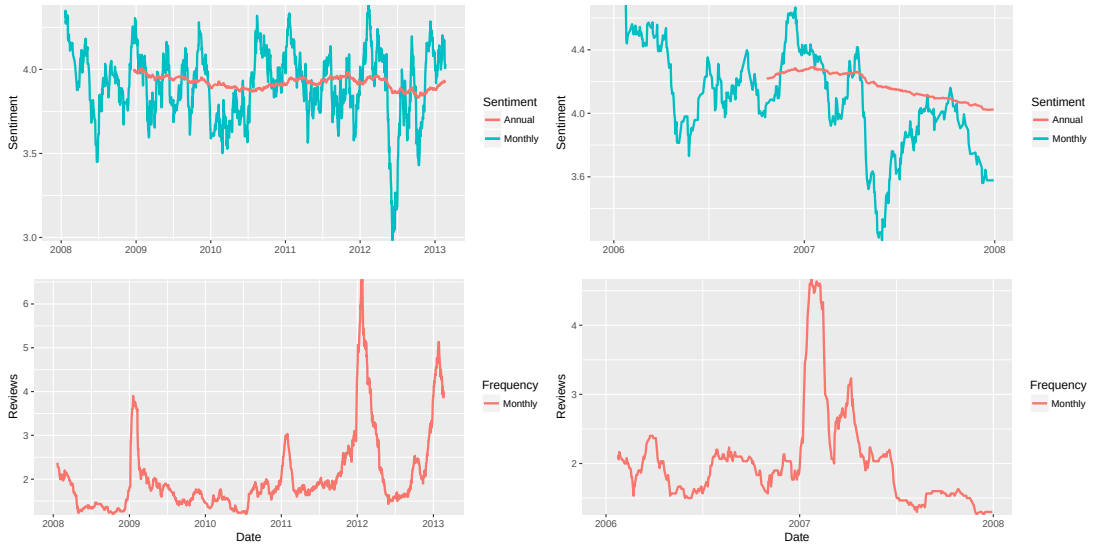


(a) Blade Runner

(b) True Blood: Season 1, Episode 1

Fig. 6.3: Video products

remained constant throughout the covered period, with a significant spike at the end of 2011, followed by a sentiment-strength drop. The *Creative ZEN 30GB player* (see Figure 6.4b) had a very short lifespan, with a single activity spike mid-life, after which customer expectations turned around, followed by sentiment degradation. Only two out of six products over time had a constant rating. All others had a diminishing sentiment, with occasional anomalies, such as with the *Blade Runner* movie. The pattern of diminishing average sentiment was reported in several studies [265, 53] and explained



(a) Sennheiser RS120 headphones (b) Creative ZEN 30GB player

Fig. 6.4: Electronic products

by self-selection bias. It is quite common for the average sentiment in the first couple of months to be significantly higher than that in later months. Based on the evidence we collected, it is also possible that a product can have multiple life-cycles based on such factors as promotions or price changes. Each cycle starts with a significant increase in popularity and average rating, and diminishes within a couple of months.

6.2.3 Sentiment seasonality

As the previous section identified, in most cases sentiment does not have a constant value. Moreover, even if some products have a stable yearly moving average, their monthly trend is typically less stable. Fluctuations in monthly averages can indicate that sentiment has strong seasonal influence. As we already mentioned, average sentiment drop frequently follows an increase in the number of reviews, typically around the end of the year (see Figure 6.4a) and is consistent with self-selection bias theory. Furthermore, looking into all electronic product moving averages (see Figure 6.1), we can see that they are similar to a harmonic signal with mean reversion (yearly moving average) (see Figures 6.2a and 6.4a). To confirm our conjecture regarding the seasonality of sentiment, we performed time-series decomposition analysis as additional experiments.

First, we investigated monthly sentiment fluctuation for each year (see Figure 6.5). Based on our previous analysis, we expected to see the sentiment drop in January. However, that was not the case. Every year followed a different trend and did not correlate with well-known retail cycles.

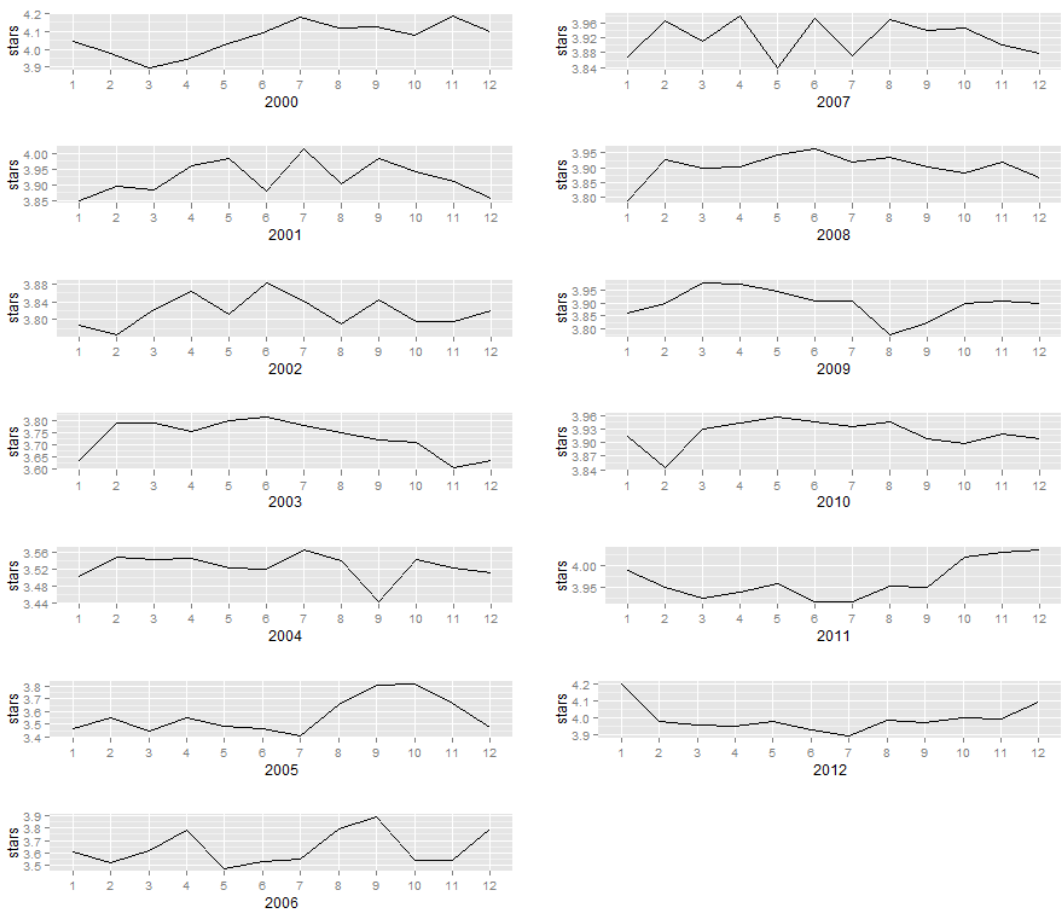


Fig. 6.5: Electronic product sentiment fluctuation by year

Following initial time-series decomposition, we performed a more in-depth seasonality analysis. The seasonal dependency is a general component of the time-series pattern that can be examined using autocorrelation diagrams [251]. Therefore, we performed autocorrelation with a three-year window (see Figure 6.6) and found the presence of a long-lasting and robust correlation, similar to the long-memory process. The long-memory process is a class of stationary processes where the autocorrelations decay much more slowly over time than in the case of the ARMA processes [16]. We also observed strong negative autocorrelation with a significant lag, pointing to the previously mentioned diminishing sentiment and self-selection bias. These findings indicate that the average sentiment should be somewhat predictable.

One of the conventional methods in seasonal pattern analysis is using exponential smoothing. De Livera et al. [52] developed the so-called TBATS model (Trigonometric Seasonal, Box-Cox Transformation, ARMA residuals, Trend and Seasonality), which is among the most wildly used exponential smoothing methods. After we established the correlation, we performed TBATS analysis using models in R, and our experimental results (see Table 6.3) confirmed seasonality presence for five out of the eleven periods.

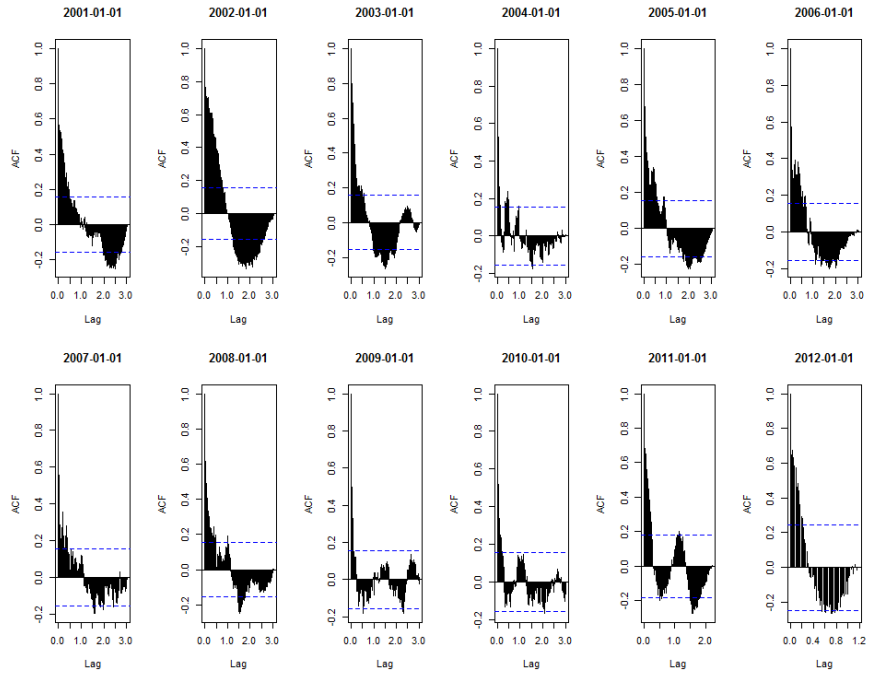


Fig. 6.6: Average rating autocorrelation

Date	Seasonal
2001-01-01 - 2004-01-01	False
2002-01-01 - 2005-01-01	True
2003-01-01 - 2006-01-01	False
2004-01-01 - 2007-01-01	False
2005-01-01 - 2008-01-01	True
2006-01-01 - 2009-01-01	True
2007-01-01 - 2010-01-01	False
2008-01-01 - 2011-01-01	True
2009-01-01 - 2012-01-01	True
2010-01-01 - 2013-01-01	False
2011-01-01 - 2014-01-01	False

Table 6.3. Seasonality TBATS analysis

We do not have a full explanation for why some years produced anomalous results, however as we can see in the example of the *Blade Runner* (see Figure 6.3a), the launch of a new product or its enhancement can introduce a significant anomaly. Also, as we highlighted before, promotions, discounts, fake reviews and other events can create new artificial product cycles or trigger sentiment bias. Amazon is launching new products daily, and vendors start promotions and social marketing events, which can naturally have a significant impact on the shape of seasonal patterns.

6.2.4 *Temporal-hybrid* temporal sentiment analysis with autoregressive sentiment

Even if seasonality investigation failed to pinpoint well-established seasonal patterns, we found that product reviews followed various patterns. One of the most visible and present in every set of product reviews is the mean reversion pattern, also called regression to the mean [243]. It is typically observable at the more granular level in aggregated daily and monthly data. This information may be exploited to predict future sentiment.

A number of researchers have established that machine-learning techniques are more effective than the statistical methods in time-series forecasting [195, 29]. Thus we attempted to create a machine-learning-based model using various sentiment time-series-based features.

The first version of the sentiment-analysis method is based on plain linear regression with only four features (see Equation (6.2)). We call this method *pSenti Regression*, as it is based on regression in *pSenti* results. All features are generated using *pSenti* output: $X_{Predicted}$ is the sentiment prediction for the current review, and the other three are simple moving averages of past sentiment (i.e. X_{Weekly} is weekly, $X_{Monthly}$ is monthly and X_{Daily} is daily). Each of the β coefficients is calculated during the training phase. We selected linear regression rather than a more complex method, as it is the most straightforward proof of concept machine-learning approach. Our aim was not to achieve the best result but instead to confirm that such an approach, in principle, improves sentiment analysis.

$$\hat{Y}_r = \hat{\beta}_1 X_{Predicted} + \hat{\beta}_2 X_{Weekly} + \hat{\beta}_3 X_{Monthly} + \hat{\beta}_4 X_{Daily} \tag{6.2}$$

To validate the proposed model, we executed experiments on the Amazon dataset (see Table 6.1) for each year from 2003 to 2011. In the experiment, we evaluated several sentiment-detection approaches and assessed how their performance compared to the proposed *pSenti Regression* sentiment-detection model. As a baseline, we selected *pSenti* in *lexicon-based* and *learning-based* modes. All *learning-based* approaches use the training dataset, which included reviews from the same domain before the year 2002. The *pSenti Regression* model uses learning-based *pSenti* to calculate both past and current sentiments. In a sense, it uses the past knowledge to help future sentiment learning. Tables 6.4 and 6.5 show that the proposed model outperformed both baselines. Comparing to the baseline, simple regression over past *pSenti* sentiment data produced significantly better results. The best improvement was achieved in the electronics dataset, in some years with up to 16% gains in prediction compared to the

best *learning-based* baseline. Results indicate that integration of the past knowledge improves future sentiment learning.

Year	RMSE				*Hybrid* Improvement vs.		
	Lexicon	*pSenti*	*pSenti Regression*	*Hybrid*	*pSenti*	*pSenti Regression*	Lexicon
2003	1.37	1.19	1.11	**1.11**	6.75%	0.60%	19.00%
2004	1.36	1.23	1.13	**1.12**	9.19%	1.17%	17.67%
2005	1.32	1.23	1.08	**1.06**	13.88%	1.72%	19.27%
2006	1.35	1.26	1.08	**1.07**	14.83%	0.71%	20.59%
2007	1.40	1.31	1.11	**1.10**	15.89%	0.58%	21.34%
2008	1.41	1.31	1.11	**1.10**	15.95%	0.50%	21.68%
2009	1.43	1.27	1.10	**1.09**	14.01%	0.38%	23.48%
2010	1.42	1.19	1.07	**1.07**	10.12%	0.22%	24.96%
2011	1.42	1.21	1.02	**1.01**	16.53%	0.51%	28.58%

Table 6.4. Method comparison on electronic products

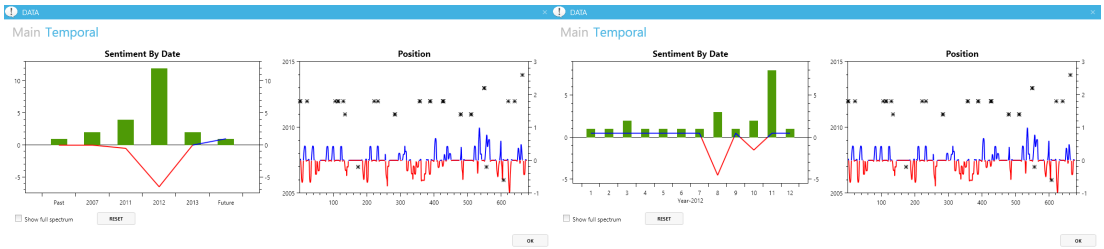
Year	RMSE				*Hybrid* Improvement vs.		
	Lexicon	*pSenti*	*pSenti Regression*	*Hybrid*	*pSenti*	*pSenti Regression*	Lexicon
2003	1.59	1.28	1.28	**1.26**	1.58%	1.71%	20.80%
2004	1.56	1.32	1.30	**1.27**	3.49%	2.53%	18.61%
2005	1.54	1.22	1.20	**1.19**	2.89%	0.86%	22.97%
2006	1.45	1.17	1.18	**1.16**	0.62%	1.52%	19.99%
2007	1.45	1.21	1.20	**1.18**	2.44%	1.67%	18.29%
2008	1.48	1.32	1.30	**1.27**	3.96%	2.66%	13.95%
2009	1.46	1.27	1.26	**1.24**	2.49%	2.14%	15.16%
2010	1.43	1.22	1.19	**1.17**	4.59%	1.58%	18.11%
2011	1.37	1.18	1.08	**1.04**	11.41%	3.53%	23.92%

Table 6.5. Method comparison on video products

Following the improvement, we expanded *pSenti Regression* and created the so-called *temporal-hybrid* model with eleven additional features as in Equation (6.3), making use of eight Plutchik [189] mood dimensions (anger, fear, sadness, disgust, surprise, anticipation, trust, and joy) and four sentiment temporal orientations. Mood dimensions were extracted using the NRC sentiment lexicon [166] and calculated as a relative frequency.

$$\hat{Y}_h = \hat{Y}_r + \hat{\beta}_1 X_{Present} + \hat{\beta}_2 X_{Future} + \hat{\beta}_3 X_{Past} + \hat{\beta}_4 X_{Anger} + \hat{\beta}_5 X_{Anticipation} + \hat{\beta}_6 X_{Disgust} + \hat{\beta}_7 X_{Fear} + \hat{\beta}_8 X_{Joy} + \hat{\beta}_9 X_{Sadness} + \hat{\beta}_{10} X_{Surprise} + \hat{\beta}_{11} X_{Trust} \tag{6.3}$$

In this experiment *pSenti* sentiment output is transformed from a single dimension into a multi-dimension output. We tag each sentence part with one of four time dimensions (past/present/future/unknown) and calculate their sentiment strength. Temporal orientation is resolved using two different methods. Using the first method we identify



(a) Yearly granularity (b) Monthly granularity

Fig. 6.7: *pSenti* article temporal sentiment analysis

temporal expressions using the *SuTime* temporal tagger [40]. *SuTime* is a rule-based tagger built on regular-expression patterns to recognise and normalise temporal expressions in English text in the form of TIMEX3 tags. TIMEX3 is part of the TimeML annotation language [193] for marking up events, times and their temporal relations in documents. It recognises both relative times, such as next Month, as well as absolute times, such as 12 January 2000. Relative time can be transformed into absolute time using the underlying document creation time. As an example, in the sentence 'I hope next year they will release an improved version' we would identify *'next year'* as a future pointer with a positive sentiment *'improved'*. Such an occurrence would generate a positive output for the *'future-sentiment'* feature.

Temporal detection seamlessly integrates into the *pSenti* output and illustrates sentiment dynamics across the time axis. In Figure 6.7a we can see temporal sentiment for the financial news article, extracted as part of Section 6.4 experiments. In the sample article, most of the negative sentiments were expressed in sentences referring to the year 2012, the date on which the article was published, with some positive sentiment pointing to the future. The application can handle almost any time granularity level. As an example Figure 6.7b illustrates sentiment distribution with monthly granularity. Another method, which we employ in the temporal context, is to identify sentence tense and use it to decide which sentiment–time dimension should be assigned to the part.

Incorporation of additional features (see Equation (6.3)) further improves performance (i.e. the last method outperformed all other methods), but by a small margin only (see Tables 6.4 and 6.5). The additional features proved more useful for sentiment analysis of electronic product reviews, most likely due to the specific domain differences. As an example, they also have different review sizes: video product reviews have an average length of eighty-nine words, and electronic reviews are shorter at sixty-six words on average. Our results indicate that due to its simple design and similar performance, a regression over past values is preferable to the hybrid model. It can be integrated into any sentiment-analysis method and exploit historical data to improve sentiment-analysis performance.

6.3 Temporal Dependency

As we already covered in Chapter 2, more specifically in Section 2.3, most sentiment-analysis approaches perform well only if targeted at a specific domain, and they suffer significant performance loss once domain boundaries are crossed. There are three main types of sentiment boundaries: *style*, *domain* and *temporal*. Both *style* and *domain* dependencies were covered in previous chapters, and in this section we expand our investigation by conducting the *temporal* domain dependency experiment.

For this experiment, we looked at Amazon's electronics and video product reviews dataset (see Table 6.1). This dataset covers a significant period, starting from the year 2000 and ending with 2012, and concentrates on consumer products. A lexical word meaning can change over time, but this change requires a significant period to happen. We hope that twelve years is a long enough span for that process to happen.

The dataset was split into two parts: reviews before 2003, and reviews after 2003. All experiments on this dataset we performed using learning-based *pSenti* in two different modes. The '*static*' system was trained once using the same domain reviews before the year 2003. The '*dynamic*' was re-trained for each run using the previous two years' reviews. A two-year batch was selected to make sure that only recent reviews participated in the training, and that its size would be comparable to the *static* system and large enough for reliable training. Natural expectations were that, due to the nature of the underlying domain, there should not be a significant drift in sentiment definition, and that the performance of both *static* and *dynamic* systems should be similar.

Results in Table 6.6 show that in the case of electronic reviews, each following year the gap between the two models increases, with the most significant jump in the final 2011 step. In the second experiment, using the video reviews dataset, the performance of both models was very similar up until the end, and only at the last two stages did the dynamic model significantly improve its performance. It is possible that the jump observed in 2011 was due to some significant change in the underlying sentiment lexicon or just a one-off occurrence, as the static system performance was constant over all periods, and we did not observe any drops in its performance. It is difficult to draw a conclusion from our results; more experiments are needed. Though it does look that temporal dependencies can indeed be observed in Amazon reviews, yet it requires a significant period to detect drift in sentiment values and is not necessarily observed in all products.

6.4 Market Sentiment Case Study

Many portfolio managers and traders are using so-called *'Trade the news'* strategy, which is broadly divided into two main categories: periodic trading and unexpected

Year	Electronics			Video		
	Dynamic	Static	Improvement	Dynamic	Static	Improvement
2003	1.11	1.11	0%	1.26	1.26	0%
2004	**1.12**	1.12	0.25%	**1.27**	1.29	1.49%
2005	**1.06**	1.08	1.36%	**1.19**	1.21	1.51%
2006	**1.07**	1.10	2.82%	1.16	**1.14**	-1.28%
2007	**1.10**	1.13	2.95%	**1.18**	1.19	0.63%
2008	**1.10**	1.14	3.09%	**1.27**	1.28	1.07%
2009	**1.09**	1.13	3.50%	**1.24**	1.24	0.20%
2010	**1.07**	1.12	5.10%	**1.17**	1.20	2.66%
2011	**1.01**	1.12	9.37%	**1.04**	1.16	10.12%

Table 6.6. Dynamic vs. static sentiment analysis (RMSE)

news trading [187]. Periodic news are issued at regular intervals and usually contain general economic data, such as interest rate announcements and company-specific reports (e.g. quarterly earnings). On the other hand, unexpected news is typically related to some adverse developments in the world or economy, or it could be company specific. Examples of such news might be a terrorist attack, Brexit or the BP oil spill. In the case of periodic news, most traders follow a set of standard trades to hedge their portfolios for possible outcomes. It is interesting to note that, using these strategies, good news is not always a signal to buy, and it is quite common that shares drop, sometimes significantly, after positive news [125]. Too high expectations can be followed by a disappointment, even if an announcement is positive.

Market sentiment is one of the most significant drivers in *bull* and *bear* runs. Thus, the capability to detect sudden shifts could provide a competitive advantage over other market participants. Detecting too high or too low expectations before periodic announcements, as well as monitoring for unexpected news, can be processed by an automatic sentiment- analysis system and classified by machine-learning models. In this section, we describe our investigation into the relationship between investors' sentiment and stock market prices.

Throughout this section, we use the term "sentiment" to describe all kinds of affective states [186, 244], and we draw a distinction between sentiment *attitudes* and sentiment *emotions*, following the typology proposed by Scherer [216]. By attitude, we mean the narrow sense of sentiment (as in most research papers on sentiment analysis) — whether people are `positive` or `negative` about something. By emotion, we mean the eight "basic emotions" in four opposing pairs: — `joy-sadness`, `anger-fear`, `trust-disgust`, and `anticipation-surprise`, as identified by Plutchik [189].

Twitter market sentiment analysis is also related to the problem of *stance detection* (SD) [218]. Most of the existing research on SD is focused on the area of politics [119,

120, 226]. Financial market participants also often express strong stances towards particular stocks (which can be divided into the so-called "bulls" and "bears"). However, there are non-trivial differences between political sentiment and market sentiment, as the financial market is usually more cyclical and dynamic, has different sentiment drivers, and can be impacted by various external factors (e.g., company performances and geopolitical events). Moreover, market sentiment extracted from news articles rather than social media would exhibit different characteristics: the former is less about the authors' stance and more about the facts and interpretation of events in a significantly richer context.

In this section, we aim to re-examine the application of sentiment analysis in the financial domain. Specifically, we try to answer the following research question:

Can market sentiment really help to predict stock price movements?

Although our intuition and experience both tell us that sentiment and price are correlated, it is not clear which is the cause and which is the effect. Furthermore, we also have little idea of what exact types of sentiment are really relevant.

The source code for our implemented market-prediction system is open to the research community[1].

6.4.1 Datasets

To obtain relevant sentiment signals, we have collected three *Financial Times* (FT)[2] datasets covering different time periods (see Table 6.7). During each period, we collected all daily published articles and extracted the article text, the authors' and company names, and the time stamp of the article. Collected FT articles have an average length of 626 words (see Figure 6.8).

In addition, from Kaggle[3] we have obtained a large set of historical news headlines from Reddit's WorldNews Channel (RWNC): for each date in the time period we picked the top twenty-five headlines ranked by Reddit users' votes. They have an average length of eighteen words and can be illustrated by the following example:

"Four oil giants to return to Iraq: Exxon Mobil, Shell, Total and BP"

Moreover, we have also gathered from Twitter a large collection of financial tweets which contain in their text one or more "cashtags". A cashtag is simply a '$' sign followed by a stock symbol (ticker). For example, the cashtag for the company Apple

[1]https://github.com/AndMu/Market-Wisdom
[2]http://www.ft.com
[3]https://www.kaggle.com/rootuser/worldnews-on-reddit


FINANCIAL TIMES myFT

Opinion Markets Insight + Add to myFT

Oil price rally presents a tricky predicament for Opec

The producers' cartel is unlikely to back away from its strategy of production cuts

With oil prices up by almost one-third since the start of the year, questions are once again arising over when Opec will be able to declare victory and refocus on securing market share by restoring production. After all, for many publicly traded energy companies, a price of about $60 a barrel is enough to generate decent returns, given that they used the oil collapse of 2014-15 as a catalyst to rationalise spending, cut waste and improve efficiency.

Fig. 6.8: The FT article snapshot

Source	From	To	Count
Financial Times I	2011-04-01	2011-12-25	11 978
Financial Times II	2014-04-01	2014-10-26	9731
Financial Times III	2014-10-26	2015-03-08	6037
Reddit	2008-06-08	2016-07-01	76 600
Twitter	2014-05-01	2015-02-01	1 145 784

Table 6.7. Financial market datasets used in our experiments.

Inc., whose ticker is `AAPL` on the stock market, would be `$AAPL`. Here, we have collected only the tweets mentioning stocks from the S&P 500 index.

For the stock price data, we have used the *end of day* (EOD) adjusted close price. In our experiments, we have focused on several representative companies, Apple (AAPL), Google (GOOGL), Hewlett-Packard (HPQ), and JPMorgan Chase & Co. (JPM), a couple of the most liquid FX currency pairs, EUR/USD and GBP/USD and the Dow Jones Industrial Average (DJIA) index. All financial market data was acquired from public datasets published by Quandl[4], Kaggle[5], and Bloomberg[6].

6.4.2 Causality

To verify whether market sentiments can indeed be useful for predicting stock price movements, we started the investigation with a *Granger causality* test [81], which is a time-series data-driven method for identifying causality based on a statistical hypothesis test that determines whether one time series is instrumental in forecasting the other. The Granger causality test has been widely accepted in econometrics as a technique to

[4]https://www.quandl.com
[5]https://www.kaggle.com
[6]https://www.bloomberg.com/

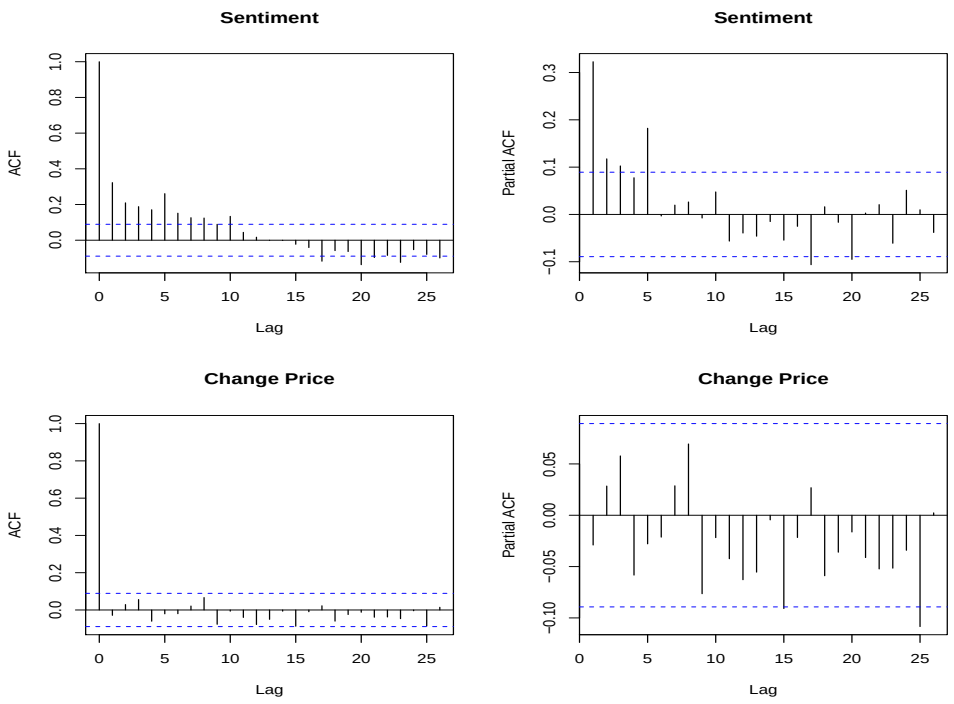


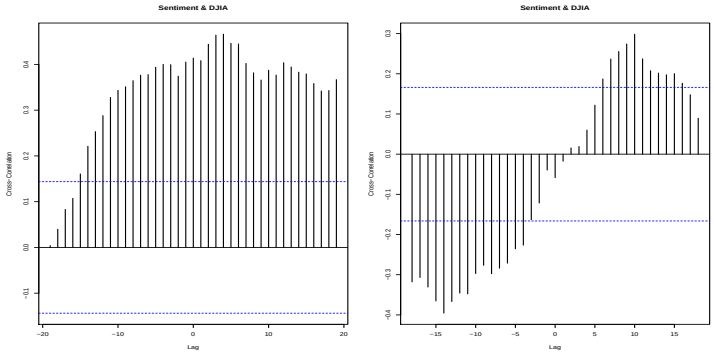
Fig. 6.9: Stationary analysis for DJIA close prices on the FT I dataset.

discover causality in time-series data. In the sense of Granger causality, x is a cause of y if it is instrumental in forecasting y, where 'instrumental' means that x can be used to increase the accuracy of y's prediction compared with considering only the past values of y itself. Essentially, a Granger causality test is a *null hypothesis significance test* (NHST): the null hypothesis is that the lagged x-values do not explain the variation in y. If the p-value given by the test is less than 0.10, we would be able to reject the null hypothesis and claim that x indeed Granger-causes y.

Through our experiments, we try to find the answers to two questions: *Does market sentiment cause changes in stock price?*, and conversely, *Does stock price cause changes in market sentiment?*.

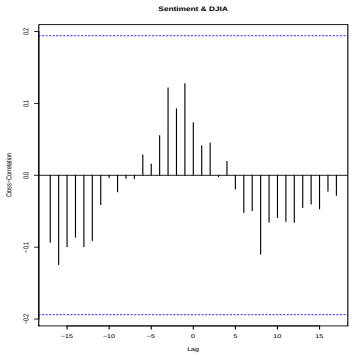
6.4.3 Time series

Before performing causality tests, it is necessary to ensure that both time series are *stationary*, because otherwise the results can lead to spurious causality [92]. The stationarity check is typically done by analysing the autocorrelation (ACF) and partial autocorrelation (PACF) functions, and performing the Ljung-Box [139] or the augmented Dickey-Fuller (ADF) [56] t-statistic tests. A market price is typically a non-stationary process, which is also true in our case. As we will explain later, the price of a stock is less important than change direction or trend. Thus, in our analysis, we replace EOD with a price-change (delta) time-series. Using the above methods, all our

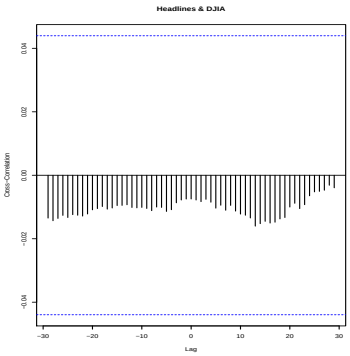


(a) *Financial Times* I

(b) *Financial Times* II



(c) *Financial Times* III



(d) Reddit Headlines

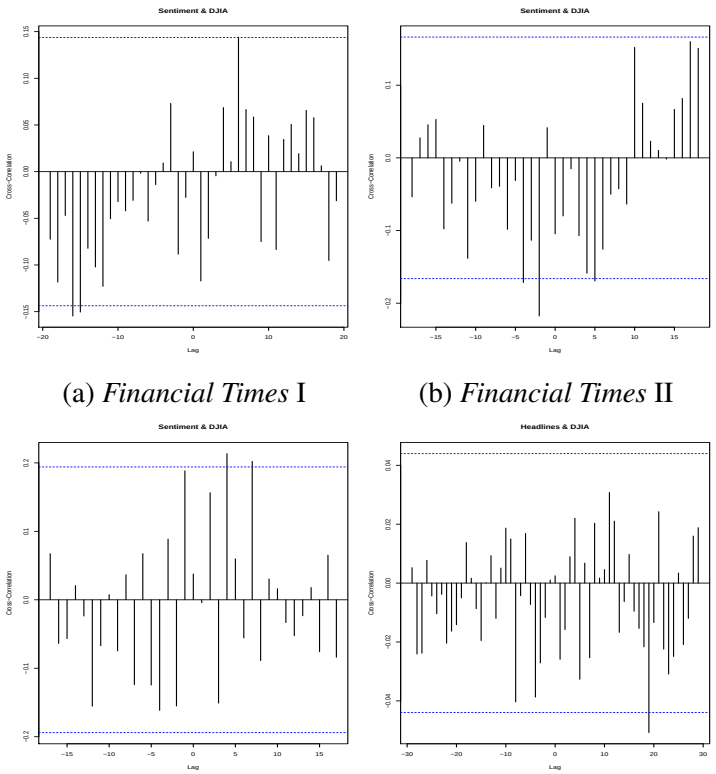
Fig. 6.10: The cross-correlation between sentiment attitudes and S&P 500 prices.

Fig. 6.11: The market close price changes (%).

selected time series were verified to be stationary. Figure 6.9 shows the stationarity check results of the FT-I and DJIA market datasets.

The next step in our investigation into the relationship between sentiment and price time series is to look at their *cross-correlation function* (CCF). Although "correlation

does not imply causation", it is frequently used as a test to discover possible causal relationship from data. In Figure 6.10 we present the cross-correlation analysis results for the S&P 500 index on all three FT news and RWNC headlines datasets. In the first dataset (see Figure 6.10a) we have a strong CCF between sentiment attitudes and stock prices; in the second (see Figure 6.10b) the CCF is significant in the lower left and upper right quadrants. However, in the FT III (see Figure 6.10c) dataset, the CCF is not significant (below the confidence threshold). This seems to suggest that the relationship between market sentiments and stock prices can be quite complex and may exist only in certain time periods. It is unsurprising that the financial market exhibited different behaviours in different time periods. As shown in Figure 6.11, from 2011 to mid-2013 we had a volatile market without a clear trend, whereas from 2013 to 2015 we saw a strong *bull* run with continual rising prices. Then we calculated the CCF between the sentiment attitudes found in RWNC headlines and the index prices for a longer time span from 2008 to 2016 (see Figure 6.10d), but still could not detect any long-term correlation.



(a) *Financial Times* I (b) *Financial Times* II

(c) *Financial Times* III (d) News Headlines

Fig. 6.12: The cross-correlation between sentiment attitudes and S&P 500 price changes.

Most of the real-life automated trading systems need to make **BUY** or **SELL** decisions for the given stocks. Therefore, from the trading perspective, the actual price of a stock is less crucial, and the profit relies on the price changes (often measured

in percentages). Similar to the previous experiments on the cross-correlation between sentiment attitudes and stock prices (see Figure 6.10), additional experiments on the cross-correlation between sentiments and stock price changes were performed (see Figure 6.12). Contrary to the previous results, we found correlations in all the FT news articles (see Figures 6.12a to 6.12c) and the RWNC headlines (see Figure 6.12d). This suggests that the percentage changes of stock prices would have higher predictability than stock prices themselves. However, the correlations are often present only with a substantial lag. Therefore, it is still valid that sentiment attitudes are unlikely to be useful in market trend prediction.

6.4.4 Experimental setup

To further analyse the relationship, we performed a set of *Granger causality* tests for the S&P 500 index and four selected stocks using the FT I dataset in which we detected the strongest correlation. The lags in the causality tests were set to just one or two days, considering that the financial market usually reacts to relevant news events almost instantaneously.

The sentiment analysis was performed using both a standard model and an enhanced temporal model. In the latter, we associate the sentiments with the corresponding temporal orientations by labelling each sentence with one of the four temporal categories (past/present/future/unknown) and calculate the sentiment strength accordingly. Here, we use a similar methodology to extract temporal orientation as in the section above. Intuitively, only the sentiments about the present and the future value of the stock would have significant impacts on its price. Therefore, we would filter out all sentiment scores with the *past* tag.

In each of our causality test experiments, two competing hypotheses would be examined: market sentiments cause stock price changes, and vice versa.

6.4.5 Experimental results

The results obtained from the experiments (see Table 6.8) show a mixed picture. In all the experiments, we failed to discover any sign that sentiment attitudes Granger-cause stock price changes, which would suggest that in general sentiment attitudes probably cannot be useful for the prediction of stock price movements. However, in many cases, we found that the opposite was true: stock price changes Granger-cause sentiment attitudes in the news, with the strongest causality found using the temporal sentiment-analysis model.

The individual stocks also produced mixed results, with each company behaving differently. For the Apple stock, we failed to detect any causality. For the Google stock,

Stock	Model	Lag	Attitude ⇒Price	Price⇒ Attitude
S&P 500	Standard	1	0.1929	0.1105
		2	0.2611	**0.0780**
	Temporal	1	0.2689	**0.0495**
		2	0.1692	**0.0940**
AAPL	Standard	1	0.7351	0.4253
		2	0.9117	0.6426
	Temporal	1	0.9478	0.6725
		2	0.9715	0.8245
GOOGL	Standard	1	0.5285	0.4035
		2	0.8075	**0.0418**
	Temporal	1	0.6920	0.5388
		2	0.8516	**0.0422**
HPQ	Standard	1	0.1534	0.3996
		2	0.1877	0.5322
	Temporal	1	0.4069	**0.0836**
		2	0.5097	0.1180
JPM	Standard	1	0.8991	**0.0461**
		2	0.9963	**0.0435**
	Temporal	1	0.9437	0.1204
		2	0.7722	0.2720

Table 6.8. Sentiment attitude Granger causality on the FT I dataset.

we identified that the prices would Granger-cause sentiment attitudes, but only with a two-day lag. For the HP stock, we detected causality only in temporal sentiment and only with a one-day lag. For the JPM stock, we found causality using standard sentiment, but it was absent using temporal sentiment. It is difficult to draw a general conclusion from such varying results. According to the Granger causality test with a one-day or two-day lag, sentiment attitudes do not seem to be useful for predicting stock price movements. However, the opposite seems to be true: the sentiment attitudes should be predictable using stock price movements. It is still possible that the Granger causality from sentiment attitudes to stock price changes is present at a finer time granularity (e.g., minutes), but we are unable to perform such an analysis using our current datasets.

Bollen et al. [28] attempted to predict the behaviour of the stock market by measuring the sentiment emotion of people on Twitter and identified that some of the emotion dimensions have predictive power. To verify their findings, we employed a similar model based on Plutchik's emotion dimensions extracted using the NRC sentiment lexicon [166] and *pSenti*. In the S&P 500 index analysis (see Table 6.9), we found that only `sadness` could Granger-cause stock price changes, which is different from the results of Bollen et al. [28]. Such a discrepancy might be explained by the fact that

Emotion	Lag	Standard		Temporal	
		Emotion ⇒Price	Price⇒ Emotion	Emotion ⇒Price	Price⇒ Emotion
anger	1	0.3815	0.6299	0.2555	0.4155
	2	0.3402	0.9153	0.3097	0.6886
anticipation	1	0.5320	0.2650	0.9216	0.9389
	2	0.4989	0.5765	0.4930	0.7173
disgust	1	0.6668	**0.0688**	0.2482	0.2031
	2	0.7166	0.3118	0.1160	0.2852
fear	1	0.5821	0.1255	0.8698	**0.0591**
	2	0.8934	0.2601	0.9888	0.1604
joy	1	0.6972	0.5549	0.3521	0.1530
	2	0.5567	0.8451	0.4045	0.4089
sadness	1	0.3885	0.1067	**0.0258**	0.1019
	2	0.6166	0.2027	**0.0983**	0.1423
surprise	1	0.5866	0.7022	0.3830	0.2315
	2	0.9802	0.8414	0.8445	0.3838
trust	1	0.9983	0.6892	0.9490	0.1124
	2	0.5534	0.8523	0.9586	0.2239

Table 6.9. Sentiment emotion Granger causality: S&P 500.

Emotion	Lag	Standard		Temporal	
		Emotion ⇒Price	Price⇒ Emotion	Emotion ⇒Price	Price⇒ Emotion
anger	1	0.9452	0.3512	0.6490	0.2352
	2	0.9851	0.4367	0.7703	0.1461
anticipation	1	0.5237	0.8272	0.3032	0.1245
	2	0.6368	0.3331	0.595	0.1518
disgust	1	0.2412	0.4128	0.1376	0.9851
	2	0.5154	0.5877	0.3392	0.3130
fear	1	0.3717	**0.0867**	0.2727	0.5577
	2	0.5609	0.1698	0.4139	0.1114
joy	1	0.3301	0.9946	0.7916	0.2580
	2	0.6657	0.5264	0.9843	0.4312
sadness	1	0.2217	0.8139	0.2280	0.6620
	2	0.1669	0.4266	0.1710	0.3245
surprise	1	0.9413	0.1960	0.1083	0.6093
	2	0.9733	0.2433	0.2033	0.2609
trust	1	0.5663	0.8439	0.3219	0.3539
	2	0.8760	0.5520	0.4473	0.2608

Table 6.10. Sentiment emotion Granger causality: AAPL.

Bollen et al. [28] used different emotion dimensions and lexicons, and a different time period in their analysis.

Emotion	Lag	Standard		Temporal	
		Emotion ⇒Price	Price⇒ Emotion	Emotion ⇒Price	Price⇒ Emotion
anger	1	0.2706	0.4460	0.4420	0.1530
	2	0.1709	0.7454	0.2457	0.2677
anticipation	1	0.1137	0.1951	0.2720	0.4348
	2	0.1487	0.4839	0.3363	0.7986
disgust	1	0.4459	0.3250	0.4000	0.5865
	2	0.7031	0.4294	0.6608	0.8880
fear	1	0.3362	0.1020	0.2874	0.1211
	2	0.2763	**0.0757**	0.3011	0.1765
joy	1	0.4718	**0.0417**	0.8350	**0.0959**
	2	0.7855	**0.0998**	0.9755	0.2282
sadness	1	0.4184	0.1316	0.3917	0.1782
	2	0.6599	0.1236	0.5286	0.3844
surprise	1	0.6551	**0.0606**	0.6869	**0.0755**
	2	0.7604	0.1166	0.6626	0.2156
trust	1	0.5008	**0.0727**	0.7541	**0.0680**
	2	0.5991	0.1302	0.8334	0.1052

Table 6.11. Sentiment emotion Granger causality: GOOGL.

Emotion	Lag	Standard		Temporal	
		Emotion ⇒Price	Price⇒ Emotion	Emotion ⇒Price	Price⇒ Emotion
anger	1	0.2129	0.8639	0.1300	0.9466
	2	0.4084	0.9521	0.2234	0.9689
anticipation	1	**0.0757**	0.6288	0.1316	0.7853
	2	0.2279	0.9059	0.3371	0.8986
disgust	1	0.4868	0.8126	0.2001	0.4536
	2	0.3803	0.9353	0.2252	0.6722
fear	1	0.2679	0.4841	0.1214	0.8193
	2	0.5361	0.4741	0.2371	0.9319
joy	1	**0.0399**	0.8186	**0.0902**	0.6261
	2	0.1255	0.8945	0.2410	0.7273
Sadness	1	**0.0106**	0.8669	**0.0110**	0.9208
	2	**0.0416**	0.9365	**0.0388**	0.8456
surprise	1	**0.0217**	0.6825	**0.0010**	0.3890
	2	**0.0759**	0.7830	**0.0064**	0.3034
trust	1	**0.0447**	0.8620	**0.0766**	0.7693
	2	0.1340	0.9034	0.2158	0.7948

Table 6.12. Sentiment emotion Granger causality: HPQ.

An interesting finding we obtained from the experimental results is that some individual stocks, such as HP (see Table 6.12) and JPM (see Table 6.13), have significantly

Emotion	Lag	Standard		Temporal	
		Emotion ⇒Price	Price⇒ Emotion	Emotion ⇒Price	Price⇒ Emotion
anger	1	0.1788	0.0488	0.1796	0.2349
	2	0.4903	0.1223	0.3155	0.3713
anticipation	1	0.3893	0.1360	0.1389	0.5989
	2	0.7729	0.3145	0.3389	0.4732
disgust	1	0.2168	0.1260	0.2208	0.2267
	2	0.5297	0.2913	0.3611	0.1637
fear	1	**0.0298**	0.1565	**0.0214**	0.2072
	2	0.1173	0.2495	**0.0210**	0.1187
joy	1	0.3417	0.2169	0.1073	0.9574
	2	0.8544	0.3905	0.3293	0.9086
sadness	1	0.6079	0.3038	0.3985	0.5781
	2	0.9297	0.5856	0.4495	0.4194
surprise	1	0.1351	**0.0303**	**0.0498**	0.1145
	2	0.4296	**0.0593**	**0.0850**	**0.0461**
trust	1	**0.0991**	0.2218	**0.0458**	0.6664
	2	0.1232	0.2066	**0.0165**	0.6645

Table 6.13. Sentiment emotion Granger causality: JPM

more emotion dimensions with predictive power than others. It could be seen in those cases that some emotion dimension other than `sadness`, including `surprise`, `fear`, `joy` and `trust`, also demonstrated predictive power. On both Google (see Table 6.11) and Apple (see Table 6.10) stock price data, we failed to find any emotion causality on their stock price. These results indicate that even if in some cases there is substantial Granger causality from sentiment emotions to stock price changes, it is not a general pattern and should be looked at on a case-by-case basis. To find out why that is happening, it would be necessary to perform a further investigation, which is beyond the scope of this thesis.

6.4.6 Prediction

The causality analysis in Section 6.4.2 has revealed that in some cases sentiment emotions could be good indicators of stock price changes. In the next set of experiments, we would like to investigate how sentiment attitudes and/or sentiment emotions could be exploited in a machine-learning model for market trend prediction to improve its accuracy.

Basically, there are two types of stock market analysis: fundamental and technical. The former evaluates a stock based on its corresponding company's business performance, whereas the latter evaluates a stock based on its volume and price on the financial market, as measured by a number of so-called *technical indicators* [129]. Both

types of analysis generate trading signals, which would be monitored by human traders or automated trading systems who then use that information to execute trades. In our experiments, only technical analysis has been utilised. It is likely that incorporating fundamental analysis and employing more technical indicators would improve the predictive model's performance. However, our research objective is not to create an optimal market trend prediction system but to analyse and understand the predictive power of sentiments on the financial market. For this purpose, a baseline model with several common technical indicators should be good enough.

6.4.7 Baseline

We first built a baseline machine-learning model to predict stock price changes with a number of selected technical indicators, and then tried to incorporate additional sentiment-based features (i.e. sentiment attitudes and sentiment emotions).

In order to construct a decent baseline model, we made use of ten common technical indicators, which led to a total of fifteen features, as follows.

- Moving Averages (MA). A moving average is frequently defined as a support or resistance level [129]. Many basic trading strategies are centred around breaking support and resistance levels. In a rising market, a 50-day, 100-day or 200-day moving average may act as a support level and, in a falling market, as resistance. We calculated 50-day, 100-day and 200-day moving averages and included each of them as a feature.

- Williams %R. This indicator was proposed by Larry Williams to detect when a stock was overbought or oversold [129]. It tells us how the current price compares with the highest price over the past period (10 days).

- Momentum (MOM) [129]. This indicator measures how the price has changed over the last N trading days. We used two momentum-based features: one-day momentum and five-day momentum.

- Relative Strength Index (RSI). This is yet another indicator to find overbought and oversold stocks [129]. It compares the magnitude of gains and losses over a specified period. We used the period most commonly used: fourteen days.

- Moving Average Convergence Divergence (MACD) [129]. This is one of the most effective momentum indicators and shows the relationship between two moving averages. It generates three features: MACD, signal and histogram values.

- Bollinger Bands is one of the most widely used technical indicators [129]. It was developed and introduced in the 1980s by the famous technical trader John Bollinger.

It represents two standard deviations away from a simple moving average, and can thus help price pattern recognition.

- Commodity Channel Index (CCI) is another a momentum indicator, introduced by Donald Lambert in 1980 [129]. This indicator can help to identify a new trend or warn of extreme conditions by detecting overbought and oversold stocks. Its normal movement is in the range from -100 to +100, so going beyond this range is considered a BUY/SELL signal.

- Average Directional Index (ADX) is a non-directional indicator which quantifies the price trend strength using values from 0 to 100 [129]. It is useful for identifying strong price trends.

- Triple Exponential Moving Average (TEMA) was developed by Patrick Mulloy and first published in 1994 [129]. It serves as a trend indicator and, in contrast to moving averages, does not have the associated lag.

- Average True Range (ATR) is a non-directional volatility indicator developed by Wilder [248]. The stocks and indexes with higher volatility typically have higher ATR.

The features were all normalised to zero mean and unit variance in advance.

In our context, the machine-learning model is just a binary classifier that generates two kinds of signal: **BUY** $(+1)$ and **SELL** (-1). It aims to predict whether or not the stock's price, n days in the future, will be higher $(+1)$ or lower (-1) than today's price. In the preliminary experiments, we tried to find out which machine-learning algorithm would perform best and how far into the future the model would be able to predict.

Following the research literature in this area [100, 42, 71], we evaluated the two most popular machine-learning approaches to market trend prediction, SVM (with the RBF kernel) and the LSTM recurrent neural network. Each dataset was randomly divided into two sets: two-thirds for training and one-third for testing. The parameters of the SVM and LSTM algorithms were set via grid search on the training set. The final LSTM model consists of a single LSTM layer with 400 units and utilises a dropout rate of 0.5 [235, 220].

It is common for such market trend prediction models to use a time lag of a few days and, by doing so, avoid short-term price volatility [50]. In our experiments, we tried both three- and five-day lags. Similar to the previous studies by Cao and Tay [37] and Thomason [229], using five-day lags was found to be optimal.

The preliminary experimental results, as shown in Table 6.14, indicate that SVM outperformed LSTM on all the datasets. The F_1 scores suggest that LSTM often favoured the positive class over the negative class and produced unbalanced results.

Type	Method	3-day ahead			5-day ahead		
		Acc	F_1^{up}	F_1^{down}	Acc	F_1^{up}	F_1^{down}
DJIA	SVM	0.616	0.738	0.282	**0.700**	**0.754**	**0.615**
	LSTM	0.559	0.706	0.120	0.585	0.728	0.127
AAPL	SVM	0.577	0.676	0.391	**0.685**	**0.723**	**0.634**
	LSTM	0.547	0.693	0.138	0.521	0.641	0.282
JPM	SVM	0.677	0.747	0.552	**0.673**	**0.733**	**0.578**
	LSTM	0.541	0.665	0.269	0.573	0.676	0.373
EUR/USD	SVM	0.642	0.607	0.672	**0.671**	**0.620**	**0.710**
	LSTM	0.509	0.423	0.572	0.563	0.370	0.665
GBP/USD	SVM	0.610	0.589	0.630	**0.714**	**0.705**	**0.723**
	LSTM	0.500	0.604	0.323	0.633	0.646	0.618

Table 6.14. Market trend prediction using main technical indicators — the baseline model.

The reason could be that the size of the dataset is relatively small: there are 670 data points in the analysed time period 2011–2015. Contrary to LSTM, SVM always yielded balanced and stable results.

In the end, SVM with a five-day lag was selected as the baseline model, which produced a reasonable accuracy of around 70% and similar F_1 scores for both classes.

6.4.8 Using sentiment signals in news

In the next set of experiments, we evaluated the predictive power of sentiments extracted from financial news articles/headlines. The time granularity here is a single day (i.e., all sentiment-based features –including both attitudes and emotions – would be aggregated by calculating their daily averages). If there was no sentiment information available on that day, the value zero would be assigned to the corresponding sentiment features.

The proposed new model consists of the same technical indicator features as in the baseline, plus nine additional sentiment-based features:

- Sentiment attitudes. The average daily sentiment attitudes, extracted using *pSenti*, with values in the range from -1 to +1.
- Sentiment emotions in eight categories: `anger`, `anticipation`, `disgust`, `fear`, `joy`, `sadness`, `surprise`, and `trust`, with values being the normalised occurrence frequency.

Sentiment attitudes and emotions were extracted from the FT news articles and the RWNC headlines in the time period from 2011 to 2015. The experimental results, as shown in Table 6.15, indicate that incorporating sentiment attitudes and sentiment emotions from the headlines actually had a negative impact on the predictive model's

Type	Baseline			*Financial Times*			Reddit Headlines		
	Acc	F_1^{up}	F_1^{down}	Acc	F_1^{up}	F_1^{down}	Acc	F_1^{up}	F_1^{down}
DJIA	0.700	**0.754**	0.615	**0.706**	0.752	**0.639**	0.618	0.716	0.417
AAPL	**0.685**	**0.723**	**0.634**	0.652	0.723	0.531	0.624	0.700	0.496
JPM	0.673	0.733	0.578	**0.679**	**0.739**	**0.583**	0.615	0.713	0.415
EUR/USD	0.641	0.715	0.518	**0.653**	0.691	**0.605**	0.638	0.684	0.578
GBP/USD	**0.714**	0.705	**0.723**	0.711	**0.708**	0.715	0.615	0.625	0.605

Table 6.15. Market trend prediction using FT news articles and RWNC headlines (2011–2015).

performance. This is consistent with the previous section, in which no correlation or causality link was established between headline sentiments and stock prices. It might be explained by the fact that headlines are very short text snippets and therefore provide little chance for us to reliably detect sentiment attitudes and sentiment emotions. The sentiments extracted from FT news articles painted a quite different picture. The sentiment-enriched model outperformed the baseline model in half of the scenarios: it demonstrated slightly better results for DJIA, JPM and EUR/USD, but slightly worse results for AAPL and GBP/USD. These experimental results are consistent with the previous section and confirm again that, for some stocks, sentiment emotions could be used to improve the baseline model for market trend prediction.

6.4.9 Using sentiment signals in tweets

In the last set of experiments, we created the enriched model based on sentiment attitudes and sentiment emotions extracted from financial tweets. The time period of the Twitter messages dataset is significantly shorter: 2014 to 2015. Consequently, the experiments were performed on a shorter time period with only 275 data points. In this time period, almost all stock prices were continually rising (see Figure 6.11). Such a so-called bull run makes it even more difficult to assess a predictive model's performance, as any basic strategy such as buy and hold would be a winning strategy.

Let us consider three different scenarios. In the first scenario ("all+attitude+emotion"), both sentiment attitudes and sentiment emotions were extracted from all financial tweets and used as additional features. This allowed us to verify how useful sentiment information is for market trend prediction. In the second scenario ("all+emotion"), only those eight sentiment emotions were used as additional features. This provided an opportunity to validate the usefulness of sentiment emotions alone. For the last scenario ("filtering+emotion"), only the Twitter messages (tweets) mentioning the company of our interest were utilised to extract sentiment emotions as additional features.

Type	baseline			all+attitude+emotion			all+emotion			filtering+emotion		
	Acc	F_1^{up}	F_1^{down}	Acc	F_1^{up}	F_1^{down}	Acc	F_1^{up}	F_1^{down}	Acc	F_1^{up}	F_1^{down}
DJIA	**0.810**	**0.854**	0.727	0.810	0.846	**0.750**	0.778	0.829	0.682	-	-	-
AAPL	**0.889**	**0.918**	**0.829**	0.810	0.860	0.700	0.794	0.847	0.683	0.794	0.831	0.735
JPM	0.746	0.800	0.652	0.730	0.779	0.653	0.746	0.789	0.680	**0.778**	**0.829**	**0.682**
GBP/USD	**0.708**	**0.387**	**0.808**	0.662	0.389	0.766	0.631	0.294	0.750	-	-	-
EUR/USD	0.685	**0.627**	0.727	0.685	**0.627**	0.727	**0.692**	0.626	**0.739**	-	-	-

Table 6.16. Market trend prediction using financial tweets from Twitter (01/04/2014 – 01/04/2015).

Fig. 6.13: Twitter Market Bot

The experimental results, as shown in Table 6.16, indicate that, most of the time, the baseline model would actually outperform the expanded model with sentiment attitudes, sentiment emotions or both as additional features. Only for the JPM stock did we see noticeable performance improvements in the "filtering+emotion" setting. Once again these results are consistent with the causality analysis in Section 6.4.2 and the market trend prediction experiments using financial news in Section 6.4.8 — the JPM stock demonstrated that integrating sentiment emotions has the potential to enhance the baseline model. Our results have also confirmed that sentiment attitudes on their own are probably not very useful for market trend prediction, but at least for some particular stocks sentiment emotions could be exploited to improve machine-learning models such as SVM to achieve better market trend prediction.

Our findings are mostly in line with other researchers' results [28]. However, there are still many questions remaining unanswered in this area. As part of our analysis, we also created the prototype Twitter bot (see Figure 6.13), which is continuously monitoring Twitter, various news outlets and market situations, and provides trading recommendations.

6.5 Political Sentiment Case Study

In this section, we present the results of political-sentiment analysis using the 2016 US presidential election Twitter data. More specifically, together with Birkbeck Department of Politics, we used Twitter analytics and sentiment analysis to examine questions of *white flight*, and the contextual basis of support for anti-immigration politicians. Our principal methodology was to categorise individuals on Twitter into pro- and anti-Trump categories using geotagged social media data. We used Trump support as a proxy, albeit an imperfect one, for immigration sentiment, thereby surmounting some of the problems presented by the absence of large-scale American longitudinal data on attitudes and voting. Though just 0.06% of tweets are tagged with coordinates, the scale of the data was sufficient to compare the ethnic contexts of Trump opponents and supporters.

Political sentiment in the research literature is also known as *stance detection* (SD) [119, 120, 226]. It can be defined as a speaker's opinion towards a particular target and is closely related to sentiment analysis. As defined by Mohammad et al. [167], a typical sentiment-detection system classifies a text into positive, negative or neutral categories, while in SD the task is to detect a text that is favourable or unfavourable to a specific given target. By analysing how attitudes and actual residential behaviour interact over time, geolocation technology and SD open new possibilities for research and provide an opportunity to examine the demographics of right-wing populists.

6.5.1 Datasets

Type	Values
Tags	#MakeAmericaGreatAgain, #NeverTrump; #DonaldTrump, #Trump2016, #Trump
Users	@realDonaldTrump

Table 6.17. 2016 US presidential election dataset

In order to carry out the analysis, we gathered a large collection of tweets related to the 2016 presidential campaign of Donald Trump. From April to November 2016, using the Twitter API, we collected all tweets expressing a strong stance towards Trump (see Table 6.17). The collected dataset consists of:

- 142 million unique messages from 7.6 million users (including re-tweets);
- 55 million original messages;
- 49 million messages with geographical tags;

Fig. 6.14: 2016 US presidential election Twitter messages

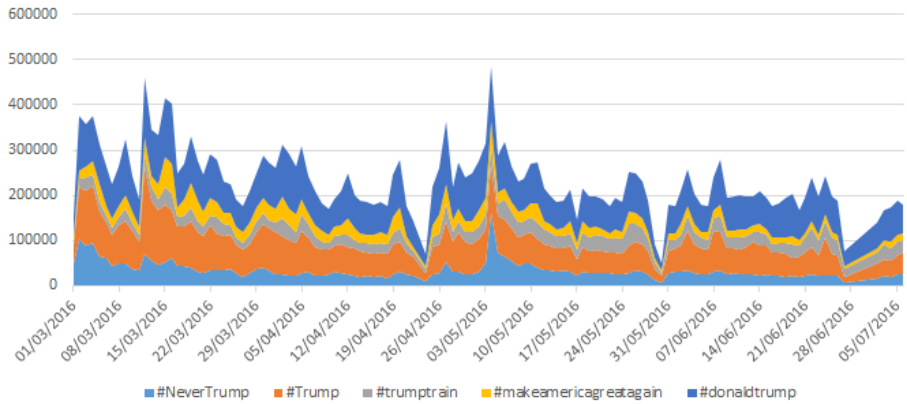


Fig. 6.15: Twitter messages by tag

- 53 million messages with user location tags;
- 3 million messages with place tags;
- 82 thousand messages with coordinates;
- around 12 thousand user profiles with an identifiable profile image.

All the tweets were automatically filtered for duplicates. Only tweets from users with less than 50 thousand followers were included. All tweets were pre-processed by replacing emoticons with their corresponding text representations and encoding URLs with tokens. Not all functionality was available via the Twitter API. Therefore, to collect user profile photos, we also employed Web scraping. In Figure 6.14 we presented tweets' distribution by date and in Figure 6.15 by hashtags.

To create the SD test set, we randomly selected 2488 tweets and annotated them using the MTurk crowdsourcing platform. In MTurk, a single annotation task is called

Source	*FAVOUR*	*AGAINST*	*NONE*
mTurk	562	852	176
SemEval	148	299	260

Table 6.18. Trump stance classification test datasets

a Human Intelligence Task (HIT). In each of the HITs, the annotators were asked to annotate three tweets with the following tags:

- *IN FAVOUR*: a tweet expresses positive stance towards Donald Trump;
- *AGAINST*: a tweet expresses negative stance towards Donald Trump;
- *NONE*: a tweet is neutral or not related to Donald Trump.

The annotation methodology followed established natural language processing standards. Only annotators with at least 80% approval rate participated. Annotators worked to agreed guidelines, with at least two annotators annotating each message. In the final dataset, only tweets where agreement passed a threshold were included. We also used a "*Known Answer Review Policy*". It was implemented by including a single "*Known Answer*" question in each of HITs. *Known answer* questions are questions where we know the answer, and if an annotator failed to answer it correctly, we rejected his annotation as unreliable. From original 2488 messages, only 1590 satisfied quality requirements. The final dataset consists of 562 messages in *favour*, 852 *against* and 176 *not related* to Donald Trump (see Table 6.18).

We also annotated the profiles of all users who posted geotagged tweets (11 969). To annotate user profiles, we collected their public profile images, extracted profile names and asked annotators to indicate if they could classify them by age, sex and race. Out of 11 969, only 8403 profiles were successfully annotated.

In the last stance detection experiment, we use the dataset from SemEval-2016 Task 6B [167]. The dataset consists of 707 tweets annotated into in *favour*, *against* or *not related* to Donald Trump (see Table 6.18). Compared to our collected dataset, this dataset has a larger proportion of non-related messages.

In the Trump supporter demographic analysis, we processed all geotagged tweets using a geographic information system (GIS). For each tweet, we extracted its coordinates and located the corresponding zip code tabulation area (ZCTA). This information allowed us to link each message with the most recent US census data.

6.5.2 Stance detection

To evaluate our proposed stance detection approach in the political domain, we carried out three experiments. In the first experiment, we classified all tweets from the first

For Trump	Against Trump
#trumpnight, #trumpolympics, #trumpforpresident, #teamtrumppa, #gotrump, #lovetrump, #godblesstrump, #trumpiswithus, #usafortrump, #trump2016, #takebackamerica, #trumpforpresident2016, #trumpcare, #trump4president, #trumpcares, #trump16, #thedonald, #trumpwillwin, #deplorablesfortrump, #great, #best, #powerful, #draintheswamp, #saveusa', #trumprally, #hilaryforprison, #hillary4prison2016, #hillaryforjail, #hillarysucks, #neverclinton, #nevereverhillary, #neverhillaryclinton, #nomoreclintons, #crookedclinton	#clinton2016, #voteclinton, #votehillaryclinton, #hillaryforpresident, #hillaryforamerica, #hillaryclintonforpresident, #hillaryclinton2016, #hillaryclinton, #hiliary, #hillary, #hillary2016, #hilaryclinton, #fuckdonaldtrump, #sexualpreditor, #notmypresident, #rapisttrump, #cowardlytrump, #racisttrump, #dangerousdonald, #fuckyoutrump, #nevertrumpers, #predator

Table 6.19. Specific "seeds" for the presidential candidate political-sentiment lexicon induction.

labelled dataset into three classes: *in support*, *against* or *not related* to Donald Trump. In the second we removed *non-related* messages and performed binary classification. In the last experiment, we compared our SD method with other participants in SemEval-2016 Task 6B competition [167] using the SemEval dataset.

Following the first two experiments, we compared four different SD methods: (i) *pSenti* with the general-purpose lexicon, (ii) *pSenti* with the induced domain lexicon using generic seeds and (iii) specific seeds, and (iv) the semi-supervised approach from Chapter 5, based on a deep-learning LSTM model. Similar to Chapter 5, for calibration of our LSTM-based binary sentiment classifier, we employed the sigmoid model of Platt [188] with cross-validation on pseudo-labelled training data. To induce the domain-specific lexicon, this research followed a similar procedure to that outlined in Chapter 4. As it is common in political tweets to express a stance using hashtags, we included in the sentiment candidate list all hashtags discovered in the dataset.

As mentioned above, we evaluated two different strategies for the domain-specific lexicon induction. The first method uses generic Twitter seeds from Chapter 4. The second method is based on the so-called two-stage method. The first stage is the same as in the first method and is based on generic seeds. The second stage uses the output of the previous stage, selects the most revealing hashtags (see Table 6.19) and uses them as seeds for the second induction pass. In a sense, the first pass gave us a better understanding of the political domain, and in the second pass we exploited this knowledge to induce a better-quality stance detection lexicon. The selection process was manual and required a human annotator to verify selected seeds. It is likely that selection processes can be automated. However, that was outside the scope of this thesis. Our aim was merely to compare the performance of the generic and specific seeds.

System	F_1^{Favour}	F_1^{Against}	F_1^{None}
pSenti with existing general-purpose lexicon	0.444	0.607	0.193
pSenti with induced (generic seeds) lexicon	0.585	0.691	0.220
pSenti with induced (specific seeds) lexicon	0.593	0.691	0.222
Semi-supervised model from Chapter 5	**0.638**	**0.707**	**0.225**

Table 6.20. Trump support messages classification into three classes

System	*AUC*	F_1^{Favour}	F_1^{Against}
Random Selection Classifier (Stratified)	0.509	0.454	0.553
pSenti with existing general-purpose lexicon	0.584	0.471	0.686
pSenti with induced (generic seeds) lexicon	0.714	0.620	0.739
pSenti with induced (specific seeds) lexicon	0.723	0.629	0.734
Semi-supervised model from Chapter 5	**0.803**	**0.679**	**0.758**

Table 6.21. Trump support messages classification into two classes

As expected, in the first experiment (see Table 6.20), the semi-supervised model outperformed all other methods by a significant margin, yet all models performed poorly with the *non-related* messages class. This can be explained by the fact that those *non-related* tweets, being only a tiny fraction of the dataset (11%), are not easy to define and are more likely just sentiment anomalies. This suggests that to improve the performance further, we would need to perform additional domain adaptation, most likely with two-step classification. In the first step, we would remove all non-related messages and later classify political messages into binary classes.

The binary political support classification task (see Table 6.21) demonstrated our domain-adaptation superiority more clearly. In all experiments, we measured statistical significance using the two-tailed binomial test [259] with a confidence level of 95% and confirmed that each following model was significantly better than its previous version. In all scenarios, the performance of the general-purpose sentiment lexicon was inferior and could not be efficiently employed in the political message analysis. It was just marginally better than random selection. Domain adaptation significantly improved performance. The induced lexicon produced reasonable results, with 0.723, and the semi-supervised model achieved 0.803 AUC. Experiments also demonstrated that although a slight increase in performance may be achieved by using the two-stage lexicon induction, the improvement is marginal and requires additional effort.

6.5.3 Evaluation of the method on the SemEval 2016 dataset

Finally, to evaluate our proposed stance detection approach, we carried out experiments on the Trump SD benchmark dataset from SemEval-2016 Task 6B [167], which is to classify 707 tweets as either in favour or against Donald Trump (see Table 6.18).

System	F_1^{Favour}	F_1^{Against}	F_1^{Average}
$\text{Ours}_{\text{LSTM}}$	0.4344	0.4650	0.4497
Worst system	0.1659	0.3487	0.2573
Median system	0.1796	0.5020	0.3408
Best system	**0.5739**	**0.5517**	**0.5628**

Table 6.22. Results for SemEval-2016 Task 6B.

No training data labelled with the stance towards "Donald Trump" was provided, but participants were free to use data from SemEval-2016 Task A to generate a training dataset. Thus, many of the proposed methods were either *supervised* or *nearly supervised*. Contrary to our *nearly-unsupervised* approach, the winner's method [242] was a *supervised* approach.

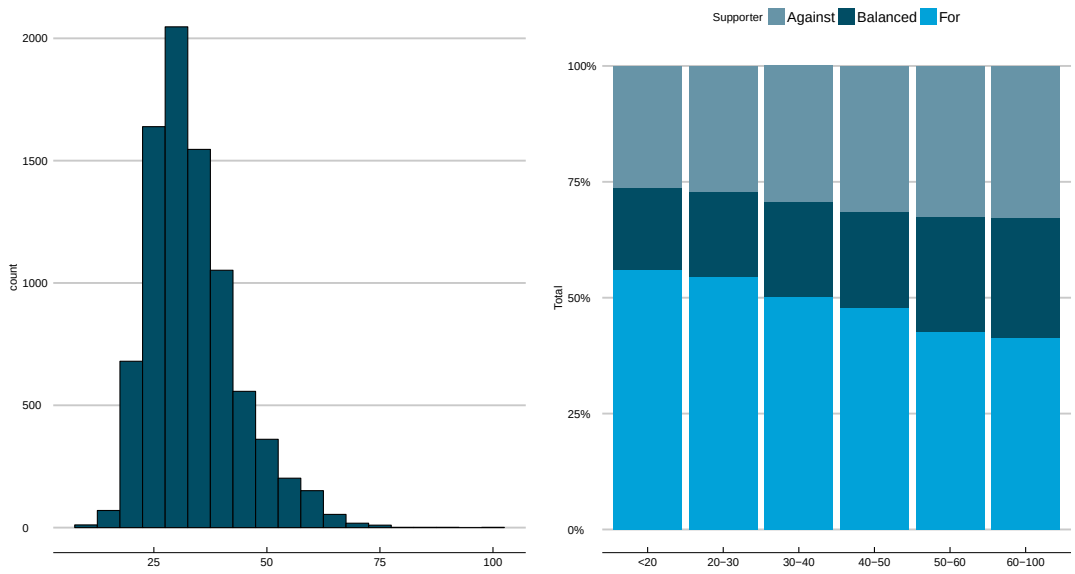
As we can see from the results shown in Table 6.22, our approach from Chapter 5, based on a deep-learning LSTM model was second best. It outperformed all participants by a significant margin and lost only to the *supervised* approach proposed by Wei et al. [242].

6.5.4 Demographics of Trump supporters

To investigate the *"white flight"* research question and the demographics of anti-immigration politicians' supporters, we selected only users who published tweets with geographical tags. From our Twitter dataset, we identified 11 969 such users, who generated more than 382 thousand messages. As mentioned above, these user profiles were annotated using MTurk to determine their gender, race and age. The annotation revealed that most Twitter profiles belong to white males (see Figures 6.17a and 6.18a) with a median age of 32 (see Figure 6.16a).

To extract the Trump political support level, we employed the SD methodology explained in the section above. All calculations were made using the best-performing deep-learning (LSTM) model, adapted to the 2016 presidential election domain. We classified all messages into two classes, such as in *favour* or *against* Donald Trump. We assigned people to the *supporters* or *opponents* of Donald Trump if at least 75% of their messages expressed the same stance. Users with a lower proportion were assigned to the *balanced stance* group.

The political-sentiment (stance) analysis results uncovered several demographic patterns, which in some instances contradict the official statistics and other research papers. Our results in this section are based on a relatively small sample and, as Kahneman and Egan [109] highlighted, experiments using small samples can suffer from a number of limitations. Such experiments have replicability issues, may yield extreme results and can lead to overinterpretation of findings. In addition, we have



(a) Age groups

(b) Political support

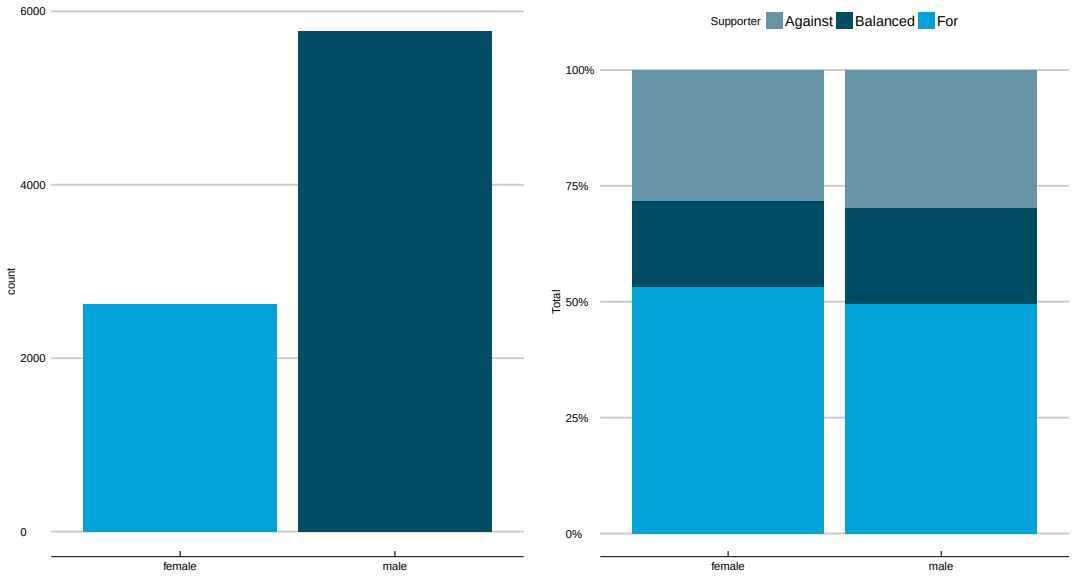
Fig. 6.16: Political demographics by age

numerous dataset sampling constraints, such as having only messages with coordinates and only with a strong stance towards Trump, which should explain some of our result outliers. Based on all this we treat our findings as indicative results only which would need to be validated and explored by further, more extensive experiments in future.

One of the first observations from our results is that Trump political support decreases with increasing supporter age (see Figure 6.16b). This is opposite from what was found in other research papers [240] and in the 2016 election exit polls [64].

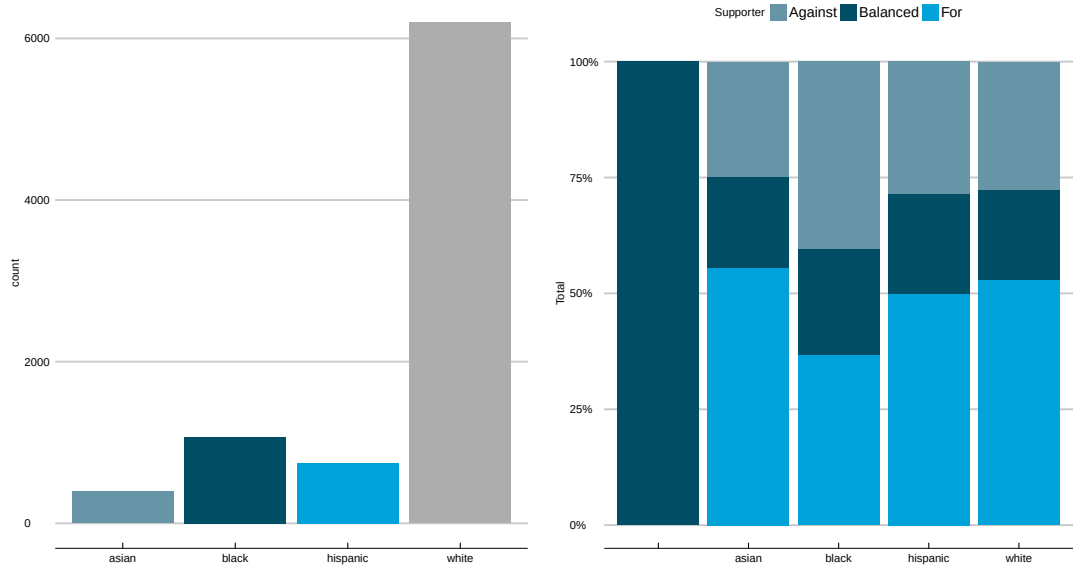
Similar surprising results have been obtained in the analysis by sex groups (see Figure 6.17b), where we found that Trump has higher support among women. In the case of race (see Figure 6.18b), our results were somewhat consistent with other sources: Trump has the lowest support among the Black population; however, contrary to the 2016 election exit polls [64], we found that he also has high support among the Asian and Hispanic population.

To analyse the distribution of Trump supporters using other demographic groups we made use of the earlier extracted ZCTA and the most recent US census data. In the case of income (see Figure 6.19), we identified that Trump has the strongest support among the lowest and highest income groups. The data analysis by education seems even more controversial (see Figure 6.20). According to the experimental results, Trump has significantly higher support in areas with the most-educated voters. Our findings contradict other researchers' findings, such as those stating that Trump supporters are predominantly middle-aged, non-Hispanic whites, especially those without a college degree, living in small cities and rural areas [168]. More research is needed to identify



(a) Sex groups (b) Political support

Fig. 6.17: Political demographics by sex

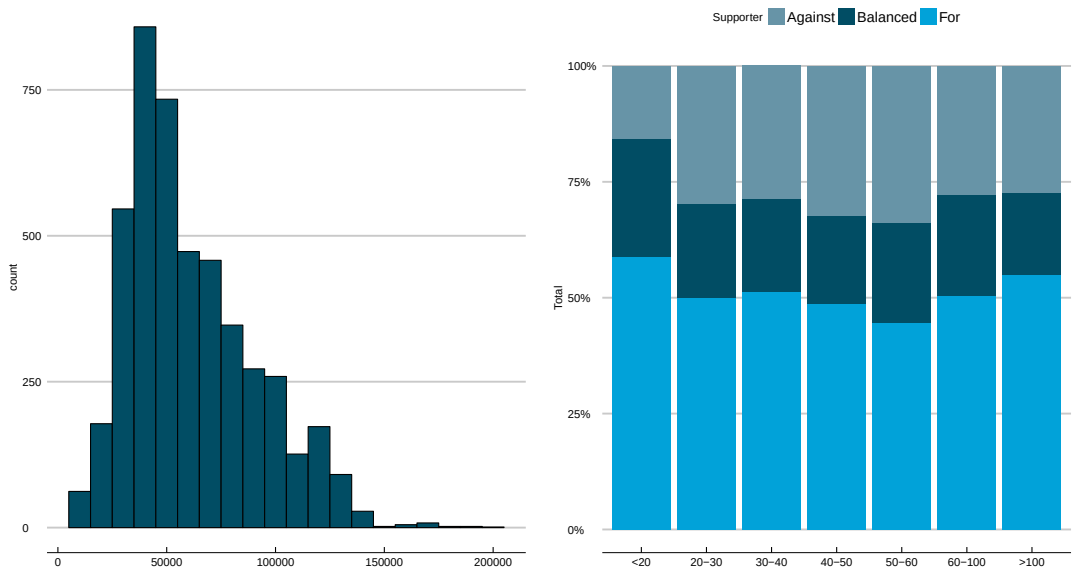


(a) Race groups (b) Political support

Fig. 6.18: Political demographics by race

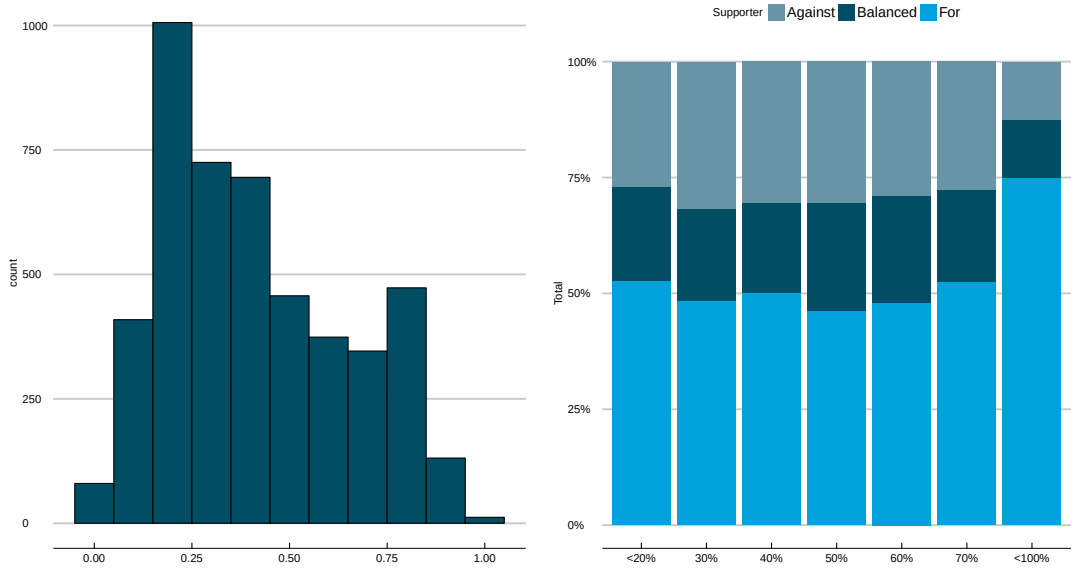
why our results are different. It might be related to our geotagging requirement or some other unidentified anomalies.

Besides that, we performed user movement analysis to answer the *"white flight"* research question. Using geotagged messages, we attempted to find out whether Trump supporters were more likely to move home to significantly whiter areas. To ascertain whether an individual has moved home or if it is just commuting/travelling/on holiday,



(a) Income groups (b) Political support

Fig. 6.19: Political demographics by income (thousands)



(a) Degree groups (b) Political support

Fig. 6.20: Political demographics by degree

we performed the temporal and spatial clustering of their message location points. Typically, people have two gravity centres (home and work), going back and forth [2]. A one-directional shift detection from one location to another one that might indicate permanent relocation. As a simplified solution, we selected out-of-work messages (between 7 p.m. and 8 a.m.), calculated distance from the centre of the USA for each message and clustered the data using K-means into two and three clusters (using

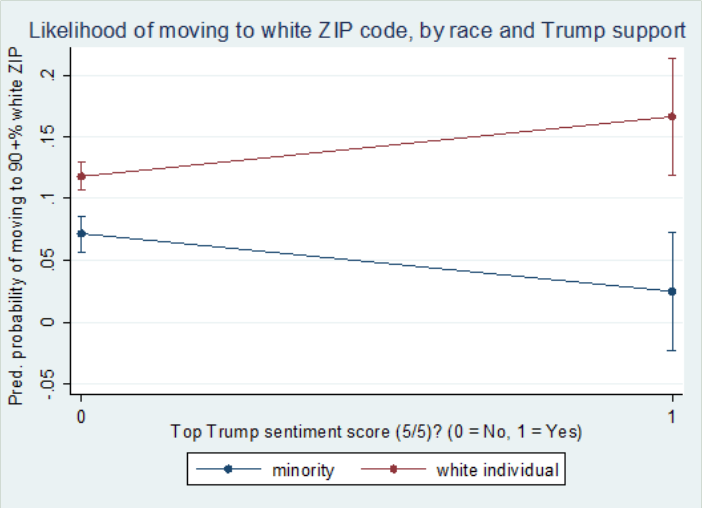


Fig. 6.21: Twitter messages by tag

normalised distance). Preliminary results (see Figure 6.21) identified that *white* Trump supporters are indeed more likely to move to white areas.

6.6 Summary and Conclusions

In this chapter, we presented four use case studies of our sentiment-analysis method *pSenti* and its application in various domains, from customer reviews to financial news articles and tweets. In Section 6.2 we focused on Amazon product reviews and sentiment time-series dynamics; in Section 6.2.3 we expanded that investigation into seasonality detection; and finally, in Section 6.2.4, a *temporal-hybrid* sentiment-analysis system was proposed. Experimental results indicate that the proposed *temporal-hybrid* system outperformed the baseline and on average achieved between 5% and 15% better results. We also identified that a simpler design using a regression over past achieved similar performance may be a more practical alternative. This indicates that aggregating past sentiment significantly boosts performance and may be employed in customer review sentiment analysis.

We found, contrary to initial expectations, that product sentiment is a dynamic process and frequently degrades over time. The sentiment value is also exposed to various anomalies and, under their influence, can significantly change strength. Our results also identified temporal dependencies and confirmed the importance of continuous sentiment system adaptation to an underlying domain.

In Section 6.4 we empirically re-examined the feasibility of applying sentiment analysis to make market trend predictions. Our experiments investigated the causal relationship between sentiment attitude/emotion signals and stock price movements

using various sentiment signal sources and different time periods. The experimental results indicate that the interaction between sentiment and price is complex and dynamic: while some stocks in some time periods exhibited strong cross-correlation, it was absent in other cases. We have discovered that in general sentiment attitudes do not seem to have any Granger causality with respect to stock prices, but sentiment emotions do seem to Granger-cause price changes for some particular stocks (e.g., JPM). Furthermore, we have attempted to incorporate sentiment signals into machine-learning models for market trend prediction. Specifically, we have compared two popular machine-learning approaches and finally selected SVM (with an RBF kernel) as the baseline, which was trained using fifteen technical indicators. On average, this method achieved 70% accuracy for five-day market trend prediction. The baseline model was then expanded using sentiment attitudes and sentiment emotions extracted from financial news or financial tweets as additional features. In some scenarios, the proposed model outperformed the baseline model and demonstrated that sentiment emotions could be employed to help predict stock price movements, whereas sentiment attitudes could not. The sentiment emotions extracted form *Financial Times* news articles yielded better performances than those extracted from Reddit news headlines.

An important research question for future work is how to identify stocks whose price changes are indeed predictable using sentiment emotions. Although the Granger causality test on the historical data could find stocks that had been predictable in the past, there is no guarantee that they would continue to be predictable in the future. It is possible that a more sophisticated classifier for this purpose could be developed.

Finally, in Section 6.5 we examined Donald Trump supporters and opponents and performed stance detection on their tweets. Using geotagged social media data, which repeatedly measures the attitudes and locations of large numbers of individuals, we shed new light on how patterns of ethnic segregation are reproduced. Some of our findings contradicted those of other researchers, such as finding that Trump has significant support in highly educated areas and among younger age voter groups. Such disparity may be explained by the fact that our results were based on a small sample, with numerous sampling limitations. This use case once more confirmed the superiority and flexibility of our domain-adaptation method, which had significantly better performance than the baseline. It also demonstrated that, using a multi-stage approach, generic seeds might be helpful in generating higher-quality domain-specific seeds. Although such an approach requires additional effort and produces only a slight increase over generic seeds, it demonstrated that this approach could, in principle, improve sentiment analysis on domain adaptation. However, more research is needed to answer all the remaining questions.

Chapter 7

Conclusion

Most of today's sentiment analysis systems are based on either a ready-made lexicon or a supervised learning technique. A typical lexicon-based sentiment analysis system is easy to understand and maintain by human users as it can provide an aspect-oriented explanation, but it cannot match supervised learning accuracy. On the contrary, learning-based sentiment analysis systems usually achieve the best performance in sentiment detection and classification, but they are by and large black boxes in the sense that no explanation or justification can be provided to the users.

A big concern in sentiment analysis is the domain dependency problem (i.e., the methods perform well only if they are targeted at a specific domain). On one hand, supervised learning solutions have superior performance but suffer a significant accuracy loss when domain boundaries are crossed. The simplest way to adapt such a sentiment analysis system to a particular domain is by collecting labelled domain-specific training data. However, that will be very expensive and time-consuming. Researchers have proposed many techniques to tackle the problem of domain adaptation, but they have various limitations. Supervised domain adaptation is not very dissimilar from supervised sentiment analysis, with the same components and requirements. Unsupervised domain adaptation methods in general have inferior performance and cannot match conventional supervised sentiment analysis methods. Moreover, many domain adaptation methods are designed with particular domain boundaries and constraints in mind. In the research literature, there are not many methods for domain adaptation between *distant domains* (e.g., between Twitter messages and newspaper articles). Those few that try to perform distant domain adaptation [158] have to make sacrifices in performance. Besides, as we have highlighted in a number of chapters of this thesis, in almost all cases, domain adaptation does not completely remove the domain dependency problem, and many of the available solutions suffer from significant sensitivity to domain boundaries, especially among *distant domains*. This issue makes them unsuitable for processing *noisy* sentiment sources containing *cross-style* or *near-domain* documents.

In order to address the problems described above, this thesis introduces a new approach to domain adaptation for sentiment analysis and presents our exploration towards answering the following research questions.

- Is it possible to overcome the above mentioned limitations of lexicon-based or learning-based methods and reduce their sensitivity to domain boundaries?
- Can lexicon-based systems improve their performance by learning a domain-specific lexicon?
- Are we able to narrow or close the performance gap between unsupervised methods and supervised methods?

Chapter by chapter, we have tried to chart the evolution of our *pSenti* sentiment analysis method along with the development of novel domain adaptation techniques. The research has concluded with the proposal of a system which can, in an nearly-unsupervised manner, adapt to the domain at hand and perform sentiment analysis with minimal loss of performance. The strengths of the system have been further demonstrated by multiple use case studies.

To report in greater detail what has been carried out within this scope, we summarise the key contributions of each chapter here.

In Chapter 3, we introduced ***pSenti*, a concept-level sentiment-analysis system** that seamlessly integrates lexicon-based and learning-based approaches to acquire adaptive sentiment analysis. The learning-based part of the proposed method was responsible for domain-specific lexicon discovery and adaptation to an underlying domain. While it is comparable to existing approaches, the experimental results from Chapter 3 illustrated one of the principal advantages of our *pSenti* algorithm (i.e. lower topic and style dependency compared to a pure bag-of-words machine-learning implementation).

In Chapter 4, we demonstrated that a high-quality **domain-specific sentiment lexicon** could be induced by using word embeddings in that domain, together with just a few seed words. The notable advantage of the proposed method over existing ones is that neither hand-crafting nor a labelled corpus are needed, and the induced lexicon quality is on a par with the handcrafted one. Lexicon induction enabled *lexicon-based pSenti* to work as an semi-supervised sentiment-detection system, which can not only be adapted to any new domain but also retains its former properties such as lower domain dependency.

As mentioned above, an induced lexicon can be applied directly to a lexicon-based algorithm for sentiment analysis; however, in Chapter 5 we achieve a higher performance through a two-stage bootstrapping approach and efficiently creating an end-to-end **semi-supervised** approach to domain-specific **sentiment analysis**. Compared to existing

methods, it has the advantage of working on any target domain, does not need labelled documents, and achieves sentiment classification accuracy comparable to that of fully supervised approaches.

Through this thesis, we also explored numerous domain-adaptation and cross-domain sentiment-analysis scenarios across a diverse set of domains. We investigated professional and customer reviews, and financial and political microblogs, and extracted sentiment from news articles and headlines. Some of our experimental choices were guided by the available datasets and we believe there is significant scope for improvement. Namely, the selection for products and product categories in customer review experiments and the use of a small dataset in the political stance detection use case. It is important to view some of the individual results in the context of other experiments, each of them presenting only small pieces of the puzzle rather than the full picture.

In Chapter 3, we explored a *near-cross-domain* environment, or, in other words, *cross-style*, as both datasets were from the same *topic domain*, although they used a different writing style. We identified that our mixed algorithm could be trained on one type of reviews and detect sentiments in another type, without a substantial penalty in its performance. We also briefly investigated and reported results on *distant cross-domain* sentiment analysis, where our proposed method outperformed the competing methods. Although some of our *cross-domain* experiments in Chapter 3 relied on a limited choice of domains and product categories, we believe that the gap was filled by our experiments from later chapters.

In Chapter 5, we compared all our proposed sentiment adaptation methods and demonstrated that they perform remarkably well in *cross-domain* sentiment analysis. As expected, *pSenti* with an induced lexicon, produced more stable results than any other method. By contrast, the deep-learning-based approach had some difficulties dealing with the change from regular to irregular language, which we believe might be resolved using additional text pre-processing and normalisation.

In Chapter 6, we presented **four case studies**, using *lexicon-based*, *supervised* and *semi-supervised* sentiment-analysis methods. Our case studies demonstrate the importance of domain adaptation and the advantages of our proposed method. They confirm that our semi-supervised domain-adaptation method might be universally applied to any target domain. We achieved good adaptation results in social media, customer and professional review, as well as with news articles. We discovered that historical sentiment information may be utilised to forecast future sentiment and proved the existence of *temporal domain dependency*. It was surprising that a *temporal sentiment* shift may be observed within a moderately short period. Moreover, the change can be observed in the **Amazon domain**, which has stringent boundaries and straightforward sentiment language. This finding confirmed that nowadays language evolution is a more

rapid process, recognising the importance of continuous sentiment system adaptation to an underlying domain.

In the **market sentiment** use case, we empirically re-examined the feasibility of applying sentiment analysis to make market trend predictions. The experimental results indicated that the interaction between sentiment and price is complex and dynamic: while some stocks in some time periods exhibited strong cross-correlation, it was absent in other cases. Furthermore, we have attempted to incorporate sentiment signals into machine-learning models for market trend prediction. In some scenarios, the proposed model outperformed the baseline model and demonstrated that sentiment emotions could be employed to help predict stock price movements, though sentiment attitudes could not.

In the **political-stance** detection use case, we examined Donald Trump supporters and opponents and performed stance detection on their tweets. The experimental results uncovered that in some domains it is not recommended to use vanilla sentiment-analysis methods, and that to get reliable results it is vital to perform domain adaptation. In the political-stance detection task, the general-purpose sentiment lexicon showed inferior performance, just fractionally better than random selection. Demographic pattern analysis has produced results which in some instances contradicted the official statistics and research papers. That may be due to the dataset size and sampling limitations and would require a further, more thorough investigation.

Admittedly, there are numerous opportunities to extend and enhance this work, and we name just a few as follows.

- Further investigation is required into the sensitivity of a learning-based approach to *distant-domain* boundaries, which is somewhat related to the ability of "translating" between the different languages spoken in different domains.

- Chapter 6 have identified some weaknesses in the current non-relevant message detection process which still need to be addressed.

- Some ideas that we have introduced in the development of our *pSenti* system still have much mileage and it is promising to expand them, e.g., the multi-stage approach to generating higher-quality domain-specific seeds, the exploitation of historical data to improve sentiment analysis, and so on.

- Recent advances in LM methods [55, 199] have brought great potential for future research. They demonstrate an improvement over existing, state-of-the-art sentiment analysis methods, and to improve our semi-supervised method from Chapter 5 we could consider replacing LSTM with a more sophisticated BERT [55]. Moreover, BERT's ability to capture context-specific information offers an opportunity to investigate new domain-independent sentiment analysis methods.

Provided with enough training data from a diverse set of domains, it may be possible to use its transfer learning capabilities and detect sentiment on any target domain without additional adaptation and with minimum loss of performance.

- Contextualised word-embeddings [55] are another interesting research topic worth considering in the future. It may further improve both the lexicon induction and the pseudo-labelled documents' bootstrap component.

- Several research directions, from Chapter 6 use case studies, are worth further investigation. A more in-depth analysis of seasonal review patterns and political stance using a larger dataset could provide a better insight into outstanding research questions and address current research limitations.

References

[1] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. "Sentiment Analysis of Twitter Data". In: *Proceedings of the Workshop on Language in Social Media*. 2011, pp. 30–38.

[2] R. Ahas, A. Aasa, S. Silm, and M. Tiru. "Daily Rhythms of Suburban Commuters' Movements in the Tallinn Metropolitan Area: Case Study with Mobile Positioning Data". In: *Transportation Research Part C: Emerging Technologies* 18.1 (2010), pp. 45–54.

[3] A. M. G. Almeida, S. Barbon, and E. C. Paraiso. "Multi-class Emotions Classification by Sentic Levels as Features in Sentiment Analysis". In: *5th Brazilian Conference on Intelligent Systems*. IEEE, Oct. 2016, pp. 486–491. ISBN: 9781509035663.

[4] N. Altrabsheh, M. Cocea, and S. Fallahkhair. "Learning Sentiment from Students' Feedback for Real-time Interventions in Classrooms". In: *International Conference on Adaptive and Intelligent Systems*. Springer. 2014, pp. 40–49.

[5] E. Altszyler, M. Sigman, S. Ribeiro, and D. F. Slezak. "Comparative Study of LSA vs Word2vec Embeddings in Small Corpora: A Case Study in Dreams Database". In: *arXiv preprint arXiv:1610.01520* (2016).

[6] A. Andreevskaia and S. Bergler. "When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2008.

[7] Apache. *OpenNLP*. 2010. URL: http://opennlp.apache.org.

[8] S. Argamon, C. Whitelaw, P. Chase, S. R. Hota, N. Garg, and S. Levitan. "Stylistic Text Classification Using Functional Lexical Features". In: *Journal of the American Society for Information Science and Technology* 58.6 (2007), pp. 802–822.

[9] M. Arias, A. Arratia, and R. Xuriguera. "Forecasting with Twitter Data". In: *ACM Transactions on Intelligent Systems and Technology* 5.1 (2013), p. 8. ISSN: 2157-6904.

[10] A. Aue and M. Gamon. "Customizing Sentiment Classifiers to New Domains: A Case Study". In: *Proceedings of Recent Advances in Natural Language Processing*. 2005.

[11] S. Baccianella, A. Esuli, and F. Sebastiani. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining". In: *International Conference on Language Resources and Evaluation*. Vol. 10. 2010, pp. 2200–2204.

[12] A. Badawy, E. Ferrara, and K. Lerman. "Analyzing the Digital Traces of Political Manipulation: The 2016 Russian Interference Twitter Campaign". In: *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE. 2018, pp. 258–265.

[13] P. T. Barthelemy, D. Guillory, and C. Mandal. "Using Twitter Data to Predict Box Office Revenues". In: *Stanford University Report* (2012).

[14] P. Beineke, T. Hastie, and S. Vaithyanathan. "The Sentimental Factor. improving review classification via human-provided information". In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Vol. 39. The AAAI Press Palo Alto, CA. Association for Computational Linguistics, 2004.

[15] D. Benikova, C. Biemann, M. Kisselew, and S. Pado. "Germeval 2014 Named Entity Recognition Shared Task: Companion Paper". In: *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*. Vol. 7. 2014, p. 281.

[16] J. Beran. *Statistics for Long-Memory Processe*. Vol. 61. Taylor & Francis, 1994. ISBN: 9780412049019.

[17] A. Bermingham and A. Smeaton. "On Using Twitter to Monitor Political Sentiment and Predict Election Results". In: *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology*. 2011, pp. 2–10.

[18] S. Bethard, H. Yu, A. Thornton, V. Hatzivassiloglou, and D. Jurafsky. "Automatic Extraction of Opinion Propositions and their Holders". In: *AAAI Spring Symposium on Exploring Attitude and Affect in Text*. 2004.

[19] S. Bird. "NLTK. the natural language toolkit". In: *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics. ACM Press, 2006, pp. 69–72.

[20] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN: 978-0387-31073-2.

[21] A. Blank and P. Koch. *Historical Semantics and Cognition*. Vol. 13. Walter de Gruyter, 1999. ISBN: 3-11-016614-3.

[22] J. Blitzer, M. Dredze, and F. Pereira. "Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 2007, pp. 440–447.

[23] Bloomberg. *Bloomberg Twitter Data Research Report*. URL: https://developer.twitter.com/content/dam/developer-twitter/pdfs-and-files/Bloomberg-Twitter-Data-Research-Report.pdf.

[24] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146.

[25] D. Bollegala, T. Maehara, and K. Kawarabayashi. "Unsupervised Cross-Domain Word Representation Learning". In: *arXiv preprint arXiv: 1505.07184* (2015).

[26] D. Bollegala, T. Mu, and J. Y. Goulermas. "Cross-domain Sentiment Classification Using Sentiment Sensitive Embeddings". In: *IEEE Transactions on Knowledge and Data Engineering* 28.2 (2016), pp. 398–410.

[27] D. Bollegala, D. J. Weir, and J. A. Carroll. "Cross-domain Sentiment Classification Using a Sentiment Sensitive Thesaurus". In: *IEEE Transactions on Knowledge and Data Engineering* 25.8 (2013), pp. 1719–1731. ISSN: 1041-4347.

[28] J. Bollen, H. Mao, and X. Zeng. "Twitter Mood Predicts the Stock Market". In: *Journal of Computational Science* 2.1 (Mar. 2011), pp. 1–8. ISSN: 1877-7503.

[29] G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne. "Machine Learning Strategies for Time Series Forecasting". In: *European Business Intelligence Summer School*. Springer. 2012, pp. 62–77.

[30] J. Borondo, A. J. Morales, J. C. Losada, and R. M. Benito. "Characterizing and Modeling an Electoral Campaign in the Context of Twitter: 2011 Spanish Presidential Election as a Case Study". In: *Chaos: an Interdisciplinary Journal of Nonlinear Science* 22.2 (2012), p. 023138.

[31] L. Borrajo, R. Romero, E. L. Iglesias, and C. M. R. Marey. "Improving Imbalanced Scientific Text Classification Using Sampling Strategies and Dictionaries". In: *Journal of Integrative Bioinformatics* 8.3 (2011), pp. 90–104.

[32] E. Breck, Y. Choi, and C. Cardie. "Identifying Expressions of Opinion in Context". In: *International Joint Conferences on Artificial Intelligence*. Vol. 7. 2007, pp. 2683–2688.

[33] G. W. Brier. "Verification of Forecasts Expressed in Terms of Probability". In: *Monthey Weather Review* 78.1 (1950), pp. 1–3.

[34] S. Brody and N. Diakopoulos. "Cooooooooooooooollllllllllllll!!!!!!!!!!!!!!: Using Word Lengthening to Detect Sentiment in Microblogs". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2011, pp. 562–570.

[35] S. Brody and N. Elhadad. "An Unsupervised Aspect-sentiment Model for Online Reviews". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, 2010, pp. 804–812. ISBN: 1-932432-65-5.

[36] E. Cambria, A. Livingstone, and A. Hussain. "The Hourglass of Emotions". In: *Cognitive Behavioural Systems* (2012), pp. 144–157.

[37] L. Cao and F. E. H. Tay. "Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting". In: *IEEE Transactions on Neural Networks* 14.6 (2003), pp. 1506–1518.

[38] G. Carenini, R. T. Ng, and E. Zwart. "Extracting Knowledge from Evaluative Text". In: *Proceedings of the 3rd International Conference on Knowledge Capture*. Association for Computational Linguistics. ACM Press, 2005, pp. 11–18. ISBN: 1595931635.

[39] G. Carenini, R. Ng, and A. Pauls. "Multi-document Summarization of Evaluative Text". In: *Proceedings of the European Chapter of the Association for Computational Linguistics*. 2006, pp. 305–312.

[40] A. X. Chang and C. Manning. "SUTime: A library for recognizing and normalizing time expressions". In: *Proceedings of the Eight International Conference on Language Resources and Evaluation*. Istanbul, Turkey: European Language Resources Association (ELRA), 2012. ISBN: 978-2-9517408-7-7.

[41] C. Chatfield. *The Analysis of Time Series: An Introduction*. Chapman & Hall/CRC Texts in Statistical Science. CRC press, 2016. ISBN: 9780203491683.

[42] K. Chen, Y. Zhou, and F. Dai. "A LSTM-based Method for Stock Returns Prediction: A Case Study of China Stock Market". In: *Proceedings of the 2015 IEEE International Conference on Big Data*. IEEE. 2015, pp. 2823–2824.

[43] Y. Chen and J. Xie. "Online Consumer Review: Word-of-mouth as a New Element of Marketing Communication Mix". In: *Management Science* 54.3 (2008), pp. 477–491.

[44] E. Ş. Chifu, T. Ş. Leţia, and V. R. Chifu. "Unsupervised Aspect Level Sentiment Analysis Using Ant Clustering and Self-organizing Maps". In: *International Conference on Speech Technology and Human-Computer Dialogue*. IEEE. 2015, pp. 1–9.

[45] Y. Choi, E. Breck, and C. Cardie. "Joint Extraction of Entities and Relations for Opinion Recognition". In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. ACM Press, 2006, pp. 431–439. ISBN: 1932432736.

[46] M. Choy, M. L. F. Cheong, M. N. Laik, and K. P. Shung. "A Sentiment Analysis of Singapore Presidential Election 2011 Using Twitter Data with Census Correction". In: *arXiv preprint arXiv:1108.5520* (2011).

[47] M. Cliche. "BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs". In: *Proceedings of the 11th International Workshop on Semantic Evaluation*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 573–580.

[48] N. A. C. Cressie and H. J. Whitford. "How to Use the Two Sample T-test". In: *Biometrical Journal* 28.2 (1986), pp. 131–148.

[49] A. M. Dai and Q. V. Le. "Semi-Supervised Sequence Learning". In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems*. Montreal, Quebec, Canada, 2015, pp. 3079–3087.

[50] S. P. Das and S. Padhy. "Support Vector Machines for Prediction of Futures Prices in Indian Stock Market". In: *International Journal of Computer Applications* 41.3 (2012).

[51] J. Dastin. *Amazon Trounces Rivals in Battle of the Shopping 'bots'*. 2017. URL: https://www.reuters.com/article/us-amazon-com-bots-insight/amazon-trounces-rivals-in-battle-of-the-shopping-bots-idUSKBN1860FK.

[52] A. M. De Livera, R. J. Hyndman, and R. D. Snyder. "Forecasting Time Series with Complex Seasonal Patterns Using Exponential Smoothing". In: *Journal of the American Statistical Association* 106.496 (2011), pp. 1513–1527.

[53] C. Dellarocas, X. M. Zhang, and N. F. Awad. "Exploring the Value of Online Product Reviews in Forecasting Sales: The Case of Motion Pictures". In: *Journal of Interactive Marketing* 21.4 (2007), pp. 23–45.

[54] S. Deng, T. Mitsubuchi, K. Shioda, T. Shimada, and A. Sakurai. "Combining Technical Analysis with Sentiment Analysis for Stock Price Prediction". In: *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*. IEEE. 2011, pp. 800–807.

[55] J. Devlin, M. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[56] D. A. Dickey and W. A. Fuller. "Distribution of the Estimators for Autoregressive Time Series with a Unit Root". In: *Journal of the American Statistical Association* 74.366a (1979), pp. 427–431.

[57] X. Ding, Y. Zhang, T. Liu, and J. Duan. "Deep Learning for Event-driven Stock Prediction". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, 2015, pp. 2327–2333. ISBN: 978-1-57735-738-4.

[58] X. Ding, B. Liu, and P. S. Yu. "A Holistic Lexicon-based Approach to Opinion Mining". In: *Proceedings of the International Conference on Web Search and Web Data Mining*. WSDM '08. Palo Alto, California, USA: ACM Press, 2008, pp. 231–240.

[59] X. Dong and G. Melo. "A Helping Hand: Transfer Learning for Deep Sentiment Analysis". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2018, pp. 2524–2534.

[60] J. Eisenstein. "Unsupervised Learning for Lexicon-Based Classification". In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. San Francisco, CA, USA, 2017, pp. 3188–3194.

[61] P. Ekman. "An Argument for Basic Emotions". In: *Cognition and Emotion* 6.3-4 (May 1992), pp. 169–200. ISSN: 1464-0600.

[62] P. Ekman, W. V. Friesen, and P. Ellsworth. *Emotion in the Human Face: Guidelines for Research and an Integration of Findings*. Pergamon general psychology series. Elsevier Science, 2013. ISBN: 9781483147635.

[63] A. Esuli and F. Sebastiani. "SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining". In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation*. Genoa, Italy: European Language Resources Association (ELRA), 2006.

[64] *Exit Polls 2016*. Nov. 2016. URL: https://edition.cnn.com/election/2016/results/exit-polls.

[65] S. Faralli and R. Navigli. "A New Minimally-supervised Framework for Domain Word Sense Disambiguation". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Jeju Island, Korea: Association for Computational Linguistics, 2012, pp. 1411–1422.

[66] M. Farhadloo and E. Rolland. "Multi-class Sentiment Analysis with Clustering and Score Representation". In: *IEEE 13th International Conference on Data Mining Workshops*. IEEE, 2013, pp. 904–912.

[67] M. Finnegan. *Thomson Reuters Adds Twitter Sentiment Analysis to Eikon Trading Terminal*. 2014. URL: https://www.computerworlduk.com/it-vendors/thomson-reuters-adds-twitter-sentiment-analysis-eikon-trading-terminal-3499978.

[68] E. Frank and M. A. Hall. "A Simple Approach to Ordinal Classification". In: *Proceedings of the 12th European Conference on Machine Learning*. Freiburg, Germany, 2001, pp. 145–156.

[69] N. H. Frijda. “The Laws of Emotion”. In: *American Psychologist* 43.5 (1988), p. 349.

[70] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera. “DRCW-OVO: Distance-based Relative Competence Weighting Combination for one-vs-one Strategy in Multi-class Problems”. In: *Pattern Recognition* 48.1 (2015), pp. 28–42.

[71] Q. Gao. “Stock Market Forecasting Using Recurrent Neural Network”. PhD thesis. University of Missouri–Columbia, 2016.

[72] D. Geeraerts. *Diachronic Prototype Semantics: A Contribution to Historical Lexicology*. Oxford Studies in Lexicography. Clarendon Press, 1997. ISBN: 9780198236528.

[73] D. Geeraerts and H. Cuyckens. “Introducing Cognitive Linguistics”. In: *The Oxford Handbook of Cognitive Linguistics*. 2007.

[74] A. Giachanou and F. Crestani. “Tracking Sentiment by Time Series Analysis”. In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. Association for Computational Linguistics. ACM Press, 2016, pp. 1037–1040. ISBN: 9781450340694.

[75] F. Gieseke, A. Airola, T. Pahikkala, and O. Kramer. “Sparse Quasi-Newton Optimization For Semi-Supervised Support Vector Machines”. In: *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*. Vilamoura, Algarve, Portugal: SciTePress - Science, 2012, pp. 45–54. ISBN: 9789898425997.

[76] X. Glorot, A. Bordes, and Y. Bengio. “Domain Adaptation for Large-scale Sentiment Classification: A Deep Learning Approach”. In: *Proceedings of the 28th international conference on machine learning*. 2011, pp. 513–520.

[77] A. Go, R. Bhayani, and L. Huang. “Twitter Sentiment Classification Using Distant Supervision”. In: *CS224N Project Report, Stanford* 1.2009 (2009), p. 12.

[78] N. Godbole, M. Srinivasaiah, and S. Skiena. “Large-Scale Sentiment Analysis for News and Blogs”. In: *The International AAAI Conference on Web and Social Media* 7.21 (2007), pp. 219–222.

[79] A. B. Goldberg and X. Zhu. “Seeing Stars when there aren’t many Stars: Graph-based Semi-supervised Learning for Sentiment Categorization”. In: *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*. TextGraphs-1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 45–52.

[80] Y. Goldberg. “Neural Network Methods for Natural Language Processing”. In: *Synthesis Lectures on Human Language Technologies* 10.1 (2017), pp. 1–309. ISSN: 1947-4059.

[81] C. W. J. Granger. “Investigating Causal Relations by Econometric Models and Cross-spectral Methods”. In: *Econometrica: Journal of the Econometric Society* (1969), pp. 424–438.

[82] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber. “LSTM: A Search Space Odyssey”. In: *IEEE Transactions on Neural Networks* 28.10 (Oct. 2017), pp. 2222–2232. ISSN: 2162-2388.

[83] S. Günnemann, N. Günnemann, and C. Faloutsos. "Detecting Anomalies in Dynamic Rating Data: A Robust Probabilistic Model for Rating Evolution". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. Association for Computational Linguistics. ACM Press, 2014, pp. 841–850.

[84] J. Haidt. "The Moral Emotions". In: R. J. Davidson, K. R. Scherer, and H. H. Goldsmith. *Handbook of Affective Science*. Oxford University Press, 2003. ISBN: 9780198029120.

[85] Y. Hamilakis. "The Past As Oral History". In: *Thinking through the Body*. Springer, 2002, pp. 121–136.

[86] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky. "Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, TX, USA: Association for Computational Linguistics, 2016, pp. 595–605.

[87] P. J. Hardin and J. M. Shumway. "Statistical Significance and Normalized Confusion Matrices". In: *Photogrammetric Engineering and Remote Sensing* 63.6 (1997), pp. 735–739.

[88] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. 2nd ed. Springer New York, 2009. ISBN: 9780387848587.

[89] V. Hatzivassiloglou and K. R. McKeown. "Predicting the Semantic Orientation of Adjectives". In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. ACL '98. Madrid, Spain: Association for Computational Linguistics, 1997, pp. 174–181.

[90] V. Hatzivassiloglou and J. M. Wiebe. "Effects of Adjective Orientation and Gradability on Sentence Subjectivity". In: *Proceedings of the 18th Conference on Computational Linguistics*. COLING '00. Saarbrücken, Germany: Association for Computational Linguistics, 2000, pp. 299–305. ISBN: 155860717X.

[91] Y. He. "Incorporating Sentiment Prior Knowledge for Weakly Supervised Sentiment Analysis". In: *ACM Transactions on Asian Language Information Processing* 11.2 (2012), p. 4.

[92] Z. He and K. Maekawa. "On Spurious Granger Causality". In: *Economics Letters* 73.3 (2001), pp. 307–313.

[93] J. M. Hilbe. "Logistic Regression". In: *International Encyclopedia of Statistical Science*. Springer, 2011, pp. 755–758.

[94] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780. ISSN: 1530-888X.

[95] A. Hogenboom, D. Bal, F. Frasincar, M. Bal, F. de Jong, and U. Kaymak. "Exploiting Emoticons in Sentiment Analysis". In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. Association for Computational Linguistics. ACM Press, 2013, pp. 703–710.

[96] J. Hong and M. Fang. "Sentiment Analysis with Deeply Learned Distributed Representations of Variable Length Texts". In: *Stanford University Report* (2015). URL: https://cs224d.stanford.edu/reports/HongJames.pdf.

[97] C. Hsu and C. Lin. “A Comparison of Methods for Multiclass Support Vector Machines”. In: *IEEE Transactions on Neural Networks* 13.2 (2002), pp. 415–425.

[98] M. Hu and B. Liu. “Mining and Summarizing Customer Reviews”. In: *Proceedings of the 2004 ACM Sigkdd International Conference on Knowledge Discovery and Data Mining*. KDD ’04. Seattle, WA, USA: ACM Press, 2004, pp. 168–177.

[99] M. Hu and B. Liu. “Opinion Feature Extraction Using Class Sequential Rules”. In: *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*. Stanford, CA, USA, 2006, pp. 61–66.

[100] W. Huang, Y. Nakamori, and S.-Y. Wang. “Forecasting Stock Market Movement Direction with Support Vector Machine”. In: *Computers & Operations Research* 32.10 (2005), pp. 2513–2522.

[101] X. Huang, Y. Huang, M. Wen, A. An, Y. Liu, and J. Poon. “Applying Data Mining to Pseudo-Relevance Feedback for High Performance Text Retrieval”. In: *Proceedings of the 6th IEEE International Conference on Data Mining*. Hong Kong, China, 2006, pp. 295–306.

[102] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. “Twitter Power: Tweets As Electronic Word of Mouth”. In: *Journal of the Association for Information Science and Technology* 60.11 (2009), pp. 2169–2188.

[103] S. Jansen. *Hands-On Machine Learning for Algorithmic Trading: Design and implement investment strategies based on smart algorithms that learn from data using Python*. Packt Publishing, 2018. ISBN: 9781789342710.

[104] S. Jebbara and P. Cimiano. “Aspect-Based Sentiment Analysis Using a Two-Step Neural Network Architecture”. In: *Semantic Web Evaluation Challenge*. Springer. 2016, pp. 153–167.

[105] Y. Jo and A. H. Oh. “Aspect and Sentiment Unification Model for Online Review Analysis”. In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. WSDM ’11. Hong Kong, China: ACM Press, 2011, pp. 815–824. ISBN: 9781450304931.

[106] T. Joachims. “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”. In: *Proceedings of the 10th European Conference on Machine Learning*. Chemnitz, Germany, 1998, pp. 137–142.

[107] T. Joachims. “Transductive Inference for Text Classification using Support Vector Machines”. In: *Proceedings of the 16th International Conference on Machine Learning*. Bled, Slovenia, 1999, pp. 200–209.

[108] T. Joachims. “Transductive Learning via Spectral Graph Partitioning”. In: *Proceedings of the 20th International Conference on Machine Learning*. Washington, DC, USA, 2003, pp. 290–297.

[109] D. Kahneman and P. Egan. *Thinking, Fast and Slow*. Vol. 1. Penguin Books Limited, 2011, p. 512. ISBN: 9780141918921.

[110] N. Kaji and M. Kitsuregawa. “Building Lexicon for Sentiment Analysis from Massive Collection of HTML Documents”. In: *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 2007, pp. 1075–1083.

[111] J. Kamps, M. Marx, R. J. Mokken, M. De Rijke, et al. "Using WordNet to Measure Semantic Orientations of Adjectives". In: *International Conference on Language Resources and Evaluation*. Vol. 4. 2004, pp. 1115–1118.

[112] J. Kaur and J. R. Saini. "Emotion Detection and Sentiment Analysis in Text Corpus: A Differential Study with Informal and Formal Writing Styles". In: *International Journal of Computer Applications* 101.9 (2014).

[113] S. Kim and E. Hovy. "Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text". In: *Proceedings of the Workshop on Sentiment and Subjectivity in Text*. SST '06. Sydney, Australia: Association for Computational Linguistics, 2006, pp. 1–8. ISBN: 1-932432-75-2.

[114] Y. Kim. "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1746–1751.

[115] S. Kohail. "Unsupervised Topic-Specific Domain Dependency Graphs for Aspect Identification in Sentiment Analysis". In: *Recent Advances in Natural Language Processing*. 2015, pp. 16–23.

[116] M. Koppel and J. Schler. "The Importance of Neutral Examples for Learning Sentiment". In: *Computational Intelligence* 22.2 (2006), pp. 100–109.

[117] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena. "Statistically Significant Detection of Linguistic Change". In: *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2015, pp. 625–635.

[118] M. L McHugh. "Interrater Reliability: The Kappa Statistic". In: *Biochemia Medica* 22 (Oct. 2012), pp. 276–82.

[119] M. Lai, D. I. H. Farías, V. Patti, and P. Rosso. "Friends and Enemies of Clinton and Trump: Using Context for Detecting Stance in Political Tweets". In: *Mexican International Conference on Artificial Intelligence*. Springer. 2016, pp. 155–168.

[120] M. Lai, V. Patti, G. Ruffo, and P. Rosso. "Stance Evolution and Twitter Interactions in an Italian Political Debate". In: *International Conference on Applications of Natural Language to Information Systems*. Springer. 2018, pp. 15–27.

[121] H. Lakkaraju, R. Socher, and C. Manning. "Aspect Specific Sentiment Analysis Using Hierarchical Deep Learning". In: *NIPS Workshop on Deep Learning and Representation Learning*. 2014.

[122] R. Lawson. *Web Scraping with Python*. Packt Publishing Ltd, 2015, p. 174. ISBN: 9781782164371.

[123] Y. LeCun, Y. Bengio, and G. Hinton. "Deep Learning". In: *Nature* 521.7553 (May 2015), pp. 436–444.

[124] H. Lee, M. Surdeanu, B. MacCartney, and D. Jurafsky. "On the Importance of Text Analysis for Stock Price Prediction". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation*. 2014, pp. 1170–1175.

[125] M. Lee. *Why Would My Stock's Value Decline Despite Good News Being Released?* 2018. URL: https://www.investopedia.com/ask/answers/06/stockdeclinegoodnews.asp.

[126] H. Li, N. Guevara, N. Herndon, D. Caragea, K. Neppalli, C. Caragea, A. C. Squicciarini, and A. H. Tapia. "Twitter Mining for Disaster Response: A Domain Adaptation Approach". In: *Information Systems for Crisis Response and Management*. 2015.

[127] S. Li, C. Huang, and C. Zong. "Multi-Domain Sentiment Classification with Classifier Combination". In: *Journal of Computer Science and Technology* 26.1 (2011), pp. 25–33. ISSN: 1860-4749.

[128] S. Li, Z. Wang, G. Zhou, and S. Y. M. Lee. "Semi-Supervised Learning for Imbalanced Sentiment Classification". In: *Twenty-Second International Joint Conference on Artificial Intelligence*. 2011.

[129] M. A. Lim. *The Handbook of Technical Analysis+ Test Bank: The Practitioner's Comprehensive Guide to Technical Analysis*. John Wiley & Sons, 2015, p. 800. ISBN: 9781118498934.

[130] C. Lin and Y. He. "Joint Sentiment/Topic Model for Sentiment Analysis". In: *Proceedings of the 18th ACM Conference on Information and knowledge management*. CIKM '09. Hong Kong, China: ACM Press, 2009, pp. 375–384. ISBN: 978-1-60558-512-3.

[131] X. Lin, Z. Yang, and Y. Song. "Short-term Stock Price Prediction Based on Echo State Networks". In: *Expert Systems with Applications* 36.3 (2009), pp. 7313–7317.

[132] B. Liu. *Web Data Mining*. Data-Centric Systems and Applications. Springer Berlin Heidelberg, 2011. ISBN: 9783642194603.

[133] B. Liu. "Sentiment Analysis and Opinion Mining". In: *Synthesis lectures on human language technologies* 5.1 (2012), pp. 1–167.

[134] B. Liu. *Sentiment Analysis and Subjectivity*. Morgan & Claypool, 2012, p. 167. ISBN: 9781608458844.

[135] B. Liu. *Sentiment Analysis. Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2015. ISBN: 9781139084789.

[136] D. Liu and L. Lei. "The Appeal to Political Sentiment: An Analysis of Donald Trump's and Hillary Clinton's Speech Themes and Discourse Strategies in the 2016 US Presidential Election". In: *Discourse, Context & Media* 25 (Oct. 2018), pp. 143–152. ISSN: 2211-6958.

[137] J. Liu, Y. Cao, C. Lin, Y. Huang, and M. Zhou. "Low-Quality Product Review Detection in Opinion Summarization". In: *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Vol. 7. 2007, pp. 334–342.

[138] Z. Liu, Y. Lin, M. Wang, and Z. Lu. "Discovering Opinion Changes in Online Reviews via Learning Fine-Grained Sentiments". In: *IEEE 2nd International Conference on Collaboration and Internet Computing*. IEEE. 2016, pp. 1–10.

[139] G. M. Ljung and G. E. P. Box. "On a Measure of Lack of Fit in Time Series Models". In: *Biometrika* 65.2 (1978), pp. 297–303.

[140] M. Loog. "Contrastive Pessimistic Likelihood Estimation for Semi-supervised Classification". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.3 (2016), pp. 462–475. ISSN: 2160-9292.

[141] W. Lowe, K. Benoit, S. Mikhaylov, and M. Laver. "Scaling Policy Preferences from Coded Political Texts". In: *Legislative Studies Quarterly* 36.1 (2011), pp. 123–155.

[142] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai. "Automatic Construction of a Context-aware Sentiment Lexicon: An Optimization Approach". In: *Proceedings of the 20th international conference on World wide web*. Association for Computational Linguistics. ACM Press, 2011, pp. 347–356.

[143] Y. Lu and C. Zhai. "Opinion Integration through Semi-supervised Topic Modeling". In: *Proceedings of the 17th International Conference on World Wide Web*. Association for Computational Linguistics. ACM Press, 2008, pp. 121–130. ISBN: 9781605580852.

[144] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon: Association for Computational Linguistics, 2011, pp. 142–150. ISBN: 978-1-932432-87-9.

[145] L. Maaten and G. Hinton. "Visualizing Data Using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[146] H. Maeda, K. Shimada, and T. Endo. "Twitter Sentiment Analysis Based on Writing Style". In: *Advances in Natural Language Processing*. Springer, 2012, pp. 278–288.

[147] B. G. Malkiel. "The Efficient Market Hypothesis and its Critics". In: *Journal of Economic Perspectives* 17.1 (2003), pp. 59–82.

[148] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. ISBN: 9780521758789.

[149] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. "The Stanford CoreNLP Natural Language Processing Toolkit". In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, 2014, pp. 55–60.

[150] M. V. Mäntylä, D. Graziotin, and M. Kuutila. "The Evolution of Sentiment Analysis—a Review of Research Topics, Venues, and Top Cited Papers". In: *Computer Science Review* 27 (Feb. 2018), pp. 16–32. ISSN: 1574-0137.

[151] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. "Building a Large Annotated Corpus of English: The Penn Treebank". In: *Computational Linguistics* 19.2 (June 1993), pp. 313–330. ISSN: 0891-2017.

[152] E. Marx and Z. Yellin-Flaherty. "Aspect Specific Sentiment Analysis of Unstructured Online Reviews". In: *Stanford University Report* (2015). URL: https://cs224d.stanford.edu/reports/MarxElliot.pdf.

[153] D. Matsumoto. "More Evidence for the Universality of a Contempt Expression". In: *Motivation and Emotion* 16.4 (1992), pp. 363–368.

[154] J. McAuley and J. Leskovec. "Hidden Factors and Hidden Topics". In: *Proceedings of the 7th ACM conference on Recommender systems*. RecSys '13. Hong Kong, China: ACM Press, 2013, pp. 165–172. ISBN: 9781450324090.

[155] W. Medhat, A. Hassan, and H. Korashy. "Sentiment Analysis Algorithms and Applications: A Survey". In: *Ain Shams Engineering Journal* 5.4 (2014), pp. 1093–1113.

[156] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. "Topic Sentiment Mixture. modeling facets and opinions in weblogs". In: *Proceedings of the 16th International Conference on World Wide Web*. Association for Computational Linguistics. ACM Press, 2007, pp. 171–180. ISBN: 9781595936547.

[157] Q. Mei, C. Liu, H. Su, and C. Zhai. "A Probabilistic Approach to Spatiotemporal Theme Pattern Mining on Weblogs". In: *Proceedings of the 15th International Conference on World Wide Web*. Association for Computational Linguistics. ACM Press, 2006, pp. 533–542. ISBN: 1595933239.

[158] Y. Mejova and P. Srinivasan. "Crossing Media Streams with Sentiment: Domain Adaptation in Blogs, Reviews and Twitter". In: *The International AAAI Conference on Web and Social Media*. 2012.

[159] J. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, T. G. B. Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. "Quantitative Analysis of Culture Using Millions of Digitized Books". In: *Science* 331.6014 (2011), pp. 176–182.

[160] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26: Annual Conference on Neural Information Processing Systems*. Vol. 26. Lake Tahoe, NV, USA, 2013, pp. 3111–3119.

[161] G. A. Miller and W. G. Charles. "Contextual Correlates of Semantic Similarity". In: *Language and Cognitive Processes* 6.1 (1991), pp. 1–28.

[162] T. Miller, D. Benikova, and S. Abualhaija. "GermEval 2015: Lexsub–a shared task for German-language lexical substitution". In: *Proceedings of the First Workshop on German Lexical Substitution*. 2015, pp. 1–9.

[163] M. Minsky. *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon and Schuster, 2007, p. 400. ISBN: 9781416579304.

[164] D. Mladenić, J. Brank, M. Grobelnik, and N. Milic-Frayling. "Feature Selection Using Linear Classifier Weights: Interaction with Classification Models". In: *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. Association for Computational Linguistics. ACM Press, 2004, pp. 234–241.

[165] S. M. Mohammad, S. Kiritchenko, and X. Zhu. "NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets". In: *arXiv preprint arXiv: 1308.6242* (2013).

[166] S. M. Mohammad and P. D. Turney. "Crowdsourcing a Word–emotion Association Lexicon". In: *Computational Intelligence* 29.3 (2013), pp. 436–465.

[167] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry. "SemEval-2016 Task 6: Detecting Stance in Tweets". In: *Proceedings of the 10th International Workshop on Semantic Evaluation*. 2016, pp. 31–41.

[168] S. M. Monnat. "Deaths of Despair and Support for Trump in the 2016 Presidential Election". In: *Pennsylvania State University Department of Agricultural Economics Research Brief* (2016).

[169] A. Mudinas, D. Zhang, and M. Levene. "Combining Lexicon and Learning Based Approaches for Concept-level Sentiment Analysis". In: *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*. WISDOM '12. Beijing, China: ACM Press, 2012. ISBN: 9781450315432.

[170] K. Mulligan. "Moral Emotions". In: D. Sander and K. Scherer. *The Oxford Companion to Emotion and the Affective Sciences*. Oxford University Press, 2008. ISBN: 9780198569633.

[171] P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson. "SemEval-2013 Task 2: Sentiment Analysis in Twitter". In: *Proceedings of the Seventh International Workshop on Semantic Evaluation*. 2013.

[172] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov. "SemEval-2016 Task 4: Sentiment Analysis in Twitter". In: *Proceedings of the 10th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2016, pp. 1–18.

[173] A. Niculescu-Mizil and R. Caruana. "Predicting Good Probabilities with Supervised Learning". In: *Proceedings of the 22nd International Conference on Machine Learning*. Bonn, Germany: ACM Press, 2005, pp. 625–632.

[174] R. Ortega, A. Fonseca, Y. Gutiérrez, and A. Montoyo. "Improving Subjectivity Detection Using Unsupervised Subjectivity Word Sense Disambiguation". In: *Procesamiento del Lenguaje Natural* 51 (2013).

[175] S. Owsley, S. Sood, and K. J. Hammond. "Domain Specific Affective Classification of Documents". In: *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*. Stanford, CA, USA, 2006, pp. 181–183.

[176] P. Pai and C. Lin. "A Hybrid ARIMA and Support Vector Machines Model in Stock Price Forecasting". In: *Omega* 33.6 (2005), pp. 497–505.

[177] S. J. Pan, X. Ni, J. Sun, Q. Yang, and Z. Chen. "Cross-domain Sentiment Classification via Spectral Feature Alignment". In: *Proceedings of the 19th International Conference on World Wide Web*. WWW '10. Raleigh, North Carolina, USA: ACM Press, 2010, pp. 751–760. ISBN: 9781605587998.

[178] S. J. Pan and Q. Yang. "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359.

[179] B. Pang and L. Lee. "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. ACL '04. Barcelona, Spain: Association for Computational Linguistics, 2004.

[180] B. Pang and L. Lee. "Seeing Stars. exploiting class relationships for sentiment categorization with respect to rating scales". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics. Association for Computational Linguistics, 2005, pp. 115–124.

[181] B. Pang and L. Lee. "Opinion Mining and Sentiment Analysis". In: *Foundations and Trends in Information Retrieval* 2.1–2 (2008), pp. 1–135. ISSN: 1554-0677.

[182] B. Pang, L. Lee, and S. Vaithyanathan. "Thumbs Up? sentiment classification using machine learning techniques". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*. EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 79–86.

[183] W. G. Parrott. *Emotions in Social Psychology: Essential Readings*. Psychology Press, 2001, p. 378. ISBN: 9780863776823.

[184] V. M. K. Peddinti and P. Chintalapoodi. "Domain Adaptation in Sentiment Analysis of Twitter". In: *Analyzing Microtext* 11 (2011), p. 05.

[185] J. Pennington, R. Socher, and C. Manning. "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2014, pp. 1532–1543.

[186] R. W. Picard. "Affective Computing". In: M.I.T. Media Laboratory Perceptual Computing Section technical report. The MIT Press, 1995. ISBN: 0-262-16170-2.

[187] E. Picardo. *How To Trade The News*. 2013. URL: https://www.investopedia.com/articles/active-trading/111313/how-trade-news.asp.

[188] J. Platt. "Advances in Large Margin Classifiers". In: A. J. Smola and P. J. Bartlett. Cambridge, MA, USA: MIT Press, 2000. Chap. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. ISBN: 0262194481.

[189] R. Plutchik. "A General Psychoevolutionary Theory Of Emotion". In: *Theories of Emotion*. New York: Elsevier, 1980, pp. 3–33. ISBN: 9780125587013.

[190] R. Plutchik. "The Nature of Emotions: Human Emotions Have Deep Evolutionary Roots, a Fact That May Explain Their Complexity and Provide Tools for Clinical Practice". In: *American Scientist* 89.4 (2001), pp. 344–350.

[191] A. Popescu and O. Etzioni. "Extracting Product Features and Opinions from Reviews". In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, 2005, pp. 339–346.

[192] J. Prusa, T. M. Khoshgoftaar, D. J. Dittman, and A. Napolitano. "Using Random Undersampling to Alleviate Class Imbalance on Tweet Sentiment Data". In: *IEEE International Conference on Information Reuse and Integration*. IEEE, Aug. 2015, pp. 197–202. ISBN: 9781467366564.

[193] J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, et al. "The Timebank Corpus". In: *Corpus Linguistics*. Vol. 2003. Lancaster, UK. 2003, p. 40.

[194] B. Qian and K. Rasheed. "Stock Market Prediction with Multiple Classifiers". In: *Applied Intelligence* 26.1 (2007), pp. 25–33.

[195] X. Qian and S. Gao. "Financial Series Prediction: Comparison Between Precision of Time Series Models and Machine Learning Methods". In: *arXiv preprint arXiv:1706.00948* (2017).

[196] G. Qiu, B. Liu, J. Bu, and C. Chen. "Opinion Word Expansion and Target Extraction through Double Propagation". In: *Computational Linguistics* 37.1 (Mar. 2011), pp. 9–27. ISSN: 1530-9312.

[197] A. Radford, R. Józefowicz, and I. Sutskever. "Learning to Generate Reviews and Discovering Sentiment". In: *arXiv preprint arXiv:1704.01444* (2017).

[198] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. *Improving Language Understanding by Generative Pre-training*. URL: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

[199] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. *Language Models Are Unsupervised Multitask Learners*. Tech. rep. OpenAi, 2018.

[200] J. Read. "Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification". In: *Proceedings of the ACL Student Research Workshop*. ACLstudent '05. Ann Arbor, Michigan: Association for Computational Linguistics, 2005, pp. 43–48.

[201] F. N. Ribeiro, M. Araújo, P. Gonçalves, M. A. Gonçalves, and F. Benevenuto. "Sentibench — A Benchmark Comparison of State-of-the-Practice Sentiment Analysis Methods". In: *EPJ Data Science* 5.1 (2016), pp. 1–29.

[202] E. Riloff and J. Wiebe. "Learning Extraction Patterns for Subjective Expressions". In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Sapporo, Japan: Association for Computational Linguistics, 2003, pp. 105–112.

[203] S. Rodriguez. *U.S. Judge Says LinkedIn Cannot Block Startup from Public Profile Data*. 2017. URL: https://uk.reuters.com/article/us-microsoft-linkedin-ruling/u-s-judge-says-linkedin-cannot-block-startup-from-public-profile-data-idUKKCN1AU2BV.

[204] S. Rosenthal, N. Farra, and P. Nakov. "SemEval-2017 Task 4: Sentiment Analysis in Twitter". In: *Proceedings of the 11th International Workshop on Semantic Evaluation*. SemEval '17. Vancouver, Canada: Association for Computational Linguistics, 2017.

[205] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter, and V. Stoyanov. "SemEval-2015 Task 10: Sentiment Analysis in Twitter". In: *Proceedings of the 9th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2015.

[206] S. Rosenthal, A. Ritter, P. Nakov, and V. Stoyanov. "SemEval-2014 Task 9: Sentiment Analysis in Twitter". In: *Proceedings of the 8th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2014, pp. 73–80. ISBN: 978-1-941643-24-2.

[207] S. Rothe, S. Ebert, and H. Schütze. "Ultradense Word Embeddings by Orthogonal Transformation". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, CA, USA: Association for Computational Linguistics, 2016, pp. 767–777.

[208] M. A. Russell. *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More*. 2nd ed. O'Reilly Media, Inc., 2013, p. 448. ISBN: 9781449368210.

[209] H. Sack, S. Dietze, A. Tordai, and C. Lange. *Semantic Web Challenges. Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*. 1st ed. Springer International Publishing, 2016. ISBN: 9783319465647.

[210] A. Salinca. "Convolutional Neural Networks for Sentiment Classification on Business Reviews". In: *arXiv preprint arXiv:1710.05978* (2017).

[211] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice Hall, 1971.

[212] G. Salton and C. Buckley. "Term-weighting Approaches in Automatic Text Retrieval". In: *Information Processing & Management* 24.5 (Jan. 1988), pp. 513–523. ISSN: 0306-4573.

[213] S. L. Salzberg. "On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach". In: *Data Mining and Knowledge Discovery* 1.3 (1997), pp. 317–328. ISSN: 1384-5810.

[214] R. Samdani and W. Yih. "Domain Adaptation with Ensemble of Feature Groups". In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*. IJCAI'11. Barcelona, Catalonia, Spain: AAAI Press, 2011, pp. 1458–1464. ISBN: 978-1-57735-514-4.

[215] P. Schauble. *Multimedia Information Retrieval: Content-based Information Retrieval from Large Text and Audio Databases*. Norwell, MA, USA: Kluwer Academic Publishers, 1997. ISBN: 0792398998.

[216] K. R. Scherer. "Psychological Models of Emotion". In: *The Neuropsychology of Emotion*. Ed. by J. C. Borod. Series in Affective Science. Oxford University Press, 2000, pp. 137–162. ISBN: 9780198027409.

[217] K. Singhal, B. Agrawal, and N. Mittal. "Modeling Indian General Elections: Sentiment Analysis of Political Twitter Data". In: *Information Systems Design and Intelligent Applications*. Springer, 2015, pp. 469–477.

[218] P. Sobhani, S. Mohammad, and S. Kiritchenko. "Detecting Stance in Tweets and Analyzing its Interaction with Sentiment". In: *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, 2016, pp. 159–169.

[219] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning. "Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2011, pp. 151–161.

[220] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[221] P. J. Stone, D. C. Dunphy, M. S. Smith, and D. M. Ogilvie. *The General Inquirer: A Computer Approach to Content Analysis*. Cambridge, MA: MIT Press, 1966, p. 651.

[222] C. Strapparava and R. Mihalcea. "SemEval-2007 Task 14. affective text". In: *Proceedings of the 4th International Workshop on Semantic Evaluations*. SemEval '07. Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 70–74.

[223] S. Straube and M. M. Krell. "How to Evaluate an Agent's Behavior to Infrequent Events? Reliable Performance Estimation Insensitive to Class Distribution". In: *Frontiers in Computational Neuroscience* 8 (2014), p. 43. ISSN: 1662-5188.

[224] N. Tabari, P. Biswas, B. Praneeth, A. Seyeditabari, M. Hadzikadic, and W. Zadrozny. "Causality Analysis of Twitter Sentiments and Stock Market Returns". In: *Proceedings of the First Workshop on Economics and Natural Language Processing*. 2018, pp. 11–19.

[225] J. P. Tangney, J. Stuewig, and D. J. Mashek. "Moral Emotions and Moral Behavior". In: *The Annual Review of Physiology* 58.1 (Jan. 2007), pp. 345–372. ISSN: 1545-2085.

[226] M. Taulé, M. A. Martí, F. M. Rangel, P. Rosso, C. Bosco, V. Patti, et al. "Overview of the Task on Stance and Gender Detection in Tweets on Catalan Independence at IberEval 2017". In: *2nd Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval 2017*. Vol. 1881. CEUR-WS. 2017, pp. 157–177.

[227] M. Thelwall, K. Buckley, and G. Paltoglou. "Sentiment Strength Detection for the Social Web". In: *Journal of the American Society for Information Science and Technology* 63.1 (2012), pp. 163–173. ISSN: 1532-2882.

[228] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. "Sentiment Strength Detection in Short Informal Text". In: *Journal of the American Society for Information Science and Technology* 61.12 (2010), pp. 2544–2558. ISSN: 1532-2890.

[229] M. Thomason. "The Practitioner Methods and Tool". In: *Journal of Computational Intelligence in Finance* 7.3 (1999), pp. 36–45.

[230] S. S. Tomkins. "Affect Theory". In: K. R. Scherer and P. Ekman. *Approaches to Emotion*. 163–195. L. Erlbaum Associates, 1984. ISBN: 9780898594065.

[231] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment". In: *The International AAAI Conference on Web and Social Media* 10.1 (2010), pp. 178–185.

[232] P. D. Turney. "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA, USA, 2002, pp. 417–424.

[233] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[234] I. Vegas, T. Tian, and W. Xiong. "Characterizing the 2016 U.S. Presidential Campaign using Twitter Data". In: *International Journal of Advanced Computer Science and Applications* 7.10 (2016). ISSN: 2158-107X.

[235] S. Wager, S. I. Wang, and P. Liang. "Dropout Training as Adaptive Regularization". In: *Advances in Neural Information Processing Systems 26: Annual Conference on Neural Information Processing Systems*. Lake Tahoe, NV, USA, 2013, pp. 351–359.

[236] G. Wang, T. Wang, B. Wang, D. Sambasivan, Z. Zhang, H. Zheng, and B. Y. Zhao. "Crowds on Wall Street: Extracting Value from Collaborative Investing Platforms". In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. Association for Computational Linguistics. ACM Press, 2015, pp. 17–30.

[237] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan. "A System for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle". In: *Proceedings of the ACL 2012 System Demonstrations*. ACL '12. Jeju Island, Korea: Association for Computational Linguistics, 2012, pp. 115–120.

[238] H. Wang and J. A. Castanon. “Sentiment Expression Via Emoticons on Social Media”. In: *arXiv preprint arXiv:1511.02556* (2015).

[239] J. Wang and J. Leu. “Stock Market Trend Prediction Using ARIMA-based Neural Networks”. In: *Proceedings of International Conference on Neural Networks*. Vol. 4. IEEE. 1996, pp. 2160–2165.

[240] Y. Wang, Y. Li, and J. Luo. “Deciphering the 2016 US Presidential Campaign in the Twitter Sphere: A Comparison of the Trumpists and Clintonists”. In: *The International AAAI Conference on Web and Social Media*. 2016, pp. 723–726.

[241] A. B. Warriner, V. Kuperman, and M. Brysbaert. “Norms of Valence, Arousal, and Dominance for 13,915 English Lemmas”. In: *Behavior Research Methods* 45.4 (Feb. 2013), pp. 1191–1207. ISSN: 1554-3528.

[242] W. Wei, X. Zhang, X. Liu, W. Chen, and T. Wang. “Pkudblab at Semeval-2016 Task 6: A Specific Convolutional Neural Network System for Effective Stance Detection”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation*. 2016, pp. 384–388.

[243] E. W. Weisstein. *Reversion to the Mean*. 2000. URL: http://mathworld.wolfram.com/ReversiontotheMean.html.

[244] J. M. Wiebe, R. F. Bruce, and T. P. O’Hara. “Development and Use of a Gold-standard Data Set for Subjectivity Classifications”. In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. ACL ’99. College Park, Maryland: Association for Computational Linguistics, 1999, pp. 246–253.

[245] J. M. Wiebe. “Recognizing Subjective Sentences: A Computational Investigation of Narrative Text”. PhD thesis. Buffalo, NY, USA: State University of New York at Buffalo, 1990.

[246] J. Wiebe and E. Riloff. “Creating Subjective and Objective Sentence Classifiers from Unannotated Texts”. In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer. 2005, pp. 486–497.

[247] J. Wiebe, T. Wilson, and C. Cardie. “Annotating Expressions of Opinions and Emotions in Language”. In: *Language Resources and Evaluation* 39.2 (2005), pp. 165–210.

[248] J. W. Wilder. *New Concepts in Technical Trading Systems*. Trend Research, 1978, p. 141. ISBN: 9780894590276.

[249] T. Wilson, J. Wiebe, and P. Hoffmann. “Recognizing Contextual Polarity in Phrase-level Sentiment Analysis”. In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. Association for Computational Linguistics, 2005, pp. 347–354.

[250] H. Wu, Y. Gu, S. Sun, and X. Gu. “Aspect-based Opinion Summarization with Convolutional Neural Networks”. In: *International Joint Conference on Neural Networks*. Vol. abs/1511.09128. 2015. eprint: 1511.09128.

[251] J. P. Wu and S. Wei. *Time Series Analysis*. ChangSha: Hunan Science and Technology Press, 1989.

[252] Y. Wu and P. Jin. “SemEval-2010 Task 18: Disambiguating Sentiment Ambiguous Adjectives”. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. SemEval ’10. Los Angeles, California: Association for Computational Linguistics, 2010, pp. 81–85.

[253] R. Xia and C. Zong. “A POS-based Ensemble Model for Cross-Domain Sentiment Classification”. In: *Proceedings of the 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand, 2011, pp. 614–622.

[254] M. Xiao and Y. Guo. “Feature Space Independent Semi-supervised Domain Adaptation Via Kernel Matching”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.1 (2015), pp. 54–66.

[255] F. Z. Xing, E. Cambria, and R. E. Welsch. “Intelligent Bayesian Asset Allocation via Market Sentiment Views”. In: *IEEE Computational Intelligence Magazine* (2018).

[256] F. Z. Xing, E. Cambria, and R. E. Welsch. “Natural Language based Financial Forecasting: A Survey”. In: *Artificial Intelligence Review* 50.1 (2018), pp. 49–73.

[257] H. Xu, B. Liu, L. Shu, and P. S. Yu. “Double Embeddings and CNN-based Sequence Labeling for Aspect Extraction”. In: *arXiv preprint arXiv:1805.04601* (2018).

[258] M. R. Yaakub, Y. Li, and J. Zhang. “Integration of Sentiment Analysis into Customer Relational Model: The Importance of Feature Ontology and Synonym”. In: *Procedia Technology* 11 (2013), pp. 495–501.

[259] Y. Yang and X. Liu. “A Re-examination of Text Categorization Methods”. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. Berkeley, CA, USA: ACM Press, 1999, pp. 42–49. ISBN: 1581130961.

[260] H. Yu and V. Hatzivassiloglou. “Towards Answering Opinion Questions. separating facts from opinions and identifying the polarity of opinion sentences”. In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Sapporo, Japan: Association for Computational Linguistics, 2003, pp. 129–136.

[261] X. Yu, Y. Liu, J. X. Huang, and A. An. “Mining Online Reviews for Predicting Sales Performance: A Case Study in the Movie Domain”. In: *IEEE Transactions on Knowledge and Data Engineering* 24.4 (2012), pp. 720–734.

[262] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu. “Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis”. In: *HP Laboratories, Technical Report HPL-2011* 89 (2011).

[263] X. Zhang, H. Fuehres, and P. A. Gloor. “Predicting Stock Market Indicators through Twitter "I Hope It Is Not As Bad As I Fear"”. In: *Procedia-Social and Behavioral Sciences* 26 (2011), pp. 55–62.

[264] S. Zhi, X. Li, J. Zhang, X. Fan, L. Du, and Z. Li. “Aspects Opinion Mining Based on Word Embedding and Dependency Parsing”. In: *Proceedings of the International Conference on Advances in Image Processing*. ICAIP 2017. Bangkok, Thailand: ACM Press, 2017, pp. 210–215. ISBN: 978-1-4503-5295-6.

[265] F. Zhu and X. Zhang. “Impact of Online Consumer Reviews on Sales: The Moderating Role of Product and Consumer Characteristics”. In: *Journal of Marketing* 74.2 (2010), pp. 133–148.

[266] J. Zhu, H. Wang, and J. Mao. “Sentiment Classification Using Genetic Algorithm and Conditional Random Fields”. In: *2nd IEEE International Conference on Information Management and Engineering*. IEEE. 2010, pp. 193–196.

[267] X. Zhu, Z. Ghahramani, and J. D. Lafferty. “Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions”. In: *Proceedings of the 20th International Conference on Machine Learning*. Washington, DC, USA, 2003, pp. 912–919.